ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΤΟΠΟΓΡΑΦΙΑΣ ΚΑΙ ΓΕΟΠΛΗΡΟΦΟΡΙΚΗΣ

Διπλωματική εργασία

Σημασιολογική Κατάτμηση και Ανίχνευση Αντικειμένων σε 3Δ νέφη σημείων με χρήση Νευρωνικών Δικτύων

Συγγραφέας
ΝΤΟΥΝΗΣ ΑΝΤΩΝΙΟΣ
ΑΜ: 13114

Επιβλέπων Καθηγητής
ΓΡΑΜΜΑΤΙΚΟΠΟΥΛΟΣ ΛΑΖΑΡΟΣ

ΑΘΗΝΑ, ΜΑΡΤΙΟΣ 2024

UNIVERSITY OF WEST ATTICA
SCHOOL OF ENGINEERING
DEPARTMENT OF SURVEYING AND GEOINFORMATICS ENGINEERING

Diploma Thesis

Semantic Segmentation and Object Detection in 3D point clouds using Neural Networks

Student
NTOUNIS ANTONIS
Registration Number: 13114

Supervisor
GRAMMATIKOPOULOS LAZAROS

ATHENS, MARCH 2024

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΤΟΠΟΓΡΑΦΙΑΣ ΚΑΙ ΓΕΟΠΛΗΡΟΦΟΡΙΚΗΣ

Σημασιολογική Κατάτμηση και Ανίχνευση Αντικειμένων σε 3Δ νέφη σημείων με χρήση Νευρωνικών Δικτύων

Η διπλωματική εργασία εξετάστηκε επιτυχώς από την κάτωθι Εξεταστική Επιτροπή:

Α/Α	ΟΝΟΜΑ ΕΠΩΝΥΜΟ	ΒΑΘΜΙΔΑ/ΙΔΙΟΤΗΤΑ	ΨΗΦΙΑΚΗ ΥΠΟΓΡΑΦΗ
1.	Λάζαρος Γραμματικόπουλος	Αναπληρωτής Καθηγητής	LAZAROS GRAMMATIKOPOULOS Digitally signed by LAZAROS GRAMMATIKOPOULOS Date: 2024.10.17 17:38:56 +03'00'
2.	Έλλη Πέτσα	Καθηγήτρια	ELENI PETSA Digitally signed by ELENI PETSA Date: 2024.10.17 17:00:08 +03'00'
3.	Γεώργιος Σφήκας	Επίκουρος Καθηγητής	Georgios Signed by: Georgios Sfikas

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένη Ντούνης Αντώνιος του Αλέξη, με αριθμό μητρώου 13114, φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Τοπογραφίας και Γεωπληροφορικής, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της 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 Δηλών

Ντούνης Αντώνιος

Περίληψη

Στη παρούσα διπλωματική διερευνάται ο τρόπος λειτουργίας των νευρωνικών δικτύων στο ευρύτερο φάσμα της τεχνητής νοημοσύνης, χρησιμοποιώντας ως δεδομένα τρισδιάστατα νέφη σημείων, όπου σκοπός τους είναι ο αυτοματισμός λειτουργιών κατά το βέλτιστο δυνατό τρόπο. Πέραν των παραδειγμάτων που εξετάζονται στη συνέχεια, αναλύουμε που χρησιμεύουν και πως προσεγγίζουν ένα πρόβλημα ερευνώντας κάθε στάδιο τους, με σκοπό την καλύτερη κατανόηση των συγκεκριμένων τεχνικών. Αρχικά, στα πρώτα κεφάλαια γίνεται μια γενικότερη ανασκόπηση της τεχνητής νοημοσύνης και εκμάθησης μηχανής, ώστε να μπορεί ο αναγνώστης να μπει στο πνεύμα του τι αφορούν οι τεχνικές αυτές παραθέτοντας διάφορα παραδείγματα και ιστορικά γεγονότα. Εδώ είναι πολύ σημαντικός ο τρόπος που θα γίνει αυτή η ανάλυση και θα δοθεί ιδιαίτερη προσοχή κατά την προσέγγιση τους, διότι για να μπορεί να αντιληφθεί κάποιος τα ειδικά προβλήματα των νευρωνικών δικτύων που θα αντιμετωπίσουμε αργότερα, θα πρέπει πρώτα να έχει κατανοήσει σε ένα βαθμό την γενικότερη φιλοσοφία της τεχνητής νοημοσύνης. Στη συνέχεια εξετάζονται οι διαφορετικές περιπτώσεις των νευρωνικών δικτύων και αναλύονται ξεχωριστά τα δίκτυα της σημασιολογικής κατάτμησης και της ανίχνευσης αντικειμένων ως προς τον σκοπό και την τον τρόπο αντιμετώπισης της κάθε μεθόδου. Σε επόμενο κεφάλαιο εξετάζουμε το περιβάλλον που χρειάζεται για την ανάπτυξη τέτοιων δικτύων, καθώς και των εργαλείων που χρειάζονται για την υποστήριξη τέτοιων μεθόδων. Αργότερα, αναλύονται τα δεδομένα που χρησιμοποιήθηκαν για τα πειράματα και πως αυτά ερμηνεύονται στα νευρωνικά δίκτυα που υλοποιήθηκαν. Η διαδικασία όλων των αναλύσεων θα γίνει με τέτοια σειρά ώστε να μην υπάρξει κάποια σύγχυση μεταξύ των μεθόδων και να γίνουν απόλυτα κατανοητά το καθ’ ένα ξεχωριστά. Τέλος, εξετάζουμε τα αποτελέσματα των μεθόδων που εφαρμόστηκαν, καταλήγοντας σε κάποιες παρατηρήσεις και θέτοντας κάποιες παραδοχές όπου θα βοηθούσαν σε μελλοντικές βελτιώσεις των συγκεκριμένων μοντέλων.

Λέξεις κλειδιά: νευρωνικά δίκτυα, τεχνητή νοημοσύνη, τρισδιάστατα νέφη σημείων, σημασιολογική κατάτμηση, ανίχνευση αντικειμένων

Abstract

In the present diploma thesis, it investigates the operations of neural networks in the broad field of artificial intelligence, using as input data three-dimensional point clouds, where their purpose is to automate operations in the best possible way. In addition to the examples and experiments examined below, we analyze what they are used for and how to approach a problem, investigating each of their stages in order to understand the specific methods. Initially, in the first chapters a more general review of artificial intelligence and machine learning, so that the reader can get into the spirit of what these techniques are about, citing various examples and historical events. The manner in which this analysis will be done is very important, and special care will be taken in approaching them, because in order to be able to perceive the special problems of neural networks which we shall face later, one must first have understood to some extent the philosophy of artificial intelligence. Next, the different cases of neural networks are examined and the semantic segmentation and object detection networks are analyzed separately in terms of the purpose and approach of each method. Next chapter we examine the environment needed to develop such networks, as well as the tools needed to support such methods. Later, the data that will be used for the experiments are analyzed and how they are interpreted in the neural networks that will be used. The process of all analyses will be done in such an order so as not to create any confusion between the methods and that each one is completely understood separately. Initially, it integrates the different cases of neural networks and at the same time analyzes their process in terms of approaching the problem. Finally, we examine the results of the methods that will be used, concluding with some observations and making some assumptions that would help in future improvements of the specific models.

Keywords: neural networks, artificial intelligence, 3D point clouds, semantic segmentation, object detection

Περιεχόμενα

Περίληψη 3
Abstract 4
1. Εισαγωγή 10
2. Τεχνητή Νοημοσύνη 11
2.1 Μηχανική Μάθηση 13
2.1.1 Επιβλεπόμενη μάθηση 14
2.1.2 Μη Επιβλεπόμενη μάθηση 16
2.1.3 Λοιποί μέθοδοι μηχανικής μάθησης 18
3. Εισαγωγή Νευρωνικών Δικτύων 20
3.1 Δομή Τεχνητών Νευρωνικών Δικτύων 22
3.2 Συναρτήσεις Ενεργοποίησης 24
3.3 Δίκτυα πρόσθιας τροφοδότηση (Feedforward Networks) 29
3.4 Συναρτήσεις απώλειας και Τακτοποίηση 31
3.5 Κανόνες Βελτιστοποίησης 36
3.5.1 Αλγόριθμοι Gradient Descent 36
3.5.2 Αλγόριθμος Momentum 42
3.5.3 Αλγόριθμοι βελτιστοποίησης AdaGrad και RMSprop 43
3.5.4 Αλγόριθμος βελτιστοποίησης Adam 44
3.6 Οπισθοδιάδοση (Backpropagation) 45
3.7 Συνελικτικά νευρωνικά δίκτυα 48
4. Μοντέλα τρισδιάστατης σημασιολογικής κατάτμησης και ανίχνευσης αντικειμένων 51
4.1 State of the art μέθοδοι στη 3Δ επεξεργασία δεδομένων 52
4.2 Ανάλυση μοντέλου σημασιολογικής κατάτμησης RandLA-Net και παραμετροποίηση 58
4.3 Ανάλυση μοντέλου σημασιολογικής κατάτμησης KPConv και παραμετροποίηση 62
4.4 Μοντέλο ανίχνευσης αντικειμένων PointPillars 69
5. Από τα Νέφη Σημείων στις Προβλέψεις: Υλοποίηση ΝΔ για 3Δ Συμπεράσματα Δεδομένων 79
5.1 Ρύθμιση του περιβάλλοντος για τα ΝΔ. Εξερευνώντας τη βιβλιοθήκη Open3D-ML και τα βασικά του στοιχεία 79
5.2 Ανάλυση δεδομένων 80

5.3 Σύνθεση δεδομένων και υπερπαραμέτρων για την εκπαίδευση και αξιολόγηση των μοντέλων 84
5.4 Αξιολόγηση των μεθόδων και αποτελέσματα προβλέψεων 88
5.4.1 Αξιολόγηση μοντέλων 89
5.4.2 Πρόβλεψη σε νέα δεδομένα με χρήση των προεκπαιδευμένων μοντέλων 94
6. Συμπεράσματα 102
Βιβλιογραφία 106

Κατάλογος Εικόνων

Εικόνα 1. Νευρώνας Perceptron του Frank Rosenblatt..... 21
Εικόνα 2. Αποκλειστική διάζευξη..... 21
Εικόνα 3. Δομή Βιολογικού Νευρώνα..... 22
Εικόνα 4. Δομή Τεχνητού Νευρωνικού Δικτύου..... 23
Εικόνα 5. Δυαδικό βήμα συνάρτησης..... 24
Εικόνα 6. Σιγμοειδής καμπύλη και tanh..... 25
Εικόνα 7. ReLU (Rectified Linear Unit)..... 26
Εικόνα 8. Leaky ReLU vs. Parametric ReLU..... 27
Εικόνα 9. Παράδειγμα συνάρτησης ενεργοποίησης Softmax..... 28
Εικόνα 10. Τεχνητό νευρωνικό δίκτυο πρόσθιας τροφοδοσίας..... 29
Εικόνα 11. Περιπτώσεις τακτοποίησης..... 33
Εικόνα 12. Παράγωγος και κυρτότητα συνάρτησης..... 37
Εικόνα 13. Υψηλός και χαμηλός ρυθμός μάθησης..... 37
Εικόνα 14. Σύγκλιση συνάρτησης απώλειας batch gradient descent και stochastic gradient descent. 39
Εικόνα 15. Ρυθμός σύγκλισης batch gradient descent, stochastic gradient descent και mini-bath gradient descent..... 40
Εικόνα 16. Πρόβλημα Vanishing gradient και Exploding gradient..... 41
Εικόνα 17. SGD με και χωρίς Momentum..... 42
Εικόνα 18. Υπόδειγμα οπισθοδιάδοσης..... 48
Εικόνα 19. Convolution και pooling layer..... 49
Εικόνα 20. Υπόδειγμα CNN με φίλτρο f, βήμα s kai padding p..... 50
Εικόνα 21. Σημασιολογική κατάτμηση (αριστερά), ανίχνευση αντικειμένων (δεξιά)..... 51
Εικόνα 22. Διαφορά κοινού CNN με ένα Sparse Conv Net με χρήση φίλτρου..... 53
Εικόνα 23. Υπόδειγμα Point Transformer για Classification (πάνω δεξιά) και Segmentation (κάτω). 54
Εικόνα 24. Υπόδειγμα PAconv (πράσινο) με χρήση σχετικών θέσεων (pi, pk) και προσαρμοστικών πυρήνων K, σε σχέση με την 2Δ συνέλιξη..... 55
Εικόνα 25. Αρχιτεκτονική μοντέλου PV-RCNN..... 56
Εικόνα 26. Αρχιτεκτονική μοντέλου Voxel R-CNN..... 57
Εικόνα 27. Αρχιτεκτονική μοντέλου CenterPoint..... 57
Εικόνα 28. Απεικόνιση του dilated residual block με αύξηση του δεκτικού πεδίου 60
Εικόνα 29. Αρχιτεκτονική RandLA-Net..... 60
Εικόνα 30. Τα μπλε σημεία αποτελούν τα kernel points και με κόκκινο το κέντρο της σφαίρας. Το κεντρικό σημείο επίσης αποτελεί kernel point..... 63
Εικόνα 31. Αναπαράσταση 2Δ kernel points με αντίστοιχες τιμές βαρών..... 63
Εικόνα 32. 2Δ αναπαράσταση deformable kernel..... 64
Εικόνα 33. 2Δ αναπαράσταση KPConv..... 66
Εικόνα 34. Αντιπαραβολή 3Δ σημείων σε 2Δ πυλώνες..... 70
Εικόνα 35. Δίκτυο εξαγωγής χαρακτηριστικών από πυλώνες..... 71
Εικόνα 36. Διαδικασία συνέλιξης και αποσυνέλιξης για εξαγωγή χαρακτηριστικών σε διάφορες κλίμακες..... 73
Εικόνα 37. Κεφαλή ανίχνευσης 3Δ αντικειμένων..... 74
Εικόνα 38. Ρύθμιση αισθητήρων..... 81
Εικόνα 39. Κατανομή ετικετών συνόλου δεδομένων SemanticKITTI..... 82

Εικόνα 40. Μορφή και δομή δεδομένων. 82
Εικόνα 41. Προσεγγιστική Οριοθέτηση τμημάτων του συνόλου των δεδομένων. 83
Εικόνα 42. Υπόδειγμα yml αρχείου. 85
Εικόνα 43. Υπόδειγμα ρύθμισης βαρών κάθε κλάσης. 85
Εικόνα 44. Πίνακας σύγχυσης/ σφαλμάτων 90
Εικόνα 45. Παράδειγμα υλοποίησης διαφορετικών κατωφλίων, για διαφορετικά αντικείμενα κλάσης. 93
Εικόνα 46. Υλοποίησης δοκιμής του RandLA-Net και KPConv σε διαφορετικά υποσύνολο δεδομένων και χρόνοι. 94
Εικόνα 47. Απεικόνιση προβλέψεων RandLA-Net και KPConv σε διαφορετικό δείγμα του SemanticKITTI. 95
Εικόνα 48. Απεικόνιση προβλέψεων RandLA-Net και KPConv στο δείγμα Toronto 3D. 96
Εικόνα 49. Προβλέψεις και σύγκριση με ground truth (σειρά 08 σάρωση 3095). 97
Εικόνα 50. Προβλέψεις και σύγκριση με ground truth (σειρά 08 σάρωση 800). 98
Εικόνα 51. Προβλέψεις (κόκκινο) και σύγκριση με ground truth (πράσινο) στο KITTI. 100

Κατάλογος Πινάκων

Πίνακας 1. Συναρτήσεις απώλειας και περιπτώσεις εφαρμογής. 32
Πίνακας 2. Πίνακας με τις υπερπαραμέτρους μοντέλου RandLA-Net. 62
Πίνακας 3. Πίνακας με τις υπερπαραμέτρους του μοντέλου KPConv. 68
Πίνακας 4. Πίνακας με τις υπερπαραμέτρους του μοντέλου PointPillars 78
Πίνακας 5. Πίνακας υπερπαραμέτρων βελτιστοποίησης και σύγκλισης των δικτύων σημασιολογική κατάτμησης. 87
Πίνακας 6. Πίνακας υπερπαραμέτρων βελτιστοποίησης και σύγκλισης των δικτύων ανίχνευσης αντικειμένων. 88
Πίνακας 7. Ακρίβειες μοντέλων 93
Πίνακας 8. Αξιολόγηση πρόβλεψης για όλες τις κλάσεις και συνολικά. 99
Πίνακας 9. Αξιολόγηση ανίχνευσης αντικειμένων mAP BEV 101
Πίνακας 10. Αξιολόγηση ανίχνευσης αντικειμένων mAP 3D 101

1. Εισαγωγή

Ένα μεγάλο βήμα της επιστημονικής κοινότητας τα τελευταία χρόνια ήταν να μπορεί στο μέγιστο βαθμό να εκμεταλλευτεί τον μεγάλο όγκο ψηφιακών δεδομένων που μπορούσε να αντλήσει από διάφορες πηγές. Το μεγάλο στοίχημα ήταν να βρεθούν τρόποι με τους οποίους θα μπορούσαν να εξαγάγουν τις σημαντικές πληροφορίες από τα δεδομένα για περεταίρω ανάλυση. Αφού βρέθηκαν αυτοί οι τρόποι, στη συνέχεια εμφανίστηκε η ανάγκη πως να εκμεταλλευτούν στο μέγιστο τις πληροφορίες που μπορούσαν να εξάγουν από αυτά τα δεδομένα. Δηλαδή, η ανάγκη να βρεθούν μέθοδοι και μοντέλα που να αναλύουν και ερμηνεύουν με τον καλύτερο δυνατό τρόπο μοτίβα και δομές των δεδομένων. Με την ταυτόχρονη εξέλιξη των υπολογιστών, αυτό είχε ως αποτέλεσμα και της ραγδαίας εξέλιξης της τεχνητής νοημοσύνης (ΤΝ). Η επιχειρηματικότητα εκμεταλλεύτηκε τις τεχνολογίες της ΤΝ, λόγω της προσομοίωσης του ανθρώπινου μυαλού και χρησιμοποιώντας προγράμματα υπολογιστών που είναι ικανά να κατανοήσουν την ανθρώπινη συμπεριφορά και να μαθαίνουν από τα δεδομένα αυτόματα και πιο αποτελεσματικά. Αυτό το γεγονός δείχνει ότι ο υπολογιστής και η ανθρώπινη νοημοσύνη θα έχουν τεράστια και εμφανή επιρροή στους ανθρώπους. Μέχρι τότε, οι υπολογιστές ήταν προγραμματισμένοι τι να σκεφτούν, ενώ τώρα με την εξέλιξη της μηχανικής μάθησης (machine learning) οι υπολογιστές κατασκευάζονται για να αναλύουν, να διαβάζουν δεδομένα, να παρατηρούν, να μαθαίνουν από τα λάθη τους και να παίρνουν αποφάσεις. Αυτό έχει φέρει μια ολόκληρη επανάσταση στην επιστήμη των υπολογιστών και στην καθημερινή μας ζωή. Με τη χρήση απλά ενός έξυπνου κινητού, έχουμε πρόσβαση μέσω εφαρμογών σε διάφορες υπηρεσίες που μας διευκολύνουν καθημερινά. Σχεδόν όλες οι ιστοσελίδες και βιομηχανικοί κλάδοι χρησιμοποιούν ΤΝ για την βελτίωση της αποτελεσματικότητας. Επειδή, πολλές λειτουργίες μπορούν πια να αυτοματοποιηθούν, αυτό έχει ως αποτέλεσμα η διαδικασία παραγωγής να έχει γίνει πιο οικονομική και να απαιτείται πολύ λιγότερος χρόνο. Παρακάτω εξηγούνται τα υποπεδία της τεχνητή νοημοσύνη και μηχανικής μάθησης (machine learning), κάνοντας μια ιστορική αναδρομή και εξερευνώντας τον δρόμο που ακολούθησαν για να φτάσουμε στο σήμερα και πως έχει διαμορφωθεί η κατάσταση.

2. Τεχνητή Νοημοσύνη

Η Τεχνητή Νοημοσύνη ή Artificial Intelligence (AI) στην Αγγλική, είναι ένα ευρύ πεδίο που αναφέρεται στην προσομοίωση της ανθρώπινης νοημοσύνης και των ικανοτήτων επίλυσης προβλημάτων από μηχανές που προγραμματίστηκαν να σκέφτονται και να εκτελούν ποικίλες προηγμένες λειτουργίες. Αυτό περιλαμβάνει εργασίες όπως ο συλλογισμός, η λήψη αποφάσεων, η επίλυση προβλημάτων, η κατανόηση φυσικής γλώσσας και η αναγνώριση προτύπων. Η τεχνητή νοημοσύνη στόχευε στην αυτοματοποίηση των εργασιών για τη βελτίωση της αποδοτικότητας και της παραγωγικότητας σε διάφορους τομείς, όπως τη βιομηχανία και την εφοδιαστική (logistics) έως την υγειονομική περίθαλψη και τα οικονομικά.

Το ταξίδι της ΤΝ ξεκινά στα μέσα του 20ου αιώνα από τον Άλαν Τιούρινγκ που θεωρείται πατέρας των σύγχρονων υπολογιστών, όπου πρότεινε την ιδέα ενός καθολικού μηχανήματος που θα μπορούσε να προσομοιώσει οποιαδήποτε υπολογιστική διαδικασία και τη δυνατότητα των μηχανών να μαθαίνουν από την εμπειρία. Η θεμελιώδης έρευνά του Turing, εισήγαγε την ιδέα ενός μηχανήματος που θα μπορούσε να εκτελέσει καθήκοντα που θεωρούνται ως ανθρώπινη νοημοσύνη. Αυτό διατυπώθηκε μέσω του Turing Test, όπου ένας ανακριτής θα έπρεπε μέσω χρήσης φυσικής γλώσσας να μπορεί να διακρίνει μεταξύ ενός ανθρώπου και μιας μηχανής (Imitation Game). Η συνομιλία γινόταν με γραπτό λόγο, μέσω μιας οθόνης υπολογιστή και πληκτρολογίου ενώ οι συμμετέχοντες βρίσκονταν σε ξεχωριστούς χώρους. Εάν ο ανακριτής δεν κατάφερνε να διακρίνει ποιος από τους δυο είναι η μηχανή, τότε πέρναγε την δοκιμασία (Turing, 1950).

Η επίσημη γέννηση της ΤΝ όμως ως καθορισμένου ακαδημαϊκού πεδίου συνέβη κατά τη διάρκεια του θερινού ερευνητικού προγράμματος Dartmouth (Dartmouth Summer Research Project on Artificial Intelligence) για την τεχνητή νοημοσύνη το 1955, που οργανώθηκε από τους Marvin Minsky, Nathaniel Rochester, Claude Shannon και John McCarthy ο οποίος ήταν ο κύριος διοργανωτής του πρώτου συνεδρίου και ήταν αυτός που επινόησε τον όρο «τεχνητή νοημοσύνη» για πρώτη φορά (McCarthy et al., 2006). Μία από τις πιο σημαντικές συνεισφορές του ήταν η ανάπτυξη της γλώσσας προγραμματισμού LISP, η οποία έγινε μία από τις κύριες γλώσσες για την έρευνα της τεχνητής νοημοσύνης (Mccarthy, 1979). Αυτό το γεγονός κατέλυσε την έρευνα AI, οδηγώντας σε ενθουσιώδεις προβλέψεις για μηχανές που θα μπορούσαν να μιμηθούν την ανθρώπινη σκέψη. Οι Allen Newell και Herbert A. Simon είχαν την ευκαιρία να δείξουν στους συμμετέχοντες το

Logic Theorist τους, ένα πρόγραμμα υπολογιστή που σχεδιάστηκε σκόπιμα για να εκτελεί αυτοματοποιημένους συλλογισμούς, το οποίο αποτελεί και το πρώτο πρόγραμμα τεχνητής νοημοσύνης. Με λίγα λόγια καθιέρωσαν τον ευριστικό (heuristic) προγραμματισμό, ένα σύστημα συμβόλων που θα δημιουργεί και θα τροποποιεί επανειλημμένα γνωστές δομές συμβόλων έως ότου η δημιουργημένη δομή ταιριάζει με τη δομή της λύσης. Κάθε επόμενο βήμα εξαρτάται από το προηγούμενο βήμα, επομένως η ευριστική αναζήτηση μαθαίνει ποιες οδούς πρέπει να ακολουθήσει και ποιες να αγνοήσει μετρώντας πόσο κοντά είναι το τρέχον βήμα στη λύση. Επομένως, ορισμένες πιθανότητες δεν θα δημιουργηθούν ποτέ, καθώς μετράται ότι είναι λιγότερο πιθανό να ολοκληρώσουν τη λύση (Gugerty, 2006).

Αργότερα, υπήρξε μεγάλη άνθηση της ΤΝ όπου η πρόσβαση ήταν πιο εφικτή και όχι πολύ δαπανηρή όπως παλαιότερα. Το διάστημα 1970, ήταν διαδεδομένα τα εξειδικευμένα συστήματα (expert systems), όπου είχαν ως στόχο να αναπαράγουν απαντήσεις σε εξειδικευμένες ερωτήσεις. Από τα πρώτα συστήματα ήταν το ELIZA (Weizenbaum, 1966), όπου αποτελούσε και το πρώτο chatbot στο ψυχοθεραπευτικό πλαίσιο. Ωστόσο, σε σχέση με μεταγενέστερα εξειδικευμένα συστήματα, το ELIZA δεν βασιζόταν σε σαφείς κανόνες ή κάποια δομή πληροφοριών, αλλά σε απλές αντιστοιχίες προτύπων και αντικατάστασης που έδιναν μια ψευδαίσθηση κατανόησης. Το DENDRAL θεωρείται ως το πρώτο εξειδικευμένο σύστημα, όπου στόχος του ήταν η χαρτογράφηση της δομής των οργανικών μορίων, βοηθώντας τους χημικούς στον εντοπισμό άγνωστων ενώσεων με βάση τη φασματική ανάλυση (Lindsay et al., 1993). Ένα άλλο έργο ήταν το MYCIN, το οποίο διήρκεσε από το 1972 έως το 1980. Το MYCIN βοήθησε τους γιατρούς να διαγνώσουν μολυσματικές ασθένειες του αίματος αναλύοντας κλινικά συμπτώματα και παρέχοντας συστάσεις (van Melle, 1978).

Την περίοδο εκείνη ξεκινά ο λεγόμενος χειμώνας της ΤΝ. Αυτό συνέβη επειδή οι αρχικές ανακαλύψεις που είχαν παρουσιαστεί ως τότε, όπως προαναφέρθηκαν, δεν κατάφεραν να ανταποκριθούν στις προσδοκίες που υπήρχαν, με αποτέλεσμα να μην υπάρχει ο ίδιος ενθουσιασμός και να μειώνονται οι επενδύσεις. Αυτό συνέβη για τους λόγους ότι μέχρι τότε οι υπολογιστές ήταν περιορισμένοι ως προς τη δυνατότητα αποθήκευσης μεγάλου όγκου δεδομένων και την ταχύτητα επεξεργασίας τους, τα οποία ήταν ελάχιστα σε σχέση με τους σημερινούς υπολογιστές. Η περίοδος αυτή έληξε το 1996 με το Deep Blue της IBM. Ένα σύστημα υπολογιστή όπου συναγωνίστηκε με τον παγκόσμιο πρωταθλητή σκάκι Γκάρι Κασπάροβ σε αναμέτρηση έξι αγώνων σκάκι. Ο αγώνας έγινε δυο φορές, την πρώτη όπου ο Κασπάροβ κέρδισε με 4-2 και την δεύτερη ένα χρόνο μετά όπου ο σουπερ

υπολογιστής κέρδισε με 3.5-2.5 (δυο νίκες, τρεις ισοπαλίες, μια ήττα). Στον ένα χρόνο διάστημα μέχρι τον επαναληπτικό αγώνα, η ομάδα της IBM εργάστηκε στη βελτίωση της βάσης δεδομένων του Deep Blue και ισχυροποιώντας τις λειτουργίες αξιολόγησης κάθε θέσης πιονιού. Επίσης, προσέλαβαν αυθεντίες σκακιστές για να συμβουλευτούν στρατηγικές. Το αποτέλεσμα ήταν να δημιουργηθεί ένας ισχυρός υπολογιστής με 32 επεξεργαστές που μπορούσε να αξιολογήσει 200 εκατομμύρια θέσεις πιονιού ανά δευτερόλεπτο και ταυτόχρονα μπορούσε να εκτελέσει περίπου 11,38 δισεκατομμύρια αριθμητικές πράξεις κάθε δευτερόλεπτο. Ο αλγόριθμος που χρησιμοποιήθηκε δεν ήταν κάποιος εξελιγμένος μηχανικής μάθησης, αλλά ένας brute force algorithm όπου κάθε πιθανή λύση σε ένα πρόβλημα ελέγχεται μέχρι να βρεθεί η σωστή, ανεξάρτητα από το κόστος υπολογισμού (Campbell et al., 2002).

2.1 Μηχανική Μάθηση

Στην πορεία, άρχισαν να εμφανίζονται λειτουργίες οι οποίες μπορούσαν να μαθαίνουν από τα δεδομένα, χωρίς την ρητή χρήση προγραμματισμού (Samuel, 1959). Αυτό μπορούσε να επιτευχθεί από την αναγνώριση μοτίβων στα δεδομένα, με αποτέλεσμα να μπορούν να κάνουν προβλέψεις και να παίρνουν αποφάσεις αυτόνομα. Μαθαίνοντας από τα διαθέσιμα δεδομένα, η πρόβλεψη και ταξινόμηση γίνονταν ολοένα και πιο ακριβείς με την πάροδο του χρόνου επεκτείνοντας τις ικανότητες των μηχανών, που μέχρι πριν από λίγα χρόνια μόνο ο άνθρωπος μπορούσε να κάνει. Δηλαδή, πέραν της δυνατότητας να μαθαίνουν από την εμπειρία (Mitchell, n.d.), κατάφεραν και να προσαρμόζονται σε νέα εισαγωγή δεδομένων και να εκτελούν εργασίες παρόμοιες με τον άνθρωπο πιο αποτελεσματικά. Στη μηχανική μάθηση (machine learning) διακρίνονται τέσσερεις μέθοδοι:

- Επιβλεπόμενη μάθηση (supervised learning)
- Μη επιβλεπόμενη μάθηση (Unsupervised learning)
- Ημιεπιβλεπόμενη (Semi-Supervised learning)
- Ενισχυτική μάθηση (Reinforcement learning)

2.1.1 Επιβλεπόμενη μάθηση

Η πιο κοινή μέθοδος που χρησιμοποιείται πιο συχνά στη μηχανική μάθηση λόγο της ευκολίας να εφαρμοστεί και επειδή ασχολείται με απλές εργασίες. Ένα πρόβλημα χαρακτηρίζεται ως επιβλεπόμενη μάθηση όταν δίνονται τα δεδομένα εισόδου - εξόδου και ο αλγόριθμος μαθαίνει να χαρτογραφεί συσχετισμούς μεταξύ τους. Δηλαδή, προσπαθεί ο αλγόριθμος να βρει μοτίβα μεταξύ των δεδομένων εισόδου και των αντίστοιχων εξόδου, ώστε να βρεθεί ένα μοντέλο που θα προσαρμόζεται στα δεδομένα όσο το δυνατό καλύτερα. Για παράδειγμα, ένας αλγόριθμος που αποφασίζει αν μια εικόνα αφορά σκύλο ή γάτα, έχει δοθεί αρχικά ένα σύνολο εικόνων που αφορούν γάτες και σκύλους και μαζί με τους αντίστοιχους χαρακτηρισμούς. Όποτε, μέσα από αυτό το σύνολο δεδομένων εισόδου-εξόδου μαθαίνει τις συσχετίσεις μεταξύ τους. Αφού έχει εκπαιδευτεί μπορεί πλέον να κάνει προβλέψεις σε νέα δεδομένα εισόδου. Αυτό θεωρείται και ως αλγόριθμος ταξινόμησης (classification), μια από τις κύριες μεθόδους επιβλεπόμενης μάθησης. Η άλλη κύρια μέθοδος αφορά την παλινδρόμηση (regression), όπου ο αλγόριθμος μαθαίνει από τα δεδομένα να προβλέπει συνεχείς τιμές, όπως οι προβλέψεις τιμών ακινήτων με βάση τα χαρακτηριστικά του κάθε ακινήτου. Συγκεκριμένα, στοχεύει στη λειτουργική σχέση μεταξύ ανεξάρτητων μεταβλητών και μιας εξαρτημένης. Στο προηγούμενο παράδειγμα η ανεξάρτητες μεταβλητές θα μπορούσαν να είναι το μέγεθος του ακινήτου, ο όροφος, η παλαιότητα, η τοποθεσία κ.ά., ενώ η εξαρτημένη μεταβλητή θα ήταν η τιμή του. Στόχος της μεθόδου αυτής είναι ή ελαχιστοποίηση της διαφοράς μεταξύ προβλεπόμενης και πραγματικής τιμής. Οι αλγόριθμοι που χρησιμοποιούνται σε αυτές τις περιπτώσεις ποικίλουν και ανάλογα επιλέγεται ο ποιο κατάλληλος.

- Γραμμική παλινδρόμηση (**linear regression**): είναι ένας από τους απλούστερους και πιο διαδεδομένη μέθοδος, όπου ο αλγόριθμος προσπαθεί να βρει μια γραμμική σχέση μεταξύ των δεδομένων εισόδου και της τιμής εξόδου (Stanton, 2001).
- Λογιστική παλινδρόμηση (**logistic regression**): χρησιμοποιείται για πρόβλεψη δυαδικής τιμής εξόδου, όπου η τιμή μπορεί να είναι αληθής ή ψευδείς. Και εδώ ο αλγόριθμος προσπαθεί να βρει μια γραμμική σχέση μεταξύ δεδομένων εισόδου και τιμής εξόδου. Η τιμή εξόδου μετασχηματίζεται μέσω μιας λογιστικής συνάρτησης για να παράγει μια τιμή πιθανότητας μεταξύ των τιμών 0 και 1 (Cramer, 2003).
- Δέντρα αποφάσεων (**Decision Tree**): μια δενδροειδής δομή αποφάσεων όπου κάθε εσωτερικός κόμβος αντιπροσωπεύει μια απόφαση και κάθε κόμβος φύλλο

(leaf node) αντιπροσωπεύει ένα πιθανό αποτέλεσμα (Rokach & Maimon, 2005). Συνήθως χρησιμοποιούνται όταν η σχέση μεταξύ τιμών εισόδου και εξόδου είναι πιο σύνθετη. Αυτός ο τύπος αλγορίθμου χρησιμοποιείται για εργασίες ταξινόμησης και παλινδρόμησης.

- **Random Forest**: στη συγκεκριμένη περίπτωση δημιουργείται ένα πλήθος δέντρων αποφάσεων όπου περιέχουν ένα τυχαίο υποσύνολο του συνόλου των δεδομένων και ένα υποσύνολο χαρακτηριστικών. Αυτή η τυχαιότητα δημιουργεί με μεταβλητότητα μεταξύ των δέντρων όπου μειώνει την υπερπροσαρμογή του μοντέλου και βελτιώνει τη συνολική απόδοση πρόβλεψης. Στη περίπτωση ταξινόμησης ο αλγόριθμος συγκεντρώνει τα αποτελέσματα όλων των δέντρων και αποφασίζει με βάση ψήφους, ενώ στη παλινδρόμηση με βάση τον μέσο όρο (Breiman, 2001). Η μέθοδος αυτή μπορεί να χειρίζεται πολύπλοκα δεδομένα και παρέχουν αξιόπιστες προβλέψεις.
- **Support Vector Machine** (SVM): στόχος του συγκεκριμένου αλγορίθμου είναι να βρεθεί το βέλτιστο υπερεπίπεδο (hyperplane) σε ένα χώρο *N* διαστάσεων που μπορεί να διαχωρίσει τα δεδομένα στις διαφορετικές κατηγορίες χαρακτηριστικών. Στην πραγματικότητα προσπαθεί το υπερεπίπεδο να μεγιστοποιήσει την απόσταση του από τα κοντινότερα σημεία των διαφορετικών κατηγοριών. Αυτό έχει ως αποτέλεσμα ένα ισχυρό μοντέλο όπου δεν επηρεάζεται από θόρυβο. Ο βαθμός του υπερεπίπεδου προσδιορίζεται με βάση το πλήθος που θέλουμε να εντοπίσουμε μείον ένα (Cortes & Vapnik, 1995). Εάν για παράδειγμα έχουμε δυο χαρακτηριστικά, τότε το υπερεπίπεδο θα τα χωρίζει με μια γραμμή. Για τρία χαρακτηριστικά τότε το υπερεπίπεδο γίνεται ένα δισδιάστατο επίπεδο κλπ.
- **K-Nearest Neighbors** (KNN): πρόκειται για ένα ευρύ διαδεδομένο απλό αλγόριθμο λόγω του ότι είναι μη παραμετρικός, δηλαδή δεν κάνει υποθέσεις σχετικά με την κατανομή των δεδομένων και δεν χρειάζεται εκπαίδευση. Αφού ο αλγόριθμος αποθηκεύσει τα δεδομένα και τις κατηγορίες χαρακτηρισμών, μπορεί να παίρνει νέο άγνωστο δεδομένο και με τη χρήση κάποιας μετρικής απόστασης, να υπολογίζει πόσο απέχει από τα *K* κοντινότερα δεδομένα. Στη ταξινόμηση, εκεί που γειτνιάζει με τα περισσότερα *K* δεδομένα θα λάβει και τον αντίστοιχο χαρακτήρα. Δηλαδή, αν *K=1* τότε το νέο δεδομένο εκχωρείται στον χαρακτηρισμό του κοντινότερου σημείου δεδομένου. Στη παλινδρόμηση, εκχωρείται η μέση τιμή

των *K* κοντινότερων σημείων. Υπάρχει και ο σταθμισμένος KNN όπου στην ίδια λογική προστίθενται βάρος ανάλογα με την απόσταση (Cover & Hart, 1967).

- **Gradient Boosting**: επίσης αλγόριθμος που χρησιμοποιείται για ταξινόμηση και παλινδρόμηση. Πρόκειται για μια μέθοδο συνόλου όπου η εκπαίδευση του μοντέλου γίνεται διαδοχικά και κάθε νέο μοντέλο διορθώνει το προηγούμενο του. Δηλαδή, συνδυάζει ανίσχυρες εκπαιδεύσεις και ισχυρές με χρήση δέντρων αποφάσεων. Σκοπός είναι να ελαχιστοποιήσει το συνολικό σφάλμα προβλέψεων με το συνδυασμό των ανίσχυρων εκπαιδευμένων μοντέλων. Η διαδικασία ξεκινάει με ίση κατανομή βαρών σε όλα τα δεδομένα και αφού γίνει η πρώτη εκπαίδευση, θα υπάρξουν κάποια εσφαλμένα δεδομένα ταξινόμησης. Στη συνέχεια τα βάρη των σφαλμάτων τροφοδοτούνται στο επόμενο μοντέλο όπου εκεί γίνεται διόρθωση τους και ενημερώνονται. Η διαδικασία αυτή συνεχίζεται μέχρι να ελαχιστοποιηθούν τα σφάλματα. Όταν έρθει νέο δεδομένο, θα περάσει από όλα τα στάδια και η κατηγορία με το υψηλότερο ποσοστό είναι η τιμή που θα πάρει (Friedman, 2001).

Τα θετικά της επιβλεπόμενης μάθησης είναι ότι χρησιμοποιεί χαρακτηρισμένα δεδομένα όπου καθιστά μια σαφήνεια σε ποια έξοδο αντιστοιχεί κάθε δεδομένο. Επίσης, είναι πιο εύκολη η εκπαίδευση μοντέλων όταν είναι γνωστή η σωστή απάντηση. Μειονεκτήματα της είναι ότι χρειάζεται χαρακτηρισμένα δεδομένα το όποιο μπορεί να είναι δαπανηρό και χρονοβόρο. Επίσης, τα δεδομένα πολλές φορές μπορεί να κουβαλάνε και ένα είδος σφάλμα (bias) το οποίο θα το κληρονομεί και το μοντέλο εκπαίδευσης.

