ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΤΟΠΟΓΡΑΦΙΑΣ ΚΑΙ ΓΕΩΠΛΗΡΟΦΟΡΙΚΗΣ

Διπλωματική εργασία

Χρήση Τεχνικών Βαθιάς Μάθησης για την Αναγνώριση Γεωγραφικών Χαρακτηριστικών σε Ιστορικούς Χάρτες

Χρήστος Ξύδας

ΑΜ: 13098

Επιβλέπων

Αναστάσιος Λ. Κεσίδης

Αθήνα, Οκτώβριος 2021

UNIVERSITY OF WEST ATTICA

SCHOOL OF ENGINEERING

DEPARTMENT OF SURVEYING AND GEOINFORMATICS ENGINEERING

Diploma Thesis

Deep learning techniques for the recognition of geographic features in historical maps

Christos Xydas

RN: 13098

Supervisor

Anastasios L. Kesidis

Athens, October 2021

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΤΟΠΟΓΡΑΦΙΑΣ ΚΑΙ ΓΕΩΠΛΗΡΟΦΟΡΙΚΗΣ

Χρήση Τεχνικών Βαθιάς Μάθησης για την Αναγνώριση Γεωγραφικών Χαρακτηριστικών σε Ιστορικούς Χάρτες

Μέλη Εξεταστικής Επιτροπής συμπεριλαμβανομένου και του Εισηγητή

Η πτυχιακή/διπλωματική εργασία εξετάστηκε επιτυχώς από την κάτωθι Εξεταστική Επιτροπή:

Α/α	ΟΝΟΜΑ ΕΠΩΝΥΜΟ	ΒΑΘΜΙΔΑ/ΙΔΙΟΤΗΤΑ	ΨΗΦΙΑΚΗ ΥΠΟΓΡΑΦΗ
	Αναστάσιος Κεσίδης (Επιβλέπων)	Αναπληρωτής Καθηγητής ΠΑΔΑ	Anastasios Kesidis Digitally signed by Anastasios Kesidis Date: 2021.11.14 21:28:31 +02'00'
	Έλλη Πέτσα	Καθηγήτρια ΠΑΔΑ	Eleni Petsa Digitally signed by Eleni Petsa Date: 2021.11.15 12:47:09 +02'00'
	Βασίλειος Κρασανάκης	Επίκουρος Καθηγητής ΠΑΔΑ	Vasileios Krasanakis Digitally signed by Vasileios Krasanakis Date: 2021.11.14 21:39:37 +02'00'

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Ξύδας Χρήστος, του Γρηγορίου, με αριθμό μητρώου 509130980227, φοιτητής του Πανεπιστημίου Δυτικής Αττικής, της Σχολής Μηχανικών, του Τμήματος Μηχανικών 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

Περιεχόμενα

Περίληψη

Οι ιστορικοί χάρτες είναι πηγές πολύτιμης πληροφορίας η οποία μπορεί να αξιοποιηθεί σε ένα πλήθος ερευνητικών πεδίων και συνεπώς αποτελούν σημαντικό επιστημονικό εργαλείο, εξαιτίας των συμπερασμάτων που μπορούν να εξαχθούν, σε ιστορικό, οικολογικό και κοινωνικό-οικονομικό επίπεδο. Η εξαγωγή πληροφορίας από χάρτες είναι εν γένει μια σύνθετη διαδικασία η οποία γίνεται ακόμη πιο δύσκολη όταν αφορά υλικό του παρελθόντος λόγω των ιδιαιτεροτήτων που τους χαρακτηρίζουν. Η παρούσα διπλωματική εργασία πραγματεύεται την επίλυση του προβλήματος εντοπισμού γεωγραφικών χαρακτηριστικών, και συγκεκριμένα κτιρίων, σε εικόνες από ιστορικούς τοπογραφικούς χάρτες. Για το σκοπό αυτό μελετήθηκαν σύγχρονα συστήματα μηχανικής μάθησης και ειδικότερα τα τεχνητά νευρωνικά δίκτυα. Ιδιαίτερη έμφαση δόθηκε στις τεχνικές βαθιάς μάθησης που βασίζονται στα υπολογιστικά μοντέλα των Συνελικτικών Νευρωνικών Δικτύων. Στην συνέχεια, υλοποιήθηκε ένα σύστημα βαθιάς μάθησης για τον εντοπισμό των κτιρίων και παράλληλα αναπτύχθηκαν οι κατάλληλες διεργασίες με σκοπό την δημιουργία του κατάλληλου συνόλου δεδομένων εκπαίδευσης από κατάλληλα επιλεγμένα δείγματα εικόνων του χάρτη. Διερευνήθηκε επιπλέον η επίδραση του μεγέθους των δεδομένων εκπαίδευσης καθώς και των διαφορετικών μεθόδων δειγματοληψίας τους στην τελική απόδοση του συστήματος. Επιπλέον, εξετάσθηκαν κρίσιμες παράμετροι της δομής του συνελικτικού δικτύου και κατά πόσον η μεταβολή τους επηρεάζει θετικά την ακρίβεια της προτεινόμενης μεθόδου. Σε κάθε περίπτωση, τα αποτελέσματα των πειραμάτων αξιολογήθηκαν τόσο ποιοτικά όσο και ποσοτικά μέσω ενός συνόλου μετρικών αποτίμησης ώστε να εξαχθούν τα απαραίτητα συμπεράσματα.