2.1.2 Μη Επιβλεπόμενη μάθηση

Σε αντίθεση με την επιβλεπόμενη, τώρα τα δεδομένα δεν έχουν κάποιο χαρακτηρισμό κάποιας κατηγορίας. Καλούνται να εντοπίσουν μοτίβα και σχέσεις μεταξύ των δεδομένων. Τέτοιες μέθοδοι είναι ισχυρές για ανάλυση ώστε να κατανοηθεί μια δομή δεομένων. Στα μοντέλα εδώ δεν δίνονται τιμές εξόδου, αλλά μόνο παραμέτρους εισόδου και ανακαλύπτει από μόνο του τις κατηγορίες και ομάδες. Στη μη επιβλεπόμενη μάθηση συναντάμε τρεις κύριες εργασίες:

- Ομαδοποίηση (**Clustering**): πρόκειται για τεχνική που όπως δηλώνει και το όνομα, ομαδοποιεί τα δεδομένα, με βάση τις ομοιότητες τους. Δηλαδή, χωρίζει τα μη κατηγοριοποιημένα δεδομένα σε ομάδες βάση ομοιοτήτων στη δομή τους. Υπάρχουν τέσσερις μέθοδοι ομαδοποίησης όπου περιλαμβάνουν την αποκλειστική, την ιεραρχική, την επικαλυπτόμενη και την πιθανολογική. Στην αποκλειστική τα δεδομένα μπορούν να ανήκουν μόνο σε μια ομάδα. Η κατηγοριοποίηση αυτή γίνεται με τη τεχνική K-means όπου τα δεδομένα εκχωρούνται σε Κ ομάδες και αναφέρεται στο σύνολο των ομάδων που θα αντιστοιχηθούν με βάση την απόσταση από το κεντροειδές της κάθε ομάδας (Macqueen, 1967). Από την άλλη υπάρχει και η επικαλυπτόμενη όπου εκεί κάποια δεδομένα μπορούν να υπάρχουν σε πολλές ομάδες, με ξεχωριστό βαθμό συμμετοχής. Στην ιεραρχική, τα δεδομένα αρχικά απομονώνονται σε ξεχωριστές ομάδες και στη συνέχεια συγχωνεύονται επαναληπτικά με βάση τις ομοιότητες, έως ότου επιτευχθεί η ομάδα. Στη πιθανολογική διαφέρουν λίγο τα πράγματα, καθώς δεν χρησιμοποιείται η ομοιότητα που παρουσιάζουν τα δεδομένων, αλλά από τη πιθανότητα να ανήκει ένα δεδομένο στην αντίστοιχη ομάδα. Αυτό μπορεί να επιτευχθεί μέσω της κανονικής κατανομής Gauss (Gaussian Mixture Model) όπου εάν ο μέσος όρος ή διακύμανση είναι γνωστά, τότε μπορεί να προσδιοριστεί σε ποια κατανομή ανήκει ένα δεδομένο σημείο.
- Κανόνες συσχέτισης (**Association Rules**): είναι μια μέθοδος που ανακαλύπτει συσχετίσεις ανάμεσα στα δεδομένα. Για παράδειγμα για τα ψώνια ενός καταστήματος θα μπορούσε να φανεί η σχέση μεταξύ της πώλησης ενός προϊόντος με τη πώληση ενός άλλου.
- Μείωση διαστάσεων (**Dimensionality reduction**): τεχνική που χρησιμεύει στη μύωση χαρακτηριστικών ή διαστάσεων σε ένα σύνολο δεδομένων, διατηρώντας τη σημαντική πληροφορία. Χρησιμοποιείται στη προεπεξεργασία των δεδομένων για αντιμετώπιση προβλημάτων όπως υπερπροσαρμογής αλγορίθμου και μείωση χρόνου υπολογισμών. Κύρια μέθοδος που χρησιμοποιείται είναι Principal Component Analysis (PCA) όπου παρουσιάστηκε από τον μαθηματικό Καρλ Πίρσον το 1901. Είναι ένας ορθογώνιος μετασχηματισμός συσχετισμένων μεταβλητών σε ένα σύνολο γραμμικά ασυσχέτιστων μεταβλητών κύριων στοιχείων (principal components). Το πρώτο principle component είναι αυτό που παρουσιάζει τη μεγαλύτερη διακύμανση στα δεδομένα, το δεύτερο το αμέσως επόμενο και πάει

λέγοντας. Λόγο της ορθογωνικότητας τα principle components δεν συσχετίζονται μεταξύ τους (Pearson, 1901).

Τα πλεονεκτήματα της μη επιβλεπόμενης μάθησης είναι ότι δεν χρειάζονται χαρακτηρισμένα δεδομένα που μπορεί να είναι χρονοβόρα και να έχουν κάποιο κόστος. Επίσης, μπορούν να εντοπίσουν μοτίβα μεταξύ των δεδομένων, τα οποία μπορεί να μην είναι αντιληπτά από τον άνθρωπο. Στα μειονεκτήματα είναι ότι δύσκολα μπορεί να αξιολογηθεί η απόδοση τους, αφού δεν υπάρχουν προκαθορισμένα δεδομένα στα οποία μπορεί να γίνει η σύγκρισή των αποτελεσμάτων. Σε περίπτωση που τα δεδομένα έχουν θόρυβο ή είναι ελλιπή τότε μπορεί να παραπλανηθεί το μοντέλο και να καταλήξει σε ανακριβή αποτελέσματα.

2.1.3 Λοιποί μέθοδοι μηχανικής μάθησης

Υπάρχουν και άλλοι μέθοδοι μηχανικής μάθησης οι οποίοι δεν είναι το ίδιο διαδεδομένοι με τις προηγούμενες δυο. Η πρώτη ονομάζεται ενισχυτική μάθηση (**reinforcement learning**) όπου είναι ένα αυτοδίδακτο σύστημα που μαθαίνει από δοκιμές και λάθη. Εκτελεί ενέργειες ώστε με στόχο να πετύχει το καλύτερο αποτέλεσμα. Σε αντίθεση με την επιβλεπόμενη μάθηση που υπάρχει η απάντηση και το μοντέλο μπορεί να εκπαιδευτεί από αυτό, στη ενισχυτική δεν υπάρχει και πρέπει να αποφασίσει τι θα πρέπει να κάνει για να εκτελέσει μια εργασία. Λόγο της έλλειψης δεδομένων, θα μαθαίνει από εμπειρία. Δηλαδή, από τα αποτελέσματα θα αποφασίζει ποια ενέργεια θα ακολουθήσει. Μετά από κάθε ενέργεια, ο αλγόριθμος ανατροφοδοτείται για να καταλάβει αν η απόφαση που πήρε ήταν σωστή, ουδέτερη ή λανθασμένη. Είναι καλή τεχνική για αυτόματα συστήματα που λαμβάνουν πολλές μικρές αποφάσεις χωρίς ανθρώπινη καθοδήγηση. Είναι κατάλληλες για επίλυση σύνθετων προβλημάτων που δεν μπορούν να επιλυθούν από συμβατικές τεχνικές. Τα δεδομένα εκπαίδευσης συλλέγονται από την άμεση αλληλεπίδραση με το περιβάλλον. Αντίθετα, η ενισχυτική μάθηση χρειάζεται πολλά δεδομένα και πολλούς υπολογισμούς. Επίσης, εξαρτάται σε μεγάλο βαθμό από την ποιότητα της συνάρτησης ανταμοιβής, όπου σε περίπτωση κακού σχεδιασμού ενδέχεται να μάθει την επιθυμητή συμπεριφορά. Γενικά είναι δύσκολα να ερμηνευτεί, γεγονός που τη καθιστά δύσκολη τη διάγνωση και την επίλυση προβλημάτων.

Άλλη μέθοδος είναι η ημί-επιβλεπόμενη μάθηση (**semi-supervised learning**) όπου είναι μια μέθοδος που βρίσκεται ανάμεσα στην επιβλεπόμενη και μη επιβλεπόμενη μάθηση, καθώς περιέχει ένα μικρό πλήθος χαρακτηρισμένων δεδομένων και ένα μεγάλο μη χαρακτηρισμένων δεδομένα. Στη συγκεκριμένη περίπτωση τα χαρακτηρισμένα δεδομένα είναι αυτά που καθοδηγούν την εκπαίδευση του μοντέλου για το χαρακτηρισμό του υπόλοιπου αγνώστου συνόλου. Ένα τέτοιο μοντέλο μπορεί χρησιμοποιεί μη επιβλεπόμενη μάθηση για να εντοπίσει πιθανές ομάδες (clusters) και αργότερα μέσω επιβλεπόμενης μάθησης να χαρακτηρίσει τα δεδομένα αντιστοίχως. Αυτό καταλαβαίνει κανείς ότι ήρθε για να συμπληρώσει τις δυο αυτές μεθόδους ώστε να μπορεί να αξιολογηθεί το μοντέλο έχοντας γνωστές τιμές. Αλλά και αυτές οι τιμές να περιορίζονται αρκετά σε σχέση με την επιβλεπόμενη μάθηση, όπου δεν χρειάζεται να είναι χαρακτηρισμένο το σύνολο των δεδομένων.

Τέτοιες εργασίες μηχανικής μάθησης παρατηρείται να υπάρχουν σε πολλούς τομείς και κλάδους συμπεριλαμβανομένων:

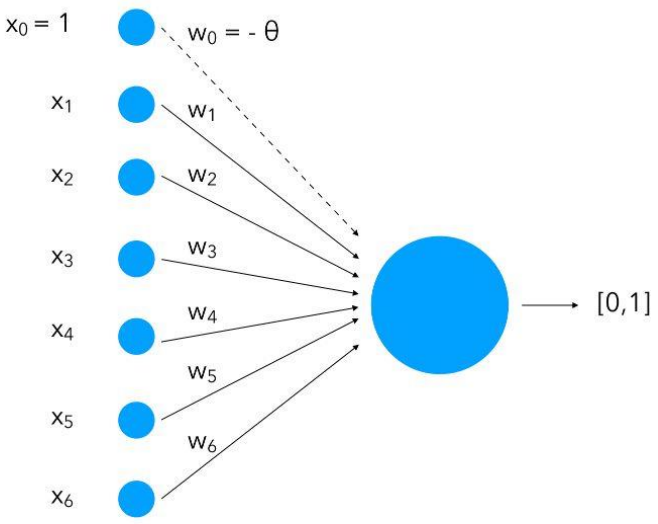
- **Υγειονομική Περίθαλψη,** όπου βελτιώνει την ακρίβεια των διαγνώσεων, βοηθά στις χειρουργικές επεμβάσεις, και προσωποποιεί τη φροντίδα των ασθενών μέσω της προγνωστικής ανάλυσης.
- **Οικονομικά**, όπου χρησιμοποιούνται για υψηλής συχνότητας συναλλαγές, ανίχνευση απάτης και αυτοματοποίηση της εξυπηρέτησης πελατών, ενισχύοντας την αποδοτικότητα και την ασφάλεια.
- **Αυτοκίνηση**, όπου εταιρείες πρωτοπορούν στην ανάπτυξη οχημάτων που μπορούν να πλοηγηθούν σε περίπλοκα περιβάλλοντα με ελάχιστη ανθρώπινη επίβλεψη.
- **Λιανικό εμπόριο**, όπου χρειάζεται στη διαχείριση των αποθεμάτων και στη βελτιστοποίηση των στρατηγικών εμπορευματοποίησης, στη εξυπηρέτηση πελατών με εικονικούς βοηθούς που μπορούν να χειριστούν ερωτήσεις των πελατών, παρέχοντας γρήγορες απαντήσεις
- **Βιομηχανία**, όπου αυτοματοποιούνται πολύπλοκες διαδικασίες που μπορούν να βελτιώσουν την αποδοτικότητα, να μειώσουν το κόστος και να αυξήσουν τα ποσοστά παραγωγής.

3. Εισαγωγή Νευρωνικών Δικτύων

Ένα νευρωνικό δίκτυο είναι ένας τύπος μηχανικής μάθησης, που ονομάζεται βαθιά μάθηση και μιμείται τον τρόπο με τον οποίο οι βιολογικοί νευρώνες συνεργάζονται. Χρησιμοποιεί διασυνδέσεις νευρώνων σε μια πολυεπίπεδη δομή που μοιάζει με τον ανθρώπινο εγκέφαλο. Η πρώτη αναφορά στο όρο νευρώνας έγινε από τον νευροφυσιολόγο Warren McCulloch και τον μαθηματικό Walter Pitts, απ' όπου πήρε και το όνομα του (McCulloch-Pitts neuron). Η ερμηνεία ήταν ότι οι νευρώνες ενεργοποιούνται όταν η είσοδος ξεπερνούσε τη τιμή ενός κατωφλίου και ότι αυτό μπορούσε να εκφραστεί χρησιμοποιώντας λογικούς κανόνες όπως στα μαθηματικά. Υποστήριξαν ότι διαφορετικές υποθέσεις των νευρώνων μπορούσαν και πάλι να καταλήξουν στην ίδια συμπεριφορά του δικτύου (McCulloch & Pitts, 1943). Αργότερα έγινε γνωστό το Hebbian Learning από τον ψυχολόγο Donald Hebb, όπου εξήγησε πως τα συναπτικά βάρη θα προσαρμοστούν ανάλογα με τη δραστηριότητα των νευρώνων, τα οποία στο προηγούμενο μοντέλο ήταν σταθερά και προκαθορισμένα. Δηλαδή, όταν δυο νευρώνες είναι ταυτόχρονα ενεργά, τα συναπτικά βάρη τους αυξάνονται, ενώ όταν ένας νευρώνας αναστέλλει τον άλλον, τότε τα συναπτικά βάρη μειώνονται (Hebb, 2005). Αυτός ο νευρώνας είχε πολλούς περιορισμούς όπως το γεγονός ότι δεχόταν μόνο δυαδικές τιμές εισόδου, χρησιμοποιούσε μια απλή συνάρτηση κατωφλίου και το μοντέλο δεν μπορούσε να μάθει ή να προσαρμόσει τις παραμέτρους του. Ήταν πιο πολύ ένα θεωρητικό μοντέλο λειτουργίας ενός νευρώνα. Το πρώτο νευρωνικό δίκτυο που δημιουργήθηκε και έδινε λύση στα προηγούμενα προβλήματα, ήταν από τον Frank Rosenblatt όπου εισήγαγε τον νευρώνα Perceptron, που αποτελούνταν από ένα επίπεδο νευρώνων. Κάθε νευρώνας υπολόγιζε ένα σταθμισμένο άθροισμα των εισόδων και εφάρμοζε μια συνάρτηση ενεργοποίησης για να παράγει μια έξοδο (Rosenblatt, 1958). Στο στάδιο υπολογισμού του συγκεκριμένου μοντέλου, πρώτα προστίθεται ένα βάρος w_i για κάθε δεδομένο εισόδου και μετά υπολογίζεται το άθροισμα του συνόλου των δεδομένων $\sum_{i=1}^{n} w_i x_i$. Στο δεύτερο βήμα το αποτέλεσμα τροφοδοτείται σε μια μοναδιαία βηματική συνάρτηση (Heaviside step function) ενεργοποίησης, που στη συγκεκριμένη περίπτωση είναι μια απλή συνάρτηση βήματος που μπορεί να αντιστοιχήσει την τιμή εισόδου, σε τιμή εξόδου *0* ή *1*. Στο άθροισμα αυτό προστίθεται και μια μεροληψία (bias) που αποτελεί ένα συστηματικό σφάλμα λόγω λανθασμένων υποθέσεων κατά τη διάρκεια της μάθησης του μοντέλου.

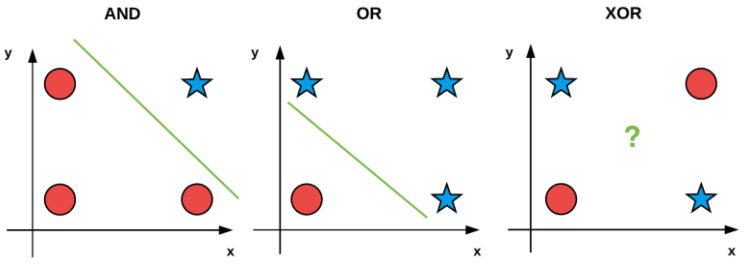
$$\boldsymbol{y = 1}, \text{εάν } \boldsymbol{\sum_i w_i x_i \geq 0} \qquad \textbf{(3.1)}$$

$$\boldsymbol{y = 0}, \text{εάν } \boldsymbol{\sum_i w_i x_i < 0} \qquad \textbf{(3.2)}$$



Εικόνα 1. Νευρώνας Perceptron του Frank Rosenblatt.

Αργότερα όμως, δημοσιεύτηκε ένα βιβλίο που ανέφερε τις ικανότητες και περιορισμούς των συγκεκριμένων νευρωνικών δικτύων. Οι Marvin Minsky και Seymour Papert αναφέρουν στο βιβλίο τους ότι ένα μονοδιάστατο επίπεδο νευρώνων δεν μπορούσε να δώσει λύσεις στο λογικό πρόβλημα XOR (αποκλειστικό OR). Το συγκεκριμένο πρόβλημα αφορά ταξινόμηση δυαδικών ζευγών εισόδου σε δυο κατηγορίες (*0-1* ή αρνητικές-θετικές). Η πρόκληση ήταν στην εύρεση ενός ορίου που θα διαχώριζε τα θετικά από τα αρνητικά. Το μονό επίπεδο νευρώνων με το γραμμικό όριο απόφασης, δεν μπορούσε να διαχειριστεί τέτοιου είδους μη γραμμικών προβλημάτων όπως είναι το XOR (Minsky & Papert, 2017).

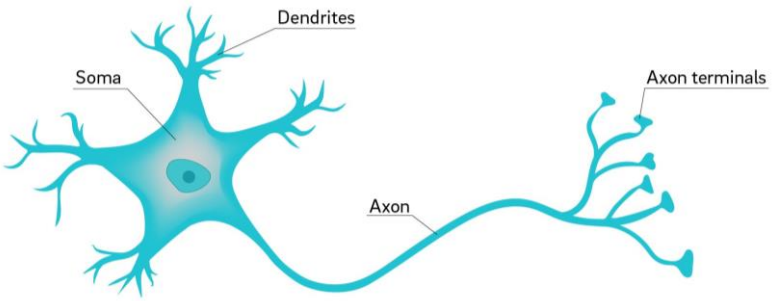


Εικόνα 2. Αποκλειστική διάζευξη.

Εκείνη τη περίοδο ξεκινά και ο πρώτος χειμώνας AI που αναφέρθηκε και στο προηγούμενο κεφάλαιο. Μια επίσης σημαντική διαφορά και μεγάλη έμπνευση που άνοιξε το δρόμο στη μεταγενέστερη εξέλιξη αλγόριθμων μάθησης, ήταν η χρήση μιας συνεχής συνάρτησης ως συνάρτηση ενεργοποίησης (activation function), όπου σε σχέση με τη βηματική συνάρτηση, που αποτελεί μια απλοποίηση της πυροδότησης των νευρώνων, επιτρέπουν μια πιο διαφοροποιημένη έξοδο και πλουσιότερες αναπαραστάσεις δεδομένων. Η λύση στο πρόβλημα έρχεται αργότερα με την είσοδο των πολυεπίπεδων νευρώνων Multilayer Perceptron (MLP), που δίνουν λύση στα μη γραμμικά προβλήματα και της οπισθοδιάδοσης (backpropagation), ενός αλγόριθμου εκπαίδευσης των βαρών του MLP.

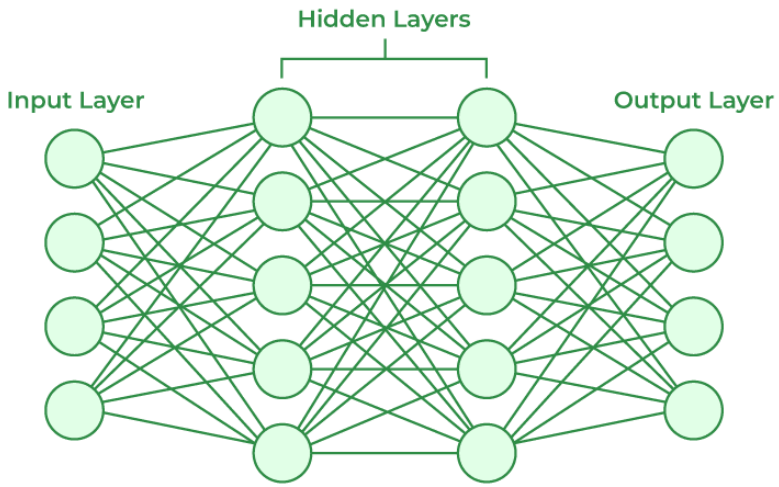
3.1 Δομή Τεχνητών Νευρωνικών Δικτύων

Όπως προαναφέρθηκε στη προηγούμενη ενότητα, ένα νευρωνικό δίκτυο μιμείται τον ανθρώπινο εγκέφαλο. Δηλαδή, νευρικά κύτταρα που είναι πρωταρχικές μονάδες τόσο του εγκεφάλου όσο και του νευρωνικού συστήματος. Αυτά τα νευρωνικά κύτταρα λαμβάνουν πληροφορίες μέσω αισθητήρων από εξωτερικούς παράγοντες, οι οποίες επεξεργάζονται και δίνουν μια έξοδο όπου με τη σειρά της μπορεί να λειτουργήσει ως είσοδος σε έναν επόμενο νευρώνα. Κάθε νευρώνας αποτελείται από ένα κυτταρικό σώμα, έναν αριθμό από δενδρίτες, που είναι οι δίοδοι εισόδου πληροφοριών στο σώμα του κυττάρου και έναν άξονα όπου είναι η έξοδος της πληροφορίας που μεταφέρει το κύτταρο. Οι δενδρίτες συνδέουν το σώμα του κυττάρου με άλλα νευρωνικά κύτταρα. Όταν τα νευρωνικά κύτταρα λαμβάνουν ή εκπέμπουν κάποια πληροφορία, μεταδίδουν ηλεκτρικά ερεθίσματα κατά μήκος του άξονα που βοηθούν στην εκτέλεση λειτουργιών.



Εικόνα 3. Δομή Βιολογικού Νευρώνα.

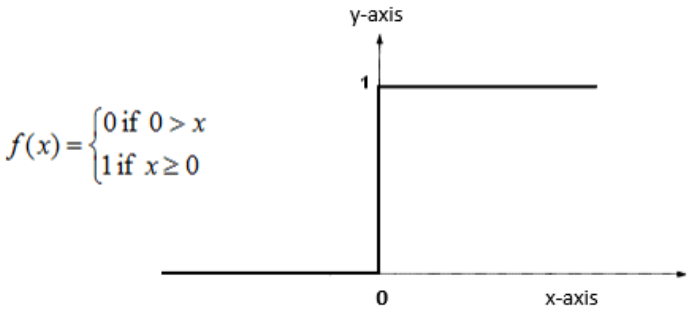
Με έναν αντίστοιχο τρόπο λειτουργούν και τα τεχνητά νευρωνικά δίκτυα. Τα νευρωνικά δίκτυα αποτελούνται από επίπεδα διασυνδεδεμένων νευρώνων (κόμβων) σε τρεις κύριους τύπους. Αρχικά είναι το επίπεδο εισόδου όπου αποτελείται από κόμβους που αντιπροσωπεύουν τα χαρακτηριστικά των δεδομένων εισόδου και σε κάθε κόμβο αντιστοιχείται η τιμή ενός χαρακτηριστικού. Το επίπεδο εξόδου είναι το στάδιο όπου γίνεται η τελική πρόβλεψη του μοντέλου. Ο αριθμός των κόμβων εξόδου μπορεί να διαφέρει σε κάθε περίπτωση, ανάλογα τη φύση του προβλήματος. Για ένα πρόβλημα δυαδικής ταξινόμησης θα υπάρχει ένας κόμβος για κάθε κλάση και θα εξάγει μια πιθανότητα. Ανάμεσα στα επίπεδο εισόδου και εξόδου μπορεί να υπάρχουν ένα ή περισσότερα κρυφά επίπεδα όπου εκεί γίνονται οι υπολογισμοί των παραμέτρων του νευρωνικού δικτύου. Τα δίκτυα που έχουν ένα ή παραπάνω κρυφά επίπεδα ονομάζονται multilayer perceptrons (MLP) και έχουν την ιδιότητα ότι όλοι οι κόμβοι είναι πλήρη συνδεδεμένοι μεταξύ τους (fully connected). Κάθε κόμβος στα κρυφά επίπεδα λαμβάνει ως είσοδο τιμές όλων των κόμβων του προηγούμενου επιπέδου, εφαρμόζει ένα σταθμισμένο μέσο όρο όπως αναφέρθηκε προηγούμενος, προστίθεται μια μεροληψία και προωθεί το αποτέλεσμα μέσω μιας συνάρτησης ενεργοποίησης.



Εικόνα 4. Δομή Τεχνητού Νευρωνικού Δικτύου.

3.2 Συναρτήσεις Ενεργοποίησης

Οι συναρτήσεις ενεργοποίησης (activation functions) έλυσαν το πρόβλημα μοντελοποίησης πολύπλοκών μη γραμμικών σχέσεων. Χωρίς αυτά ένα νευρωνικό δίκτυο δεν θα μπορούσε να ανακαλύψει πολύπλοκα μοτίβα στα δεδομένα και θα συμπεριφερόταν όπως ένα απλό μοντέλο γραμμικής παλινδρόμησης. Μια συνάρτηση ενεργοποίησης μπορεί να καθορίσει αν ένας κόμβος πρέπει να ενεργοποιηθεί, με βάση το σταθμισμένο άθροισμα των εισόδων και μιας μεροληψίας. Η πιο απλή περίπτωση είναι η γραμμική συνάρτηση ενεργοποίησης $f(x) = x$, όπου στην ουσία δεν γίνεται κάποια διεργασία και απλά περνά τη τιμή του σταθμισμένου αθροίσματος. Προηγουμένως εξετάσθηκε η απλή μορφή της συνάρτησης βήματος, που ενεργοποιείται όταν η τιμή ξεπερνά μια τιμή κατωφλίου. Όμως αυτή η μέθοδος παρουσιάζει πρόβλημα σε περιπτώσεις ταξινόμησης, όπου μπορεί να υπάρχουν πολλαπλές έξοδοι.



Εικόνα 5. Δυαδικό βήμα συνάρτησης.

Η σιγμοειδής συνάρτηση, γνωστή και ως λογιστική συνάρτηση, ήταν η πρώτη που χρησιμοποιήθηκε στα νευρωνικά δίκτυα και χρησιμοποιείται ακόμα. Αυτό διότι αντιστοιχεί οποιαδήποτε είσοδο πραγματικής τιμής στο εύρος *0* έως *1*. Όπως αναδεικνύει και το όνομα της, είναι μια καμπύλη σχήματος *S*, όπου γίνεται πιο επίπεδη όσο πλησιάζει τις ελάχιστες και τις μέγιστες τιμές της. Οι τιμές της υπολογίζονται από τον ακόλουθο τύπο:

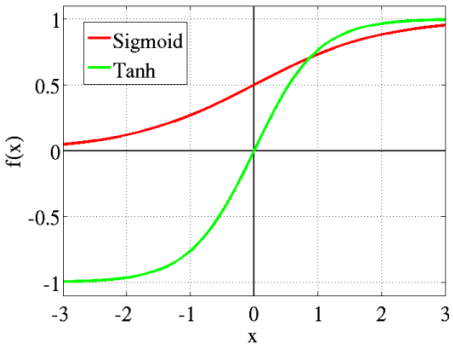
$$\boldsymbol{\sigma(x) = \frac{1}{1 + e^{-x}}} \qquad \textbf{(3.3)}$$

Η είσοδος (τιμή *x*) όταν είναι μεγάλη, ο εκθετικός όρος στο παρονομαστή μικραίνει πλησιάζοντας το *0*, με αποτέλεσμα η συνάρτηση να προσεγγίζει την μονάδα. Αντιθέτως, για μικρό *x*, ο εκθετικός όρος μεγαλώνει πλησιάζοντας τη μονάδα, με αποτέλεσμα η συνάρτηση να προσεγγίζει το *0*. Η συνάρτηση επίσης είναι παραγωγίσιμη και έχει ομαλή κλίση, αποτρέποντας μεγάλα άλματα στις τιμές εξόδου. Ο υπολογισμός της παραγώγου είναι:

$$\boldsymbol{\sigma'(x) = \sigma(\chi) \cdot (1 - \sigma(\chi))} \qquad \textbf{(3.4)}$$

Για πολύ μεγάλες ή πολύ μικρές τιμές εισόδου η παράγωγος πλησιάζει το μηδέν, με αποτέλεσμα να χάνονται οι κλίσεις (vanishing gradient) και να δυσκολεύει την εκπαίδευση του μοντέλου. Για αυτό το λόγο η συγκεκριμένη συνάρτηση χρησιμοποιείται συνήθως για προβλήματα δυαδικής ταξινόμησης όπου η έξοδος αντιπροσωπεύει μια πιθανότητα.

Μια παρόμοια συνάρτηση είναι η υπερβολική εφαπτομένη (hyperbolic tangent function ή tanh), όπου και αυτή είναι παρόμοια καμπύλη, με τη διαφορά ότι έχει εύρος τιμών εξόδου *-1* έως *1*. Εδώ, όσο πιο μεγάλη και θετική η τιμή, τόσο πιο κοντά στο *1* θα είναι η έξοδος, ενώ όσο πιο μικρή και αρνητική είναι η τιμή εισόδου, τόσο πιο κοντά στο *-1* θα είναι η τιμή εξόδου. Σε αυτή τη περίπτωση η καμπύλη είναι κεντραρισμένη στο μηδέν με αποτέλεσμα να μπορούν να χαρακτηριστούν οι τιμές ως έντονα αρνητικές, ουδέτερες ή έντονα θετικές. Είναι πιο απότομη καμπύλη και έχει επίσης το πρόβλημα εξαφάνισης κλίσης.



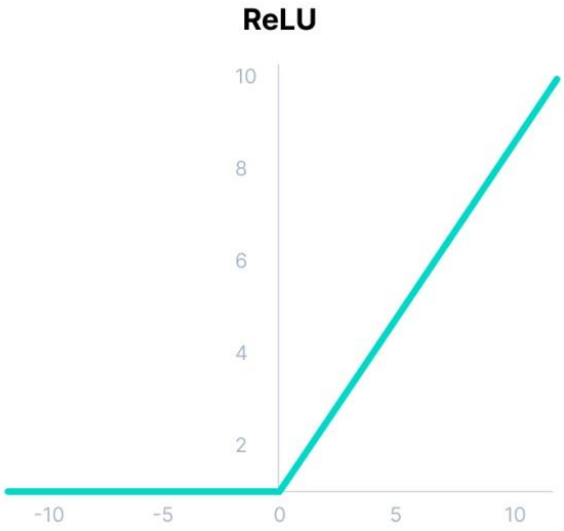
Εικόνα 6. Σιγμοειδής καμπύλη και tanh.

Και εδώ η συνάρτηση είναι παραγωγίσιμη, με συνάρτηση και παράγωγο:

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \qquad (3.5)$$

$$f'(x) = 1 - tan\,h^{2}\,(x) \qquad (3.6)$$

Η πιο διαδεδομένη μέθοδος και πιο συχνή στα νευρωνικά δίκτυα, είναι η ReLU (Rectified Linear Unit) (Fukushima, 1975). Ο λόγος της κυριαρχίας οφείλεται στο γεγονός ότι οι τιμές που είναι από μηδέν και κάτω να μηδενίζονται αυτόματα και τις τιμές μεγαλύτερες του μηδέν να μένουν ως έχουν. Δηλαδή, οι νευρώνες που εξάγουν μηδενική τιμή θα παραμένουν απενεργοποιημένοι. Αυτό καταλαβαίνει κανείς ότι είναι λιγότερο κοστοβόρο υπολογιστικά με την απόδοση της εκπαίδευσης να αυξάνεται σημαντικά.



Εικόνα 7. ReLU (Rectified Linear Unit).

$$f(x) = \begin{cases} 0, & για\ x < 0 \\ x, & για\ x \geq 0 \end{cases} \qquad (3.7)$$

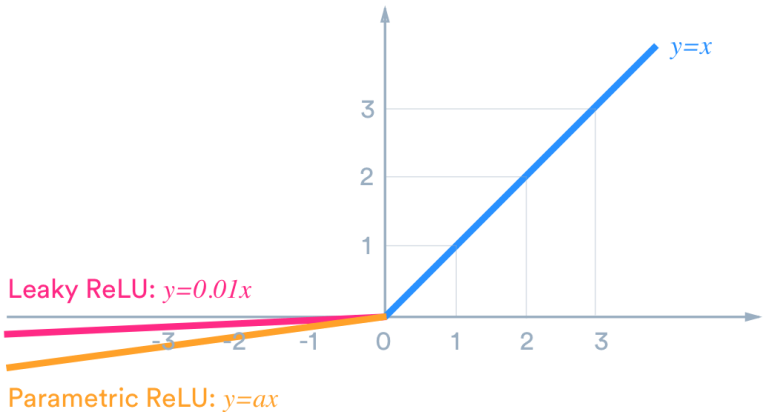
$$f'(x) = \begin{cases} 0, & για\ x < 0 \\ 1, & για\ x \geq 0 \end{cases} \qquad (3.8)$$

Βέβαια, αυτό δημιουργεί και ένα πρόβλημα, καθώς οι νευρώνες που δεν ενεργοποιούνται, δεν ενημερώνονται τα βάρη τους (dying ReLU), με αποτέλεσμα την μείωση της ικανότητα του μοντέλου να προσαρμοστεί και να εκπαιδευτεί από τα δεδομένα (Lu, 2020). Το πρόβλημα αυτό αντιμετωπίζεται με διάφορες παραλλαγές της συγκεκριμένης συνάρτησης. Ένας τρόπος είναι με τη μέθοδο Leaky ReLU, όπου σε σχέση με την απλή μορφή, έχει μια θετική κλίση στην αρνητική περιοχή. Δηλαδή, αντί να μηδενίζονται οι τιμές στη περιοχή $x < 0$, χρησιμοποιείται μια μικρή, μη μηδενική διαβάθμιση α(συνήθως $\alpha = 0.01$) (B. Xu et al., 2015). Όποτε, σε σχέση με πριν, το εύρος αλλάζει και γίνεται από αρνητικό άπειρο έως θετικό άπειρο.

$$f(x) = \begin{cases} ax, & \text{για } x < 0 \\ x, & \text{για } x \geq 0 \end{cases} \quad (3.9)$$

$$f'(x) = \begin{cases} 0.01, & \text{για } x < 0 \\ 1, & \text{για } x \geq 0 \end{cases} \quad (3.10)$$

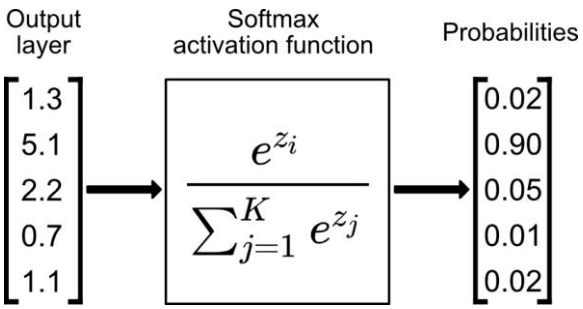
Επειδή, η παράμετρος α είναι δεδομένη από πριν, μπορεί να επηρεάσει αρνητικά την εκπαίδευση του μοντέλου. Οπότε, δίνεται η δυνατότητα να θεωρηθεί ως άγνωστη παράμετρος και να υπολογισθεί κατάλληλα μέσω του νευρωνικού, μαζί με τις υπόλοιπες παραμέτρους και βάρη. Ο υπολογισμός της συνάρτησης και της παραγώγου του είναι ακριβώς ίδια με τη Leaky ReLU, με τη διαφορά ότι η παράμετρος α είναι άγνωστη.



Εικόνα 8. Leaky ReLU vs. Parametric ReLU.

Άλλη μια μέθοδος που χρησιμοποιείται πάρα πολύ είναι η συνάρτηση softmax (Bridle, 1989). Η συγκεκριμένη συνάρτηση μετατρέπει τις τιμές εισόδου σε ένα εύρος τιμών πιθανοτήτων 0 έως 1. Μπορεί να θυμίζει τη σιγμοειδή συνάρτηση υπολογισμού τιμών πιθανότητας, με τη διαφορά ότι το άθροισμα των πιθανοτήτων όλων των τιμών εξόδων είναι ίσο με μηδέν. Συγκεκριμένα, αφού υπολογιστούν οι τιμές του προηγούμενου επιπέδου του νευρωνικού δικτύου, για κάθε τιμή υπολογίζεται μια εκθετική συνάρτηση με τη χρήση της σταθεράς Euler ($e \approx 2.718$), που διασφαλίζει όλες οι τιμές να είναι θετικές και τέλος όλες οι τιμές να διαιρούνται με το άθροισμα όλων των εκθετικών τιμών ώστε να γίνει μια κανονικοποίηση (normalization) όλων των τιμών εξόδου. Για αυτό το λόγο είναι σύνηθες και συναντάται στο τελευταίο επίπεδο ενός νευρωνικού δικτύου σε περιπτώσεις ταξινόμησης πολλαπλών κλάσεων. Η μαθηματική έκφραση της συνάρτησης είναι:

$$\boldsymbol{S(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}} \qquad \textbf{(3.11)}$$

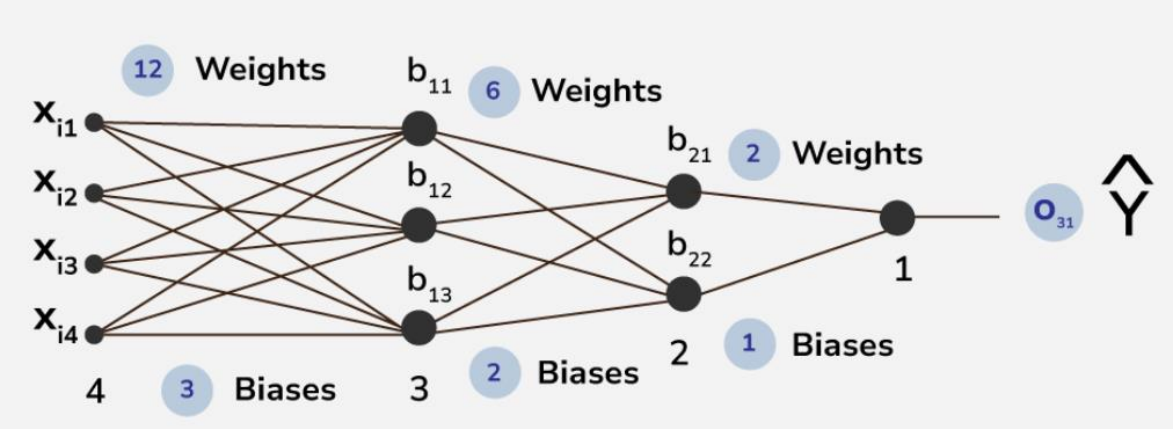


Εικόνα 9. Παράδειγμα συνάρτησης ενεργοποίησης Softmax.

Πέρα από αυτές, υπάρχουν και άλλες συναρτήσεις ενεργοποίησης και η επιλογή εξαρτάται από διάφορους παράγοντες, όπως τα δεδομένα και η φύση του προβλήματος αντιμετώπισης. Ανάλογα το πρόβλημα, διαφορετικές συναρτήσεις αποδίδουν καλύτερα, όπως για παράδειγμα η συνάρτηση softmax σε περιπτώσεις ταξινόμησης πολλαπλών κλάσεων. Επίσης, ο τύπος του νευρωνικού δικτύου επηρεάζει την επιλογή της κατάλληλης συνάρτησης ενεργοποίησης. Για παράδειγμα, στα συνελικτικά νευρωνικά δίκτυα είναι προτιμότερο η επιλογή της ReLU ή κάποια παραλλαγή της, ενώ στα αναδρομικά νευρωνικά δίκτυα είναι πιο κατάλληλα η σιγμοειδής ή tanh.

3.3 Δίκτυα πρόσθιας τροφοδότηση (Feedforward Networks)

Τα δίκτυα πρόσθιας τροφοδότησης αποτελούν τη θεμελιώδης αρχιτεκτονική των νευρωνικών δικτύων. Είναι μια διαδικασία υπολογισμών των ενδιάμεσων μεταβλητών και των τιμών εξόδου, από το επίπεδο εισόδου έως το επίπεδο εξόδου. Η διαδικασία ξεκινάει με την εισαγωγή των δεδομένων στο πρώτο επίπεδο του νευρωνικού δικτύου. Τα δεδομένα επεξεργάζονται στα κρυφά επίπεδα, όπου οι νευρώνας κάθε επίπεδου λαμβάνουν ως είσοδο τιμή υπολογισμένη από το προηγούμενο επίπεδο. Σε κάθε νευρώνα υπολογίζεται ο σταθμισμένος μέσος όρος των εισόδων και εφαρμόζεται μια συνάρτηση ενεργοποίησης, με το αποτέλεσμα να τροφοδοτείται στο επόμενο επίπεδο. Το σταθμισμένο άθροισμα στα νευρωνικά είναι γνωστό με τον όρο dot product. Στο τέλος τα επεξεργασμένα δεδομένα περνούν από το τελικό επίπεδο εξόδου όπου και εκεί εφαρμόζεται μια κατάλληλη συνάρτηση ενεργοποίησης, ώστε τελικά να παραχθεί η τελική πρόβλεψη ή ταξινόμηση των δεδομένων εισόδου. Η διαδικασία αυτή είναι απαραίτητη για να πραγματοποιηθεί πρόβλεψη. Στη συνέχεια, εάν υπάρχουν οι πραγματικές τιμές εξόδου, γίνεται σύγκριση ώστε να υπολογιστεί η απώλεια. Η απώλεια είναι αναγκαία για την ενημέρωση των βαρών και της μεροληψίας του νευρωνικού κατά την εκπαίδευση του.



Εικόνα 10. Τεχνητό νευρωνικό δίκτυο πρόσθιας τροφοδοσίας.

Στο παράδειγμα της εικόνας 10, παρουσιάζεται ένα τυπικό νευρωνικό δίκτυο τεσσάρων επιπέδων. Το πρώτο αφορά το επίπεδο εισόδου που αποτελείται από τέσσερεις τιμές. Το τελευταίο επίπεδο είναι η έξοδος του δικτύου και αποτελείται από έναν κόμβο. Τα κρυφά επίπεδα στο συγκεκριμένο παράδειγμα είναι δύο, με το πρώτο να αποτελείται από τρεις

κόμβους και το δεύτερο από δυο. Επειδή αποτελεί μια πλήρη σύνδεση επιπέδων (fully connected layer), κάθε κόμβος συνδέεται με όλους τους κόμβους στο προηγούμενο και επόμενο επίπεδο. Συνολικά δηλαδή, υπάρχουν 26 παράμετροι εκπαίδευσης που αφορούν τα βάρη ($w_{111,}w_{112}, w_{113}, \ldots$) και biases ($b_{11}, b_{12}, b_{13}, \ldots$). Η διαδικασία αυτή γίνεται με πράξεις πινάκων και για το πρώτο κρυφό επίπεδο ισχύουν τα εξής:

$$\boldsymbol{\sigma}\left(\begin{bmatrix} \boldsymbol{W_{111}} & \boldsymbol{W_{112}} & \boldsymbol{W_{113}} \\ \boldsymbol{W_{121}} & \boldsymbol{W_{122}} & \boldsymbol{W_{123}} \\ \boldsymbol{W_{131}} & \boldsymbol{W_{132}} & \boldsymbol{W_{133}} \\ \boldsymbol{W_{141}} & \boldsymbol{W_{142}} & \boldsymbol{W_{143}} \end{bmatrix}^{\boldsymbol{T}} \begin{bmatrix} \boldsymbol{X_{i1}} \\ \boldsymbol{X_{i2}} \\ \boldsymbol{X_{i3}} \\ \boldsymbol{X_{i4}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{b_{11}} \\ \boldsymbol{b_{12}} \\ \boldsymbol{b_{13}} \end{bmatrix}\right) =$$

$$\boldsymbol{\sigma}\left(\begin{bmatrix} \boldsymbol{W_{111}X_{i1} + W_{121}X_{i2} + W_{131}X_{i3} + W_{141}X_{i4}} \\ \boldsymbol{W_{112}X_{i1} + W_{122}X_{i2} + W_{132}X_{i3} + W_{142}X_{i4}} \\ \boldsymbol{W_{113}X_{i1} + W_{123}X_{i2} + W_{133}X_{i3} + W_{143}X_{i4}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{b_{11}} \\ \boldsymbol{b_{12}} \\ \boldsymbol{b_{13}} \end{bmatrix}\right) = \quad \textbf{(3.12)}$$

$$\boldsymbol{\sigma}\left(\begin{bmatrix} \boldsymbol{W_{111}X_{i1} + W_{121}X_{i2} + W_{131}X_{i3} + W_{141}X_{i4} + b_{11}} \\ \boldsymbol{W_{112}X_{i1} + W_{122}X_{i2} + W_{132}X_{i3} + W_{142}X_{i4} + b_{12}} \\ \boldsymbol{W_{113}X_{i1} + W_{123}X_{i2} + W_{133}X_{i3} + W_{143}X_{i4} + b_{13}} \end{bmatrix}\right) = \begin{bmatrix} \boldsymbol{O_{11}} \\ \boldsymbol{O_{12}} \\ \boldsymbol{O_{13}} \end{bmatrix}$$

Αντίστοιχα, στο δεύτερο κρυφό επίπεδο, η έξοδος του πρώτου και τα βάρη του δεύτερου θα δώσουν την έξοδο του δεύτερου κρυφού επιπέδου:

$$\boldsymbol{\sigma}\left(\begin{bmatrix} \boldsymbol{W_{211}} & \boldsymbol{W_{212}} \\ \boldsymbol{W_{221}} & \boldsymbol{W_{222}} \\ \boldsymbol{W_{231}} & \boldsymbol{W_{232}} \end{bmatrix}^{\boldsymbol{T}} \begin{bmatrix} \boldsymbol{O_{11}} \\ \boldsymbol{O_{12}} \\ \boldsymbol{O_{13}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{b_{21}} \\ \boldsymbol{b_{22}} \end{bmatrix}\right) =$$

$$\boldsymbol{\sigma}\left(\begin{bmatrix} \boldsymbol{W_{211}O_{11} + W_{221}O_{12} + W_{231}O_{13}} \\ \boldsymbol{W_{212}O_{11} + W_{222}O_{12} + W_{232}O_{13}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{b_{21}} \\ \boldsymbol{b_{22}} \end{bmatrix}\right) = \quad \textbf{(3.13)}$$

$$\boldsymbol{\sigma}\left(\begin{bmatrix} \boldsymbol{W_{211}O_{11} + W_{221}O_{12} + W_{231}O_{13} + b_{21}} \\ \boldsymbol{W_{212}O_{11} + W_{222}O_{12} + W_{232}O_{13} + b_{22}} \end{bmatrix}\right) = \begin{bmatrix} \boldsymbol{O_{21}} \\ \boldsymbol{O_{22}} \end{bmatrix}$$

Τέλος, γίνεται η αντίστοιχη διαδικασία για τον υπολογισμό της τελικής πρόβλεψης Y ως έξοδο:

$$\boldsymbol{\sigma}\left(\begin{bmatrix}\boldsymbol{W_{311}}\\ \boldsymbol{W_{321}}\end{bmatrix}^{\boldsymbol{T}}\begin{bmatrix}\boldsymbol{O_{21}}\\ \boldsymbol{O_{22}}\end{bmatrix}+[\boldsymbol{b_{31}}]\right)=$$

$$\boldsymbol{\sigma}([\boldsymbol{W_{311}O_{21}}+\boldsymbol{W_{321}O_{22}}]+[\boldsymbol{b_{31}}])= \quad \textbf{(3.14)}$$

$$\boldsymbol{\sigma}([\boldsymbol{W_{311}O_{21}}+\boldsymbol{W_{321}O_{22}}+\boldsymbol{b_{31}}])=[\boldsymbol{O_{31}}]=\boldsymbol{Y}$$

Από τα δεδομένα του προηγούμενου επιπέδου, κάθε κόμβος χρησιμοποιεί έναν μετασχηματισμό ομοιότητας (affine transformation) της μορφής $z = W^T X + b$ και την εφαρμογή μιας συνάρτησης ενεργοποίησης g(z). Κατά τη διάρκεια όλης της διαδικασίας, αποθηκεύονται όλοι οι παράμετροι που υπολογίστηκαν σε κάθε επίπεδο, διότι χρησιμεύουν αργότερα στην οπισθοδιάδοση (backpropagation) για την εκπαίδευση του νευρωνικού δικτύου.

3.4 Συναρτήσεις απώλειας και Τακτοποίηση

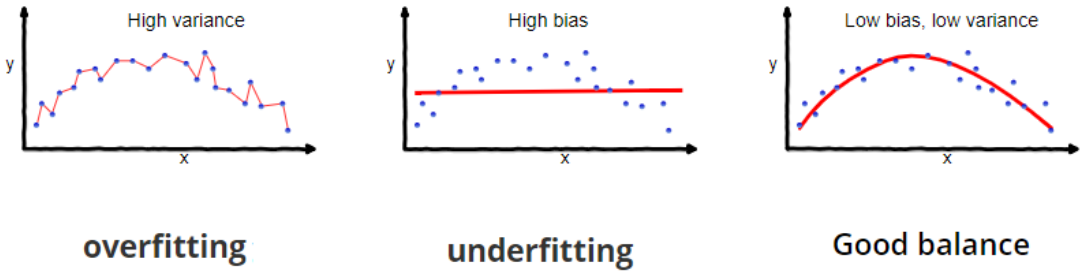
Η συνάρτηση απώλειας (loss functions) είναι η διαδικασία που έρχεται να γίνει αφού υπολογιστούν οι προβλέψεις του δικτύου. Οι συναρτήσεις αυτές συγκρίνουν την πραγματική τιμή στόχος, με τη τιμή πρόβλεψης του δικτύου και υπολογίζού την απόκλιση των δυο τιμών. Σκοπός του είναι να καθοδηγήσει το μοντέλο κατά την εκπαίδευση για το πόσο καλά αποδίδει, ελαχιστοποιώντας την απόκλιση της πραγματικής και εκτιμώμενης τιμής με επαναληπτική διαδικασία. Έτσι, με τη μορφή ποσοστού μπορεί να δοθεί η ακρίβεια πρόβλεψης του μοντέλου. Όσο μικρότερη η τιμή του ποσοστού, τόσο μεγαλύτερη η επιτυχία του μοντέλου και ταυτόχρονα η πρόβλεψή του. Υπάρχουν πολλές διαφορετικές συναρτήσεις απώλειας και η επιλογή εξαρτάται από το τύπο του μοντέλου που χρησιμοποιείται. Για παράδειγμα, το μέσο τετραγωνικό σφάλμα (mean squared error - MSE) επιλέγεται στις εργασίες παλινδρόμησης, ενώ η μέθοδος cross-entropy για εργασίες ταξινόμησης. Κατά τη διάρκεια της εκπαίδευσης εκτελείται ένας αλγόριθμος εκμάθησης οπισθοδιάδοσης (backpropagation), όπου χρησιμοποιεί την παράγωγο της συνάρτησης απώλειας (κλίση), ώστε να τις προσαρμόσει κατάλληλα και να ελαχιστοποιήσει την απώλεια, βελτιώνοντας την απόδοση του μοντέλου. Οι συναρτήσεις απώλειας κατηγοριοποιούνται με βάση την εργασία. Στις περιπτώσεις που οι προβλέψεις του

μοντέλου αφορούν συνεχείς τιμές εξόδους, τότε χρησιμοποιούνται οι συναρτήσεις απώλειας που αφορούν προβλήματα παλινδρόμησης. Αντίθετα, στις προβλέψεις που αποδίδουν διακριτές ετικέτες σε δεδομένα, τότε επιλέγονται συναρτήσεις απώλειας που αφορούν προβλήματα ταξινόμησης. Παρακάτω παρουσιάζονται ο πίνακας διαφόρων συναρτήσεων απώλειας και τα πεδία εφαρμογής τους.

Loss functions	**Περίπτωση Επιλογής**	**Εξίσωση**
Mean Square Error (MSE) / L2 Loss	Για εργασίες παλινδρόμησης. Χρησιμοποιείται όταν χρειάζεται να τιμωρήσει σημαντικά μεγάλα σφάλματα πρόβλεψης.	$\boldsymbol{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
Mean Absolute Error (MAE) / L1 Loss	Για εργασίες παλινδρόμησης. Είναι πιο ανθεκτική σε ακραίες τιμές και παρέχει μια ισορροπία των σφαλμάτων	$MAE = \frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y}_i\rvert$
Binary Cross-Entropy Loss / Log Loss	Για εργασίες ταξινόμησης. Ενθαρρύνει το μοντέλο να αντιστοιχίσει τις υψηλές πιθανότητες στη σωστή κλάση	$\boldsymbol{BCE} = -\frac{1}{n}\sum_{i=1}^{n}(y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i))$
Categorical Cross-Entropy Loss	Για εργασίες ταξινόμησης. Γενίκευση της *BCE* για πολλαπλές κλάσεις	$\boldsymbol{CCE} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{C} y_{ij} \log \hat{y}_{ij}$
Hinge Loss	Για εργασίες δυαδικής ταξινόμησης. Χρησιμοποιείται στα SVM τιμωρώντας τις λανθασμένες ταξινομήσεις γραμμικά	$\boldsymbol{Hinge\ Loss} = \max(0, 1 - y_i\hat{y}_i)$
Huber Loss / Smooth Mean Absolute Error	Για εργασίες ισχυρής παλινδρόμησης. Συνδυάζει τη στιβαρότητα της *MAE* στα μηδενικά σφάλματα και σαν *MSE* για μεγαλύτερα σφάλματα	$\boldsymbol{Huber} = \begin{cases} \frac{1}{2}(y_i - \hat{y}_i)^2 \text{ if } \lvert y_i - \hat{y}_i\rvert \le \delta \\ \delta\lvert y_i - \hat{y}_i\rvert - \frac{1}{2}\delta^2 \text{ if } \lvert y_i - \hat{y}_i\rvert > \delta \end{cases}$

Πίνακας 1. Συναρτήσεις απώλειας και περιπτώσεις εφαρμογής.

Εκτός από την ένδειξη για το πόσο καλά αποδίδει ένα μοντέλο, μπορεί να παρέχει και επιπλέον πληροφορία σχετικά με τη προσαρμογή του μοντέλου στα δεδομένα. Εάν το μοντέλο προσαρμόζει υπερβολικά στα δεδομένα (overfitting) συμπεριλαμβανομένου και του θορύβου, η απώλεια των δεδομένων θα είναι πάρα πολύ μικρή με αποτέλεσμα το μοντέλο να αποτυγχάνει να γενικεύσει σε νέα δεδομένα. Η τακτοποίηση (regularization) είναι μια τεχνική που βοηθά να αποφευχθεί αυτή η υπερπροσαρμογή και να μπορεί να γενικεύει τα μοντέλα. Η τακτοποίηση οδηγεί σε λιγότερο ακριβείς προβλέψεις στα δεδομένα εκπαίδευσης, ώστε να πετύχουν πιο ακριβείς προβλέψεις στα δεδομένα δοκιμών (test data). Διαφορετικά, σε νέα δεδομένα θα υπήρχαν μεγάλες διακυμάνσεις με αποτέλεσμα το μοντέλο να είναι ευαίσθητο σε μικρές αλλαγές δεδομένων. Στη περίπτωση αυτή θα συνέβαινε το αντίθετο, δηλαδή χαμηλό σφάλμα στα δεδομένα εκπαίδευσης αλλά υψηλό σφάλμα στα δεδομένα δοκιμής. Αυτή η διαδικασία της αύξησης του σφάλματος εκπαίδευσης ώστε να μειωθεί το σφάλμα δοκιμής, είναι γνωστή ως αντάλλαγμα μεροληψίας και διακύμανσης (bias – variance tradeoff). Η μεροληψία μετρά τη μέση διαφορά μεταξύ πρόβλεψης και πραγματικής τιμής. Όταν αυξάνεται η τιμή του, το μοντέλο κάνει προβλέψεις λιγότερο ακριβείς σε ένα σύνολο δεδομένων εκπαίδευσης. Η διακύμανση από την άλλη, αφορά στο πόσο αλλάζει η απόδοση ενός μοντέλου όταν εκπαιδεύεται σε διαφορετικά υποσύνολα δεδομένων. Όσο η διακύμανση αυξάνεται, το μοντέλο κάνει προβλέψεις λιγότερο ακριβή σε μη ορατά δεδομένα. Άρα, καταλαβαίνει κανείς ότι η μεροληψία και η διακύμανση αντιπροσωπεύουν αντιστρόφως την ακρίβεια του μοντέλου, στα σύνολα εκπαίδευσης και δοκιμών αντίστοιχα. Σκοπός είναι η ταυτόχρονη μείωση της μεροληψίας και διακύμανσης, για αυτό είναι αναγκαία η τακτοποίηση. Η τακτοποίηση μειώνει τη διακύμανση του μοντέλου, με κόστος να αυξάνει έως ένα σημείο τη μεροληψία, ώστε να πετύχει μια καλή γενίκευση.



Εικόνα 11. Περιπτώσεις τακτοποίησης.

Για να γίνει πιο κατανοητή η μέθοδος τακτοποίησης, παρατίθεται ένα παράδειγμα πρόβλεψης γραμμικής παλινδρόμησης. Η γραμμική παλινδρόμηση ή αλλιώς ελάχιστα τετράγωνα (λόγω του ότι δίνει την ελάχιστη διακύμανση), όπως προαναφέρθηκε, θα προσπαθήσει να προβλέψει τις παραμέτρους του μοντέλου, βρίσκοντας τη καλύτερη προσαρμοσμένη γραμμή μέσω των δεδομένων σημείων κατά την εκπαίδευση. Καθώς ο αριθμός των προβλέψεων αυξάνεται στο μοντέλο, η σχέση δεδομένων εισόδου εξόδου γίνεται περίπλοκη και τότε είναι που εισάγεται η τακτοποίηση. Υπάρχουν διάφορες μέθοδοι regularization που χρησιμοποιούνται, αλλά οι βασικότερες είναι οι L1, L2 και το ελαστικό δίκτυο τακτοποίησης (Elastic net regularization). Πρακτικά προσθέτουν μια ποινή στη συνάρτηση απώλειας, με βάση το μέγεθος των παραμέτρων του μοντέλου.

- Στην L1 τακτοποίηση ή και λάσο παλινδρόμηση (Lasso regression), η ποινή εισάγεται στη συνάρτηση απώλειας του μοντέλου αθροίσματος των τετραγώνων. Η ποινή αυτή βοηθά στη μείωση των βαρών ορισμένων παραμέτρων στο μηδέν. Αυτό έχει ως αποτέλεσμα την εξάλειψη μεταβλητών που δεν έχουν κάποια πληροφορία (multicollinearity). Η μορφή της εξίσωσης είναι:

$$\boldsymbol{R(w) = \sum_{i=1}^{n} |w_i|} \qquad \textbf{(3.15)}$$

- Στην L2 τακτοποίηση (Ridge regression), επίσης εισάγεται ποινή στους συντελεστές που έχουν μεγάλη τιμή στη συνάρτηση απώλειας τους. Ωστόσο, η διαφορά με τη L1 τακτοποίηση είναι ότι εδώ η ποινή είναι το τετραγωνικό άθροισμα των συντελεστών και όχι η απόλυτη τιμή τους. Άλλη μια διαφορά είναι ότι δεν συρρικνώνει τις τιμές των συντελεστών στο μηδέν ώστε να μπορεί να αφαιρεί χαρακτηριστικά, αλλά τις συρρικνώνει κοντά στο μηδέν. Η εξίσωση έχει τη μορφή:

$$\boldsymbol{R(w) = \sum_{i=1}^{n} w_i^2} \qquad \textbf{(3.16)}$$

- Το ελαστικό δίκτυο τακτοποίησης συνδυάζει τις L1, L2 παλινδρομήσεις, εισάγοντας και τους δυο όρους ποινής στη συνάρτηση απώλειας του τετραγωνικού αθροίσματος. Με αυτό τον τρόπο συνδυάζει αυτόματη επιλογή μεταβλητών και ισορροπία του μοντέλου.

Συνοψίζοντας, η L1 τακτοποίηση επειδή μπορεί να αφαιρεί ορισμένες παραμέτρους από το μοντέλο, αυτό μπορεί να βοηθά στη μείωση της πολυπλοκότητας του μοντέλου, αλλά μπορεί να υπάρξει αστάθεια στο μοντέλο καθώς μικρές αλλαγές στα δεδομένα μπορούν να προκαλέσουν μεγάλες αλλαγές στις παραμέτρους. Η L2 τακτοποίηση αποδίδει πιο ομαλά, καθώς οι παράμετροι συρρικνώνονται σταθερά αλλά δεν εξαλείφονται. Αυτό μπορεί να βοηθήσει στη γενίκευση και ανθεκτικότητα του μοντέλου, καθώς μειώνει τις μεγάλες διακυμάνσεις και τον θόρυβο στις παραμέτρους. Ωστόσο, αυτό δημιουργεί ένα πλεόνασμα στο μοντέλο καθώς διατηρεί όλα τα χαρακτηριστικά, ακόμη και αυτά που είναι αχρείαστα.
Το συνολικό κόστος του μοντέλου τώρα θα είναι η συνάρτηση μαζί με τον όρο της τακτοποίησης, πολλαπλασιασμένο με μια υπερπαράμετρο λ που ορίζει πόσο σοβαρή θα είναι η ποινή. Η υπερπαράμετρος αυτή μπορεί να υπολογιστεί μέσω της διασταυρούμενης επικύρωσης (cross validation). Μεγάλη τιμή του λ, μεγαλύτερη απλούστευση όπου μπορεί να οδηγήσει σε κακή προσαρμογή. Μικρότερη τιμή του λ δίνει περισσότερη έμφαση στη προσαρμογή των δεδομένων με αποτέλεσμα να οδηγήσει σε υπερπροσαρμογή. Τελικά, το συνολικό κόστος θα είναι της μορφής:

$$\boldsymbol{J(w) = Loss(w) + \lambda R(w)} \qquad \textbf{(3.17)}$$

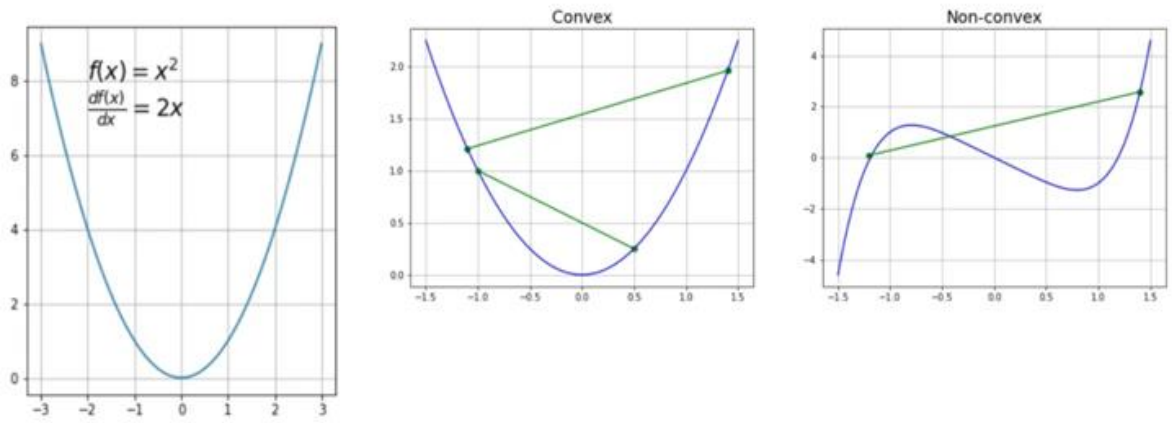
Όπου, *Loss(w)* η συνάρτηση απώλειας (πχ MSE, MAE, BCE), λ η υπερπαράμετρος που ορίζει την ισχύ της τακτοποίησης και *R(w)* ο όρος της τακτοποίησης (πχ L1 ή L2). Άλλες μέθοδοι που χρησιμοποιούνται και αξίζει να αναφερθούν είναι η early stopping, data augmentation και dropout. Η early stopping είναι ουσιαστικά μια υπερπαράμετρος εποχής, όπου διακόπτει την εκπαίδευση νωρίτερα για να αποτραπεί η υπερβολική προσαρμογή του μοντέλου στα δεδομένα εκπαίδευσης. Το πόσο νωρίς, αποφασίζεται παρακολουθώντας την ακρίβεια επικύρωσης. Στο data augmentation δημιουργείται ένα έξτρα σύνολο δεδομένων εκπαίδευσης από τα ήδη υπάρχοντα δεδομένα, μετασχηματίζοντας τα κατά μια στροφή, προσθέτοντας θόρυβο ή και κόβοντας τα. Η συγκεκριμένη μέθοδος βοηθά στη καλύτερη γενίκευση του μοντέλου λόγο της ποικιλίας των δεδομένων.

3.5 Κανόνες Βελτιστοποίησης

Εκτός των συναρτήσεων απώλειας με την αντίστοιχη τακτοποίηση, υπάρχει και άλλο ένα συστατικό που βοηθά στη βελτίωση της απόδοσης ενός μοντέλου και αφορά τους βελτιστοποιητές (optimizers). Καθώς υπολογίζεται η διαφορά μεταξύ πρόβλεψης και πραγματικής τιμής ενός μοντέλου, η συνάρτηση απώλειας αξιολογεί την αποτελεσματικότητα του μοντέλου. Όταν τροποποιούνται οι παράμετροι του μοντέλου για τη μείωση της συνάρτησης απώλειας, τότε ο βελτιστοποιητής συμβάλει στη πρόοδο του μοντέλου. Ουσιαστικά αυτό που κάνουν είναι να μαθαίνουν πως να μεταβάλουν το βάρος και με τι ρυθμό μάθησης (learning rate). Ένας βελτιστοποιητής με βάση τη συνάρτηση απώλειας, μπορεί να μαθαίνει πότε κινείται προς τη σωστή ή λάθος κατεύθυνση. Η επιλογή της κατάλληλης μεθόδου, εκτός για τη βελτίωση της ακρίβειας του μοντέλου, παίζει ρόλο και στο πόσο γρήγορα θα γίνει η εκπαίδευση. Υπάρχουν διάφοροι αλγόριθμοι βελτιστοποίησης, αλλά στη παρούσα διπλωματική αναλύονται αυτοί που χρησιμοποιούνται στα πειράματα που υλοποιήθηκαν, που είναι και πιο διαδεδομένοι.

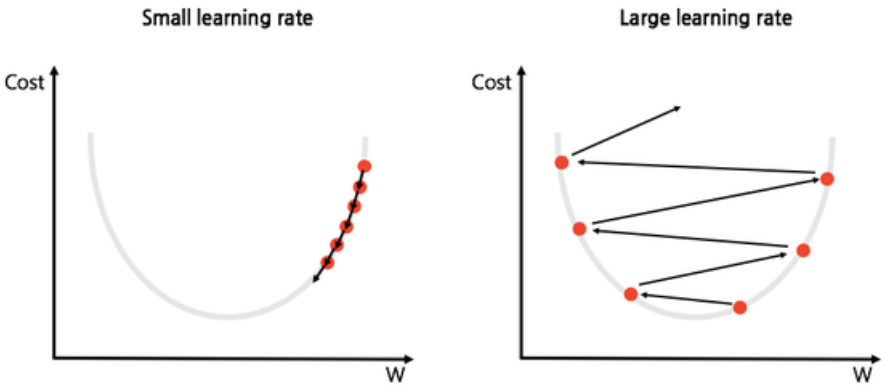
3.5.1 Αλγόριθμοι Gradient Descent

Πρώτη σημαντική μέθοδος είναι ο gradient descent ή αλγόριθμος κατάβασης κλίσης, όπου αποτελεί τη πιο συνηθισμένη μέθοδο βελτιστοποίησης στα νευρωνικά δίκτυα (Ruder, 2016), λόγο της ταχύτητας, της ευελιξίας και της ισχύς του. Ο gradient descent (ή και batch gradient descent - BGD) είναι ένας επαναληπτικός αλγόριθμος που χρησιμοποιείται για την εύρεση ενός τοπικού ελάχιστου (local minimum) μιας συνάρτησης απώλειας. Αρχικά για να μπορεί να συμβεί αυτό θα πρέπει η συνάρτηση να είναι παραγωγίσιμη και κυρτή. Αυτό σημαίνει ότι θα πρέπει να έχει μια παράγωγο για κάθε σημείο στο πεδίο ορισμού της και να είναι κυρτή καμπύλη ώστε ένα ευθύγραμμο τμήμα που ενώνει δυο σημεία της καμπύλης, να βρίσκεται πάνω από το γράφημα ανάμεσα των δυο σημείων.



Εικόνα 12. Παράγωγος και κυρτότητα συνάρτησης.

Με τον όρο gradient εννοούμε τη κλίση μιας καμπύλης σε ένα δεδομένο σημείο. Το σημείο εκκίνησης θα είναι ένα αυθαίρετο σημείο που θα αφορά την αξιολόγηση της απόδοσης. Από αυτό το σημείο υπολογίζεται η παράγωγος (ή κλίση) της συνάρτησης απώλειας και από την εφαπτόμενη γραμμή πόσο απότομη είναι. Στο σημείο εκκίνησης η κλήση θα είναι πιο απότομη και όσο επαναλαμβάνεται η διαδικασία θα αρχίσει να γίνεται πιο ομαλή, έως ότου φτάσει στο χαμηλότερο σημείο της καμπύλης γνωστό ως σημείο σύγκλισης. Καταλαβαίνει κανείς για να μειωθεί η τιμή της συνάρτησης απώλειας θα πρέπει η κατεύθυνση της παραγώγου να είναι αρνητική. Αυτό γίνεται με την χρήση της υπερπαραμέτρου ρυθμού μάθησης (learning rate). Η συγκεκριμένη υπερπαράμετρος καθορίζει τη κλίμακα ενημέρωσης των βαρών. Μεγάλη τιμή ρυθμού μάθησης μπορεί να οδηγήσει σε ταλάντωση, ενώ πολύ μικρή τιμή οδηγεί σε πολύ αργή σύγκλιση.



Εικόνα 13. Υψηλός και χαμηλός ρυθμός μάθησης.

Ο υπολογισμός της συνάρτησης δίνεται από το τύπο:

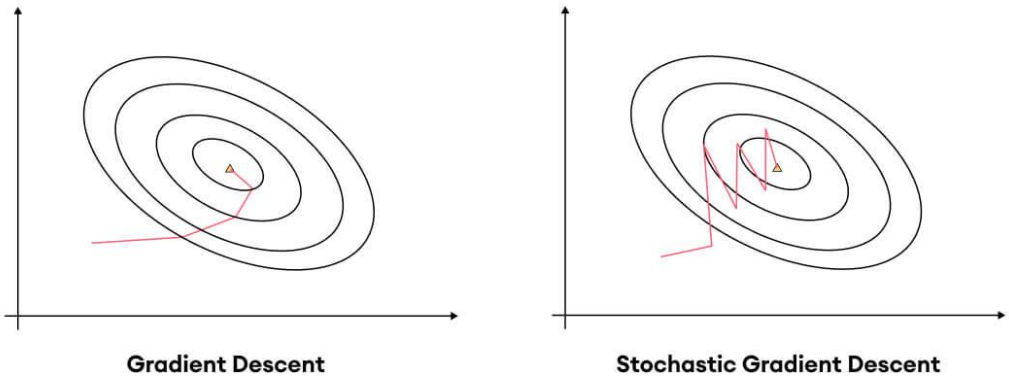
$$\boldsymbol{\theta = \theta - \eta \nabla_{\theta} J(\theta)} \qquad \textbf{(3.18)}$$

Όπου θ τα βάρη του μοντέλου (weights + biases), η ο ρυθμός εκμάθησης που καθορίζει το βήμα για κάθε επανάληψη και η παράγωγος της συνάρτησης απώλειας ως προς τη τρέχουσα τιμή των βαρών που δίνει τη κλίση. Η διαδικασία αυτή γίνεται λαμβάνοντας υπόψη το σύνολο των δεδομένων εκπαίδευσης. Δηλαδή, γίνεται υπολογισμός του μέσου όρου όλων των κλίσεων ώστε η μέση κλίση να χρησιμοποιηθεί για την ενημέρωση των παραμέτρων. Η κλίση μιας συνάρτησης αποτελείται από το διάνυσμα των μερικών παραγώγων σε σχέση με όλες τις ανεξάρτητες μεταβλητές της. Αν για παράδειγμα έχουμε μια συνάρτηση που αποτελείται από δυο ανεξάρτητες μεταβλητές, τότε θα ισχύει:

$$\boldsymbol{\nabla f(x,y) = \left(\frac{\partial f}{\partial x}(x,y), \frac{\partial f}{\partial y}(x,y)\right)} \qquad \textbf{(3.19)}$$

Θετικό πρόσημο της κλίσης δίνει τη κατεύθυνση της μεγαλύτερης αύξησης της συνάρτησης, αντίθετα αρνητικό πρόσημο δίνει τη κλίση της κατεύθυνσης της μεγαλύτερης μείωσης της συνάρτησης. Συνήθως μια συνάρτηση έχει πολλές μεταβλητές και γίνεται προσπάθεια ελαχιστοποίησης του κόστους, ακολουθώντας το αρνητικό της κλίσης. Μια επανάληψη της διαδικασίας αυτής αφορά μια εποχή, η οποία είναι επίσης μια υπερπαράμετρος. Είναι μια αποτελεσματική μέθοδος που λαμβάνει υπόψη όλα τα δεδομένα σε κάθε επανάληψη και καταλήγει πάντα σε τοπικό ελάχιστο της συνάρτησης κόστους. Τα αρνητικά της μεθόδου είναι ότι ακριβώς επειδή χρησιμοποιεί για κάθε επανάληψη όλα τα δεδομένα, το μοντέλο γίνεται υπολογιστικά δαπανηρό και χρονοβόρο. Μια πιθανή ερώτηση που μπορεί να κάνει κάποιος, είναι τι γίνεται αν το τοπικό ελάχιστο δεν είναι το καθολικό ελάχιστο; Η απάντηση είναι ότι ο αλγόριθμος δεν μπορεί να το ξέρει αυτό, καθώς όταν πετύχει το τοπικό ελάχιστο δεν μπορεί να συγκλίνει παραπάνω. Μια λύση σε αυτό το πρόβλημα είναι να γίνεται επανάληψη της διαδικασίας για διαφορετικά σημεία εκκινήσεις και να επιλέγεται τελικά η καλύτερη περίπτωση, δηλαδή αυτή που θα δίνει τη μικρότερη τιμή κόστους. Γίνεται αντιληπτό πως αυτός ο τρόπος εισάγει παραπάνω υπολογισμούς που σημαίνει παραπάνω υπολογιστική ισχύς.

Άλλη μια μέθοδος που αποτελεί παραλλαγή του batch gradient descent είναι η stochastic gradient descent (SGD). Η μεγάλη διαφορά είναι ότι αντί να χρησιμοποιηθεί το σύνολο των δεδομένων σε κάθε επανάληψη, χρησιμοποιείται μια τυχαία παρατήρηση για τον υπολογισμό της κλίσης σε κάθε επανάληψη. Για ένα σύνολο δεδομένων που αποτελείται από χίλια δεδομένα, ο αλγόριθμος θα ενημερώσει τις παραμέτρους χίλιες φορές, δηλαδή μια επανάληψη για κάθε δεδομένο. Ο λόγος που γίνεται αυτό είναι για να επιτευχθεί πιο γρήγορα ο υπολογισμός των βαρών του μοντέλου και ταυτόχρονα το καθιστά πιο κατάλληλη μέθοδο για μεγάλα σύνολα δεδομένων. Αυτή η μέθοδος μπορεί να οδηγήσει σε θορυβώδεις κλίσεις λόγο της τυχαιότητας, γεγονός που μπορεί να κάνει τη διαδρομή βελτιστοποίησης λιγότερο ομαλή. Παρά τον θόρυβο, η μέθοδος βοηθά τον αλγόριθμο να ξεφύγει από τα τοπικά ελάχιστα και να εξερευνήσει τον χώρο των παραμέτρων πιο αποτελεσματικά.



Εικόνα 14. Σύγκλιση συνάρτησης απώλειας batch gradient descent και stochastic gradient descent.

Αντίστοιχα η εξίσωση υπολογισμού της συνάρτησης κόστους είναι:

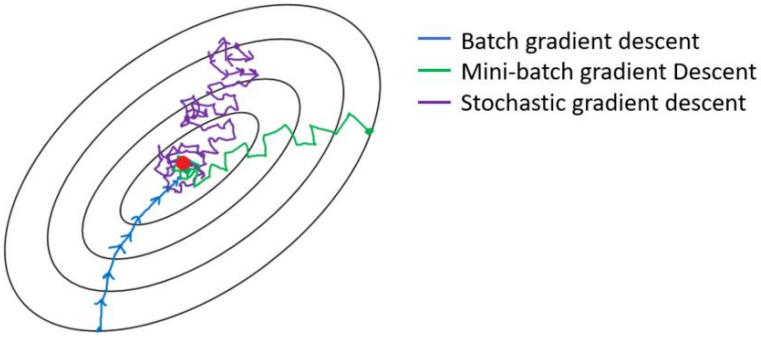
$$\boldsymbol{\theta = \theta - \eta \nabla_{\theta} J(\theta; x^{(i)}, y^{(i)})} \qquad \textbf{(3.20)}$$

Και εδώ ισχύουν τα αντίστοιχα με την εξίσωση (3.18), με τη διαφορά ότι στον όρο $\nabla_{\theta} J(\theta; x^{(i)}, y^{(i)})$ υπολογίζεται η κλίση της συνάρτησης κόστους $J(\theta)$ ως προς τη παράμετρο θ, για ένα μόνο παράδειγμα εκπαίδευσης $(x^{(i)}, y^{(i)})$.

Άλλη μια μέθοδος είναι η mini-batch stochastic gradient descent όπου αποτελεί συνδυασμό των δυο προηγούμενων μεθόδων. Ουσιαστικά διαιρεί το σύνολο των δεδομένων σε μικρότερα υποσύνολα, και ενημερώνει τις παραμέτρους σε κάθε ένα από αυτά καθώς εκτελούν SGD. Με αυτή τη μέθοδο επιτυγχάνεται μια ισορροπία της αποτελεσματικότητας του BGD και της ταχύτητας του SGD, με τη σύγκλιση να γίνεται πιο ομαλή και σταθερή. Η μορφή της εξίσωσης αλλάζει σε σχέση με την (3.20), διαιρώντας με το μέγεθος m, που αφορά τον αριθμό των υποσυνόλων.

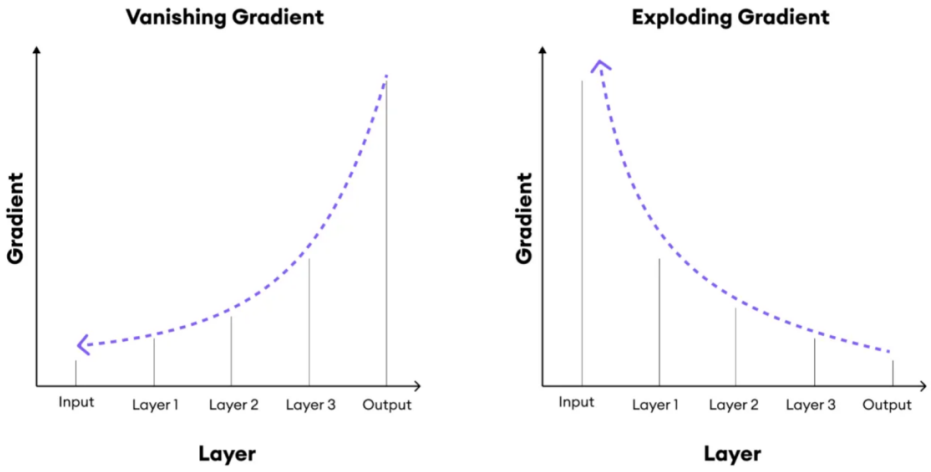
$$\boldsymbol{\theta = \theta - \eta \frac{1}{m} \nabla_{\theta} J(\theta; x^{(i)}, y^{(i)})} \qquad \textbf{(3.21)}$$

Εκτός των άλλων, η μέθοδος επειδή κάθε φορά επεξεργάζεται ένα υποσύνολο, επιτυγχάνει καλύτερη απόδοση στη μνήμη σε σχέση με τη BGD. Είναι αντιληπτό ότι με τον συνδυασμό των δυο προηγούμενων μεθόδων, επιτυγχάνεται μια ομαλή διαδρομή σύγκλισης και ταυτόχρονα λιγότερο θορυβώδης. Μπορεί η συγκεκριμένη μέθοδος συνδυάζοντας τα πλεονεκτήματα των άλλων δυο να φαντάζει ιδανική, αλλά στη πράξη έχει αποδειχτεί ότι σε πολύ μεγάλα σύνολα δεδομένων παραμένει υπολογιστικά δαπανηρή. Επίσης, απαιτείται προσεκτική ρύθμιση των υπερπαραμέτρων ρυθμού εκμάθησης (learning rate) και μέγεθος υποσυνόλων (mini-batch), διότι μπορεί να καταλήξει σε αστοχίες.



Εικόνα 15. Ρυθμός σύγκλισης batch gradient descent, stochastic gradient descent και mini-bath gradient descent.

Γενικότερα όταν χρησιμοποιούνται οι μέθοδοι κατάβασης κλίσης μπορεί να δημιουργηθούν δυο προβλήματα που επηρεάζουν σημαντικά την διαδικασία της εκπαίδευσης άρα και την απόδοση του νευρωνικού δικτύου. Αυτά είναι η εξαφανιζόμενη κλίση (vanishing gradient) και η έκρηξη κλίσης (exploding gradient). Η πρώτη περίπτωση εμφανίζεται κατά τη διάρκεια της οπισθοδιάδοσης, όταν η παράγωγος ή κλίση της συνάρτησης απώλειας γίνεται με τον καιρό αρκετά μικρή, μέχρι και εκθετικά μικρότερη, καθώς γίνεται κίνηση προς τα πίσω επίπεδα του νευρωνικού δικτύου. Αυτό είναι εμφανές σε βαθιά δίκτυα που έχουν πολλά επίπεδα. Κατά τη διάρκεια της οπισθοδιάδοσης η κλίση γίνεται μικρότερη με αποτέλεσμα και ο ρυθμός εκμάθησης να μειώνεται στα πρώτα απ' ότι στα αργότερα επίπεδα του δικτύου. Αυτό κάνει τις ενημερώσεις των παραμέτρων αμελητέες, εμποδίζοντας το δίκτυο να μαθαίνει αποτελεσματικά. Η χρήση της σιγμοειδής ή tanh συνάρτησης ενεργοποίησης μπορεί να προκαλέσει αυτό το πρόβλημα, καθώς οι παράγωγοι τους είναι μικρότερη της τιμής 1 για τα περισσότερα δεδομένα. Στην περίπτωση έκρηξης της κλίσης συμβαίνει ακριβώς το αντίστροφο. Δηλαδή, η κλίση της συνάρτησης απώλειας γίνεται υπερβολικά μεγάλη καθώς γίνεται οπισθοδιάδοση στο δίκτυο. Εδώ η βασική αιτία του προβλήματος βρίσκεται στα βάρη του δικτύου και όχι στην επιλογή συνάρτησης ενεργοποίησης. Υψηλές τιμές βαρών οδηγούν σε αντίστοιχα υψηλές τιμές παραγώγων, προκαλώντας μεγάλες αποκλίσεις στις νέες τιμές βαρών σε σχέση με τις προηγούμενες. Αυτό έχει ως αποτέλεσμα η κλίση να αποτυγχάνει να συγκλίνει και μπορεί να οδηγήσει σε ταλάντωση του δικτύου γύρω από τοπικά ελάχιστα, καθιστώντας δύσκολη την επίτευξη του καθολικού ελαχίστου.



Εικόνα 16. Πρόβλημα Vanishing gradient και Exploding gradient.

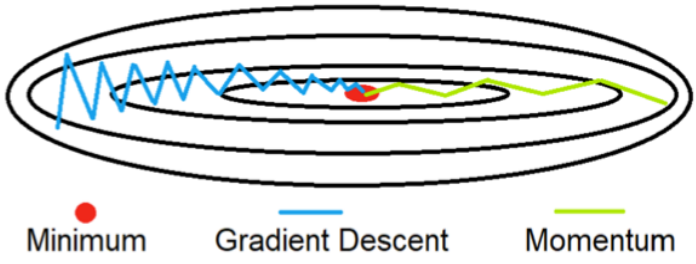
3.5.2 Αλγόριθμος Momentum

Μια επέκταση του gradient descent αλγορίθμου είναι το Momentum (Duda, 2019), όπου βοηθά στην επιτάχυνση της σύγκλισης, οδηγώντας τα διανύσματα κλίσεων προς τις σωστές κατευθύνσεις. Αυτό λύνει το πρόβλημα του gradient descent όπου η κλίση μπορεί από δεδομένα με τυχαία σφάλματα, να έχει ασυνήθιστες κατευθύνσεις ακόμη και ανηφορικά. Αφού υπολογιστεί η εξίσωση (3.20) για το θ σε μια συγκεκριμένη στιγμή, στη συνέχεια:

$$\boldsymbol{v_t = \beta v_{t-1} + (1-\beta)\nabla_\theta J(\theta; x^{(i)}, y^{(i)})}$$

$$\boldsymbol{\theta_t = \theta_{t-1} - \eta v_t} \qquad \textbf{(3.22)}$$

Εισάγεται ένας όρος v (από velocity), που αφορά την συσσώρευση των κλίσεων από τις προηγούμενες ενημερώσεις. Αυτή η συσσώρευση αυξάνει τη ταχύτητα προς μια σταθερή κατεύθυνση κλίσης, εξομαλύνοντας τις ενημερώσεις και ενδεχομένως αποφεύγοντας τοπικά ελάχιστα. Ο συντελεστής β (decay rate), είναι μια υπερπαράμετρος που αφορά τη συνεισφορά της προηγούμενης ταχύτητας v στη τρέχουσα ενημέρωση. Ο όρος αυτός είναι γνωστός και ως εκθετικός σταθμισμένος κινούμενος μέσος όρος (exponentially weighted moving average - EMWA), που ουσιαστικά εντοπίζει μακροπρόθεσμες τάσεις στα δεδομένα των κλίσεων, δίνοντας μεγαλύτερη βαρύτητα στις πιο πρόσφατες παρατηρήσεις και εξομαλύνοντας τις βραχυπρόθεσμες διακυμάνσεις μειώνοντας τον θόρυβο στα δεδομένα. Υψηλή τιμή, σημαίνει ότι δίνει μεγαλύτερο βάρος στις προηγούμενες ενημερώσεις, εξομαλύνοντας περισσότερο τη τροχιά (συνήθως κοντά 0.9).



Εικόνα 17. SGD με και χωρίς Momentum.

3.5.3 Αλγόριθμοι βελτιστοποίησης AdaGrad και RMSprop

Σημαντικοί επίσης αλγόριθμοι βελτιστοποίησης στα νευρωνικά δίκτυα, είναι αυτοί που χρησιμοποιούν προσαρμοστικό ρυθμό μάθησης (adaptive learning rates). Δηλαδή, κρατάνε ιστορική πληροφορία των κλίσεων, οι οποίες μπορούν να βελτιώσουν την ταχύτητα σύγκλισης και την ακρίβεια του αλγορίθμου. Ο αλγόριθμος AdaGrad (Duchi et al., 2011), είναι μια επέκταση του SGD. Ο αλγόριθμος κλιμακώνει τον ρυθμό εκμάθησης αντίστροφος ανάλογα με την τετραγωνική ρίζα του αθροίσματος όλων των προηγούμενων κλίσεων, για κάθε παράμετρο. Η κάθε παράμετρος ενημερώνεται:

$$\boldsymbol{\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{G_{i,t}+\epsilon}} g_t} \qquad \textbf{(3.23)}$$

Όπου θ_t η παράμετρος μια χρονική στιγμή, η ο καθολικός ρυθμός εκμάθησης, g_t η κλίση, $G_{i,t}$ ένας διαγώνιος πίνακας όπου κάθε τιμή του αντιπροσωπεύει το άθροισμα των τετραγώνων των κλίσεων ως προς την κάθε παράμετρο σε κάθε βήμα και το ϵ που είναι μια σταθερά για να αποτρέψει την διαίρεση με το μηδέν. Η μέθοδος αυτή λειτουργεί καλά με αραιά δεδομένα, όμως με τη συσσώρευση των τετραγωνικών των κλίσεων μπορεί να οδηγήσει σε ταχεία μείωση (rapid decay) του ρυθμού εκμάθησης, με αποτέλεσμα η βελτιστοποίηση να γίνεται αρκετά αργή μετά από ένα σημείο. Η λύση σε αυτό το πρόβλημα αντιμετωπίζεται από τη μέθοδο RMSprop, όπου εισάγει ένα κινητό μέσο όρο στα τετράγωνα των κλίσεων.

$$\begin{aligned} \boldsymbol{v_t} &= \boldsymbol{\beta v_{t-1} + (1-\beta) g_t^2} \\ \boldsymbol{\theta_t} &= \boldsymbol{\theta_{t-1} - \frac{\eta}{\sqrt{v_t+\epsilon}} g_t} \end{aligned} \qquad \textbf{(3.24)}$$

Η διαφορά με την εξίσωση (3.23) είναι ότι εδώ χρησιμοποιείται ο εκθετικός μέσος όρος. Αυτή η μέθοδος έχει δείξει ότι στη πράξη είναι πιο αποτελεσματική, καθώς βοηθά στη διατήρηση μιας πιο ελεγχόμενης και συνεπούς ρυθμού μάθησης με τον καιρό. Απαιτεί προσεκτική ρύθμιση των υπερπαραμέτρων (learning – decay rate) λόγο ευαισθησίας του. Ο αλγόριθμος εξετάζει πόσο απότομη είναι η επιφάνεια του σφάλματος και προσαρμόζει την ενημέρωση του ρυθμού εκμάθησης. Για παράδειγμα, παράμετροι με υψηλές κλίσεις, λαμβάνουν μικρότερα βήματα ενημέρωσης και στις χαμηλές κλίσεις το αντίστροφο.

3.5.4 Αλγόριθμος βελτιστοποίησης Adam

Ένας αλγόριθμος βελτιστοποίησης που συναντάται παντού και μπορεί να υλοποιηθεί με τρομερή επιτυχία είναι ο Adam (Adaptive Moment Estimation) (Kingma & Ba, 2014). Ο συγκεκριμένος αλγόριθμος αποτελεί έναν συνδυασμό των αλγορίθμων RMSprop και SGD με χρήση Momentum, όπου κληρονομεί τα θετικά χαρακτηριστικά των μεθόδων για να αποδώσει μια πιο βελτιστοποιημένη κλίση. Πιο συγκεκριμένα, υπολογίζει τον προσαρμοστικό ρυθμό εκμάθησης με βάση τον μέσο όρο των πρόσφατων κλίσεων για κάθε παράμετρο, παρόμοια με τον RMSprop και διατηρεί έναν εκθετικά κινούμενο μέσο όρο παλαιότερων κλίσεων που βοηθά στην εξομάλυνση θορύβου και επιτάχυνση της σύγκλισης, όπως το Momentum. Οπότε χρησιμοποιούνται δυο στιγμές εκτίμησης, μια για τη κάθε μέθοδο. Αρχικά υπολογίζονται οι κλίσεις της συνάρτησης απώλειας, στη συνέχεια υπολογίζεται η πρώτη στιγμή εκτίμησης που αφορά την εξίσωση (3.22) όπου συμβολίζεται με το γράμμα m και η δεύτερη στιγμή εκτίμησης που αφορά την εξίσωση (3.24) όπου συμβολίζεται με το γράμμα v. Επειδή γίνεται μια αρχικοποίηση των παραμέτρων αυτών στο μηδέν, υπολογίζεται κάθε φορά και η μεροληψία (bias) που αφορά την διόρθωση αυτών των εκτιμήσεων. Τελικά, γίνεται η ενημέρωση των παραμέτρων χρησιμοποιώντας τις διορθωμένες εκτιμήσεις πρώτης και δεύτερης στιγμής.

$$\boldsymbol{g_t = \nabla_{\theta_{t-1}} J(\theta_{t-1})} \qquad \textbf{(3.25)}$$

$$\boldsymbol{m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t} \qquad \textbf{(3.26)}$$

$$\boldsymbol{v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2} \qquad \textbf{(3.27)}$$

$$\boldsymbol{\hat{m}_t = \frac{m_t}{1 - \beta_1^t}} \qquad \textbf{(3.28)}$$

$$\boldsymbol{\hat{v}_t = \frac{v_t}{1 - \beta_2^t}} \qquad \textbf{(3.29)}$$

$$\boldsymbol{\theta t = \theta_{t-1} - a \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}} \qquad \textbf{(3.30)}$$

Σε αυτές τις σχέσεις πρέπει να γίνει ορισμός των υπερπαραμέτρων a που αφορά τον ρυθμό εκμάθησης, της σταθεράς ϵ που αποτρέπει την διαίρεση με το μηδέν και των ποσοστών αποσύνθεσης β_1 και β_2. Οι συνήθεις τιμές των παραμέτρων αυτών είναι 0.9 και 0.999 αντίστοιχα. Αυτό διότι θέλει να δείξει ότι η συνεισφορά της προηγούμενης κλίσης είναι πολύ σημαντική σε σχέση με παλιότερες ή τη συγκεκριμένη χρονική στιγμή.

3.6 Οπισθοδιάδοση (Backpropagation)

Η οπισθοδιάδοση αποτελεί έναν αλγόριθμο επιβλεπόμενης μάθησης των νευρωνικών δικτύων. Όπως αναλύεται και στο κεφάλαιο 3.3, σε ένα νευρωνικό πρόσθιας τροφοδότησης, τα δεδομένα εισόδου εισάγονται στο επίπεδο εισόδου και κινούνται προς το επίπεδο εξόδου. Η οπισθοδιάδοση έχει σημαντικό ρόλο στη βελτίωση των προβλέψεων που γίνονται στο νευρωνικό δίκτυο. Αυτό συμβαίνει διαδίδοντας το σφάλμα αντίστροφα, δηλαδή από το επίπεδο εξόδου προς το επίπεδο εισόδου. Για να μπορεί να ελαχιστοποιηθεί το σφάλμα ενός μοντέλου, καταλαβαίνει κανείς ότι θα πρέπει και τα βάρη των παραμέτρων να αναπροσαρμοστούν. Ο συνδυασμός αυτών είναι που επιτρέπει στο νευρωνικό δίκτυο να εκτελεί την εκπαίδευση. Δηλαδή, ο αλγόριθμος υπολογίζει τις κλίσεις της συνάρτησης απώλειας σε σχέση με τα βάρη προς την αντίστροφη κατεύθυνση, όπου στη συνέχεια χρησιμοποιούνται από έναν αλγόριθμο κατάβασης κλίσης, ώστε να ελαχιστοποιηθεί η συνάρτηση απώλειας (Rumelhart et al., 1986). Για τον υπολογισμό των κλίσεων της συνάρτησης απώλειας χρησιμοποιείται ο κανόνας αλυσίδας (chain rule). Ο συγκεκριμένος κανόνας εξετάζει τη μερική παράγωγο κάθε παραμέτρου. Σε κάθε υπολογισμό της παραγώγου ενός βάρους, τα υπόλοιπα βάρη αντιμετωπίζονται ως σταθερές. Σκοπός του κανόνα αλυσίδας είναι η απλοποίηση των σύνθετων συναρτήσεων που αποτελούνται από άλλες συναρτήσεις. Στη περίπτωση των νευρωνικών, δείχνει πως προσαρμογές των βαρών, για παράδειγμα στην είσοδο των δεδομένων, μπορεί να ελαχιστοποιήσει τη διαφορά της πρόβλεψης από την πραγματική τιμή. Για να μην υπάρξει σύγχυση μεταξύ των εννοιών οπισθοδιάδοσης και κατάβασης κλίσης, στη πρώτη περίπτωση υπολογίζονται οι κλίσεις της συνάρτησης απώλειας, όπου με βάση αυτές ένας αλγόριθμος κατάβασης κλίσης εντοπίζει τα βάρη που θα ελαχιστοποιήσουν τη συνάρτηση απώλειας. Γίνεται έτσι αντιληπτό ότι οι αλγόριθμοι κατάβασης κλίσης εξαρτούνται από την οπισθοδιάδοση. Όλη η διαδικασία αυτή επαναλαμβάνεται συνεχώς για πολλές εποχές, έως ότου το μοντέλο συγκλίνει στο ελάχιστο τη συνάρτηση απώλεια. Τα βάρη w και b

αρχικοποιούνται συνήθως με μικρές τυχαίες τιμές. Κατά το πρόσθιο πέρασμα υπολογίζεται ο σταθμισμένος μέσος όρος μαζί με τη συνάρτηση ενεργοποίησης.

$$\boldsymbol{z = W \cdot x + b}$$

$$\boldsymbol{\alpha = \sigma(z)} \qquad \textbf{(3.31)}$$

Η διαδικασία αυτή επαναλαμβάνεται για κάθε σύνδεση νευρώνων, από το επίπεδο εισόδου έως το επίπεδο εξόδου. Στο επίπεδο εξόδου, υπολογίζεται η συνάρτηση απώλειας, όπου συγκρίνει την τιμή πρόβλεψης του δικτύου με την πραγματική τιμή. Για παράδειγμα, αν χρησιμοποιηθεί η MSE:

$$\boldsymbol{C = \frac{1}{2}\sum (y_{pred} - y_{true})^2} \qquad \textbf{(3.32)}$$

Στο επόμενο βήμα ξεκινά η οπισθοδιάδοση, υπολογίζοντας την κλίση της συνάρτησης απώλειας σε σχέση με κάθε βάρος του δικτύου. Η διαδικασία αυτή γίνεται με τον κανόνα αλυσίδας. Αρχικά υπολογίζεται το σφάλμα του επιπέδου εξόδου:

$$\boldsymbol{\delta^{(C)} = \nabla_{\alpha} C \cdot \sigma'(z^{(C)})} \qquad \textbf{(3.33)}$$

όπου $\nabla_{\alpha} C$ η παράγωγος της απώλειας σε σχέση με τη συνάρτηση ενεργοποίησης α και σ' η παράγωγος της συνάρτησης ενεργοποίησης. Στη συνέχεια, για κάθε κρυφό επίπεδο l μέχρι το επίπεδο εισόδου υπολογίζεται αντίστοιχα το σφάλμα:

$$\boldsymbol{\delta^{(l)} = (W^{(l+1)})^T \delta^{l+1} \cdot \sigma'(z^{(l)})} \qquad \textbf{(3.34)}$$

Για το τρέχον σφάλμα λαμβάνονται υπόψιν τα βάρη του επόμενου επιπέδου, το σφάλμα που υπολογιστικέ στο προηγούμενο επίπεδο και η παράγωγος της συνάρτησης ενεργοποίησης στο συγκεκριμένο επίπεδο. Να διευκρινιστεί ότι στην οπισθοδιάδοση, το $l+1$ επίπεδο αφορά το ίδιο επίπεδο με αυτό στο πρόσθιο πέρασμα. Άρα, στην οπισθοδιάδοση αποτελεί τα στοιχεία υπολογισμού του προηγούμενου επιπέδου. Η εξίσωση αυτή χρησιμοποιείται επαναληπτικά για κάθε κρυφό επίπεδο έως το πρώτο. Στη

συνέχεια ενημερώνονται τα βάρη, αφαιρώντας από την αρχική τους τιμή το γινόμενο του σφάλματος που υπολογίστηκε, με την τιμή εισόδου (μπορεί να είναι η τιμή στο επίπεδο εισόδου ή η τιμή μετά τη συνάρτηση ενεργοποίησης του εκάστοτε κρυφού επιπέδου) και τον συντελεστή ρυθμού εκμάθησης.

$$\boldsymbol{W^{(l)} = W^{(l)} - \eta \frac{\partial C}{\partial W^l} = W^{(l)} - \eta \cdot (\delta^l \cdot (\alpha^{l-1})^T)}$$

$$\boldsymbol{b^{(l)} = b^{(l)} - \eta \frac{\partial C}{\partial b^l} = b^{(l)} - \eta \cdot \delta^l} \qquad \textbf{(3.35)}$$

Η πρώτη σχέση από την εξίσωση (3.35), προκύπτει από το γεγονός ότι η παράγωγος της συνάρτησης κόστους ως προς τα βάρη, με βάση του κανόνα αλυσίδας, ισοδυναμεί με τη παράγωγο της συνάρτησης κόστους ως προς τη τιμή εισόδου πριν την ενεργοποίηση πολλαπλασιαζόμενο με τη παράγωγο της τιμής πριν την ενεργοποίησης ως προς τα βάρη.

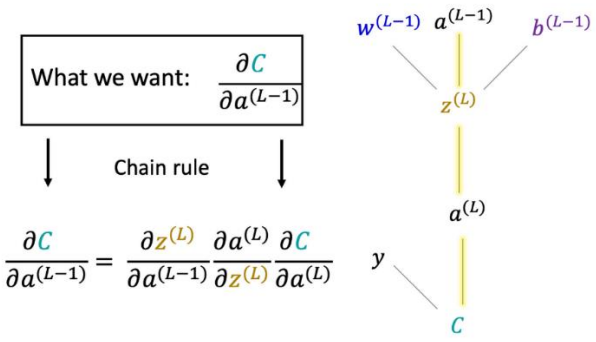
$$\boldsymbol{\frac{\partial C}{\partial W^l} = \frac{\partial C}{\partial z^l} \cdot \frac{\partial z^l}{\partial W^l}}$$

$$\boldsymbol{\frac{\partial C}{\partial z^l} = \frac{\partial C}{\partial \alpha^l} \cdot \sigma'(z^l)} \qquad \textbf{(3.36)}$$

όπου, το δεύτερο μέρος ισούται με $\boldsymbol{\delta^{(C)}}$, από την εξίσωση (3.33). Αντίστοιχα:

$$\boldsymbol{\frac{\partial z}{\partial W^l} = \alpha^{l-1}} \qquad \textbf{(3.37)}$$

αφού η παράγωγος της εξίσωσης $\boldsymbol{z^l = W^l \cdot \alpha^{l-1} + b^l}$ ως προς το βάρος, παραμένει ο συντελεστής $\boldsymbol{\alpha^{l-1}}$. Έτσι προκύπτει η εξίσωση (3.35) για την ενημέρωση των βαρών και κατά συνέπεια τη μείωση της σύγκλισης.



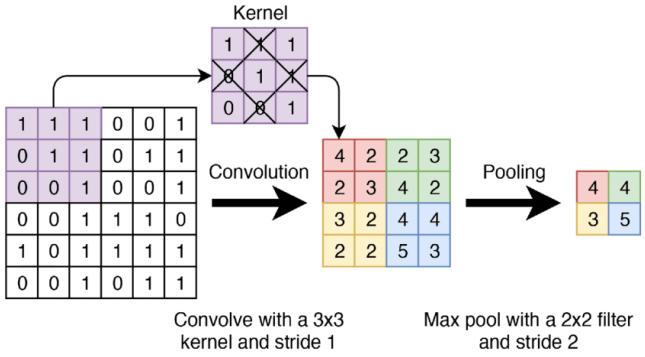
Εικόνα 18. Υπόδειγμα οπισθοδιάδοσης.

3.7 Συνελικτικά νευρωνικά δίκτυα

Ένα από τα πιο διαδεδομένα είδη τεχνητών νευρωνικών δικτύων που χρησιμοποιούνται είναι τα συνελικτικά νευρωνικά δίκτυα (Convolutional Neural Networks - CNN), όπου χρησιμεύουν στην επεξεργασία και ανάλυση οπτικών δεδομένων (O'Shea & Nash, 2015). Αυτό γιατί σε σχέση με τα απλά νευρωνικά δίκτυα, μπορούν αυτόματα να μαθαίνουν χωρικές ιεραρχίες από τα δεδομένα. Αυτό το είδος νευρωνικών χρησιμοποιείται κατά κύριο λόγο για επεξεργασία δομημένων δεδομένων όπως είναι οι εικόνες (πλέγμα), αλλά υπό κάποια προεπεξεργασία μπορούν να χρησιμοποιηθούν και σε άλλες δομές, όπως τα τρισδιάστατα που είναι και το ζητούμενο της διπλωματικής. Για αυτό το λόγο θα γίνει μια γενικότερη ανάλυση των CNN, χωρίς ιδιαίτερη έμφαση στο μαθηματικό περιεχόμενο τους.

Η δομή των CNN δεν έχει μεγάλη διαφορά με αυτά των κοινών νευρωνικών δικτύων. Και στις δυο περιπτώσεις αποτελούνται από επίπεδα νευρώνων, όπου στο καθ' ένα εισέρχεται ένα δεδομένο, το επεξεργάζονται και το προωθούν στο επόμενο επίπεδο. Και στα δύο χρησιμοποιούνται τα βάρη και η μεροληψία για τους υπολογισμούς και την εκπαίδευση του μοντέλου με χρήση της εμπρόσθιας τροφοδότησης και οπισθοδιάδοσης, ώστε να ενημερώνονται οι παράμετροι του δικτύου. Το ίδιο ισχύει και για την εισαγωγή μιας συνάρτησης ενεργοποίησης, για την αποφυγή της γραμμικότητας του μοντέλου. Επίσης, στη τελευταία ένωση, ανάμεσα στο τελευταίο κρυφό επίπεδο και το επίπεδο εξόδου, χρησιμοποιείται μια συνάρτηση απώλειας για τον υπολογισμό της σύγκλισης της πρόβλεψης του δικτύου σε σχέση με τις πραγματικές τιμές. Αυτό που τα διαφοροποιεί ως προς τη δομή τους είναι η εισαγωγή της συνέλιξης. Η συνέλιξη ουσιαστικά είναι μια

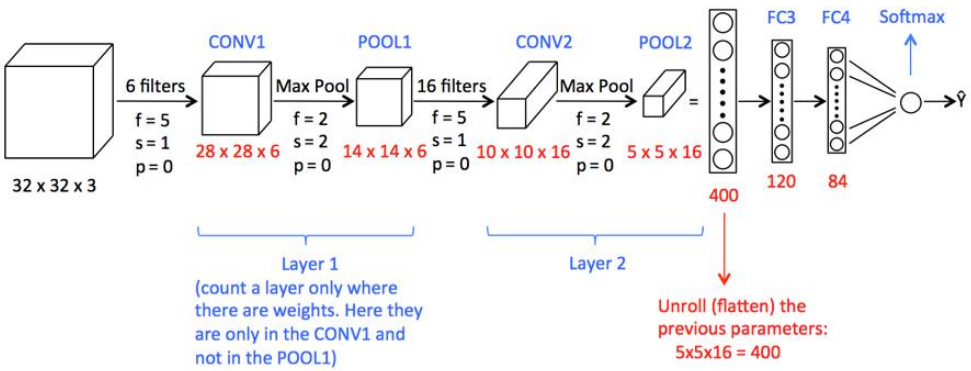
εισαγωγή τριών επιπέδων στο νευρωνικό δίκτυο. Αυτά αποτελούν το convolution layer, pooling layer και fully connected layer (όπως και στα απλά νευρωνικά δίκτυα). Στα convolution layer χρησιμοποιείται ένα φίλτρο (kernel) που διαπερνά από τα δεδομένα εισόδου. Αυτό βοηθά στην εκμετάλλευση της χωρικής δομής των δεδομένων, αλλά και στη κοινή χρήση παραμέτρων, καθώς το φίλτρο χρησιμοποιεί το ίδιο σετ βαρών σε όλα τα δεδομένα εισόδου. Αυτό έχει ως αποτέλεσμα να μειώνονται δραματικά το πλήθος των παραμέτρων, καθιστώντας τα CNN πιο αποτελεσματικά ειδικά σε μεγάλα δεδομένα εισόδου. Αντίθετα με τα κοινά νευρωνικά δίκτυα όπου κάθε νευρώνας έχει δικό του βάρος, που οδηγεί σε μεγάλο πλήθος παραμέτρων, άρα και περισσότερες υπολογιστικές απαιτήσεις. Αυτά τα φίλτρα λειτουργούν σαν ανιχνευτές, όπου μπορούν για παράδειγμα να εντοπίζουν διάφορα χαρακτηριστικά των δεδομένων, όπως τις ακμές ενός αντικειμένου. Να σημειωθεί ότι σε κάθε convolution μπορεί να υπάρχουν παραπάνω από ένα φίλτρο που θα μπορεί να εντοπίζει διαφορετικά χαρακτηριστικά. Το pooling layer, από τη πλευρά του έχει ως στόχο τη μείωση των δεδομένων, κρατώντας τα σημαντικότερα χαρακτηριστικά (feature map) που εξάγεται από την συνέλιξη, όπου και αυτό με τη σειρά του μειώνει το υπολογιστικό φορτίο του δικτύου. Η πιο κοινή μέθοδος αφορά το max pooling όπου διατηρεί μόνο τις μέγιστες τιμές του εξαγόμενου feature map.



Εικόνα 19. Convolution και pooling layer.

Η διαδικασία αυτή μπορεί να επαναληφθεί αρκετές φορές και να αυξάνεται ο αριθμός των φίλτρων. Αυτό έχει ως αποτέλεσμα το CNN να αναγνωρίζει υφές και σχήματα και όσο προχωράει πιο βαθιά στο νευρωνικό να εντοπίζει πιο περίπλοκες δομές, όπως μέρη αντικειμένων. Τα κοινά νευρωνικά δίκτυα δυσκολεύονται σε τέτοιες εργασίες δεδομένων μεγάλων διαστάσεων, καθώς εκτός από το γεγονός ότι πρέπει να υπολογίζουν πολλές

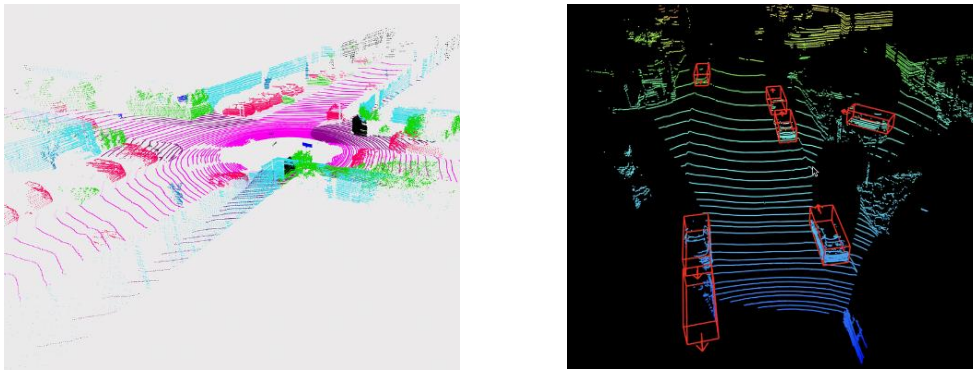
περισσότερες παραμέτρους, όπως προαναφέρθηκε, ταυτόχρονα ακριβώς επειδή είναι πάρα πολλές οι παράμετροι αυτό οδηγεί σε overfitting του μοντέλου όπου οδηγεί σε μια όχι τόσο καλή εκπαίδευση του μοντέλου. Όσο για τις υπερπαραμέτρους των CNN είναι τα ίδια με τα κοινά, επιπλέον έχουν το μέγεθος του φίλτρου, το πλήθος των φίλτρων, το βήμα που αφορά τη κίνηση του φίλτρου (μεγάλο βήμα σημαίνει μείωση των χωρικών διαστάσεων του feature map άρα γρηγορότερος υπολογισμός, αλλά ταυτόχρονα οδηγεί σε μείωση πληροφορίας), το padding που αφορά τη προσθήκη μηδενικών στοιχείων γύρω από το όριο της εικόνας εισόδου για τον έλεγχο των διαστάσεων της εξόδου του feature map. Συνήθως επιλέγεται τιμή που θα οδηγεί στην ίδια χωρική διάσταση της εισόδου με την έξοδο. Τέλος, το μέγεθος του παραθύρου pooling και το είδος του, όπως φαίνονται στην εικόνα 19. Συνήθως επιλέγεται ένα παράθυρο 2X2 όπου μειώνει κατά το ήμισυ τις διαστάσεις, ενώ μεγαλύτερα μεγέθη αποτελούν επιθετική δειγματοληψία, με αποτέλεσμα να χαθούν σημαντικά χαρακτηριστικά. Το είδος συνήθως αποτελεί το max pooling, ώστε να διατηρεί τις μέγιστες τιμές που θα αποτελούν και τα σημαντικότερα χαρακτηριστικά. Η διαδικασία μετά τη συνέλιξη συνεχίζει όπως στα απλά νευρωνικά δίκτυα με χρήση των πλήρη συνδέσεων (fully connected) των νευρώνων. Τέλος χρησιμοποιείται η Softmax συνάρτηση, όπου μετατρέπει τις τιμές εξόδου του δικτύου σε κατανομές πιθανότητας της κάθε κατηγορίας εξόδου. Είναι ιδανική μέθοδος κατανομής πιθανοτήτων σε εργασίες ταξινόμησης πολλαπλών κατηγοριών, καθώς παρέχει κανονικοποιημένη κατανομή πιθανοτήτων σε όλες τις κατηγορίες. Αυτό όχι μόνο επιτρέπει στο μοντέλο να κάνει μια σαφή πρόβλεψη, αλλά ποσοτικοποιεί και την εμπιστοσύνη της πρόβλεψης.



Εικόνα 20. Υπόδειγμα CNN με φίλτρο f, βήμα s kai padding p.

4. Μοντέλα τρισδιάστατης σημασιολογικής κατάτμησης και ανίχνευσης αντικειμένων

Στη κεφάλαιο αυτό αναλύονται διάφορες τεχνικές νευρωνικών δικτύων, που έχουν ως σκοπό να αναγνωρίζουν μοτίβα στα δεδομένα και να ανταποδίδουν κάποιο χαρακτηρισμό. Οι πιο κοινές και διαδεδομένες τεχνικές αφορούν για δεδομένα εικόνας, πιο συγκεκριμένα εικονοστοιχεία (pixel). Η παρούσα διπλωματική ασχολείται με έναν άλλο τύπο που αφορά τα τρισδιάστατα δεδομένα. Δηλαδή, αντί για έναν δισδιάστατο κατανεμημένο χώρο pixel μιας εικόνας, σε ένα τρισδιάστατο χώρο σημείων όπου τα δεδομένα έχουν τυχαία κατανομή. Η διαδικασία που ακολουθούν τα μοντέλα σε αυτές τις περιπτώσεις, καταλαβαίνει κανείς ότι γίνεται ακόμα πιο δύσκολη, καθώς θα πρέπει να χειριστούν μια επιπρόσθετη διάσταση και τις προκλήσεις τις. Δυο βασικές μέθοδοι αφορούν τη σημασιολογική κατάτμηση και την ανίχνευση αντικειμένων. Στη πρώτη περίπτωση, το μοντέλο αποδίδει μια συγκεκριμένη ετικέτα σε κάθε δεδομένο σημείο του τρισδιάστατου χώρου. Για παράδειγμα, σε ένα σύνολο τρισδιάστατων δεδομένων που απεικονίζονται σημεία από ένα αστικό περιβάλλον, θα μπορεί να αποδώσει για κάθε σημείο μια ετικέτα που αυτή θα μπορούσε να αφορά δρόμο, πεζούς, αυτοκίνητα, κτίρια κ.λπ.. Στη δεύτερη περίπτωση, γίνεται αναγνώριση και εντοπισμός του αντικειμένου. Πιο συγκεκριμένα, η έξοδος δίνει οριοθετημένα πλαίσια γύρω από τα αντικείμενα, μαζί με την ετικέτα τους. Σε αντίθεση με την ανίχνευση αντικειμένων στο δισδιάστατο χώρο, παρέχει μια πιο ολοκληρωμένη κατανόηση του σχήματος, του μεγέθους και της θέσης του αντικειμένου στο πραγματικό κόσμο.



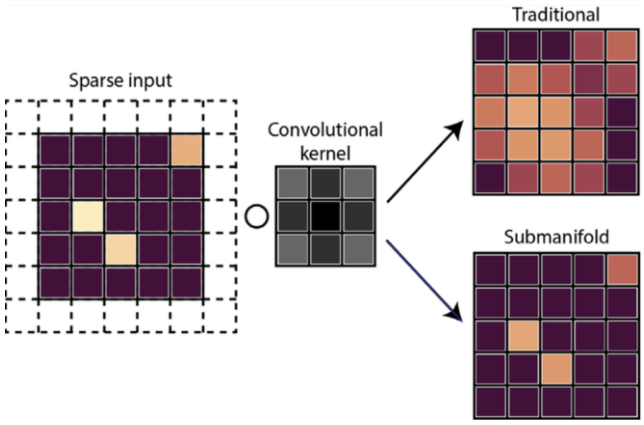
Εικόνα 21. Σημασιολογική κατάτμηση (αριστερά), ανίχνευση αντικειμένων (δεξιά).

Τα τελευταία χρόνια, οι εργασίες αυτές έχουν αναδειχθεί ως κρίσιμοι τομείς έρευνας στην όραση υπολογιστών, με γνώμονα την πρόοδο στη βαθιά μάθηση και την αυξανόμενη διαθεσιμότητα τρισδιάστατων δεδομένων. Αυτές οι εργασίες έχουν σημαντικό ρόλο σε εφαρμογές όπως η αυτόνομη οδήγηση, η επαυξημένη πραγματικότητα, η ρομποτική και άλλα, όπου η κατανόηση του περιβάλλοντος στις τρεις διαστάσεις είναι απαραίτητη. Έτσι, οι μηχανές αρχίζουν να αντιλαμβάνονται και να αλληλεπιδρούν με το περιβάλλον τους παρόμοια με τον άνθρωπο. Στη συνέχεια αναλύονται διαδεδομένες μεθοδολογίες και τεχνικές που χρησιμοποιούνται μέχρι και σήμερα στη σημασιολογική κατάτμηση και ανίχνευση αντικειμένων στο τρισδιάστατο χώρο και αργότερα, οι μέθοδοι που υλοποιήθηκαν στη παρούσα διπλωματική.

4.1 State of the art μέθοδοι στη 3Δ επεξεργασία δεδομένων

Η μεγάλη διαφορά των δεδομένων εικόνας και ενός 3Δ νέφους σημείων, είναι ότι η πρώτη περίπτωση αποτελεί ένα πυκνό (dense) σύνολο δεδομένων, ενώ η δεύτερη περίπτωση ένα αραιό (sparse) σύνολο δεδομένων. Αυτό πρακτικά σημαίνει ότι μια εικόνα έχει σε όλο το σύνολο της τιμές, κάτι που δεν συμβαίνει το ίδιο σε ένα νέφος σημείων, καθώς μεταξύ των σημείων μπορεί να μεσολαβεί χώρος που δεν έχουν τιμές. Για αυτό το λόγο οι τυπικές εφαρμογές πυκνών συνελικτικών δικτύων, όπως αναφέρονται στο προηγούμενο κεφάλαιο, είναι αναποτελεσματικές σε αραιά δεδομένα. Ένα προηγμένο μοντέλο αραιού συνελικτικού δικτύου αποτελεί το Submanifold Sparse Convolutional Network (SSCN) (Graham et al., 2017). Η μέθοδος αυτή είναι συνηθισμένη σε εφαρμογές με 3Δ νέφη σημείων ή πλέγματα voxel (voxel grid), όπου μόνο ένα μικρό μέρος των δεδομένων αντιπροσωπεύει σημαντικές πληροφορίες. Στα κοινά CNN θα εφαρμοζόταν η συνέλιξη και στα κενά κελιά, με αποτέλεσμα ένα μεγάλο μέρος των υπολογισμών να δεσμεύεται σε άδεια κελιά. Δηλαδή, το φίλτρο θα ολισθαίνει σε ολόκληρο το πλέγμα εισόδου σε κάθε θέση και θα υπολογίζει την έξοδο της περιοχής που καλύπτει. Στα SSCNs παρακάμπτονται αυτά τα κενά κελιά και εκτελεί τις λειτουργίες μόνο στις μη μηδενικές τιμές με αποτέλεσμα να μειώνεται αρκετά το υπολογιστικό κόστος. Η μέθοδος αυτή είναι κατάλληλη για εργασίες που περιλαμβάνουν μεγάλους τρισδιάστατους χώρους και η διατήρηση ενός αραιού νέφους δεδομένων βοηθά στην αποτελεσματικότητα όσον αφορά τη μνήμη και τους υπολογισμούς. Άλλη μια παρόμοια μέθοδος που διαχειρίζεται αραιά νέφη σημείων αποτελεσματικά, είναι το Minkowski Convolutional Neural Networks

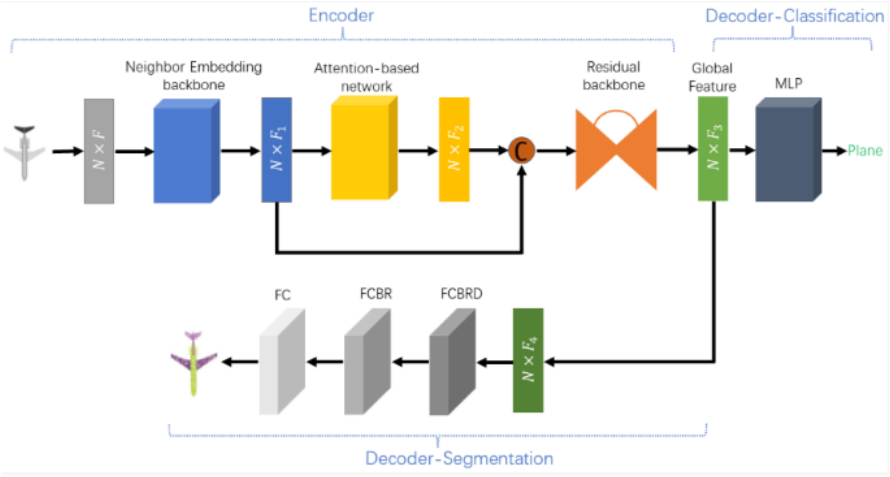
(Choy et al., 2019). Παραπάνω πλεονέκτημα της είναι ότι μπορεί να χειριστεί 4Δ χωροχρονικά δεδομένα αλλά και γενικότερα πολυδιάστατες δομές δεδομένων. Τα δεδομένα εισόδου που μπορεί να είναι σημεία με συντεταγμένες (x,y,z) και διάφορα χαρακτηριστικά (intensity, r, g, b), εισάγονται στο αραιό νευρωνικό δίκτυο και υπολογίζονται μόνο οι μη κενές περιοχές. Καθώς προχωρά σε ποιο βαθιά επίπεδα του νευρωνικού, κάθε επίπεδο εφαρμόζει συνέλιξη στα χαρακτηριστικά εισόδου, μετατρέποντας τα σε πιο σύνθετα χαρακτηριστικά, βελτιστοποιώντας σταδιακά τις πληροφορίες, ενώ η αραιή δομή βοηθά στην διατήρηση της αποτελεσματικότητας.



Εικόνα 22. Διαφορά κοινού CNN με ένα Sparse Conv Net με χρήση φίλτρου.

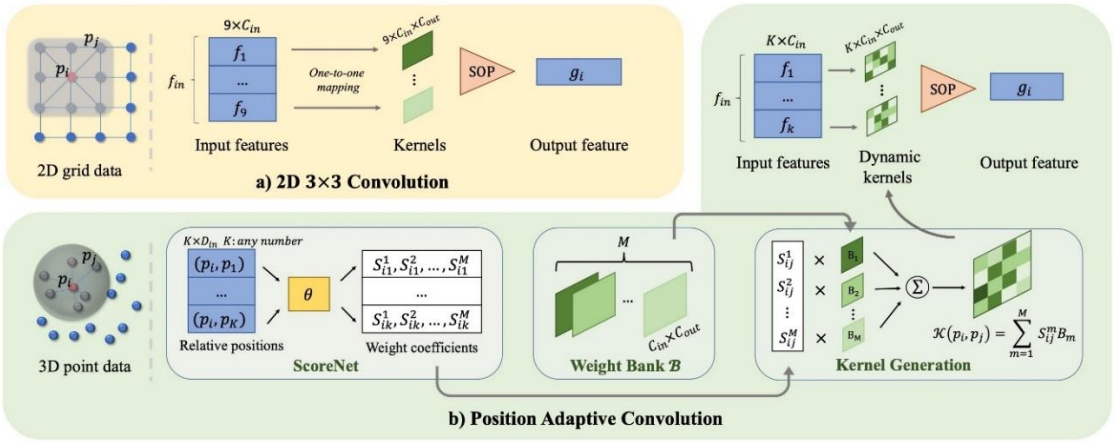
Ένα άλλο διαδεδομένο νευρωνικό δίκτυο που με τον καιρό έχουν αναπτυχθεί βελτιωμένες εκδώσεις του αφορά το Point Transformer (Zhao et al., 2020). Οι transformers αποτελούν ένα τύπο μοντέλου βαθιάς μάθησης που σχεδιάστηκε αρχικά για δεδομένα ακολουθίας (sequence data), όπως είναι εργασίες επεξεργασίας φυσικής γλώσσας (Natural Language Processing), που μπορούν να μεταφράζουν ή να συνοψίζουν ένα κείμενο. Το βασικό στοιχείο των συγκεκριμένων δικτύων είναι ο self-attention μηχανισμός, ο οποίος επιτρέπει στο μοντέλο να σταθμίζει δυναμικά τη σημασία διαφορετικών τμημάτων των δεδομένων εισόδου. Ουσιαστικά, οι transformers επεξεργάζονται ταυτόχρονα ολόκληρη την είσοδο και καταγράφουν σχέσεις μεταξύ όλων των δεδομένων, ανεξάρτητα τη θέση τους. Δηλαδή, ο μηχανισμός αυτός χρησιμοποιείται για την εκμάθηση τοπικών αλλά και καθολικών χαρακτηριστικών, τα οποία είναι κρίσιμα για την κατανόηση μιας τρισδιάστατης δομής. Χρησιμοποιώντας μια μορφή κωδικοποίησης (encoding) για να καταγράψει τις χωρικές σχέσεις μεταξύ των σημείων, βοηθάει το δίκτυο να κατανοήσει τις

σχέσεις μεταξύ των σημείων στον χώρο. Η επιρροή που μπορεί να έχει ένα σημείο σε σχέση με κάποιο γειτονικό, υπολογίζεται με ένα σκορ (attention score) που συνδυάζει τα χαρακτηριστικά των σημείων με τη σχετική τους θέση. Μόλις υπολογιστούν αυτές οι βαθμολογίες, τα χαρακτηριστικά των γειτονικών σημείων μετατρέπονται με βάση αυτές τις βαθμολογίες. Τέλος, ο Point Transformer χρησιμοποιώντας ιεραρχικές δομές, όπως pooling στα συνελικτικά νευρωνικά δίκτυα, για τη μείωση του χωρικού δείγματος (downsample), καταγράφοντας έτσι χαρακτηριστικά υψηλότερου επιπέδου, δηλαδή από μικρότερες σε ευρύτερες λεπτομέρειες.



Εικόνα 23. Υπόδειγμα Point Transformer για Classification (πάνω δεξιά) και Segmentation (κάτω).

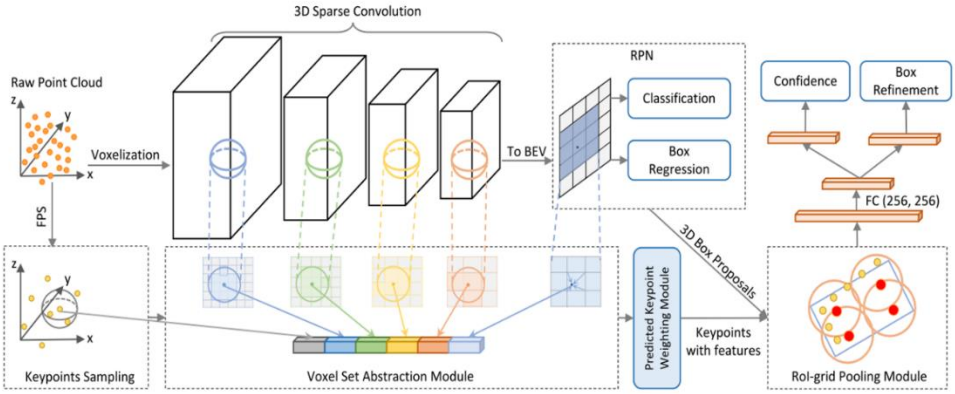
Άλλη μια διαδεδομένη μέθοδος που βασίζεται στα σημεία είναι η PAconv (Position Adaptive Convolution) (M. Xu et al., 2021). Σε σχέση με τα συνελικτικά δίκτυα στο 2Δ χώρο που χρησιμοποιείται ένα σταθερό φίλτρο, εδώ η συνέλιξη προσαρμόζεται συνεχώς ανάλογα τη διάταξη των σημείων. Έτσι, το μοντέλο μπορεί να καταγράφει γεωμετρικές λεπτομέρειες, όπως το σχήμα και το προσανατολισμό των αντικειμένων. Με βάση τις σχετικές θέσεις των σημείων, δημιουργούνται προσαρμοστικοί συνελικτικοί πυρήνες (adaptive convolutional kernels), που όπως προαναφέρθηκε θα είναι διαφορετικοί σε κάθε σημείο και στη συνέχεια ακολουθεί η συνάθροιση των χαρακτηριστικών που είναι παρόμοια με τα απλά συνελικτικά δίκτυα.



Εικόνα 24. Υπόδειγμα PAconv (πράσινο) με χρήση σχετικών θέσεων (pi, pk) και προσαρμοστικών πυρήνων K, σε σχέση με την 2Δ συνέλιξη.

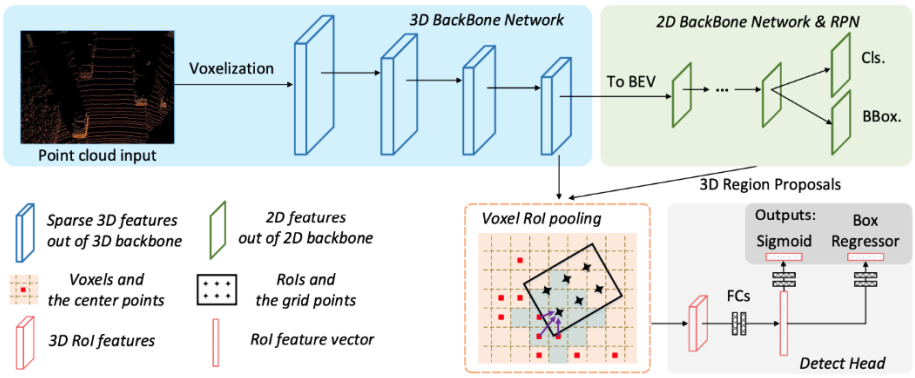
Αυτά ήταν κάποια από τα δημοφιλέστερα μοντέλα που εφαρμόζονται στη 3Δ σημασιολογική κατάτμηση που εστιάζουν στη ακριβή ταξινόμηση των σημείων. Στο πεδίο της 3Δ ανίχνευσης αντικειμένων σκοπός είναι η δημιουργία και βελτίωση του πλαισίου οριοθέτησης ενός αντικειμένου, μαζί με την κατηγορία στην οποία ανήκει. Λόγω του ότι τα 3Δ νέφη σημείων είναι συχνά αραιά, σε μεγάλες αποστάσεις καταλαβαίνει κανείς ότι είναι δύσκολη η καταγραφή επαρκών λεπτομερειών για την ανίχνευση αντικειμένων. Μια μέθοδος που μπορεί να ξεπεράσει αυτό το πρόβλημα είναι το Point-Voxel Region CNN (PV-RCNN) (Shi et al., 2019). Χρησιμοποιείται ένας συνδυασμός voxel-based και point-based μεθόδων, αρχικά για την αποτύπωση χονδρικών καθολικών πλαισίων από τα αραιά νέφη σημείων και εν συνεχεία για τις λεπτομερείς πληροφορίες από τα πρωτογενή σημεία αντίστοιχα. Κατά τη διαδικασία του voxelization, τα σημεία εισόδου διαιρούνται στο χώρο σε ένα πλέγμα voxel. Σε αυτά τα voxel υλοποιείται μια αραιή 3Δ συνέλιξη, ώστε να γίνει εξαγωγή των χαρακτηριστικών τους. Η έξοδος που δίνει είναι ένα voxel feature map όπου κωδικοποιεί χονδρικές χωρικές πληροφορίες της σκηνής. Στη συνέχεια το μοντέλο δημιουργεί προτεινόμενες 3Δ περιοχές με τη βοήθεια ενός Region Proposal Network (RPN), ένα σημαντικό στοιχείο του δικτύου για την πρόβλεψη πλαισίων οριοθέτησης υποψηφίων αντικειμένων ενδιαφέροντος. Η διαδικασία αυτή γίνεται ταυτόχρονα με τη συνέλιξη, καθώς κάθε σημείο του feature map θεωρείται σημείο “άγκυρα”. Το δίκτυο χρησιμοποιεί προκαθορισμένα πλαίσια αγκύρωσης σε διάφορες θέσεις των voxel feature map, κλίμακες και προσανατολισμούς. Έτσι, για κάθε πλαίσιο αγκύρωσης παράγεται μια τιμή πρόβλεψης για το εάν υπάρχει ή όχι αντικείμενο εντός του πλαισίου και προσαρμογές της θέσης, του μεγέθους και του προσανατολισμού του πλαισίου για καλύτερη

προσαρμογή. Η point-based μέθοδο, επιλέγει δειγματοληπτικά σημεία κλειδιά (πχ farthest point sampling, random sampling) που θα καλύπτουν ολόκληρη τη σκηνή και με τη χρήση μιας λειτουργίας συγκέντρωσης χαρακτηριστικών (Voxel Set Abstraction), κάθε σημείο (τοπικές λεπτομέρειες) συγκεντρώνει τα χαρακτηριστικά από τα κοντινά voxel (καθολικό πλαίσιο). Το αποτέλεσμα είναι ένα σύνολο διανυσμάτων χαρακτηριστικών για κάθε σημείο, το οποίο ενσωματώνει τις τοπικές λεπτομέρειες από τα αρχικά σημεία αλλά και το πλαίσιο των voxel χαρακτηριστικών. Ο συνδυασμός των δυο προηγούμενων, δηλαδή οι προτεινόμενες περιοχές μαζί με την εξαγωγή διανυσμάτων των χαρακτηριστικών, δίνει το λεγόμενο Region of Interest Pooling (RoI Pooling) και αποτελεί έναν πιο δομημένο τρόπο οριοθέτησης και αναπαράστασης των πλαισίων. Το δίκτυο αποτελεί ένα Fully Connected Layer ώστε να βελτιώνονται τα RoI με τον καιρό και ταυτόχρονα το δίκτυο να κάνει πρόβλεψη ταξινόμησης για τη κατηγορία του αντικειμένου.



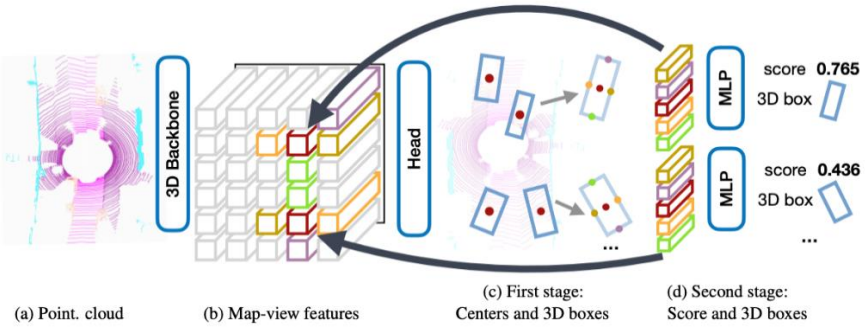
Εικόνα 25. Αρχιτεκτονική μοντέλου PV-RCNN.

Παρόμοια μέθοδος είναι η Voxel R-CNN (Deng et al., 2020), όπου μόνη διαφορά της είναι ότι εδώ εστιάζει μόνο στην επεξεργασία voxel-based και όχι στη point-based όπως στο PV-RCNN. Μπορεί να υστερεί σε ακρίβεια από τη PV-RCNN, αλλά στη πράξη το συγκεκριμένο μοντέλο έχει μεγαλύτερη απλότητα και η αποτελεσματικότητα το καθιστά ιδιαίτερα ανταγωνιστικό και δημοφιλές μοντέλο.



Εικόνα 26. Αρχιτεκτονική μοντέλου Voxel R-CNN.

Εκτός της μεθόδου αγκύρωσης, υπάρχει και η μέθοδοι ανίχνευσης βάση κέντρου (center-based) με πιο διαδεδομένη τη CenterPoint (Yin et al., 2020). Σε αυτή τη περίπτωση, αντί να προσπαθεί το μοντέλο να ανιχνεύσει ολόκληρο το αντικείμενο ταυτόχρονα, εστιάζει στον εντοπισμό του κέντρου κάθε αντικειμένου και στη συνέχεια υπολογίζει τα υπόλοιπα (πχ μέγεθος, σχήμα, προσανατολισμός), έτσι η διαδικασία να γίνεται πιο απλή, γρήγορη και ακριβής. Το πρώτο στάδιο της συγκεκριμένης μεθόδου είναι να κάνει voxelization και να το προβάλει σε ένα 2Δ πλέγμα. Η διαδικασία αυτή γνωστή ως Bird's-eye view (BEV), διατηρεί τις χωρικές πληροφορίες και επιτρέπει τη χρήση 2Δ συνελικτικών νευρωνικών δικτύων. Να σημειωθεί ότι σε κάθε κελί του πλέγματος του BEV, αντιστοιχεί σε μια περιοχή στον τρισδιάστατο χώρο που περιέχει συγκεντρωτικά χαρακτηριστικά των σημείων. Κατά την εκπαίδευση, με τη χρήση Gaussian heatmaps, το μοντέλο προβλέπει τη τιμή κορυφής που θα υποδεικνύει το ακριβές κέντρο του αντικειμένου. Επιπλέων στα θετικά, από τη στιγμή που σκοπός είναι ο εντοπισμός του κέντρου του αντικειμένου, καταλαβαίνει κανείς ότι έτσι αντιμετωπίζεται και το πρόβλημα της μερικής απόκρυψης ενός αντικειμένου.



Εικόνα 27. Αρχιτεκτονική μοντέλου CenterPoint.

4.2 Ανάλυση μοντέλου σημασιολογικής κατάτμησης RandLA-Net και παραμετροποίηση

Η πρώτη μέθοδος που χρησιμοποιήθηκε στη παρούσα διπλωματική αφορά τη RandLA-Net (Random Sampling and Local Aggregation Network), που αποτελεί μια αρχιτεκτονική νευρωνικού δικτύου κατάλληλη για σημασιολογική κατάτμηση σε μεγάλης κλίμακας νέφη σημείων (Hu et al., 2019). Η κύρια πρόκληση στην επεξεργασία 3Δ νέφους σημείων είναι ο αποτελεσματικός χειρισμός του μεγάλου αριθμού σημείων. Οι παραδοσιακές μέθοδοι που χρησιμοποιούνται σε αντίστοιχες εργασίες, είτε δυσκολεύονται με την υπολογιστική αποτελεσματικότητα είτε αποτυγχάνουν να συλλάβουν επαρκείς τοπικές γεωμετρικές λεπτομέρειες. Το μοντέλο RandLA-Net μπορεί να αντιμετωπίσει αυτές τις προκλήσεις εισάγοντας μια νέα προσέγγιση που συνδυάζει τυχαία δειγματοληψία με τοπική συγκέντρωση χαρακτηριστικών. Η βασική ιδέα είναι να μειωθεί ο αριθμός των σημείων κατά την επεξεργασία, διασφαλίζοντας ταυτόχρονα ότι τα υπόλοιπα σημεία εξακολουθούν να φέρουν αρκετές πληροφορίες για την επίτευξη υψηλής ακρίβειας. Αυτό καθιστά το μοντέλο ιδιαίτερα κατάλληλο για εφαρμογές σε πραγματικό χρόνο, όπου τόσο η ταχύτητα όσο και η ακρίβεια είναι σημαντικές.

Ο μηχανισμός του δικτύου ξεκινά πρώτα με τη χρήση τυχαίας δειγματοληψίας για να μειώσει τον αριθμό των σημείων που υποβάλλονται σε επεξεργασία σε κάθε επίπεδο του δικτύου. Το πλεονέκτημα της είναι ότι επιλέγει ομοιόμορφα τα σημεία από το αρχικό σύνολο, διασφαλίζοντας ότι το υποσύνολο είναι αντιπροσωπευτικό και σε σύγκριση με άλλες μεθόδους δειγματοληψίας έχει υψηλότερη υπολογιστική απόδοση, ανεξάρτητα του μεγέθους του νέφους σημείων, καθώς χρειάζεται μόνο 0,004 δευτερόλεπτα για την επεξεργασία 106 σημείων. Στη συνέχεια, χρησιμοποιώντας μια σειρά λειτουργιών, το μοντέλο προσπαθεί να καταγράψει και συγκεντρώσει τοπικά γεωμετρικά χαρακτηριστικά από γειτονικά σημεία. Με τη χρήση μιας τοπικής χωρικής κωδικοποίησης (Local spatial encoding) σε ένα σημείο i του υποσυνόλου, με χρήση της μεθόδου K-nearest neighbors (KNN), συλλέγονται τα γειτονικά σημεία. Από αυτή τη διαδικασία μπορεί να υπολογιστούν οι σχετικές θέσεις των γειτονικών σημείων μέσω της σχέσης:

$$\boldsymbol{r_i^k = MLP\left(p_i \oplus p_i^k \oplus \left(p_i - p_i^k\right) \oplus ||p_i - p_i^k||\right)} \qquad \textbf{(4.1)}$$

όπου p_i και p_i^k είναι η θέση (x, y, z συντεταγμένες) του σημείου ενδιαφέροντος σε σχέση με τα γειτονικά του σημεία αντίστοιχα και $||p_i - p_i^k|$ η ευκλείδεια απόσταση. Όλα αυτά τα στοιχεία συνδέονται σε ένα διάνυσμα, όπου με τη σειρά του περνά από ένα νευρωνικό δίκτυο MLP και να παραχθεί μια νέα κωδικοποιημένη θέση r_i^k στο οποίο θα συνδεθούν όλα τα χαρακτηριστικά των γειτονικών σημείων (πχ χρώμα, ένταση) διανύσματος f_i^k. Έτσι, πλέον η πληροφορία που έχει συλλεχθεί δεν είναι μόνο τα χαρακτηριστικά των γειτονικών σημείων, αλλά και πως είναι τοποθετημένο το σημείο σε σχέση με τα γειτονικά του. Με αυτό τον τρόπο μπορεί το νευρωνικό να κατανοεί καλύτερα τη τοπική γεωμετρική δομή γύρω από κάθε σημείο. Ακολουθεί ένας μηχανισμός attentive pooling ώστε το δίκτυο να εστιάσει στα πιο σχετικά ή σημαντικά από τα γειτονικά σημεία. Αυτό επιτυγχάνεται μέσω μιας συνάρτησης που αποτελεί ένα μικρό νευρωνικό δίκτυο MLP, που ακολουθείται από μια λειτουργία softmax, για τον υπολογισμό της βαθμολογίας κάθε χαρακτηριστικού σημείου.

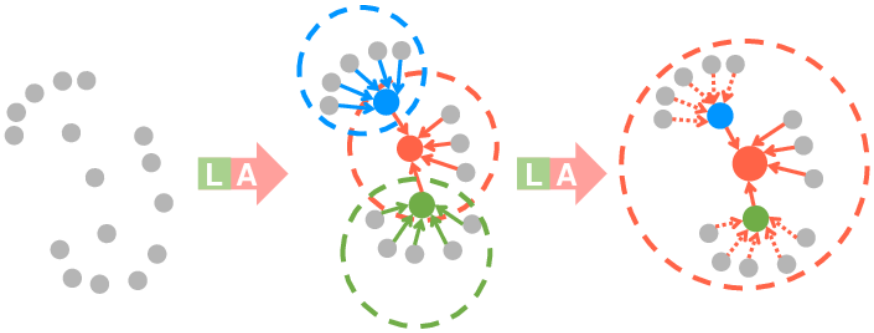
$$\boldsymbol{s_i^k = g(\hat{f}_i^k, W)} \qquad \textbf{(4.2)}$$

όπου $g()$ η συνάρτηση, $\hat{f}_i^k$ τα χαρακτηριστικά των τοπικών σημείων και W τα βάρη που μαθαίνει το νευρωνικό MLP. Οι βαθμολογίες αυτές λειτουργούν σαν “μάσκα” που επιλέγει αυτόματα τα σημαντικά χαρακτηριστικά, για αυτό το λόγο όλα τα χαρακτηριστικά των γειτονικών σημείων πολλαπλασιάζονται με τις αντίστοιχες βαθμολογίες τους και συνοψίζονται για να δημιουργηθεί ένα νέο, πιο κατατοπιστικό διάνυσμα χαρακτηριστικών.

$$\tilde{\boldsymbol{f}}_{\boldsymbol{i}} = \sum_{\boldsymbol{k=1}}^{\boldsymbol{K}} (\hat{\boldsymbol{f}}_{\boldsymbol{i}}^{\boldsymbol{k}} \cdot \boldsymbol{s}_{\boldsymbol{i}}^{\boldsymbol{k}}) \qquad \textbf{(4.3)}$$

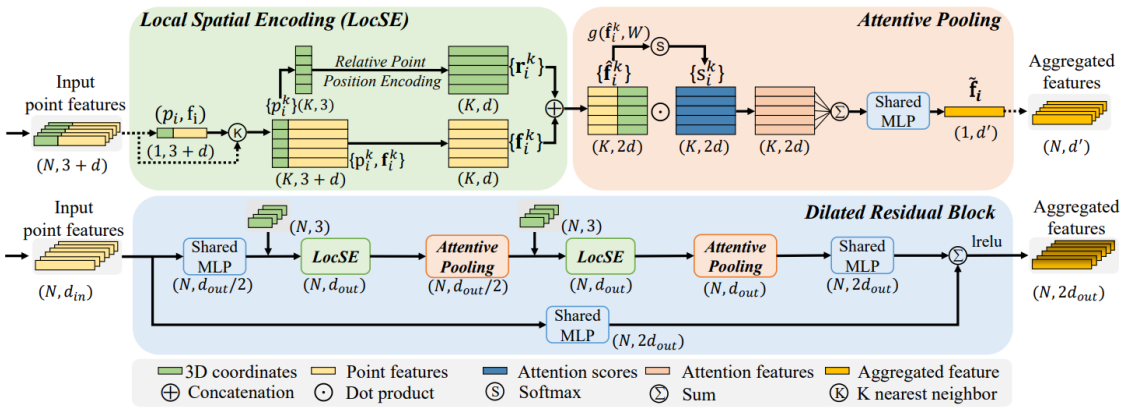
Στο τελικό στάδιο του μοντέλου χρησιμοποιείται μια μέθοδος αύξησης του δεκτικού πεδίου (receptive field) για κάθε σημείο, καθώς στα νέφη σημείων γίνεται μείωση δειγματοληψίας. Έτσι, ακόμη και στη περίπτωση που χαθούν κάποια σημεία, οι γεωμετρικές λεπτομέρειες να μπορούν να διατηρηθούν. Το δεκτικό πεδίο στα νευρωνικά δίκτυα αναφέρεται σε πόσα δεδομένα εισόδου επηρεάζει το χαρακτηριστικό ενός σημείου. Ένα μεγαλύτερο δεκτικό πεδίο σημαίνει ότι το σημείο επηρεάζεται από μεγαλύτερη περιοχή του νέφους, επιτρέποντας στο δίκτυο να συλλάβει περισσότερες γεωμετρικές λεπτομέρειες και χαρακτηριστικά. Εμπνευσμένο από ResNet (He et al., 2016), χρησιμοποιείται η μέθοδος Dilated Residual Block, όπου στοιβάζει πολλαπλές LocSE και

Attentive pooling λειτουργείες με μια skip συνδεσιμότητα (συνδέσεις που παρακάμπτουν ένα ή περισσότερα επίπεδα και προσθέτουν την είσοδο τους απευθείας σε μεταγενέστερο επίπεδο). Με αυτό τον τρόπο επιτυγχάνεται γρήγορα και αποτελεσματικά η αύξηση του δεκτικού πεδίου διατηρώντας τη χωρική ανάλυση και καταγράφοντας λεπτομέρειες σε πολλαπλές κλίμακες.



Εικόνα 28. Απεικόνιση του dilated residual block με αύξηση του δεκτικού πεδίου

Στη πράξη αποδείχτηκε ότι δυο στοίβες LocSE και Attentive pooling είναι αρκετές για να επιτευχθεί μια ικανοποιητική ισορροπία μεταξύ αποτελεσματικότητας και αποδοτικότητας, καθώς παραπάνω θα δημιουργούσε προβλήματα υπερπροσαρμογής του μοντέλου και το υπολογιστικά θα ήταν πιο απαιτητικό.



Εικόνα 29. Αρχιτεκτονική RandLA-Net.

Για τον ορισμό ενός νευρωνικού δικτύου, στην αρχή ορίζεται ο αριθμός των επιπέδων. Το βάθος αυτό καθορίζει την ικανότητα του δικτύου να μαθαίνει πολύπλοκα χαρακτηριστικά. Ένα πολυεπίπεδο δίκτυο μπορεί να εντοπίζει πολύπλοκες γεωμετρίες, αλλά υπολογιστικά θα είναι πολύ απαιτητικό. Για το downsampling κάθε επιπέδου (dilated residula block), ώστε να γίνεται μείωση της χωρικής ανάλυσης και να εντοπίζονται αυτές οι πολύπλοκες γεωμετρίες, ορίζεται μια αναλογία υποβάθμισης. Στα μεγαλύτερα επίπεδα του νευρωνικού, όπου οι γεωμετρίες είναι πιο πολύπλοκες, το διάνυσμα των χαρακτηριστικών για κάθε σημείο χρειάζεται να είναι μεγαλύτερο, για αυτό το λόγο προσδιορίζονται εξαρχής για κάθε επίπεδο το μέγεθος του διανύσματος των χαρακτηριστικών. Αυτό το διάνυσμα στην αρχή αποτελείται από τα χαρακτηριστικά των δεδομένων, όπως έχουν εξαχθεί από τη συλλογή ή από κάποια επιπλέον προεπεξεργασία. Συνήθως, αποτελούν τις συντεταγμένες θέσης και μερικές φορές μπορεί να συνδυάζονται με την ένταση της δέσμης λέιζερ ή και των χρωμάτων. Αφού εισέλθουν στο δίκτυο και πριν ολοκληρωθεί το πρώτο επίπεδο, μπορεί να ορισθεί πόσα χαρακτηριστικά θα υπολογισθούν κατά τη πρώτη επεξεργασία (πχ από aggregation γειτονικών σημείων).

Εκτός από την μείωση της χωρικής ανάλυσης που χρησιμοποιείται στα κρυφά επίπεδα, κατά την είσοδο των δεδομένων γίνεται η τυχαία επιλογή σημείων ώστε να μειωθεί αρκετά η επεξεργασία των δεδομένων, αλλά διατηρώντας ταυτόχρονα και τη συνοχή των αντικειμένων. Από αυτή τη μείωση δεδομένων επιλέγεται ένας συγκεκριμένος αριθμός σημείων που θα επεξεργαστεί το μοντέλο. Έτσι, το μοντέλο έχει καταλήξει ποια δεδομένα θα επεξεργαστεί και ξεκινά η διαδικασία του LocSE και Attentive Pooling όπου ορίζεται το πλήθος των γειτόνων που προσάπτουν στο κάθε σημείο. Τέλος, ορίζονται το πλήθος των κλάσεων που καλείται να εντοπίζει το δίκτυο αλλά και η κλάση που θα αποτελείται από όλα τα σημεία όπου μετά τη πρόβλεψη δεν θα αντιστοιχούν σε καμία από τις υπόλοιπες κλάσεις, ώστε να μην συμπεριλαμβάνονται στην αξιολόγηση.

Γίνεται αντιληπτή η προσπάθεια του συγκεκριμένου μοντέλου να μειώσει όσο το δυνατόν περισσότερο τα δεδομένα επεξεργασίας, καθώς σκοπός του είναι η αποτελεσματική πρόβλεψη σε νέφη σημείων μεγάλης κλίμακας. Ακολουθεί αναλυτικός πίνακας των παραμέτρων του μοντέλου στα σύνολα δεδομένων SemanticKITTI και Toronto-3D.

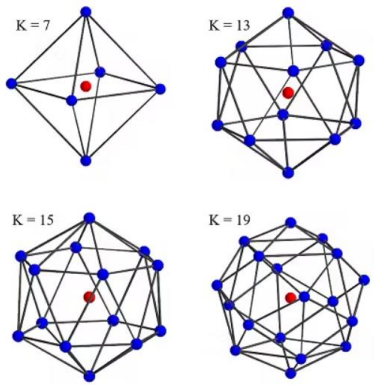
Παράμετρος	SemanticKITTI	Toronto-3D
num_layers	4	5
num_neighbors	16	16
num_points	45056	65536
num_classes	19	8
ignored_label_inds	[0]	[0]
sub_sampling_ratio	[4, 4, 4, 4]	[4, 4, 4, 4, 2]
in_channels	3	6
dim_features	8	8
dim_output	[16, 64, 128, 256]	[16, 64, 128, 256, 512]
grid_size	0.06	0.05

Πίνακας 2. Πίνακας με τις υπερπαραμέτρους μοντέλου RandLA-Net.

4.3 Ανάλυση μοντέλου σημασιολογικής κατάτμησης KPConv και παραμετροποίηση

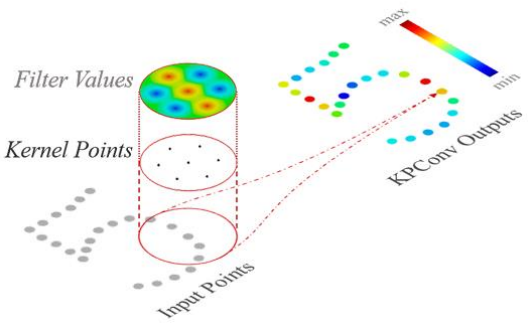
Η επόμενη μέθοδος σημασιολογικής κατάτμησης που εφαρμόστηκε είναι η KPConv (Kernel Point Convolution) (Thomas et al., 2019). Η μέθοδος είναι εμπνευσμένη από τη συνέλιξη εικόνων, με τη διαφορά ότι εδώ ορίζονται πυρήνες χώρου (spatial kernels) για τον καθορισμό της περιοχής συνέλιξης που μπορούν να εφαρμοστούν απευθείας στα νέφη σημείων, επιτρέποντας στο δίκτυο να μαθαίνει χωρικά χαρακτηριστικά από τα δεδομένα. Και οι δυο περιπτώσεις έχουν μια κοινή ιδιότητα που είναι απαραίτητη για την συνέλιξη. Αυτή αποτελεί τον χωρικό εντοπισμό όπου σε ένα πλέγμα τα χαρακτηριστικά εντοπίζονται από τον δείκτη τους σε έναν πίνακα, ενώ στα νέφη σημείων από τις συντεταγμένες. Η διαφορά είναι ότι οι εικόνες χρησιμοποιούν ένα σταθερό πλέγμα, ενώ η μέθοδος KPConv χρησιμοποιεί ένα σύνολο σημείων πυρήνα (kernel points) που κατανέμονται στο χώρο γύρο από κάθε σημείο εισόδου. Αυτά τα kernel points καθορίζουν που εφαρμόζονται τα συνελικτικά βάρη, παρόμοια με τα φίλτρα στα CNN. Επιλέγοντας μια σταθερή ακτίνα

γύρω από ένα σημείο, ορίζεται μια σφαίρα στην οποία βρίσκονται όλα τα γειτονικά σημεία.



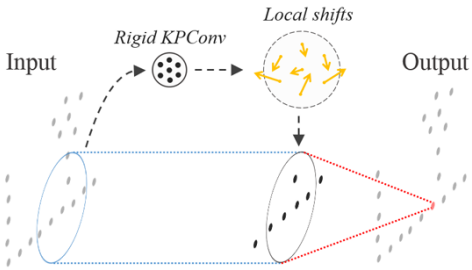
Εικόνα 30. Τα μπλε σημεία αποτελούν τα kernel points και με κόκκινο το κέντρο της σφαίρας. Το κεντρικό σημείο επίσης αποτελεί kernel point.

Τα kernel points επίσης τοποθετούνται σε αυτή τη σφαίρα όπου κάθε τέτοιο σημείο προσπαθεί να είναι όσο πιο μακριά από τα υπόλοιπα και γύρο από το κεντρικό δεδομένο σημείο, όπως φαίνεται στην Εικόνα 30. Με τη χρήση πολλαπλών kernel points επιτυγχάνεται συλλογή πληροφορίας προς όλες τις κατευθύνσεις του σημείου, ενώ με τη κανονική διάταξη παράγεται ομοιόμορφη ανάλυση προς κάθε κατεύθυνση. Κάθε kernel point αποτελείται από ένα σύνολο βαρών τα οποία εφαρμόζονται στα γειτονικά σημεία σε σχέση με την απόσταση τους. Έτσι, κάθε kernel point έχει μια περιοχή επιρροής που ορίζεται από μια συνάρτηση συσχέτισης. Αυτή η συνάρτηση συσχέτισης καθορίζει εάν ένα kernel point επηρεάζει ένα γειτονικό σημείο και πόσο συμβάλει αυτό το γειτονικό σημείο στον υπολογισμό της τιμής του kernel.



Εικόνα 31. Αναπαράσταση 2Δ kernel points με αντίστοιχες τιμές βαρών.

Με αυτόν τον τρόπο τα χαρακτηριστικά των δεδομένων σημείων υπολογίζονται από το άθροισμα των χαρακτηριστικών όλων των γειτόνων, πολλαπλασιαζόμενο με τη συσχέτιση και τα βάρη τους. Κάθε kernel point έχει ένα σύνολο βαρών με σχήμα $[D_{in},D_{out}]$ όπου αφορά τον αριθμό των χαρακτηριστικών του σημείου στο τρέχον επίπεδο και τον αριθμό των χαρακτηριστικών που εξάγονται για το επόμενο επίπεδο. Μετά από κάθε πέρασμα από το συγκεκριμένο επίπεδο του νευρωνικού, τα χαρακτηριστικά των σημείων ενημερώνονται και χρησιμοποιούνται για τον υπολογισμού του επόμενου επιπέδου KPConv. Αυτή η διαδικασία αποτελεί την άκαμπτη (rigid) kernel point, αλλά υπάρχει και η παραμορφώσιμη (deformable) εκδοχή όπου το δίκτυο μαθαίνει και δημιουργεί τοπικές μετατοπίσεις των kernel points, οι οποίες προστίθενται σε κάθε θέση για τον προσδιορισμό της νέας.



Εικόνα 32. 2Δ αναπαράσταση deformable kernel.

Αυτή η προσαρμοστικότητα επιτρέπει στη συνέλιξη να εντοπίσει πιο περίπλοκα και ακανόνιστα σχήματα. Άρα είναι και πιο πολύπλοκη σε σχέση με την rigid, καθώς απαιτούνται επιπρόσθετες παραμέτρους και υπολογισμούς για την προσαρμογή των kernel points. Αυτή η πολυπλοκότητα οδηγεί σε υψηλότερο υπολογιστικό κόστος και μεγαλύτερους χρόνους εκπαίδευσης. Για αυτό το λόγο σε περιπτώσεις που περιέχονται πολύπλοκες γεωμετρίες προτιμάται η deformable συνέλιξη, ενώ σε περιπτώσεις πιο απλών γεωμετριών, όπως ένα ομοιόμορφο περιβάλλον μιας αστικής περιοχής, η rigid συνέλιξη μπορεί να είναι επαρκής.

Ο λόγος που επιλέγεται η ακτίνα για την δημιουργία μιας σφαίρας που θα περιλαμβάνει τη περιοχή ενδιαφέροντος είναι διότι αυτός ο τρόπος παρέχει πιο συνεπείς και προσαρμοστικές τοπικές γειτονίες, ειδικά σε νέφη σημείων όπου η πυκνότητα των σημείων είναι ακανόνιστη. Η επιλογή βάσει ακτίνας διασφαλίζει ότι λαμβάνονται υπόψη μόνο σημεία σε μια ορισμένη απόσταση από το κέντρο, γεγονός που οδηγεί σε καλύτερη τοπική εξαγωγή χαρακτηριστικών, σε σύγκριση με το KNN, το οποίο μπορεί να επιλέξει

γείτονες που είναι πολύ μακριά ή πολύ κοντά ανάλογα με την κατανομή των σημείων. Στη πιο αναλυτική λειτουργία του δικτύου, ξεκινώντας από τον ορισμό σημείου x_i του νέφους σημείων $\mathcal{P} \in \mathbb{R}^{N\times 3}$ και f_i των αντίστοιχων χαρακτηριστικών με $\mathcal{F} \in \mathbb{R}^{N\times D}$. Ο γενικός τύπος συνέλιξης του $\mathcal{F}$ από ένα kernel g σε ένα σημείο $x \in \mathbb{R}^3$ ορίζεται ως:

$$(\boldsymbol{\mathcal{F}} * \boldsymbol{g})(\boldsymbol{x}) = \sum_{\boldsymbol{x_i}\in \boldsymbol{N_x}} \boldsymbol{g}(\boldsymbol{x_i} - \boldsymbol{x})\boldsymbol{f_i} \qquad \textbf{(4.4)}$$

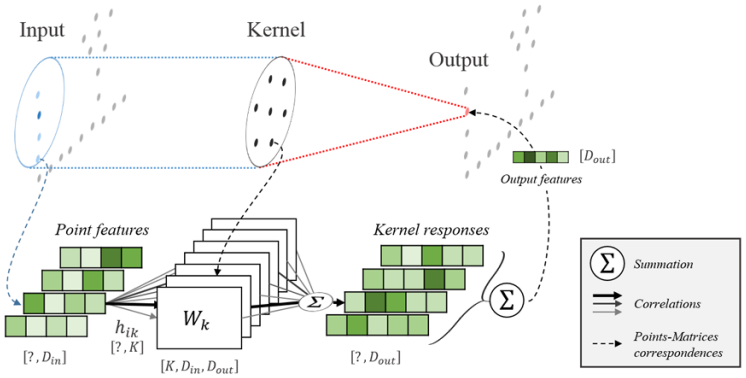
Όπου οι γείτονες ορίζονται ως η αφαίρεση των σημείων από το κεντρικό που υπάρχουν εντός της ακτίνας, $N_x \{x_i \in \mathcal{P} \mid \|x_i - x\| \leq r\}$ με $r \in \mathbb{R}$. Έστω ότι οι γείτονες ορίζονται ως $x_i - x \leq y$, οπότε ορίζεται το πεδίο ορισμού της συνάρτησης g ως σφαίρα με ακτίνα r, $B_r^3 = \{y \in \mathbb{R}^3 \mid \|y\| \leq r\}$. Έτσι, τα kernel points περιορίζονται μόνο σε αυτά εντός τις σφαίρας και ορίζονται μαζί με τους αντίστοιχους πίνακες βαρών. Δηλαδή, $\{\widetilde{x_k} \mid k < K\} \subset B_r^3$ και $\{W_k \mid k < K\} \subset \mathbb{R}^{D_{in}\times D_{out}}$ με K ο μέγιστος αριθμός kernel points. Τελικά, η συνάρτηση g ορίζεται ως:

$$\boldsymbol{g}(\boldsymbol{y_i}) = \sum_{\boldsymbol{k<K}} \boldsymbol{h}(\boldsymbol{y_i}, \widetilde{\boldsymbol{x_k}})\boldsymbol{W_k} \qquad \textbf{(4.5)}$$

Όπου h είναι η συσχέτιση μεταξύ του y_i και $\widetilde{x_k}$. Όσο πιο κοντά είναι το kernel point σε ένα γειτονικό σημείο, τόσο μεγαλύτερη θα είναι και η συσχέτιση. Υπολογίζεται μέσω μιας γραμμικής συσχέτισης από το τύπο:

$$\boldsymbol{h}(\boldsymbol{y_i}, \widetilde{\boldsymbol{x_k}}) = \mathbf{max}(\mathbf{0}, \mathbf{1} - \frac{\|\boldsymbol{y_i} - \widetilde{\boldsymbol{x_k}}\|}{\boldsymbol{\sigma}}) \qquad \textbf{(4.6)}$$

Όπου σ είναι μια υπερπαράμετρος που δείχνει την απόσταση επιρροής των kernel points και επιλέγεται ανάλογα τη πυκνότητα του νέφους σημείων.



Εικόνα 33. 2Δ αναπαράσταση KPConv.

Στη deformable εκδοχή, τη πρώτη φορά χρησιμοποιείται η rigid και στη συνέχεια μαθαίνει τις K μετατοπίσεις $\Delta(x)$.

$$(\mathcal{F} * g)(x) = \sum_{x_i \in N_x} g_{deform}(x_i - x, \Delta(x)) f_i \qquad (4.7)$$

και το deformable kernel ορίζεται:

$$g_{deform}(y_i, \Delta(x)) = \sum_{k<K} h(y_i, \widetilde{x_k} + \Delta_k(x)) W_k \qquad (4.8)$$

Η επιλογή των γειτονικών σημείων στη συγκεκριμένη μέθοδο, όπως αναλύθηκε προηγουμένως, γίνεται με τον ορισμό μιας ακτίνας, δηλαδή σε τη απόσταση kernel points επηρεάζουν το δεκτικό πεδίο για κάθε συνέλιξη. Εκτός της ακτίνας, ορίζεται και ο τρόπος που επηρεάζει. Οι δυο πιο κοινοί τρόποι είναι γραμμικά (όσο πιο μακριά, μικρότερη επιρροή) και με τη κανονική κατανομή (Gaussian distribution). Για την αρχιτεκτονική του μοντέλου ορίζεται η ακολουθία των συνελικτικών μπλοκ. Αυτά αποτελούνται από ένα απλό συνελικτικό επίπεδο, διαδοχικά ResNet, ResNet strided για μείωση της ανάλυσης και διαδοχικά nearest neighbor upsampling και unary, για αύξηση της ανάλυσης ξανά και προσαρμογή των διαστάσεων των χαρακτηριστικών. Στο συγκεκριμένο μοντέλο χρησιμοποιούνται επίσης ορισμένες τεχνικές επαύξησης (augmentation) που βοηθούν ώστε το μοντέλο να είναι πιο ανθεκτικό σε ατέλειες του πραγματικού κόσμου. Εισάγονται

έτσι παραλλαγές σε διάφορες παραμέτρους όπως το χρώμα και το μέγεθος. Για την κατάλληλη επιλογή του είδους του μοντέλου που θα χρησιμοποιηθεί (rigid ή deformable), ορίζεται παράμετρος ώστε τα kernel points να είναι κεντραρισμένα.

Παράμετρος	**SemanticKITTI**	**Toronto-3D**
KP_extent	1.2	1.0
KP_influence	linear	linear
aggregation_mode	sum	sum
architecture	'simple', 'resnetb', 'resnetb_strided', 'resnetb', 'resnetb_strided', 'resnetb', 'resnetb_strided', 'resnetb', 'resnetb_strided', 'resnetb', 'nearest_upsample', 'unary', 'nearest_upsample', 'unary', 'nearest_upsample', 'unary', 'nearest_upsample', 'unary'	'simple', 'resnetb', 'resnetb_strided', 'resnetb', 'resnetb_strided', 'resnetb', 'resnetb_strided', 'resnetb', 'resnetb_strided', 'resnetb', 'nearest_upsample', 'unary', 'nearest_upsample', 'unary', 'nearest_upsample', 'unary', 'nearest_upsample', 'unary'
augment_color	0.8	1
augment_noise	0.001	0.0001
augment_rotation	vertical	vertical
augment_scale_anisotropic	true	false
augment_scale_max	1.2	1.1
augment_scale_min	0.8	0.9
augment_symmetries	true - false - false	true - false - false
batch_limit	50000	10000
batch_norm_momentum	0.98	0.98
batcher	ConcatBatcher	ConcatBatcher
conv_radius	2.5	2.5

deform_fitting_mode	point2point	point2point
deform_fitting_power	1.0	1.0
deform_radius	6.0	6.0
density_parameter	5.0	5.0
first_features_dim	128	128
first_subsampling_dl	0.06	0.08
fixed_kernel_points	center	center
ignored_label_inds	0	0
in_features_dim	2	1
in_points_dim	3	3
in_radius	4.0	4.0
lbl_values:	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]	[0, 1, 2, 3, 4, 5, 6, 7, 8]
min_in_points	10000	5000
max_in_points	20000	10000
modulated	false	false
num_classes	19	8
num_kernel_points:	15	15
num_layers	5	5
repulse_extent	1.2	1.2
use_batch_norm	true	true

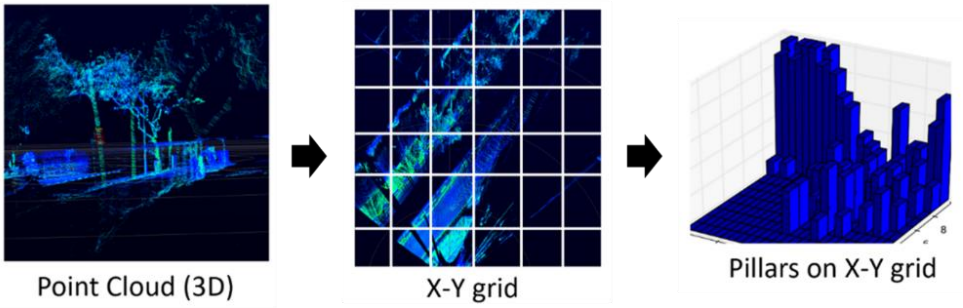
Πίνακας 3. Πίνακας με τις υπερπαραμέτρους του μοντέλου KPConv.

4.4 Μοντέλο ανίχνευσης αντικειμένων PointPillars

Στο κομμάτι της 3Δ ανίχνευσης αντικειμένων μια από τις μεθόδους που χρησιμοποιήθηκε αφορά τη PointPillars, που αποτελεί μια εργασία εκμάθησης χρησιμοποιώντας μόνο 2Δ συνέλιξη (Lang et al., 2018). Αποτελεί μια νέα μέθοδο που μέσω ενός κωδικοποιητή (encoder), μαθαίνει χαρακτηριστικά σε πυλώνες (κάθετες στήλες), για τη πρόβλεψη 3Δ πλαισίων για τα αντικείμενα. Ένα από τα πλεονεκτήματα της είναι ότι αντί να χρησιμοποιεί έναν σταθερό κωδικοποιητή προκαθορισμένου τρόπου ερμηνείας των δεδομένων που ενδεχομένως να χάνονται ορισμένες λεπτομέρειες, χρησιμοποιεί μια μέθοδο που μαθαίνει χαρακτηριστικά απευθείας από τα δεδομένα. Άρα, μπορεί να προσαρμόζει και να βελτιώνει τη κατανόηση του με βάση τα χαρακτηριστικά των δεδομένων. Επιπλέον, απλοποιεί τη προεπεξεργασία των δεδομένων των σημείων, και εξαλείφεται η ανάγκη για χειροκίνητη επεξεργασία. Επίσης, το σημαντικότερο πλεονέκτημα της είναι ότι όλες οι λειτουργίες χρησιμοποιούν 2Δ συνέλιξη, με αποτέλεσμα να είναι εξαιρετικά αποδοτικές στους υπολογισμούς.

Το μοντέλο αποτελείται από τρία κύρια στάδια. Ένα δίκτυο κωδικοποίησης χαρακτηριστικών (feature encoder network) που μετατρέπει ένα νέφος σημείων σε μια ψευδοεικόνα (pseudo-image), μια 2Δ συνέλιξη που αποτελεί το κύριο μέρος της μεθόδου (2D convolutional backbone) για την επεξεργασία της ψευδοεικόνας και μια κεφαλή ανίχνευσης που εντοπίζει τα 3Δ κουτιά.

Η βασική καινοτομία του μοντέλου είναι η δημιουργία πυλώνων όπου ο τρισδιάστατος χώρος χωρίζεται σε κάθετες στήλες και τα σημεία σε κάθε πυλώνα υποβάλλονται σε επεξεργασία για την εξαγωγή χαρακτηριστικών. Έτσι, με την χρήση πυλώνων αντί για 3Δ voxel, απλοποιείται η προεπεξεργασία των σημείων όπως προαναφέρθηκε, καθώς οι πυλώνες εκτείνονται κάθετα χωρίς να χρειάζεται να χωριστούν σε μικρότερους κύβους. Αυτοί οι πυλώνες είναι ουσιαστικά κάθετες περιοχές του τρισδιάστατου χώρου, όπου κάθε πυλώνας συγκεντρώνει όλα τα σημεία που περιέχονται στην ορισμένη χωρική έκταση του (xy όρια), ανεξάρτητα από το ύψος τους (συντεταγμένη z). Με την κατάργηση του άξονα z, αυτό πρακτικά σημαίνει ότι ο χώρος πλέον ορίζεται στην οριζόντια κατανομή των σημείων σχηματίζοντας μια ψευδοεικόνα (δηλαδή BEV), που κάθε pixel αντιστοιχεί σε ένα πυλώνα με συγκεντρωτική πληροφορία από τα σημεία που βρίσκονται μέσα σε αυτό.



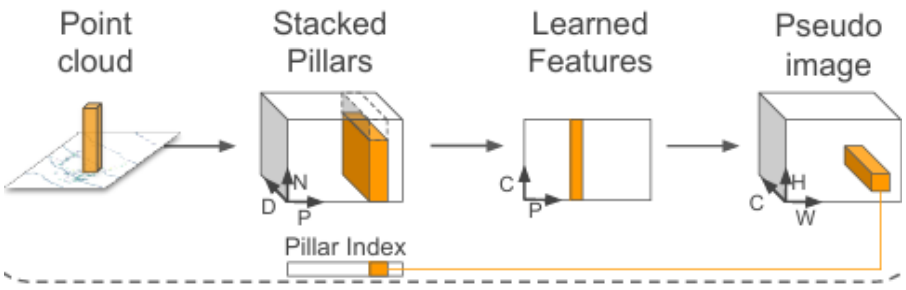
Εικόνα 34. Αντιπαραβολή 3Δ σημείων σε 2Δ πυλώνες.

Ως πρώτο βήμα, το νέφος σημείων διακρίνεται ομοιόμορφα σε ένα πλέγμα στο 2Δ επίπεδο xy, δημιουργώντας ένα σύνολο πυλώνων σταθερού μήκους και πλάτους. Τα χαρακτηριστικά των σημείων αρχικά αποτελούν την θέση τους στο 3Δ χώρο (x,y,z) και την ένταση ανάκλασης (r). Όταν όμως εντάσσονται στους πυλώνες, προσθέτουν σε αυτά η τη μέση θέση όλων των σημείων που βρίσκονται εντός του πυλώνα (x_c,y_c,z_c) και η κεντρική θέση του πυλώνα (x_p,y_p), με αποτέλεσμα τα νέα χαρακτηριστικά για κάθε σημείο να αυξάνουν στα εννιά (D=9). Με αυτό τον τρόπο το δίκτυο κατανοεί τη θέση κάθε σημείου σε σχέση με τους υπόλοιπους πυλώνες και τη θέση του σημείου σε σχέση με το κέντρο του πυλώνα που ανήκει αντίστοιχα.

Τα νέφη σημείων, όπως έχει προαναφερθεί, είναι αραιά με αποτέλεσμα οι περισσότεροι πυλώνες να είναι κενοί και οι υπόλοιποι να έχουν ελάχιστα σημεία. Παρόλα αυτά, εάν σε ένα πυλώνα υπάρχουν πάρα πολλά σημεία, τότε χρησιμοποιείται μια τυχαία δειγματοληψία, αντίθετος εάν τα σημεία είναι πολύ λίγα, τότε τα δεδομένα συμπληρώνονται εφαρμόζοντος zero-padding, για να συμπληρώσει τα υπόλοιπα κενά με μηδενικές τιμές. Αυτό συμβαίνει ακριβώς λόγο του προβλήματος των αραιών δεδομένων. Δηλαδή, ορίζεται αρχικά ο τανυστής (tensor) του πυλώνα (P) με συγκεκριμένο αριθμό σημείων (N) και συγκεκριμένο αριθμό χαρακτηριστικών (D), με αποτέλεσμα να δημιουργείται ένα εύρος τανυστών (D,P,N).

Στη συνέχεια χρησιμοποιείται μια απλουστευμένη μέθοδος του νευρωνικού PointNet σε κάθε πυλώνα για εξαγωγή χαρακτηριστικών. Αρχικά, ένα γραμμικό layer (fully connected layer) εφαρμόζεται στα σημεία. Ακολουθεί Batch-Normalization για την κανονικοποίηση των εξαγόμενων χαρακτηριστικών και η συνάρτηση ενεργοποίησης ReLU, για την εισαγωγή μη γραμμικότητας και δυνατότητας εκμάθησης πολύπλοκων μοτίβων. Το γραμμικό layer μετατρέπει τα σημεία σε ένα χώρο χαρακτηριστικών με μεγαλύτερη

διάσταση. Έτσι, τα μετασχηματισμένα δεδομένα σχηματίζουν ένα τανυστή μεγέθους (C,P,N). Ακολουθεί η διαδικασία max pooling όπου από όλα τα σημεία, επιλέγεται η μέγιστη τιμή για κάθε χαρακτηριστικό και δημιουργείται ένα μοναδικό διάνυσμα για κάθε πυλώνα (δηλαδή C,P). Αυτά τα διανύσματα των πυλώνων τώρα, επιστρέφουν πίσω στην αντίστοιχη θέση τους σε ένα 2Δ πλέγμα (2D grid). Αυτό αποτελεί και τη διαδικασία της ψευδοεικόνας, όπου κάθε σε εικονοστοιχείο αντιστοιχεί ένας πυλώνας, με τιμές χαρακτηριστικών τα εξαγόμενα διανύσματα. Οι διαστάσεις της ψευδοεικόνας θα είναι $C \times H \times W$, όπου H και W είναι το ύψος και πλάτος του 2Δ πλέγματος που αντιπροσωπεύει το νέφος σημείων στο επίπεδο xy και C ο αριθμός των χαρακτηριστικών.



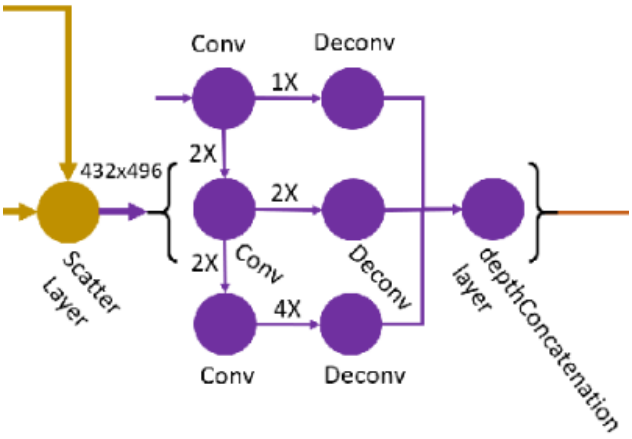
Εικόνα 35. Δίκτυο εξαγωγής χαρακτηριστικών από πυλώνες.

Το επόμενο μέρος του μοντέλου αφορά την επεξεργασίας της ψευδοεικόνας, καθώς από αυτήν θα προκύψουν χαρακτηριστικά από διαφορετικές χωρικές αναλύσεις. Αυτή η επεξεργασία περιλαμβάνει δυο διαδικασίες. Η πρώτη αφορά ένα top-down δίκτυο όπου εξάγονται χαρακτηριστικά από τα δεδομένα εισόδου σε προοδευτικά μικρότερες χωρικές αναλύσεις (downsample). Αντιθέτως, κατά τη δεύτερη διαδικασία γίνεται αύξηση της χωρικής ανάλυσης, από τη προηγουμένη διαδικασία και τα συνδυάζει μέσω συνένωσης. Η πρώτη διαδικασία αποτελείται από συνελικτικά επίπεδα που μειώνουν τη χωρική ανάλυση της εισόδου, εξάγοντας χαρακτηριστικά υψηλότερου επιπέδου. Για να επιτευχθεί αυτό χρειάζεται κατά τη συνέλιξη να χρησιμοποιείται και ένα βήμα μεγαλύτερο ίσο του ενός. Έτσι, μειώνεται το ύψος και πλάτος του χάρτη χαρακτηριστικών (feature map), επιτρέποντας στο δεκτικό πεδίο (receptive field) να καταγράφουν μεγαλύτερα πλαίσια της σκηνής. Με αυτό τον τρόπο δημιουργούνται διαφορετικά μπλοκ τα οποία ουσιαστικά ορίζονται από 3 συστατικά.

- Το βήμα S, που καθορίζει το πόσο θα μειωθεί η χωρική δειγματοληψία
- Το πλήθος L συνελίξεων (ή αλλιώς το βάθος συνέλιξης του επιπέδου) που αποτελείται από ένα kernel 3x3.
- Το πλήθος των χαρακτηριστικών εξόδου F (από τα feature maps)

Η διαδικασίες που ακολουθούν σε κάθε μπλοκ, αποτελούν τη κανονικοποίηση (BatchNorm) των ενεργοποιήσεων μετά από κάθε συνέλιξη και τη συνάρτηση ενεργοποίησης ReLU όπου εισάγει μη γραμμικότητα του μοντέλου και βοηθά το δίκτυο να μάθει πολύπλοκα χαρακτηριστικά. Η πρώτη συνέλιξη στο μπλοκ χρησιμοποιεί βηματισμό $\frac{S}{S_{in}}$, ώστε να εξασφαλισθεί ότι το μπλοκ επεξεργάζεται τα χαρακτηριστικά με τον επιθυμητό βηματισμό, καθώς S_{in} αναφέρεται στο αρχικό βήμα του feature map όταν για πρώτη φορά εισέρχεται στο δίκτυο η ψευδοεικόνα. Επόμενες συνελίξεις εντός του ίδιου μπλοκ ακολουθούν με βήμα ένα, ώστε να διατηρηθεί η μείωση της χωρικής ανάλυσης.

Στη συνέχεια ακολουθεί η αντίστροφη διαδικασία της αύξησης της χωρικής ανάλυσης, ώστε να επανέλθει η επεξεργασμένη ψευδοεικόνα στην αρχική της χωρική ανάλυση. Η διαδικασία επιτυγχάνεται με τη χρήση ανάστροφης συνέλιξης ή αποσυνέλιξη (transpose convolution ή deconvolution). Ουσιαστικά αυτό που συμβαίνει είναι ότι από τα εξαγόμενα feature map, αυξάνεται η χωρική ανάλυση εφαρμόζοντας τα εκπαιδευμένα φίλτρα ανάστροφα. Για να γίνει αυτό, χρειάζεται ο αρχικός βηματισμός S_{in} του feature map που δείχνει πόσο έχει μειωθεί η χωρική ανάλυση. Επίσης, χρειάζεται το S_{out} που είναι ο βηματισμός του εξαγόμενου feature map κατά την ανάστροφη συνέλιξη, όπου ιδανικά θα πρέπει να ταιριάζει ή να είναι κοντά στον αρχικό βηματισμό εισόδου και τέλος, ο αριθμός των καναλιών F (βάθος feature map, πόσες διαφορετικές πτυχές ενός χαρακτηριστικού) μετά τη αύξηση της χωρικής ανάλυσης. Όπως και στη προηγούμενη διαδικασία, έτσι και εδώ μετά την αύξηση της χωρικής ανάλυσης εφαρμόζεται κανονικοποίηση και ενεργοποίηση. Τέλος, τα εξαγόμενα feature map συνενώνονται. Αυτό σημαίνει ότι χαρακτηριστικά που προέρχονται από διαφορετικά βήματα χωρικής ανάλυσης, συνδυάζονται κατά μήκος των καναλιών F, με αποτέλεσμα ο τελικός χάρτης να έχει πληροφορίες από διαφορετικές κλίμακες σε μια ενιαία αναπαράσταση.



Εικόνα 36. Διαδικασία συνέλιξης και αποσυνέλιξης για εξαγωγή χαρακτηριστικών σε διάφορες κλίμακες.

Το τελικό στάδιο του PointPillars είναι η ανίχνευση, η οποία προβλέπει τα τρισδιάστατα πλαίσια οριοθέτησης και τις αντίστοιχες βαθμολογίες κλάσης αντικειμένων με βάση το χάρτη χαρακτηριστικών του BEV. Αυτό επιτυγχάνεται χρησιμοποιώντας τον αλγόριθμο Single Shot Detector (SSD) (Liu et al., 2015). Ο συγκεκριμένος αλγόριθμος αποτελεί μια δημοφιλή μέθοδο ανίχνευσης αντικειμένων στο 2Δ χώρο, άρα μπορεί και προσαρμόζεται και στο PointPillats, καθώς αξιοποιεί τη 2Δ αναπαράσταση του BEV. Η μέθοδος χρησιμοποιεί προκαθορισμένα πλαίσια (ή κουτιά αγκύρωσης όπως στο PV-RCNN) σε όλες τις θέσεις του feature map. Αυτά τα κουτιά δίνονται σε διάφορες διαστάσεις και κλίμακες για διάφορες πιθανές τοποθεσίες και μεγέθη των αντικειμένων. Κάθε πλαίσιο σχετίζεται με δυο προβλέψεις. Τη βαθμολογία της κλάσης που δηλώνουν τη πιθανότητα ένα αντικείμενο να ανήκει σε μια συγκεκριμένη κλάση και τη μετατόπιση του πλαισίου οριοθέτησης όπου βελτιώνουν το προκαθορισμένο πλαίσιο ώστε να ταιριάζει καλύτερα με την πραγματική θέση του αντικειμένου. Ο δείκτης IoU (Intersection over Union) χρησιμοποιείται για τη σύγκριση μεταξύ του προκαθορισμένου και του αληθινού (ground truth) πλαισίου όπου αναλύεται στη συνέχεια.

$$IoU = \frac{Area\ of\ Intersection}{Area\ of\ Union} \qquad (4.9)$$

Επειδή, η διαδικασία αφορά για 2Δ ανίχνευση αντικειμένων, η πληροφορία για το ύψος (άξονας Z) δεν λαμβάνεται υπόψη απευθείας κατά τη διαδικασία των προκαθορισμένων πλαισίων, αλλά αντιμετωπίζεται στη συνέχει ως πρόσθετη παράμετρος παλινδρόμησης για την απλοποίηση της διαδικασίας αντιστοίχισης.

Εφόσον ο δείκτης IoU ενός προκαθορισμένου και του αληθινού πλαισίου υπερβαίνει ένα όριο (threshold), τότε θεωρείται ότι ταιριάζει. Έτσι, το δίκτυο προβλέπει παραμέτρους του πλαισίου οριοθέτησης. Οι προβλέψεις αφορούν τη μετατόπιση του κέντρου του πλαισίου οριοθέτησης (μετατοπίσεις x,y,z του πλαισίου αγκύρωσης), το μέγεθος του πλαισίου (width, length, height), τη στροφή (θ) που δείχνει το προσανατολισμό του αντικειμένου και τη κατεύθυνση (yaw) που βοηθά στο χειρισμό αντικειμένων με προσανατολισμούς που υπερβαίνουν τις ενενήντα μοίρες. Μαζί σε αυτά προθέτονται και η ύπαρξη (occupancy) ή όχι ενός αντικειμένου εντός πλαισίου (0 ή 1) και η κλάση του αντικειμένου σε περίπτωση που υπάρχει, μαζί με τις αντίστοιχες πιθανότητες.

Εικόνα 37. Κεφαλή ανίχνευσης 3Δ αντικειμένων.

Ο συνολικός υπολογισμός της απώλειας αποτελείται από τρία κύρια μέρη. Τις απώλειες εντοπισμού, κατεύθυνσης και ταξινόμησης. Τα πραγματικά και τα προκαθορισμένα πλαίσια αποτελούνται από τις παραμέτρους (x, y, z, w, l, h, θ), όπου η διαφορά τους δίνει το πόσο καλά η πρόβλεψη ταυτίζεται με τη πραγματικότητα. Πιο συγκεκριμένα,

$$\Delta x = \frac{x^{gt} - x^a}{d^a}, \qquad \Delta y = \frac{y^{gt} - y^a}{d^a}, \qquad \Delta z = \frac{z^{gt} - z^a}{h^a}$$

$$\Delta w = \log\frac{w^{gt}}{w^a}, \qquad \Delta l = \log\frac{l^{gt}}{l^a}, \qquad \Delta h = \log\frac{h^{gt}}{h^a} \qquad (4.10)$$

$$\Delta\theta = \sin(\theta^{gt} - \theta^a)$$

όπου gt (ground truth) η πραγματική θέση και a (anchor) η προκαθορισμένη ή αγκύρωση. Η παράμετρος $d^a = \sqrt{(w^a)^2 + (l^a)^2}$, αποτελεί μια κανονικοποίηση που κάνει την απώλεια πιο αμετάβλητη ως προς τη κλίμακα, με αποτέλεσμα να μην ευνοεί παραπάνω μεγαλύτερα αντικείμενα έναντι μικρότερων. Η χρήση του λογαρίθμου βοηθά στη καλύτερη εκμάθηση του αλγορίθμου για σχετικές αλλαγές στη κλίμακα, καθώς είναι πιο σταθερή κατά τη διάρκεια της εκπαίδευσης. Για τον προσανατολισμό, η ημιτονοειδής συνάρτηση δίνει τη μικρότερη γωνιακή διαφορά. Συνολικά, η απώλεια εντοπισμού πλαισίου οριοθέτησης υπολογίζεται από μια Smooth L1 απώλεια (συνίσταται για εργασίες παλινδρόμησης), καθώς είναι λιγότερο ευαίσθητη σε ακραίες τιμές (outliers).

$$L_{loc} = \sum_{b\in(x,y,z,w,l,h,\theta)} SmoothL1(\Delta b) \qquad (4.11)$$

Στη περίπτωση της κατεύθυνσης, υπάρχει ο περιορισμός προς τα πού "βλέπει" το πλαίσιο του αντικειμένου (πχ 0° ή 180°). Για το πρόβλημα αυτό, χρησιμοποιείται μια ξεχωριστή softmax απώλειας ταξινόμησης (L_{dir}) για τη πρόβλεψη της κατεύθυνσης.

Για την απώλεια ταξινόμησης (L_{cls}), το δίκτυο αρχικά αναγνωρίζει εάν στην αγκύρωση υπάρχει αντικείμενο κλάσης. Εάν υπάρχει, χρησιμοποιεί τη μέθοδο focal loss (Lin et al., 2020), που είναι μια ιδιαίτερα αποτελεσματική μέθοδος για μη ισορροπημένα δεδομένα όπου υπάρχει περισσότερος κενός χώρος (μη αγκυρώσεις) από αγκυρώσεις αντικειμένων.

$$L_{cls} = -a_a(1 - p^a)^\gamma \log p^a \qquad (4.12)$$

Η παράμετρος p^a είναι η προβλεπόμενη πιθανότητα ότι μια άγκυρα περιέχει ένα αντικείμενο, a και γ προκαθορισμένες παράμετροι συντονισμού. Τέλος, η συνολική συνάρτηση απώλειας δίνεται από το τύπο:

$$\boldsymbol{L = \frac{1}{N_{pos}}(\beta_{loc}L_{loc} + \beta_{cls}L_{cls} + \beta_{dir}L_{dir})} \quad \textbf{(4.13)}$$

όπου N_{pos} το πλήθος των θετικών αγκυρώσεων και οι παράμετροι β, τα βάρη που ελέγχουν τη συνεισφορά κάθε απώλειας.

Κύριο χαρακτηριστικό της συγκεκριμένης μεθόδου αποτελεί η δημιουργία πυλώνων, που έχει ως αποτέλεσμα τη δημιουργία ψευδοεικόνας. Οι παράμετροι που αφορούν τους πυλώνες είναι τρεις. Συγκεκριμένα, μια για το πλήθος των σημείων που θα δέχεται κάθε πυλώνας, το μέγεθος του ως προς τις τρεις διευθύνσεις και το μέγιστο πλήθος των πυλώνων που μπορεί να έχει. Για να μην υπάρξει σύγχυση, στον πίνακα που ακολουθεί με το σύνολο των παραμέτρων, οι πυλώνες αναφέρονται ως voxel και όχι pillars. Στη συνέχεια ακολουθεί δίκτυο από το οποίο θα προκύψει η ψευδοεικόνα με την αντιστοίχιση των χαρακτηριστικών από τους πυλώνες. Σε αυτό το δίκτυο ορίζονται παράμετροι για το πλήθος των χαρακτηριστικών των δεδομένων εισόδου-εξόδου και το μέγεθος ανάλυσης της ψευδοεικόνας. Μετά ξεκινά το κυρίως μέρος του συνελικτικού νευρωνικού δικτύου από το οποίο γίνεται εξαγωγή των χαρακτηριστικών. Όπως και στα προηγούμενα μοντέλα έτσι και εδώ, ορίζονται οι παράμετροι που αφορούν το πλήθος των χαρακτηριστικών κατά την είσοδο και έξοδο σε κάθε επίπεδο, το πλήθος των επιπέδων και μια παράμετρος του βήματος για τη μείωση της χωρικής ανάλυσης. Στη συνέχεια ακολουθεί η αύξηση της ανάλυσης ώστε να γίνει ο συνδυασμός χαρακτηριστικών από τις διαφορετικές κλίμακες. Αντίστοιχα με τη μείωση ανάλυσης, δίνεται το πλήθος των χαρακτηριστικών κατά την είσοδο και έξοδο κάθε επιπέδου και το βήμα, αλλά τώρα για την αύξηση της χωρικής ανάλυσης. Τελικά για την πρόβλεψη του πλαισίου και τη βαθμολογία κλάσης ορίζονται παράμετροι που φιλτράρουν τις ανιχνεύσεις εξαλείφοντας αυτές με χαμηλή πιθανότητα και ενισχύουν τον προσδιορισμό πλαισίων. Και εδώ ορίζονται τεχνικές augmentation που ενισχύουν την ποικιλομορφία και την ποιότητα των δεδομένων εκπαίδευσης, βελτιώνοντας την ικανότητα του μοντέλου να γενικεύει σε διάφορα σενάρια του πραγματικού κόσμου.

Παράμετρος			**KITTI**
point_cloud_range			[0, -39.68, -3, 69.12, 39.68, 1]
classes			['Pedestrian', 'Cyclist', 'Car']
loss	focal	gamma	2.0
		alpha	0.25
		loss_weight	1.0
	smooth_l1	beta	0.11
		loss_weight	2.0
	cross_entropy	loss_weight	0.2
voxelize	max_num_points		2
	voxel_size		[0.16, 0.16,4]
	max_voxels		[16000, 40000]
voxel_encoder	in_channels		4
	feat_channels		[64]
	voxel_size		[0.16, 0.16,4]
scatter	in_channels		64
	output_shape		[496, 432]
backbone	in_channels		64
	out_channels		[64, 128, 256]
	layer_nums		[3, 5, 5]
	layer_strides		[2, 2, 2]
neck	in_channels		[64, 128, 256]
	out_channels		[128, 128, 128]
	upsample_strides		[1, 2, 4]
	use_conv_for_no_stride		false
head	in_channels		384
	feat_channels		384
	nms_pre		100

	score_thr		0.1
	ranges		[[0,-39.68,-0.6,70.4,39.68,-0.6], [0,-39.68,-0.6,70.4,39.68,-0.6], [0,-39.68,-1.78,70.4,39.68,-1.78]]
	sizes		[[0.6, 0.8, 1.73], [0.6, 1.76, 1.73], [1.6, 3.9, 1.56]]
	rotations		[0, 1.57]
	iou_thr		[[0.35, 0.5], [0.35, 0.5], [0.45, 0.6]]
augment	PointShuffle		True
	ObjectRangeFilter	point_cloud_range	[0, -39.68, -3, 69.12, 39.68, 1]
	ObjectSample	min_points_dict	Car: 5
			Pedestrian: 10
			Cyclist: 10
		sample_dict	Car: 15
			Pedestrian: 10
			Cyclist: 10

Πίνακας 4. Πίνακας με τις υπερπαραμέτρους του μοντέλου PointPillars.

5. Από τα Νέφη Σημείων στις Προβλέψεις: Υλοποίηση ΝΔ για 3Δ Συμπεράσματα Δεδομένων

Σε αυτό το κεφάλαιο αναλύονται παραδείγματα και αποτελέσματα που προέκυψαν από διάφορες εφαρμογές που υλοποιήθηκαν πάνω στα νευρωνικά δίκτυα, που παρουσιάστηκαν στο προηγούμενο κεφάλαιο. Το πρακτικό κομμάτι της διπλωματικής αυτής, αφορά τη χρήση νευρωνικών μοντέλων σε διάφορα σύνολα 3Δ δεδομένων, για προβλέψεις που αφορούν τη σημασιολογική κατάτμηση και ανίχνευση αντικειμένων. Να σημειωθεί ότι λόγο περιορισμένων υπολογιστικών πόρων που διατίθενται στο τρέχον σύστημα, ήταν ανέφικτοι η εκτέλεση εκπαίδευσης των νευρωνικών, οπότε χρησιμοποιήθηκαν τα βάρη που έχουν εκπαιδευτεί ήδη και είναι διαθέσιμα για χρήση προς το κοινό. Πρακτικά, έτρεξαν τα συμπεράσματα (inference) των νευρωνικών από την εκπαίδευση, σε νέα δεδομένα που το μοντέλο δεν έχει ξαναδεί. Χρησιμοποιήθηκε φορητός υπολογιστής μεσαίας κατηγορίας, με επεξεργαστή AMD Ryzen 7 4800H 8 πυρήνων κατάλληλο για πολλαπλές εργασίες, κάρτα γραφικών NVIDIA GTX 1660 Ti, μνήμη RAM 16 GB και ένας σκληρός δίσκος SSD για ισορροπία ταχύτητας και αποθήκευσης. Καθώς η κάρτα γραφικών αποτελεί κατασκευή της NVIDIA, χρησιμοποιήθηκε η παράλληλη πλατφόρμα υπολογισμών CUDA, που επιτρέπει στη GPU να επιταχύνει εργασίες, σε σχέση με έναν επεξεργαστή.

5.1 Ρύθμιση του περιβάλλοντος για τα ΝΔ. Εξερευνώντας τη βιβλιοθήκη Open3D-ML και τα βασικά του στοιχεία

Για την υλοποίηση των νευρωνικών χρησιμοποιήθηκε η βιβλιοθήκη Open3d-ML, που αποτελεί επέκταση της Open3D και είναι σχεδιασμένη για τεχνικές μηχανικής μάθησης (Zhou et al., 2018). Η Open3D αποτελεί μια βιβλιοθήκη ανοιχτού κώδικα σχεδιασμένη για εργασίες με 3Δ δεδομένα. Παρέχει εργαλεία επεξεργασίας ενός νέφους σημείων ή πλέγματος, οπτικοποίηση σκηνών, ευθυγράμμιση πολλαπλών 3Δ δεδομένων κ.ά.. Η Machine Learning επέκταση του, βασιζόμενη στη προ υπάρχουσα δομή, προσφέρει επιπλέον προκατασκευασμένα εργαλεία και μοντέλα για εργασίες όπως σημασιολογική κατάτμηση και ανίχνευση αντικειμένων. Πέρα από αυτά, έχει τη δυνατότητα

ενσωμάτωσης βιβλιοθηκών Pytorch (επιλέχθηκε) και TensorFlow, που επιταχύνουν τις διαδικασίες στα νευρωνικά μοντέλα. Επιπλέον στην οπτικοποίηση, ενσωματώνονται οι ετικέτες των προβλέψεων για την ερμηνεία των αποτελεσμάτων των μοντέλων. Η ανάπτυξη του περιβάλλοντος έγινε πάνω σε λειτουργικό σύστημα Linux Ubuntu 22.04. Αυτό περιλάμβανε την εγκατάσταση βασικών εργαλείων όπως Python και Conda για διαχείριση εικονικού περιβάλλοντος, διασφαλίζοντας ότι όλα είναι συμβατά με την Open3D-ML, όπως δίνει η επίσημη σελίδα.

Για την εκτέλεση και των εντοπισμό σφαλμάτων (debugging) του κώδικα, επιλέχθηκε το ολοκληρωμένο περιβάλλον ανάπτυξης PyCharm, που αποτελεί ισχυρό και ευέλικτο εργαλείο για προγραμματισμό σε Python. Δίνεται επίσης η δυνατότητα να τρέχουν οι κώδικες με διαφορετικό διερμηνέα (interpreter), δηλαδή η δυνατότητα επιλογής κάθε φορά ποιο περιβάλλον Python θα χρησιμοποιήσει για να εκτελέσει τον κώδικα, από τα εικονικά περιβάλλοντα που έχουν δημιουργηθεί. Αυτό χρησιμεύει σε περιπτώσεις διαφορετικών project όπου το κάθε ένα να είναι συμβατό με διαφορετική έκδοση Python, αλλά και η απομόνωση διαφορετικών εξαρτημάτων σε εικονικό περιβάλλον αποτρέπει τις συγκρούσεις μεταξύ τους και διασφαλίζει ότι κάθε έργο έχει τις ακριβείς εκδόσεις των βιβλιοθηκών που χρειάζεται.

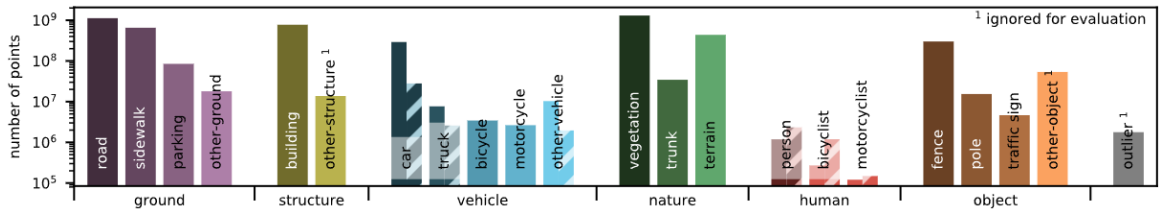
5.2 Ανάλυση δεδομένων

Ένα σετ δεδομένων που χρησιμοποιήθηκε για την ανίχνευση αντικειμένων είναι το KITTI Vision Benchmark Suite που δημιουργήθηκε από το Ινστιτούτο Τεχνολογίας της Καρλσρούη και το Τεχνολογικό Ινστιτούτο Toyota στο Σικάγο (Geiger et al., 2012). Συλλέχτηκε το 2012 χρησιμοποιώντας ένα όχημα εξοπλισμένο με αισθητήρες, που κινούνταν γύρω από τους δρόμους της Καρλσρούη. Για τη καταγραφή των δεδομένων χρησιμοποιήθηκαν συστήματα ψηφιακών αισθητήρων (μια ασπρόμαυρη μια έγχρωμη), ένας σαρωτή λέιζερ Velodyne HDL-64E που παράγει περισσότερα από ένα εκατομμύριο σημεία ανά δευτερόλεπτο και ένα σύστημα εντοπισμού OXTS RT 3003 που συνδυάζει σήματα GPS, GLONASS, IMU και RTK.

Εικόνα 38. Ρύθμιση αισθητήρων.

Χαρακτηριστικό του είναι ότι αποτελείται ένα σύνολο δεδομένων μεγάλης κλίμακας, καθώς παρέχει έναν εκτεταμένο αριθμό σαρώσεων, καλύπτοντας ένα πλήρες οπτικό πεδίο 360º. Αυτό έχει ως αποτέλεσμα να είναι ένα ολοκληρωμένο σύνολο δεδομένων που καταγράφει πολλές σκηνές. Το δείγμα που είναι διαθέσιμο στο κοινό για ελεύθερη χρήση, αποτελείται από ένα δείγμα εκπαίδευσης 7.480 στιγμιότυπων και ένα δείγμα δοκιμής 7.517 στιγμιότυπων. Για τις προβλέψεις του μοντέλου, έχουν επιλεχθεί μόνο τρεις κλάσεις και συγκεκριμένα πεζοί, ποδηλάτες και αυτοκίνητα. Να σημειωθεί για το δείγμα αξιολόγησης δεν δίνεται πάντα ξεχωριστά, αλλά πολλές φορές μέσα από το σύνολο του δείγματος εκπαίδευσης, δίνεται η δυνατότητα στο χρήστη να χωρίσει το δείγμα εκπαίδευσης και να πειραματιστεί.

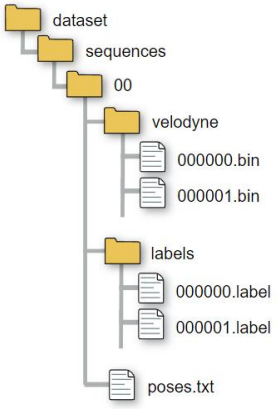
Ένα άλλο υποσύνολο δεδομένων που χρησιμοποιήθηκε και είναι ευρέως διαδεδομένο για εργασίες 3Δ σημασιολογικής κατάτμησης αφορά το SemanticKITTI, που προέρχεται από το σύνολο δεδομένων KITTI Vision Benchmark Suite (Behley et al., 2019). Το σύνολο των δεδομένων αποτελείται από 22 ακολουθίες, που καταγράφονται σε διαφορετικά σενάρια. Συγκεκριμένα, αποτελούν πραγματικές σκηνές κατά την οδήγηση σε αυτοκινητόδρομους, αστικές και επαρχιακές περιοχές. Η ακολουθία από το 00-10 αποτελεί το δείγμα εκπαίδευσης (training), εκτός του 08 που αποτελεί το δείγμα αξιολόγησης (validation), ενώ από το 11-21 το σύνολο δεδομένων δοκιμών (test). Για το σύνολο δεδομένων εκπαίδευσης και αξιολόγησης, παρέχονται οι ετικέτες 28 σημασιολογικών κλάσεων, συμπεριλαμβανομένων κλάσεων που διακρίνονται σε κινούμενα και μη, αλλά στα πειράματα θα θεωρείται ότι ανήκουν στην ίδια, με αποτέλεσμα να καταλήγει σε 19 κλάσεις.



Εικόνα 39. Κατανομή ετικετών συνόλου δεδομένων SemanticKITTI.

Υπάρχουν επίσης 11 πρόσθετες κατηγορίες στα νέφη σημείων οι οποίες όμως δεν λαμβάνονται υπόψη στην αξιολόγηση (πχ σημεία απόκρυψης ή αγνοημένες κατηγορίες). Συνολικά περιέχει πάνω από 43.000 (19.130 training, 4.071 validation, 20.351 testing) σαρώσεις, όπου κάθε σάρωση αποτελείται έως και 100.000 σημεία. Οι σαρώσεις περιέχουν μια λεπτομερή και πυκνή 3Δ αναπαράσταση του περιβάλλοντος, καλύπτοντας μια σειρά διαφορετικών επιφανειών και αντικειμένων.

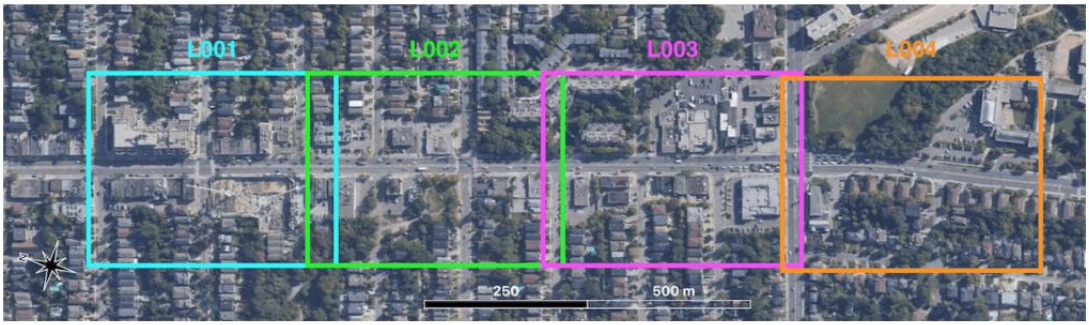
Κάθε νέφος σημείων αποθηκεύεται ως αρχείο .bin, όπου κάθε σημείο περιέχει τέσσερα χαρακτηριστικά. Τρία για την θέση και ένα για την ένταση της αντανάκλασης του λέιζερ (x, y, z, intensity). Κάθε νέφος σημείων της εκπαίδευσης συνδυάζεται με ένα .label αρχείο που εκχωρεί μια σημασιολογική ετικέτα σε κάθε σημείο. Επίσης, παρέχεται ένα αρχείο poses.txt που δίνει τη τοποθέτηση κάθε σάρωσης ώστε να μπορούν να ευθυγραμμιστούν μεταξύ τους σε ένα ενιαίο πλαίσιο, καθώς το πλαίσιο συντεταγμένων κάθε σάρωσης αποτελείται από ένα τοπικό πλαίσιο συντεταγμένων του αισθητήρα LiDAR.



Εικόνα 40. Μορφή και δομή δεδομένων.

Το δεύτερο σύνολο δεδομένων που χρησιμοποιήθηκε στα πειράματα της σημασιολογικής κατάτμησης είναι το Toronto-3D, όπου αποτελεί σάρωση αστικής περιοχής μεγάλης κλίμακας στην Avenue Road στο Τορόντο του Καναδά (Tan et al., 2020). Τα νέφη σημείων αποκτήθηκαν από ένα σύστημα Mobile Laser Scanning (MLS), συγκεκριμένα της Teledyne Optech Maverick. Το σύστημα αυτό αποτελείται από 32 αισθητήρες LiDAR, μια πανοραμική κάμερα, ένα δορυφορικό σύστημα GNSS και ένα σύστημα Simultaneous Localization and Mapping (SLAM). Ο αισθητήρας LiDAR μπορεί να καταγράψει έως και 700.000 σημεία ανά δευτερόλεπτο και το οπτικό του πεδίο στο κατακόρυφο άξονα καλύπτει από -10^{o} έως +30^{o}, με ακρίβεια καλύτερη από 3 εκατοστά. Με μια περεταίρω επεξεργασία στη συνέχεια, αποδίδεται ένα φυσικό χρώμα (RGB) που οφείλεται από τον επιπλέον αισθητήρα κάμερας που διαθέτει.

Η περιοχή ενδιαφέροντος καλύπτει περίπου 1 χιλιόμετρο του οδικού τμήματος με περίπου 78,3 εκατομμύρια σημεία. Χωρίζεται σε τέσσερα σχεδόν ισομερή τμήματα των 250 μέτρων έκαστος.



Εικόνα 41. Προσεγγιστική Οριοθέτηση τμημάτων του συνόλου των δεδομένων.

Κάθε ένα τμήμα ξεχωριστά, αποτελείται από το αντίστοιχο σύνολο δεδομένων και αποθηκεύεται σε ένα αρχείο της μορφής .ply και σε κάθε σημείο αντιστοιχούν 10 χαρακτηριστικά.

- Τρία αφορούν τη θέση (x, y, z στο σύστημα NAD83 / UTM Zone 17N)
- Τρεις τιμές RGB για την απόδοση αληθινού χρώματος
- Η ένταση της αντανάκλασης του LiDAR (κανονικοποιημένη σε 0-255)
- Ο χρόνος του GPS κατά τη καταγραφή του σημείου

- Η γωνία πρόσπτωσης της δέσμης λέιζερ
- Η ετικέτα κλάσης κάθε αντικειμένου, καταγεγραμμένο σε ακέραια τιμή (0-8)

Από το παραπάνω καταλαβαίνει κανείς ότι το Toronto-3D αποτελείται από 9 κατηγορίες που αποτελούν δρόμους, σημάνσεις δρόμων, βλάστηση, κτήρια, αυτοκίνητα, γραμμές κοινής ωφέλειας, στύλους, φράχτες και μη ταξινομημένα. Στην Εικόνα 41, φαίνεται ο διαχωρισμός των δεδομένων. Στη περίπτωση μας τα νέφη σημείων στο πρώτο, τρίτο και τέταρτο σετ αποτελούν τα δεδομένα εκπαίδευσης και το δεύτερο τα δεδομένα πρόβλεψης.

5.3 Σύνθεση δεδομένων και υπερπαραμέτρων για την εκπαίδευση και αξιολόγηση των μοντέλων

Σε αυτή την ενότητα γίνεται αναφορά στο τρόπο με τον οποίο γίνεται η σύνθεση του συνόλου δεδομένων με τις υπερπαραμέτρους του μοντέλου για την υλοποίηση εκπαίδευσης και αξιολόγησης του. Κάθε συνδυασμός μοντέλου με ένα σύνολο δεδομένων, ακολουθείτε από ένα configuration file της μορφής .yml, που ενσωματώνουν όλες τις απαραίτητες ρυθμίσεις και υπερπαραμέτρους που απαιτούνται, χωρίς την ανάγκη παρέμβασης στο πηγαίο κώδικα. Να σημειωθεί ότι οι παράμετροι και οι υπερπαράμετροι λειτουργούν σε διαφορετικά επίπεδα και διαχειρίζονται διαφορετικά. Οι παράμετροι αφορούν εσωτερικά στοιχεία ενός μοντέλου όπου κατά τη διάρκεια της εκπαίδευσης μαθαίνει από τα δεδομένα (πχ βάρη και biases κάθε νευρώνα). Από τη άλλη, υπερπαράμετροι είναι εξωτερικές διαμορφώσεις που ορίζονται πριν την εκπαίδευση. Ρυθμίζουν τη διαδικασία εκπαίδευσης της δομή του μοντέλου αλλά δεν μαθαίνονται από τα δεδομένα. Συνήθως ρυθμίζονται χειροκίνητα ή μέσω τεχνικών βελτιστοποίησης (πχ ρυθμός εκμάθησης, αριθμός κρυφών επιπέδων, πλήθος νευρώνων ανά επίπεδο, batch size). Ενώ οι παράμετροι βελτιστοποιούνται για να ταιριάζουν στα δεδομένα, οι υπερπαράμετροι πρέπει να ρυθμιστούν προσεκτικά για να καθοδηγήσουν τη διαδικασία βελτιστοποίησης, διασφαλίζοντας ότι το μοντέλο μαθαίνει αποτελεσματικά και γενικεύεται καλά σε νέα δεδομένα.

```yaml
dataset:
  name: SemanticKITTI
  dataset_path:  # path/to/your/dataset
  test_split: ['11', '12', '13', '14']
  training_split: ['00', '01', '02', '03',]
  validation_split: ['08']
model:
  name: RandLANet
  ckpt_path: # path/to/your/checkpoint
  num_neighbors: 16
  num_layers: 4
  num_points: 45056
  num_classes: 19
  ignored_label_inds: [0]
  sub_sampling_ratio: [4, 4, 4, 4]
  in_channels: 3
  dim_features: 8
  dim_output: [16, 64, 128, 256]
  grid_size: 0.06
pipeline:
  name: SemanticSegmentation
  optimizer:
    lr: 0.001
  batch_size: 4
  main_log_dir: ./logs
  max_epoch: 100
  save_ckpt_freq: 5
  scheduler_gamma: 0.9886
  test_batch_size: 1
  train_sum_dir: train_log
  val_batch_size: 2
```

Εικόνα 42. Υπόδειγμα yml αρχείου.

Αυτά τα αρχεία χρησιμεύουν ως προσχέδια, περιγράφοντας πώς θα πρέπει να συμπεριφέρονται και να αλληλεπιδρούν διαφορετικά στοιχεία του αγωγού (pipeline) μηχανικής εκμάθησης. Με αυτό τον τρόπο διαμορφώνεται ένας διαχωρισμός που διευκολύνει τη διαχείριση και τις αναπροσαρμογές υπερπαραμέτρων, δίχως αλλαγές του πηγαίου κώδικα. Για τη σύνθεση των δεδομένων δηλώνεται ο τρόπος που διαχωρίζεται ένας νέφος σημείων σε περιοχές εκπαίδευσης, αξιολόγησης και δοκιμής , όπως φαίνεται στη παραπάνω εικόνα. Επίσης, δίνεται και μια λίστα βαρών που αποτελεί υπερπαράμετρο και χρησιμεύει στην εξισορρόπηση των κλάσεων. Συγκεκριμένα, δίνονται μεγαλύτερα βάρη σε κλάσεις που έχουν μικρότερη εκπροσώπηση, για να διασφαλιστεί ότι το μοντέλο δίνει περισσότερο βάση σε αυτά κατά τη διάρκεια της εκμάθησης. Σε σύνολα δεδομένων όπως το SemanticKITTI όπου τα σημεία κλάσης δρόμου είναι πολύ παραπάνω από ότι αυτά του ποδηλάτου, καταλαβαίνει κανείς ότι επικρατεί μεγάλη ανισορροπία κλάσεων. Για αυτό το λόγο η λίστα των βαρών κλάσεων χρησιμεύει ώστε σε συνδυασμό με μια συνάρτηση απώλειας (weighted cross-entropy) να εξισορροπηθεί η επιρροή κάθε κλάσης.

```yaml
dataset:
  name: SemanticKITTI
  dataset_path:  # path/to/your/dataset
  cache_dir: ./logs/cache
  class_weights: [55437630, 320797, 541736, 2578735, 3274484, 552662, 184064,
    78858, 240942562, 17294618, 170599734, 6369672, 230413074, 101130274,
    476491114, 9833174, 129609852, 4506626, 1168181]
  test_result_folder: ./test
  test_split: ['12']
  training_split: ['00', '01', '02', '03', '04', '05', '06', '07', '09', '10']
  all_split: ['00', '01', '02', '03', '04', '05', '06', '07', '09',
  '08', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20', '21']
  validation_split: ['08']
  use_cache: true
  sampler:
    name: 'SemSegRandomSampler'
```

Εικόνα 43. Υπόδειγμα ρύθμισης βαρών κάθε κλάσης.

Στο προηγούμενο κεφάλαιο στην ανάλυση των μοντέλων, εξετάστηκαν οι υπερπαράμετροι όπου οι συντάκτες επέλεξαν για την υλοποίηση εκπαίδευσης των μοντέλων στα αντίστοιχα σύνολα δεδομένων. Υπάρχουν κάποιες άλλες υπερπαράμετροι που δεν έχουν να κάνουν με τα δεδομένα ή το μοντέλο, αλλά επηρεάζουν έμμεσα τον τρόπο με τον οποίο μαθαίνει το μοντέλο, τη πολυπλοκότητα του, τη ταχύτητα σύγκλισης και την απόδοση του. Συγκεκριμένα, στο RandLA-Net μοντέλο όπου χρησιμοποιεί τη μέθοδο βελτιστοποίησης Adam, για τη σύγκλιση του μοντέλου ρυθμίζεται η υπερπαράμετρος learning rate και scheduler gamma όπου συνδυαστικά ελέγχουν την προσαρμογή των παραμέτρων αλλά και την προσαρμογή του ρυθμού εκμάθησης αντίστοιχα, κατά τη διάρκεια της εκπαίδευσης. Οι άλλες δυο υπερπαράμετροι αφορούν το μέγεθος του δείγματος (batch size) και ο μέγιστος αριθμός εποχών (max epochs). Το batch size είναι ο αριθμός των δειγμάτων που υποβάλλονται σε επεξεργασία πριν από την ενημέρωση των παραμέτρων του μοντέλου. Όπως αναλύθηκε σε σχέση με το κεφάλαιο 3.5, όταν τα δεδομένα χωρίζονται σε μικρότερα δείγματα, πετυχαίνει πιο γρήγορη σύγκλιση καθώς οι υπολογισμοί των παραμέτρων είναι λιγότεροι σε σχέση με το σύνολο των δεδομένων, αλλά ταυτόχρονα είναι πιο απαιτητικά υπολογιστικά καθώς υλοποιεί ταυτόχρονα για όλες τις παρτίδες. Επίσης, λόγο της ταχείας σύγκλισης, υπάρχει ο κίνδυνος να καταλήξει σε τοπικό ελάχιστο στη συνάρτηση απώλειας με αποτέλεσμα να γενικεύει χειρότερα σε νέα δεδομένα. Εκτός του batch size στα δεδομένα εισόδου, ορίζονται και για τα δεδομένα αξιολόγησης αλλά και δοκιμών για αργότερες προβλέψεις. Μια εποχή αποτελεί ένα πέρασμα του νευρωνικού από το σύνολο των δεδομένων. Αυτή η υπερπαράμετρος βοηθά στον έλεγχο της διάρκειας της εκπαίδευσης ώστε να αποφευχθεί μια υπερβολική προσαρμογή.

Ίδιες υπερπαράμετροι ισχύουν και στο μοντέλο KPConv, καθώς και εδώ έχει επιλεχθεί η βελτιστοποίηση Adam, με τη διαφορά ότι συμπληρώνεται με τον αλγόριθμο momentum. Λόγο και της deformable εκδοχής του μοντέλου, ορίζεται και υπερπαράμετρος ρυθμού εκμάθησης ειδικά για τα παραμορφώσιμα επίπεδα του μοντέλου. Επίσης, αντί για την υπερπαράμετρο scheduler gamma, εδώ επιλέγεται μια άλλη φθορά ρυθμού εκμάθησης όπου ουσιαστικά εφαρμόζεται συνεχόμενα κατά την εκπαίδευση σε σχέση με την προηγούμενη όπου υλοποιείται στο τέλος κάθε εποχής. Παρακάτω παρουσιάζεται ο πίνακας των τιμών αυτών που επιλέχθηκαν από τους συντάκτες για την εκπαίδευση στα δεδομένα.

Παράμετρος	RandLA-Net		KPConv	
	SemanticKITTI	Toronto-3D	SemanticKITTI	Toronto-3D
optimizer lr	0.001	0.001	0.01	0.01
batch_size	4	2	1	1
max_epoch	100	200	800	1000
scheduler_gamma	0.9886	0.99	-	-
lr_decays	-	-	0.98477	0.98477
test_batch_size	1	1	1	1
val_batch_size	2	2	1	1
momentum			0.98	0.98
deform_lr_factor			0.1	0.1
weight_decay			0.001	0.001

Πίνακας 5. Πίνακας υπερπαραμέτρων βελτιστοποίησης και σύγκλισης των δικτύων σημασιολογική κατάτμησης.

Για τη μέθοδο PointPillars ισχύουν περίπου τα ίδια, καθώς η κύρια διαφορά είναι ότι εδώ επιλέχθηκε η μέθοδος βελτιστοποίησης AdamW που συνδυάζεται με τον αλγόριθμο momentum. Στην Adam βελτιστοποίηση το πέναλτι weigh decay προστίθεται απευθείας στη κλίση πριν την ενημέρωση των βαρών, όπου αυτό μπορεί να έχει ως αποτέλεσμα να επηρεάζεται η προσαρμογή του ρυθμού εκμάθησης, καταλήγοντας σε μη βέλτιστη σύγκλιση. Το AdamW αποσυνδέει τη μείωση του βάρους κατά την ενημέρωσης της κλίσης. Συγκεκριμένα, το πέναλτι weigh decay εξακολουθεί να είναι αναγκαίο αλλά για τον έλεγχο της τακτοποίησης που εφαρμόζεται στα βάρη του μοντέλου, με αποτέλεσμα η βελτιστοποίηση να εφαρμόζεται ομοιόμορφα. Για τον προσδιορισμό εάν μια ανίχνευση θεωρείται σωστή (true positive) κατά την αξιολόγηση, δίνονται τιμές κατωφλίου για κάθε κλάση. Επίσης, δίνεται μια αντιστοίχιση ορισμένων κλάσεων που μπορεί το μοντέλο να θεωρήσει ως “παρόμοιες”, όπως ένα βαν με ένα αυτοκίνητο. Τέλος, ορίζεται υπερπαράμετρος αποκοπή κλίσης το οποίο εμποδίζει τις κλίσεις του μοντέλου να μεγαλώσουν πολύ κατά τη διάρκεια του backpropagation. Αυτό βοηθά στη σταθεροποίηση της εκπαίδευσης αποφεύγοντας τις απότομες εκτοξεύσεις των κλίσεων.

Παράμετρος	KITTI
batch_size	6
max_epoch	200
test_batch_size	1
val_batch_size	1
grad_clip_norm	2
lr	0.001
betas	[0.95, 0.99]
weight_decay	0.01
overlaps	[0.5, 0.5, 0.7]
similar_classes	{Van: Car, Person_sitting: Pedestrian}
difficulties	[0, 1, 2]

Πίνακας 6. Πίνακας υπερπαραμέτρων βελτιστοποίησης και σύγκλισης των δικτύων ανίχνευσης αντικειμένων.

5.4 Αξιολόγηση των μεθόδων και αποτελέσματα προβλέψεων

Σε αυτό το κεφάλαιο, αναλύονται οι αξιολογήσεις των μοντέλων των νευρωνικών δικτύων που υλοποιήθηκαν από τους συντάκτες στα προαναφερόμενα νέφη σημείων. Αυτά τα μοντέλα έχουν αξιολογηθεί με βάση δυο βασικούς δείκτες. Το Intersection Over Union (IoU) για εργασίες σημασιολογικής κατάτμησης και τη μέση ακρίβεια (mAP) για την ανίχνευση αντικειμένων. Στη πρώτη ενότητα αναλύεται η απόδοση των μοντέλων όπου οι δείκτες αξιολόγησης χρησιμεύουν για να δείξουν την ικανότητα τους να ταξινομούν σωστά κάθε σημείο στα 3Δ νέφη σημείων. Το ίδιο θα ακολουθήσει για την ικανότητα των μοντέλων στον εντοπισμό αντικειμένων σε 3Δ περιβάλλοντα και στην ακριβή εκτίμηση των θέσεων, των προσανατολισμών και των διαστάσεων τους. Τέλος, θα παρουσιαστούν τα πειράματα που υλοποιήθηκαν χρησιμοποιώντας τα ίδια μοντέλα σε νέα σύνολα. Αυτά τα πειράματα δείχνουν πώς τα μοντέλα γενικεύονται σε νέα δεδομένα, προσφέροντας πληροφορίες για την ευελιξία τους σε πραγματικές εφαρμογές.

5.4.1 Αξιολόγηση μοντέλων

Για τις εργασίες 3Δ σημασιολογικής κατάτμησης όπως αυτές που εκτελούνται από το RandLA-Net και το KPConv στα σύνολα δεδομένων SemanticKITTI και Toronto3D, η ακρίβεια και το Intersection over Union (IoU) χρησιμοποιούνται συνήθως ως μετρικές αξιολόγησης. Αυτές οι μετρήσεις αξιολογούν την απόδοση της ικανότητας του μοντέλου να ταξινομεί σημεία στον τρισδιάστατο χώρο, στις σωστές σημασιολογικές κλάσεις.

Η συνολική ακρίβεια (overall accuracy – OA) είναι ο λόγος των σωστά προβλεπόμενων σημείων προς τον συνολικό αριθμό σημείων των δεδομένων. Συγκεκριμένα μετρά πόσα σημεία ταξινομήθηκαν σωστά σε σχέση με το συνολικό αριθμό σημείων, ανεξάρτητα από τη κλάση.

$$Accuracy = \frac{Number\ of\ correct\ classified\ points}{Total\ number\ of\ points} \quad \textbf{(5.1)}$$

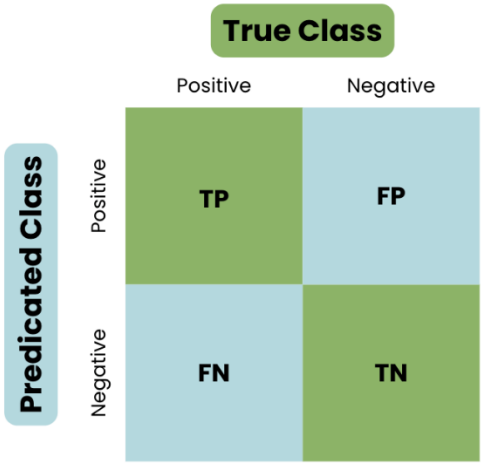
Με αυτόν το τρόπο όμως δημιουργείται ένας περιορισμός. Σε δεδομένα όπως το SemanticKITTI όπου ορισμένες κλάσεις (πχ δρόμος) αντιστοιχούν πολλά περισσότερα σημεία σε σχέση με μια μικρότερης συχνότητας (πχ ποδήλατο), αυτό μπορεί να οδηγήσει σε ανισορροπία κλάσεων. Αυτό έχει ως αποτέλεσμα η συνολική ακρίβεια μπορεί να είναι υψηλή, αλλά το μοντέλο μπορεί να είναι ανεπαρκές σε κλάσεις μειοψηφίας. Αυτό είναι και ο λόγος που δηλώνονται βάρη εξισορρόπησης για κάθε κλάση, όπως εξηγήθηκε προηγουμένως.

Το IoU είναι μια πιο αναλυτική μέθοδος που χρησιμοποιείται για τη μέτρηση της απόδοσης του μοντέλου για κάθε κατηγορία. Υπολογίζεται συγκρίνοντας τις προβλέψεις των σημείων και τις πραγματικές τιμές τους για κάθε τάξη.

$$IoU = \frac{True\ Positive}{True\ Positive + False\ Positive + False\ Negative} \quad \textbf{(5.2)}$$

Όπου True Positive (TP) ο αριθμός των σημείων που ταξινομήθηκαν σωστά ότι ανήκουν στην κλάση, False Positive (FP) ο αριθμός των σημείων που ταξινομήθηκαν εσφαλμένα ότι ανήκουν στην κλάση και False Negative (FN) ο αριθμός των σημείων που ανήκουν στην κλάση αλλά δεν ταξινομήθηκαν σε αυτή. Άρα σε σύνολα δεδομένων όπου υπάρχουν

πολλές διαφορετικές κλάσεις, όπως το SemanticKITTI και Toronto3D, είναι ένας πιο αντιπροσωπευτικός τρόπος αξιολόγησης των διαφορετικών κλάσεων του μοντέλου. Για τη συνολική αξιολόγηση δίνεται ο μέσος όρος IoU (mIoU) για όλες τις κλάσεις. Το mIoU παρέχει μια ενιαία βαθμολογία που υπολογίζει τον μέσο όρο των IoU για κάθε κλάση, η οποία χρησιμεύει στη σύγκριση της απόδοσης του μοντέλου σε σύνολα δεδομένων και παρόμοιων εργασιών. Ίδια μέθοδος αξιολόγησης χρησιμοποιείται και για το KPConv μοντέλο, με διαφορετική απόδοση λόγο διαφορετικής αρχιτεκτονικής.



Εικόνα 44. Πίνακας σύγχυσης/ σφαλμάτων.

Για το μοντέλο RandLA-Net στο σύνολο δεδομένων του SemanticKITTI, οι συντάκτες έχουν χωρίσει το δείγμα εκπαίδευσης για τις ακολουθίες 00-10 και το δείγμα δοκιμών για τις ακολουθίες 11-21. Η ακολουθία 08 αποτελεί το δείγμα αξιολόγησης, από το οποίο θα προκύψει η ακρίβεια του μοντέλου. Συγκεκριμένα, μετά την εκπαίδευση του μοντέλου στα συγκεκριμένα δεδομένα τα οποία είναι χαρακτηρισμένα με τις πραγματικές τους τιμές, το μοντέλο καλείται να αξιολογηθεί στα δεδομένα αξιολόγησης της 08 ακολουθίας όπου δεν χρησιμοποιήθηκε κατά την εκπαίδευση. Η συγκεκριμένη ακολουθία έχει επίσης τις πραγματικές τιμές, με αποτέλεσμα να κάνει σύγκριση με τις προβλέψεις. Μετά από διάφορες παραλλαγές και δοκιμές, το μοντέλο έχει καταλήξει σε μία ακρίβεια 52.8% mIoU.

Το σύνολο δεδομένων Toronto3D, αποτελείται από τέσσερις σειρές σαρώσεων παρόμοιων διαστάσεων. Οι συχνότητες 01, 03 και 04 αποτελούν τα χαρακτηρισμένα δεδομένα, από τα

οποία θα προκύψουν τα δεδομένα εκπαίδευσης και αξιολόγησης. Στη δημοσίευση του μοντέλου δεν αναφέρεται συγκεκριμένα στο τρόπο που επιλέχθηκαν, αφήνοντας να εννοηθεί ότι και οι τέσσερις πιθανοί συνδυασμοί υλοποιήθηκαν, με την ακρίβεια να φτάνει στο 74% mIoU. Στο μοντέλο KPConv αντιστοίχως, η ακρίβεια για το SemanticKITTI είναι στο 58% και στο Toronto3D περίπου 65.6%.

Για τις εργασίες 3Δ ανίχνευσης αντικειμένων που εκτελούνται από το μοντέλο PointPillars στο σύνολο δεδομένων KITTI, χρησιμοποιείται η μέση ακρίβεια (Mean Average Precision - mAP) για το BEV και το 3D, ως μετρικές αξιολόγησης. Το mAP βρίσκεται με βάση την καμπύλη ακριβείας-ανάκλασης (precision-recall curve) για κάθε κλάση και υπολογίζεται κατά μέσο όρο σε όλες τις κλάσεις, για να δώσει μια συνολική βαθμολογία απόδοσης. Η ακρίβεια είναι η αναλογία των σωστά προβλεπόμενων αντικειμένων από όλα τα προβλεπόμενα αντικείμενα, ενώ η ανάκλαση είναι η αναλογία των σωστά προβλεπόμενων αντικειμένων από όλα τα πραγματικά αντικείμενα.

$$Precision = \frac{TruePositive}{True\ Positive + False\ Positive} \quad \textbf{(5.3)}$$

$$Recall = \frac{TruePositive}{True\ Positive + False\ Negatives} \quad \textbf{(5.4)}$$

Στη συνέχεια υπολογίζεται η ακρίβεια και η ανάκλαση για διαφορετικά επίπεδα εμπιστοσύνης (threshold). Μόλις υπολογιστούν οι τιμές σχεδιάζονται σε ένα γράφημα (καμπύλη), όπου δείχνει πόσο μειώνεται η ακρίβεια (άξονας y) όσο αυξάνεται η ανάκλαση (άξονας x). Αυτό συμβαίνει διότι όσο γίνεται προσπάθεια ανίχνευσης περισσότερων αντικειμένων (υψηλότερη ανάκλαση), τόσο περισσότερα λάθη θα γίνονται (μικρότερη ακρίβεια). Το AP είναι η περιοχή κάτω από την καμπύλη ακριβείας-ανάκλησης. Εάν το μοντέλο έχει καλή ακρίβεια σε όλα τα επίπεδα ανάκλησης, το AP του θα είναι υψηλό. Για τον υπολογισμό mAP, υπολογίζεται το AP για κάθε κλάση αντικειμένων στο σύνολο δεδομένων και στη συνέχεια τον μέσο όρο αυτών των τιμών.

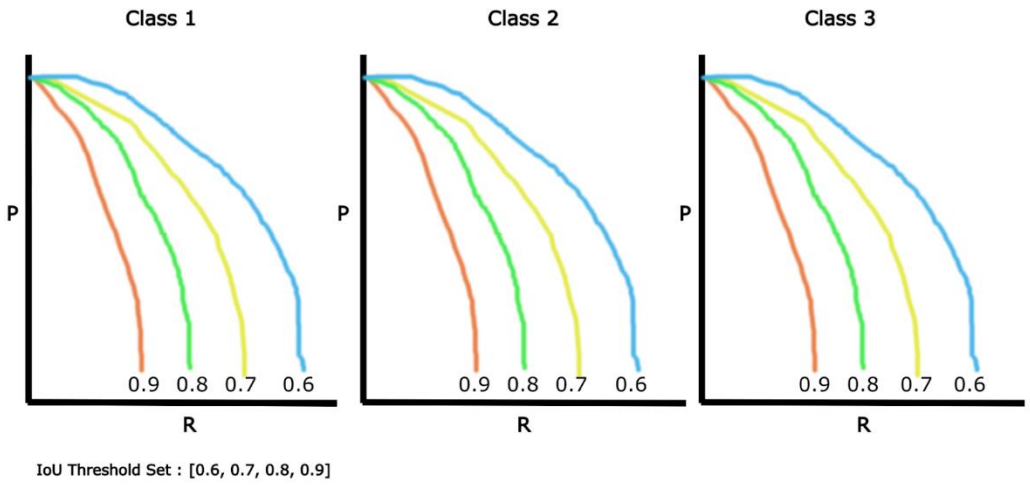
$$mAP = \frac{1}{N}\sum_{i=1}^{N} A\,P_i \qquad (5.5)$$

Όπου N ο αριθμός των κλάσεων και AP_i η μέση ακρίβεια για της κλάσης. Όπως προαναφέρθηκε το μοντέλο παρέχει δυο προβολές για αξιολόγηση. Το BEV μετρά την ικανότητα του μοντέλου να ανιχνεύει αντικείμενα στο 2Δ επίπεδο όταν το βλέπει κανείς από ψηλά. Στο BEV, η αξιολόγηση γίνεται με την προβολή των 3Δ πλαισίων οριοθέτησης στο επίπεδο της γης (επίπεδο xy). Αυτό απλοποιεί το πρόβλημα σε δισδιάστατο χώρο, εστιάζοντας στην οριζόντια θέση και αγνοώντας την κατακόρυφη διάσταση (ύψος). Η 3D μέτρηση αξιολογεί την ικανότητα του μοντέλου να ανιχνεύει αντικείμενα στο πλήρη 3Δ χώρο. Σε αυτή την περίπτωση, η απόδοση του μοντέλου αξιολογείται με βάση και τις τρεις διαστάσεις, λαμβάνοντας υπόψη το πλήρες πλαίσιο οριοθέτησης που περιλαμβάνει το ύψος του αντικειμένου εκτός από τη θέση.

Και εδώ είναι χρήσιμη η αξιολόγηση IoU, καθώς μετρά την επικάλυψη μεταξύ του προβλεπόμενου πλαισίου οριοθέτησης και του πλαισίου οριοθέτησης στη πραγματικότητα. Τιμή δείκτη ίση με ένα σημαίνει ότι τα κουτιά επικαλύπτονται τέλεια, ενώ τιμή μηδέν σημαίνει ότι δεν υπάρχει επικάλυψη. Για το σύνολο δεδομένων KITTI, το mAP υπολογίζεται για δύο κατώφλια IoU.

- IoU ≥ 0,7 για αυτοκίνητα
- IoU ≥ 0,5 για πεζούς και ποδηλάτες

Αυτά τα όρια αντικατοπτρίζουν τα αποδεκτά επίπεδα επικάλυψης πλαισίου οριοθέτησης που απαιτούνται για να θεωρηθεί μια ανίχνευση ως σωστή. Για παράδειγμα, το μοντέλο μπορεί να επιτύχει υψηλό σκορ στο BEV αλλά ελαφρώς χαμηλότερο στο 3D λόγω της πρόσθετης πρόκλησης του ακριβή εντοπισμού της κάθετης διάστασης.



Εικόνα 45. Παράδειγμα υλοποίησης διαφορετικών κατωφλίων, για διαφορετικά αντικείμενα κλάσης.

Για το μοντέλο PointPillars στο σύνολο δεδομένων KITTI, δίνεται σύνολο δεδομένων εκπαίδευσης 7481 σαρώσεων από τις οποίες οι σαρώσεις 3712 και πάνω έχουν επιλεχθεί ως το δείγμα αξιολόγησης. Δηλαδή, το δείγμα αξιολόγησης είναι οι 3769 υπόλοιπες σαρώσεις. Η ακρίβεια που δίνει το μοντέλο είναι 61.2 στο BEV και 52.8 στο 3D. Ακολουθεί ο πίνακας με τις τελικές ακρίβειες των μοντέλων, στα σύνολα δεδομένων που εξετάστηκαν.

Model/Dataset	**SemanticKITTI**	**Toronto 3D**	**KITTI [BEV/3D]**
RandLA-Net	52.8	74.0	-
KPConv	58.0	65.6	-
PointPillars	-	-	61.2 / 52.8

Πίνακας 7. Ακρίβειες μοντέλων

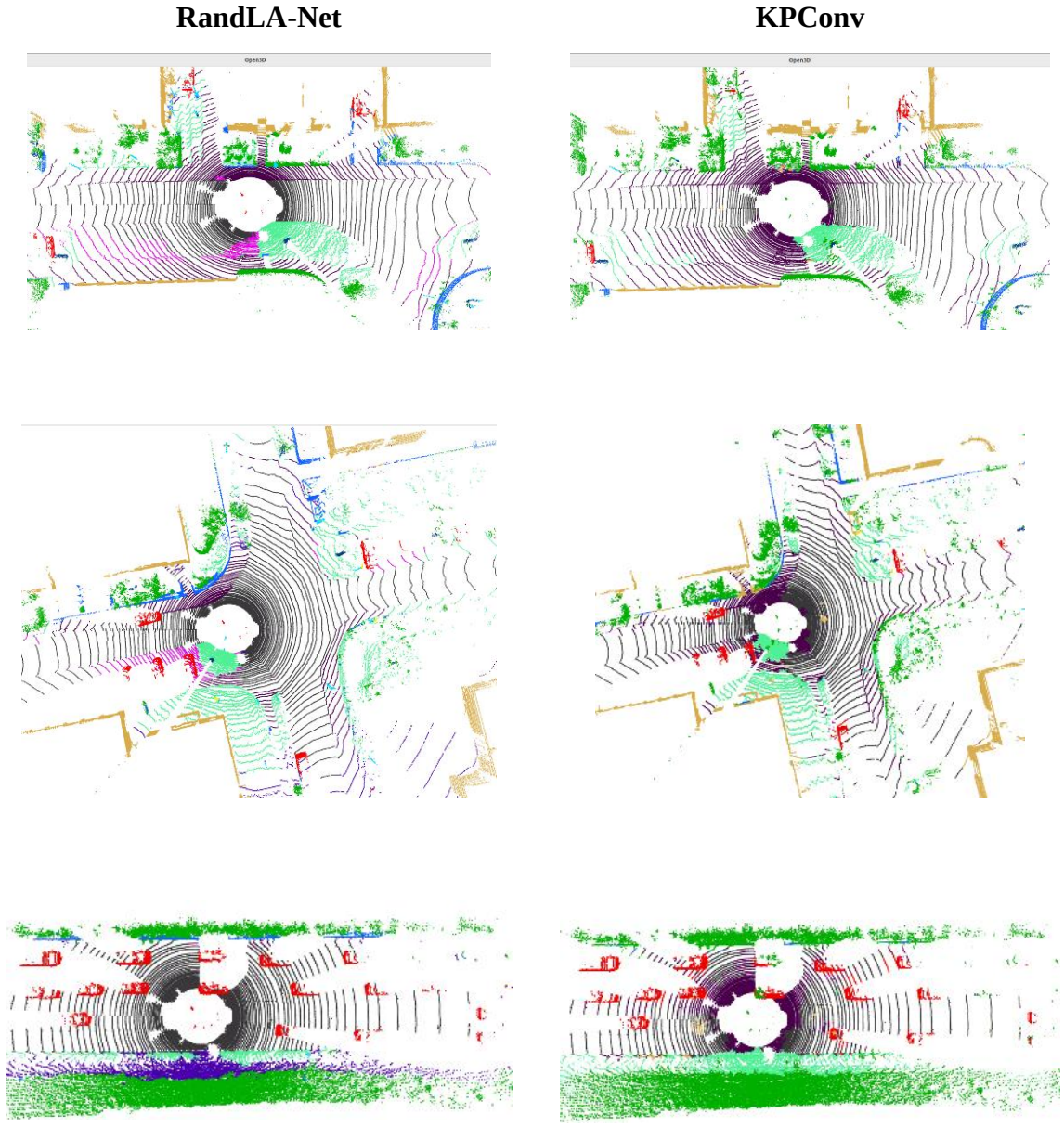
5.4.2 Πρόβλεψη σε νέα δεδομένα με χρήση των προεκπαιδευμένων μοντέλων

Για λόγους πρακτικούς και εξοικονόμησης χρόνου, η πρόβλεψη κάθε ακολουθίας στα δεδομένα δοκιμής του SemanticKITTI από το μοντέλο RandLA-Net έγινε σε ξεχωριστές στιγμές, καθώς ο απαιτούμενος χρόνος για μια πρόβλεψη σε μια σειρά χιλίων σαρώσεων ήταν περίπου 10 δευτερόλεπτα, δηλαδή περίπου δυόμιση ώρες για να δοθεί πρόβλεψη σε κάθε σημείο (1.000 σαρώσεις επί 100.000 σημεία η σάρωση, περίπου 100.000.000 προβλέψεις για μια συχνότητα). Στο αντίστοιχο μοντέλο του KPConv οι υπολογισμοί είναι πιο απαιτητικοί. Ο απαιτούμενος χρόνος που χρειάζεται μια πρόβλεψη στις αντίστοιχες χίλιες σαρώσεις είναι περίπου 44 δευτερόλεπτα, δηλαδή περίπου δώδεκα ώρες και είκοσι λεπτά για τη συνολική πρόβλεψη. Παρακάτω απεικονίζονται παραδείγματα από τους χρόνους πρόβλεψης σε διαφορετικές συχνότητες από τα δυο αυτά μοντέλα.

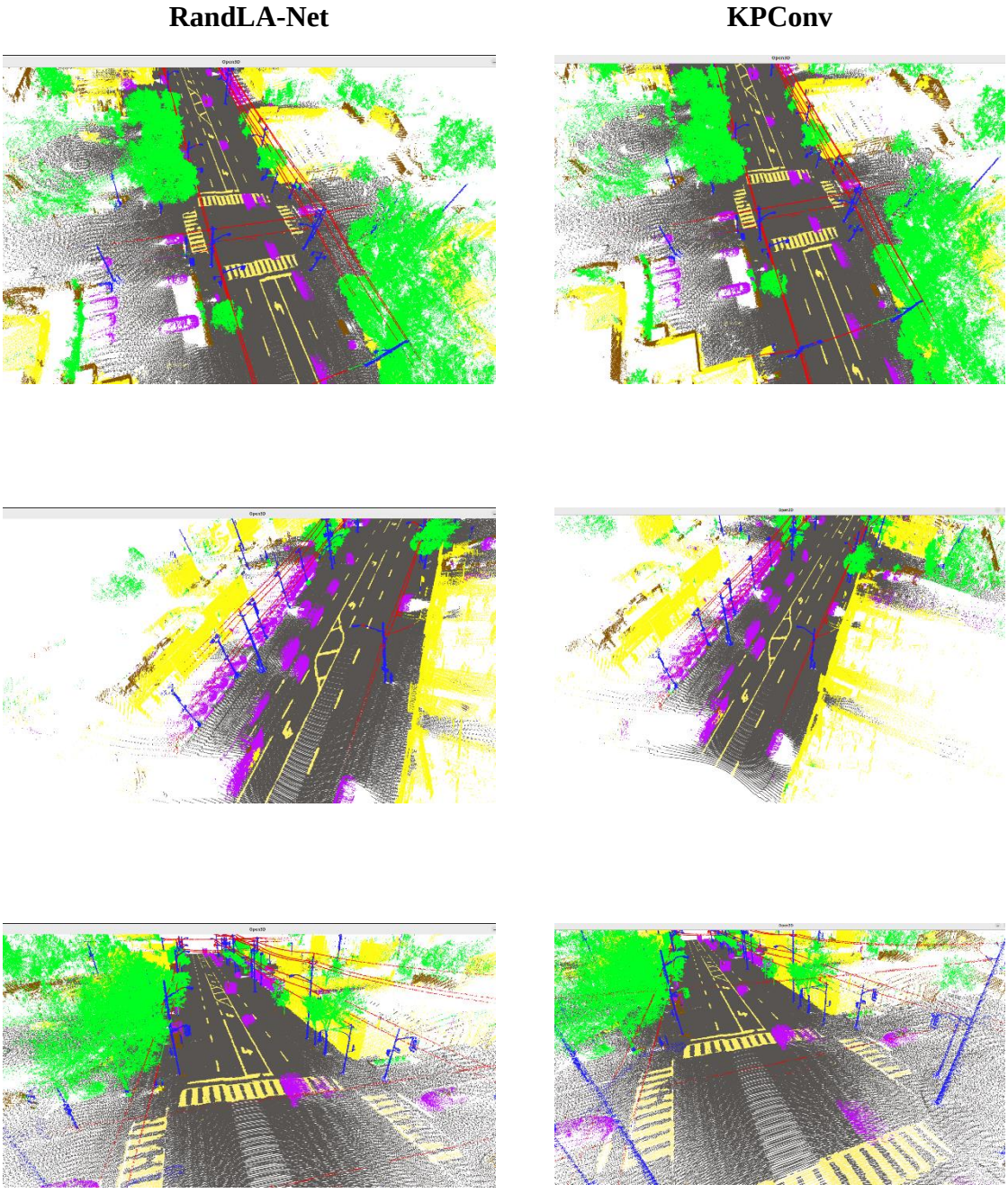
Εικόνα 46. Υλοποίησης δοκιμής του RandLA-Net και KPConv σε διαφορετικά υποσύνολο δεδομένων και χρόνοι.

Για το σύνολο δεδομένων Toronto 3D χρειάστηκε πολύ παραπάνω χρόνο καθώς ο όγκος του είναι πολύ μεγάλος και δεν είναι διαχωρισμένος όπως στο SemanticKITTI. Συγκεκριμένα, η σάρωση L002 αποτελείται από 10.283.800 σημεία, όπου συγκριτικά με το προηγούμενο σύνολο δεδομένων είναι εκατό φορές μεγαλύτερο από μια ξεχωριστή σάρωση των περίπου 100.000 σημείων. Για τις δοκιμές του μοντέλο PointPillars στα δεδομένα KITTI, τα πράγματα είναι παρόμοια με το σύνολο δεδομένων SemanticKITTI,

καθώς το δεύτερο προέρχεται από το πρώτο και η δομή των σαρώσεων είναι χωρισμένα σε μικρότερα κομμάτια που διευκολύνει τις προβλέψεις. Έτσι, έγινε και η αντίστοιχη διαμόρφωση στο configuration αρχείο ώστε να υπολογίζει κάθε φορά ξεχωριστά ένα υποσύνολο της ακολουθίας του test. Παρακάτω απεικονίζονται τρεις συγκριτικές προβλέψεις διαφορετικών συχνοτήτων, από τα δυο μοντέλα της σημασιολογικής κατάτμησης στα δεδομένα του SemanticKITTI. Παρομοίως και για τα δεδομένα Toronto 3D. Για το KITTI δεν υπήρξαν συγκριτικά δεδομένα με κάποιο άλλο μοντέλο.

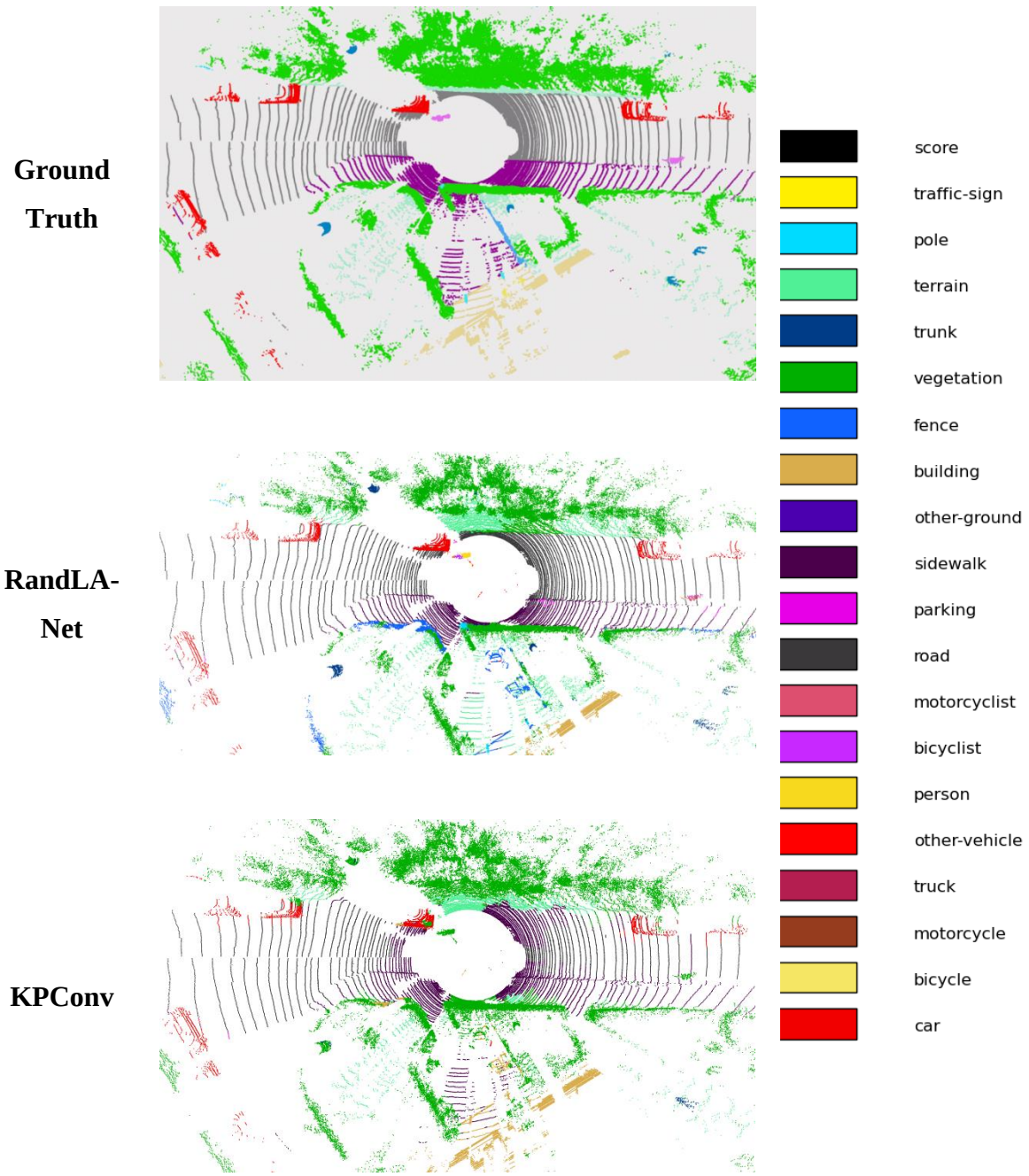


Εικόνα 47. Απεικόνιση προβλέψεων RandLA-Net και KPConv σε διαφορετικό δείγμα του SemanticKITTI.

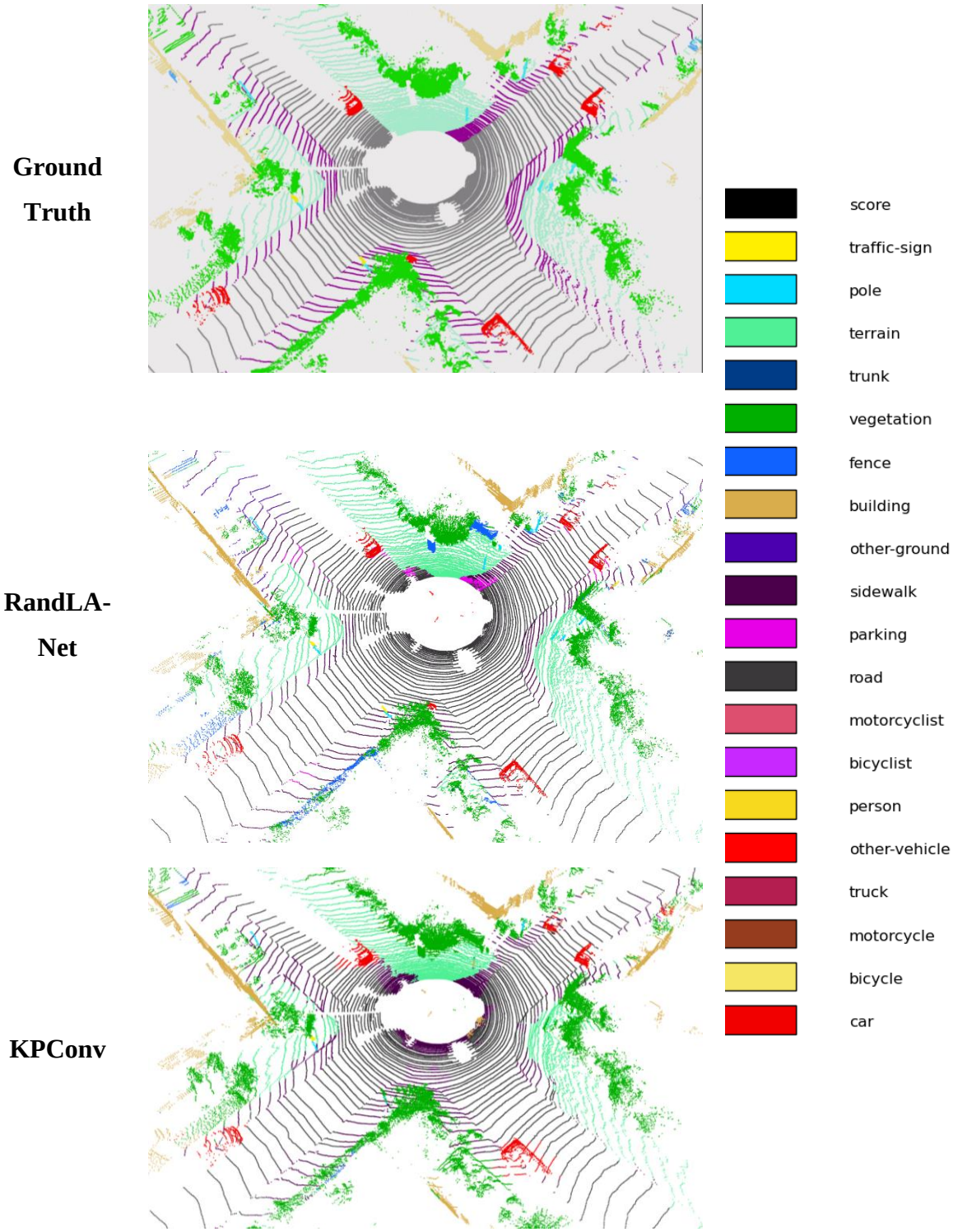


Εικόνα 48. Απεικόνιση προβλέψεων RandLA-Net και KPConv στο δείγμα Toronto 3D.

Στη συνέχεια υλοποιήθηκαν προβλέψεις στο δείγμα αξιολόγησης, ώστε πέρα από τα ποιοτικά, να υπάρξουν και ποσοτικά αποτελέσματα συγκρίσεων. Επιλέχθηκε το δείγμα αξιολόγησης για το λόγο ότι διαθέτει τις ετικέτες του ground truth. Ώστε να μπορεί να γίνει η σύγκριση της πρόβλεψης με τις πραγματικές τιμές. Δείγμα αξιολόγησης δίνεται μόνο από τα σύνολα δεδομένων SemanticKITTI και KITTI.



Εικόνα 49. Προβλέψεις και σύγκριση με ground truth (σειρά 08 σάρωση 3095).



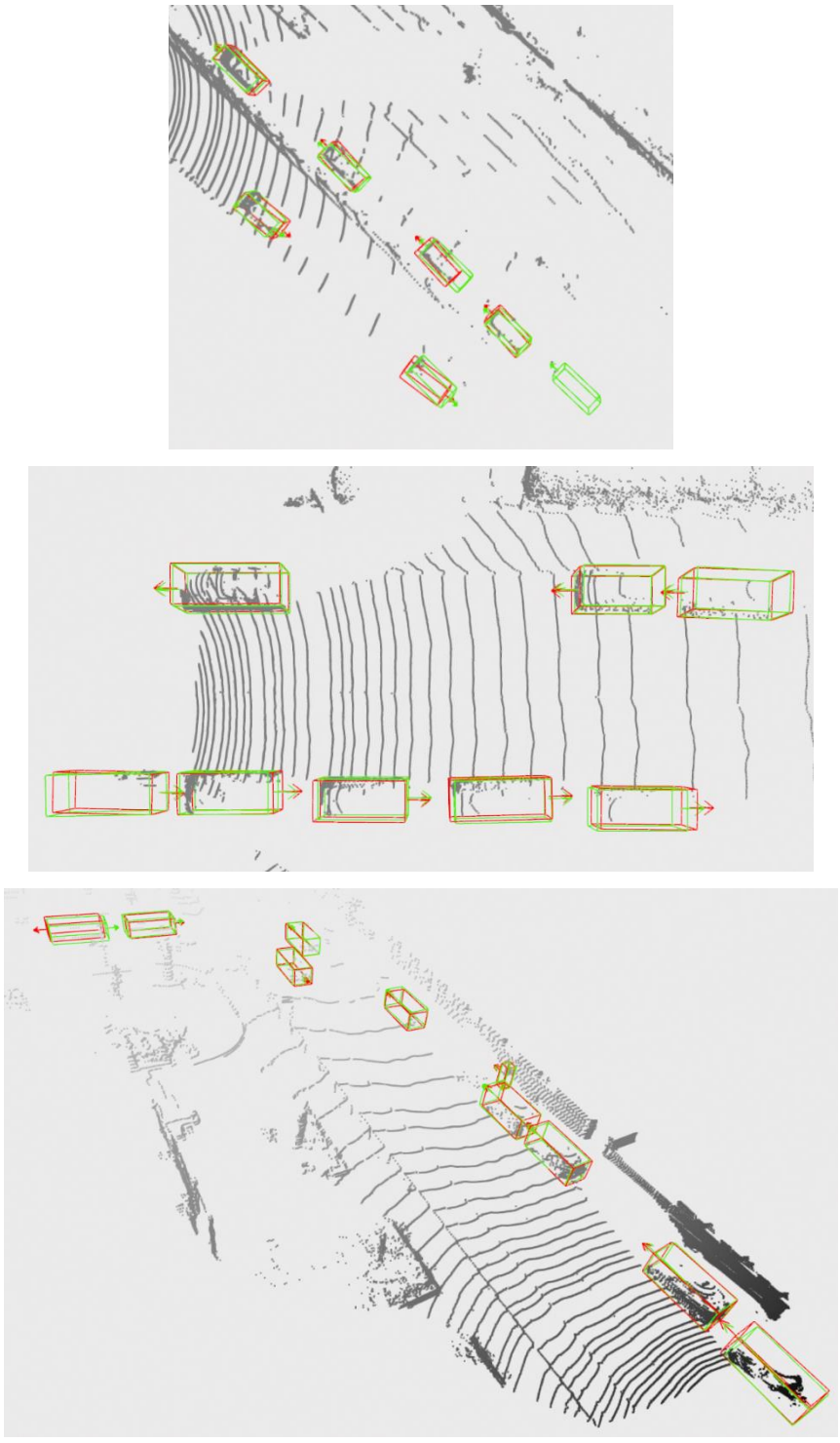
Εικόνα 50. Προβλέψεις και σύγκριση με ground truth (σειρά 08 σάρωση 800).

Παρακάτω ακολουθεί ο πίνακας αποτελεσμάτων αξιολόγησης των δυο μεθόδων σημασιολογικής κατάτμησης στο σύνολο δεδομένων SemanticKITTI από την παραπάνω πρόβλεψη της σειράς 08 σάρωση 3095, για κάθε μια κλάση από τις 19 κλάσεις (από τις αρχικές είκοσι η μια αποτελεί τα unlabeled).

Κλάση/ IoU (acc)	**RandLA-Net**	**KPConv**
Car	0.96 (0.99)	0.73 (0.75)
Bicycle	NaN	NaN
Motorcycle	NaN	NaN
Truck	NaN	NaN
Other vehicle	NaN	NaN
Person	0	0
Bicyclist	0.29 (0.29)	0
Motorcyclist	NaN	NaN
Road	0.94 (0.97)	0.42 (0.43)
Parking	0	0
Sidewalk	0.73 (0.80)	0.43 (0.71)
Other ground	0	0
Building	0.85 (0.88)	0.85 (0.91)
Fence	0.05 (0.40)	0
Vegetation	0.76 (0.80)	0.79 (0.96)
Trunk	0.79 (0.94)	0.22 (0.22)
Terrain	0.25 (0.54)	0.27 (0.49)
Pole	0.42 (0.93)	0.04 (0.42)
Traffic sign	0.76 (0.93)	0.93 (1)
mIoU	**0.48 (0.71)**	**0.33 (0.44)**

Πίνακας 8. Αξιολόγηση πρόβλεψης για όλες τις κλάσεις και συνολικά.

Αντίστοιχα για τα δεδομένα KITTI, υλοποιήθηκαν προβλέψεις σε διάφορες σαρώσεις του δείγματος αξιολόγησης, ώστε να συγκριθεί με τις πραγματικές τιμές του ground truth για τις τρεις κλάσεις πεζού, ποδηλάτη και αυτοκινήτου.



Εικόνα 51. Προβλέψεις (κόκκινο) και σύγκριση με ground truth (πράσινο) στο KITTI.

mAP BEV			
Κλάση/ difficulty	**0**	**1**	**2**
Pedestrian	69.09	68.58	69.33
Cyclist	84.78	80.29	80.27
Car	87.29	87.56	88.58
Overall	**79.39**		

Πίνακας 9. Αξιολόγηση ανίχνευσης αντικειμένων mAP BEV

mAP 3D			
Κλάση/ difficulty	**0**	**1**	**2**
Pedestrian	51.52	52.93	53.28
Cyclist	72.51	70.44	72.06
Car	74.05	76.59	77.12
Overall	**67.48**		

Πίνακας 10. Αξιολόγηση ανίχνευσης αντικειμένων mAP 3D

6. Συμπεράσματα

Συμπερασματικά, αυτή η διπλωματική εργασία έχει διερευνήσει την εφαρμογή προηγμένων μοντέλων βαθιάς μάθησης, συγκεκριμένα RandLA-Net και KPConv, για την εκτέλεση τρισδιάστατης σημασιολογικής κατάτμησης στα σύνολα δεδομένων SemanticKITTI και Toronto3D, καθώς και PointPillars για ανίχνευση 3Δ αντικειμένων στο Δεδομένα KITTI. Και τα τρία σύνολα δεδομένων χρησιμοποιούνται ευρέως για εκπαίδευση και αξιολόγηση μοντέλων, για την καλύτερη κατανόηση και ταξινόμηση του 3Δ περιβάλλοντος σε συστήματα αυτόνομης οδήγησης και ρομποτικής.

Οι μετρήσεις αξιολόγησης, συμπεριλαμβανομένου του mIoU και της συνολικής ακρίβειας για τη σημασιολογική κατάτμηση, καθώς και της μέσης ακρίβειας (mAP) για την ανίχνευση αντικειμένων, υπογράμμισαν τα δυνατά σημεία και τους περιορισμούς κάθε μοντέλου στα σύνολα δεδομένων. Μέσω της χρήσης αυτών των μοντέλων, το έργο απέδειξε με επιτυχία την ικανότητα να κάνει ακριβείς προβλέψεις σε εξωτερικά περιβάλλοντα, εντοπίζοντας διάφορες κατηγορίες αντικειμένων και χαρακτηριστικά εδάφους με υψηλό επίπεδο λεπτομέρειας. Τα αποτελέσματα αποκάλυψαν περιοχές που χρήζουν βελτίωση, με ορισμένες κατηγορίες να έχουν χαμηλή απόδοση και την ακρίβεια πρόβλεψης να είναι κατώτερη των προσδοκιών. Όπως έγινε φανερό στο παράδειγμα του SemanticKITTI στις κλάσεις άνθρωπος, ποδηλάτης, παρκινγκ, άλλο έδαφος και φράχτης, οι ακρίβειες ήταν πολύ χαμηλές έως μηδαμινές. Αυτό συμβαίνει διότι στο δείγμα τα σημεία που αποτελούν αυτές τις οντότητες ήταν πολύ λίγα, αλλά και επειδή το δείγμα για την συνολική αξιολόγηση κάθε κλάσης ήταν γενικότερα πολύ μικρό για να είναι αξιόλογο. Αυτό φαίνεται και στο γεγονός ότι άλλες κλάσεις που τα σημεία απεικονίζονται παραπάνω όπως τα αυτοκίνητα, τα κτίρια, η βλάστηση και ο δρόμος έχουν πολύ καλύτερες ακρίβειες που φτάνουν έως και 90% επιτυχής πρόβλεψη. Άρα καταλαβαίνει κανείς ότι χρειάζεται ένα μεγαλύτερο δείγμα αξιολόγησης, όπως υλοποιείται και από την βιβλιοθήκη σε πάνω από 4000 σαρώσεις. Επίσης, το μοντέλο RandLA-Net έδειξε ότι σε περιπτώσεις μεγάλης κλίμακας δεδομένων, έχει καλύτερη απόδοση και γρηγορότερη επεξεργασία, καθώς σχεδόν σε όλες τις κλάσεις πέτυχε καλύτερες βαθμολογίες και σε χρόνο ¼ από ότι το KPConv. Αυτό είναι πολύ σημαντικό για εφαρμογές σε πραγματικό χρόνο, καθώς ο σχεδιασμός του επικεντρώνεται στη μείωση της υπολογιστικής πολυπλοκότητας. Ομοίως, το μοντέλο PointPillars πέτυχε ανταγωνιστικά αποτελέσματα στην ανίχνευση 3Δ αντικειμένων τόσο στο Bird's Eye View όσο και στον 3Δ χώρο.

Μία από τις βασικές προκλήσεις που έπρεπε να αντιμετωπιστούν κατά την αξιολόγηση των μοντέλων ήταν η απουσία των ground truth σε κάποιο υποσύνολο δοκιμών. Τα ground truth είναι ζωτικής σημασίας για την ακριβή μέτρηση της απόδοσης των μοντέλων, καθώς παρέχουν την αναφορά για τη σύγκριση των προβλέψεων που γίνονται από το μοντέλο. Δυστυχώς, το σετ δοκιμών δεν περιλάμβανε ετικέτες πραγματικών τιμών, γεγονός που εμπόδιζε την αξιολόγηση των μοντέλων σε νέα δεδομένα.

Για την αντιμετώπιση του, επιλέχθηκε να χρησιμοποιηθούν τμήματα από το σύνολο αξιολόγησης, το οποίο είχε διαθέσιμες ετικέτες. Τα δεδομένα του συνόλου αξιολόγησης δεν είχαν χρησιμοποιηθεί κατά τη διαδικασία εκπαίδευσης του μοντέλου, διασφαλίζοντας ότι εξακολουθούσε να παρέχει ένα νέο σύνολο δεδομένων για σκοπούς αξιολόγησης. Ωστόσο, αυτή η προσέγγιση είχε ορισμένα μειονεκτήματα. Αν και τα δεδομένα επικύρωσης δεν χρησιμοποιήθηκαν για εκπαίδευση, είχαν χρησιμοποιηθεί στο παρελθόν για συντονισμό υπερπαραμέτρων (tuning hypeparameters), κάτι που μπορεί να επηρεάσει ελαφρώς τα αποτελέσματα της αξιολόγησης. Χρησιμοποιώντας το σύνολο επικύρωσης με αυτόν τον τρόπο, επιτεύχθηκε να αποκτηθούν σημαντικές μετρήσεις απόδοσης, αλλά η σημασία της ύπαρξης αποκλειστικών ετικετών για ένα σύνολο δοκιμών σε μελλοντικά πειράματα είναι σημαντική.

Ένα άλλο πρόβλημα ήταν η αντιμετώπιση περιορισμών που σχετίζονται με το υλικό του υπολογιστή. Το σύστημά αποδείχτηκε λίγο κατά τη διάρκεια των διαδικασιών. Δοκιμάστηκε να γίνει εκπαίδευση ξανά με διαφοροποίηση και μείωση του δείγματος εκπαίδευσης, ώστε να γίνουν διαφορετικές δοκιμές με διαφορετικά αποτελέσματα. Τα μοντέλα βαθιάς εκμάθησης όπως το RandLA-Net και το KPConv είναι απαιτητικά και η κάρτα γραφικών GTX 1660 Ti, αν και ικανή, ήταν στα όρια όταν επεξεργαζόταν μεγάλα νέφη σημείων. Αυτό οδήγησε σε πιο αργούς χρόνους συμπερασμάτων και περιστασιακά προβλήματα διαχείρισης μνήμης, ιδιαίτερα όταν εκτελούνται και τα δύο μοντέλα σε μεγάλα σύνολα δεδομένων όπως το SemanticKITTI και το Toronto3D. Παρά αυτούς τους περιορισμούς, τα αποτελέσματα εξακολουθούσαν να είναι ενημερωτικά.

Η βιβλιοθήκη Open3D είναι ένα πολύ ισχυρό και ευέλικτο εργαλείο για την επεξεργασία 3Δ δεδομένων, προσφέροντας διάφορες εφαρμογές όπως η όραση υπολογιστών και η μηχανική μάθηση. Οι δυνατότητές, πέρα από τεχνικές μηχανικής μάθησης, συμπεριλαμβάνουν την επεξεργασία νέφη σημείων, την ανακατασκευή πλέγματος και την οπτικοποίησης.

Ωστόσο, για να επιτευχθούν πιο λεπτομερή αποτελέσματα, οι χρήστες ενδέχεται να χρειαστεί μερικές φορές να εμβαθύνουν στον υποκείμενο κώδικα της βιβλιοθήκης. Ενώ το Open3D-ML παρέχει ένα ολοκληρωμένο σύνολο μοντέλων και δεδομένων, ορισμένες μετρήσεις ενδέχεται να μην είναι εύκολα προσβάσιμες. Σε τέτοιες περιπτώσεις, οι χρήστες πρέπει να διερευνήσουν παραπάνω για να εφαρμόσουν προσαρμοσμένες λειτουργίες. Ένα παράδειγμα είναι που προέκυψε, αφορούσε την εξαγωγή των βαθμολογιών πρόβλεψης σε δείγμα δοκιμής. Κανονικά, όταν γίνεται δοκιμής σε ένα υποσύνολο δείγματος με χρήση ενός προεκπαιδευμένου μοντέλου, θα έπρεπε να παρέχονται οι αντίστοιχες βαθμολογίες της κάθε πρόβλεψης για κάθε σημείο αλλά και συνολικά. Η βιβλιοθήκη δεν έδινε κώδικα για την εξαγωγή τους οπότε έπρεπε με τη σύνταξη ενός script να εξαχθούν. Ενώ έγινε αυτό το πρόβλημα στη συνέχεια ήταν ότι στις τιμές είχε πραγματοποιηθεί μια κανονικοποίηση, λογικά για κάποιες επεξεργασίες κατά την πρόβλεψη, με αποτέλεσμα οι τιμές βαθμολογίας να είναι σε ένα εύρος που δεν ήταν εφικτό να βγουν κάποια συμπεράσματα. Ο ίδιος προβληματισμός υπήρχε και στο πεδίο των προβλημάτων στη σελίδα του GitHub από συνεισφορά των χρηστών. Το πεδίο του φόρουμ μπορεί να παρέχει πρόσθετες πληροφορίες και παραδείγματα από τη κοινότητα που να βοηθούν.

Για τη βελτίωση της απόδοσης μοντέλων για την 3Δ σημασιολογική κατάτμηση και την ανίχνευση αντικειμένων, μπορούν να ληφθούν υπόψη διάφορες στρατηγικές. Ένας τρόπος είναι η συμπλήρωση του συνόλου εκπαίδευσης με πρόσθετα δεδομένα, μπορεί να βελτιώσει την ικανότητα γενίκευσης του μοντέλου. Όμως δεν είναι ο μοναδικός τρόπος αύξησης του συνόλου εκπαίδευσης η υλοποίηση νέων σαρώσεων. Τεχνικές όπως η περιστροφή, η αλλαγή κλίμακας ή προσθήκη θορύβου επίσης θεωρούνται αύξηση των δεδομένων που μπορούν να συμβάλουν στη βελτίωση του μοντέλου.

Ο πειραματισμός με διαφορετικές υπερπαραμέτρους, όπως οι ρυθμοί εκμάθησης, τα μεγέθη παρτίδων και ο αριθμός των εποχών, μπορεί να οδηγήσει σε καλύτερη σύγκλιση και βελτιωμένη απόδοση του μοντέλου.

Επίσης, η βελτίωση της αρχιτεκτονικής μοντέλων είναι άλλος ένας τρόπος. Η διερεύνηση παραλλαγών στην αρχιτεκτονική του μοντέλου, όπως η επεξεργασία πολλαπλών κλιμάκων, μπορεί να οδηγήσει σε βελτιωμένη εξαγωγή χαρακτηριστικών.

Ένας άλλος τρόπος που θα μπορούσε να βοηθήσει στη βελτίωση των μοντέλων η διασφάλιση ετικετών υψηλής ποιότητας. Σε ένα από τα παραδείγματα που υλοποιήθηκε πρόβλεψη και αξιολόγηση σε δεδομένα του KITTI, κατά την απεικόνιση φάνηκαν προβλέψεις πεζών και οχημάτων που δεν τους είχαν δοθεί ετικέτες. Από τη στιγμή που στην αξιολόγηση γίνεται σύγκριση προβλέψεων με πραγματικές τιμές, αν ξαφνικά εμφανιστεί μια πρόβλεψη που δεν έχει με τι να συγκριθεί, αυτό ρίχνει απευθείας την αξιοπιστία του μοντέλου.

Βιβλιογραφία

Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., & Gall, J. (2019). *SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences*.

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Bridle, J. (1989). Training Stochastic Model Recognition Algorithms as Networks can Lead to Maximum Mutual Information Estimation of Parameters. In D. Touretzky (Ed.), *Advances in Neural Information Processing Systems* (Vol. 2). Morgan-Kaufmann. https://proceedings.neurips.cc/paper_files/paper/1989/file/0336dcbab05b9d5ad24f4333c7658a0e-Paper.pdf

Campbell, M., Hoane, A. J., & Hsu, F. (2002). Deep Blue. *Artificial Intelligence*, *134*(1–2), 57–83. https://doi.org/10.1016/S0004-3702(01)00129-1

Choy, C., Gwak, J., & Savarese, S. (2019). *4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks*.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/BF00994018

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*(1), 21–27. https://doi.org/10.1109/TIT.1967.1053964

Cramer, J. S. (2003). The Origins of Logistic Regression. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.360300

Deng, J., Shi, S., Li, P., Zhou, W., Zhang, Y., & Li, H. (2020). *Voxel R-CNN: Towards High Performance Voxel-based 3D Object Detection*.

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, *12*, 2121–2159.

Duda, J. (2019). *SGD momentum optimizer with step estimation by online parabola model*.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5). https://doi.org/10.1214/aos/1013203451

Fukushima, K. (1975). Cognitron: A self-organizing multilayered neural network. *Biological Cybernetics*, *20*(3–4), 121–136. https://doi.org/10.1007/BF00342633

Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 3354–3361. https://doi.org/10.1109/CVPR.2012.6248074

Graham, B., Engelcke, M., & van der Maaten, L. (2017). *3D Semantic Segmentation with Submanifold Sparse Convolutional Networks*.

Gugerty, L. (2006). Newell and Simon's Logic Theorist: Historical Background and Impact on Cognitive Modeling. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *50*(9), 880–884. https://doi.org/10.1177/154193120605000904

Hebb, D. O. (2005). *The Organization of Behavior*. Psychology Press. https://doi.org/10.4324/9781410612403

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. https://doi.org/10.1109/CVPR.2016.90

Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., & Markham, A. (2019). *RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds*.

Kingma, D. P., & Ba, J. (2014). *Adam: A Method for Stochastic Optimization*.

Lang, A. H., Vora, S., Caesar, H., Zhou, L., Yang, J., & Beijbom, O. (2018). *PointPillars: Fast Encoders for Object Detection from Point Clouds*.

Lindsay, R. K., Buchanan, B. G., Feigenbaum, E. A., & Lederberg, J. (1993). DENDRAL: A case study of the first expert system for scientific hypothesis formation. *Artificial Intelligence*, *61*(2), 209–261. https://doi.org/10.1016/0004-3702(93)90068-M

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollar, P. (2020). Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(2), 318–327. https://doi.org/10.1109/TPAMI.2018.2858826

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2015). *SSD: Single Shot MultiBox Detector*. https://doi.org/10.1007/978-3-319-46448-0_2

Lu, L. (2020). Dying ReLU and Initialization: Theory and Numerical Examples. *Communications in Computational Physics*, *28*(5), 1671–1706. https://doi.org/10.4208/cicp.OA-2020-0165

Macqueen, J. (1967). *SOME METHODS FOR CLASSIFICATION AND ANALYSIS OF MULTIVARIATE OBSERVATIONS*.

Mccarthy, J. (1979). *History of Lisp*.

McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (2006). *A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence*.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115–133. https://doi.org/10.1007/BF02478259

Minsky, M., & Papert, S. A. (2017). *Perceptrons: An Introduction to Computational Geometry*. The MIT Press. https://doi.org/10.7551/mitpress/11301.001.0001

Mitchell, T. M. (Tom M. (n.d.). *Machine Learning*.

O'Shea, K., & Nash, R. (2015). *An Introduction to Convolutional Neural Networks*.

Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space . *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *2*(11), 559–572. https://doi.org/10.1080/14786440109462720

Rokach, L., & Maimon, O. (2005). Decision Trees. In *Data Mining and Knowledge Discovery Handbook* (Vol. 6, pp. 165–192). Springer-Verlag. https://doi.org/10.1007/0-387-25465-X_9

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386–408. https://doi.org/10.1037/h0042519

Ruder, S. (2016). *An overview of gradient descent optimization algorithms*.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536. https://doi.org/10.1038/323533a0

Samuel, A. L. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, *3*(3), 210–229. https://doi.org/10.1147/rd.33.0210

Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., & Li, H. (2019). *PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection*.

Stanton, J. M. (2001). Galton, Pearson, and the Peas: A Brief History of Linear Regression for Statistics Instructors. *Journal of Statistics Education*, *9*(3). https://doi.org/10.1080/10691898.2001.11910537

Tan, W., Qin, N., Ma, L., Li, Y., Du, J., Cai, G., Yang, K., & Li, J. (2020). *Toronto-3D: A Large-scale Mobile LiDAR Dataset for Semantic Segmentation of Urban Roadways*. https://doi.org/10.1109/CVPRW50498.2020.00109

Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F., & Guibas, L. J. (2019). *KPConv: Flexible and Deformable Convolution for Point Clouds*.

Turing, A. M. (1950). Computing Machinery and Intelligence. *Mind*, *59*(236), 433–460. http://www.jstor.org/stable/2251299

van Melle, W. (1978). MYCIN: a knowledge-based consultation program for infectious disease diagnosis. *International Journal of Man-Machine Studies*, *10*(3), 313–322. https://doi.org/10.1016/S0020-7373(78)80049-2

Weizenbaum, J. (1966). ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, *9*(1), 36–45. https://doi.org/10.1145/365153.365168

Xu, B., Wang, N., Chen, T., & Li, M. (2015). *Empirical Evaluation of Rectified Activations in Convolutional Network*.

Xu, M., Ding, R., Zhao, H., & Qi, X. (2021). *PAConv: Position Adaptive Convolution with Dynamic Kernel Assembling on Point Clouds*.

Yin, T., Zhou, X., & Krähenbühl, P. (2020). *Center-based 3D Object Detection and Tracking*.

Zhao, H., Jiang, L., Jia, J., Torr, P., & Koltun, V. (2020). *Point Transformer*.

Zhou, Q.-Y., Park, J., & Koltun, V. (2018). *Open3D: A Modern Library for 3D Data Processing*.