Abstract

Historical maps are sources of valuable information that can be used in a number of research fields and are therefore an important scientific tool, because of the conclusions that can be drawn, at the historical, ecological and socio-economic level. Extracting information from maps is generally a complex process which becomes even more difficult when it comes to historical material due to the peculiarities that characterize them. This thesis deals with the problem of locating geographical features, and in particular buildings, in images corresponding to historical topographic maps. For this purpose, modern machine learning systems and in particular artificial neural networks are studied. Particular emphasis is placed on deep learning techniques based on computer models of Convolutional Neural Networks. Then, a deep learning system is implemented to locate the buildings and at the same time proper processes are developed in order to create an adequate set of training data from selected image patches of the historical maps. The effect of the patch size as well as their different sampling methods on the final performance of the system are further investigated. In addition, critical parameters of the network's structure are examined and whether their change positively affects the accuracy of the proposed method. In each case, the results of the experiments are evaluated both qualitatively and quantitatively through a set of evaluation metrics in order to draw the necessary conclusions.

Πίνακας Εικόνων

1 Εισαγωγή

Η μελέτη ενός ιστορικού χάρτη μπορεί να παράσχει πληροφορίες σχετικά με την ακρίβεια, την τεχνολογία, ακόμα και την επιστημονική γνώση της εποχής κατά την οποία δημιουργήθηκε. Η σύγχρονη ψηφιακή τεχνολογία, επιτρέπει στον ερευνητή του σήμερα να “αντιγράψει” έναν παλαιό χάρτη και να αντλήσει πληροφορία συσχετίζοντας τον με το σήμερα. Κατά αυτόν τον τρόπο, μία χαρτογραφική σύνδεση του παρελθόντος, με το παρόν, είναι εφικτή, με παράλληλη οπτικοποίηση γεωγραφικών χαρακτηριστικών διαφορετικών χρονολογικών περιόδων. Πλήθος συμπερασμάτων μπορούν να εξαχθούν, πραγματοποιώντας συσχετίσεις μέσα από τη συγκριτική μελέτη χαρτογραφικών δεδομένων του παρελθόντος και του παρόντος, και μάλιστα σε τέτοιο βαθμό, που δεν θα ήταν υπερβολικός ο χαρακτηρισμός ενός παλαιού χάρτη ως κάποιου είδους χρονομηχανής, που μπορεί να εξηγήσει πολύ περισσότερα για το παρελθόν, από τα γεωγραφικά χαρακτηριστικά κάποιας περιοχής του.

Πολλές βιβλιοθήκες και υπηρεσίες που διαχειρίζονται κρατικά αρχεία, διεθνώς, έχουν ψηφιοποιήσει και ψηφιοποιούν υλικό από χαρτογραφικές συλλογές τις οποίες έχουν συγκεντρώσει. Ένα βασικό επίπεδο ψηφιοποίησης, τέτοιου είδους χαρτογραφικού αρχείου, αποτελείται από εικόνες σκαναρισμένων χαρτών, μαζί με κάποια ετικέτα πληροφοριών όπως, τίτλο, χρονολογία παραγωγής και από ποιόν δημιουργήθηκε. Παρ’όλα αυτά, για να μπορέσουν να γίνουν τέτοιου είδους σκαναριμένοι χάρτες, πιο εύκολα και χρηστικά προσβάσιμοι, είναι επιθυμητή μία πιο δομημένη και συστηματικά οργανωμένη αναπαράσταση της περιεχόμενης πληροφορίας. Αυτή η διαδικασία περιλαμβάνει την τοποθέτηση ετικετών πχ. με τις ονομασίες πόλεων, οδών, υδρογραφικού δικτύου και τον προσδιορισμό διοικητικών ορίων.

Γίνεται αντιληπτό ότι μία δομημένη και συστηματικά οργανωμένη αναπαράσταση της περιεχόμενης πληροφορίας, ενός μεγάλου όγκου σκαναρισμένων χαρτών, προσεγγίζει ένα είδος γεωχωρικής υποδομής, η οποία μπορεί να αποτελέσει σημαντικό υπόβαθρο, για διάφορα επιστημονικά πεδία. Προφανώς, ο εντοπισμός και η απόδοση ετικετών, δηλαδή η πραγματοποίηση ενός είδους σημασιολογικής κατάτμησης, χειρωνακτικά, σε γεωγραφικά χαρακτηριστικά, τόσο μεγάλων όγκων αρχείων, είναι μια ιδιαίτερα κοπιαστική διαδικασία. Η αυτοματοποίηση, λοιπόν, της διαδικασίας αναγνώρισης και εξαγωγής γεωγραφικών χαρακτηριστικών από χάρτες παλαιότερων περιόδων, έχει μεγάλη αξία, καθώς μπορεί να δώσει λύσεις σε καίρια ζητήματα, τα οποία με τη σειρά τους αποτελούν δυνάμει παράγοντες με σημαντικό αντίκτυπο, σε διάφορα επιστημονικά πεδία.

Τέτοιου είδους εγχειρήματα, χαρακτηρίζονται από αρκετές δυσκολίες, οι οποίες πρέπει να αντιμετωπιστούν αποτελεσματικά. Η βασική εξ αυτών, είναι η χαμηλή

γραφική ποιότητα, η οποία χαρακτηρίζει το περιεχόμενο των παλαιών χαρτών. Επιπρόσθετα, ανάλογα του πόσο παλαιός είναι ένας χάρτης, τόσο περισσότερες είναι οι πιθανότητες να έχει σημαντικές φθορές, ακόμα και να λείπει, σε εξαιρετικές περιπτώσεις, κάποιο τμήμα του.

Για την επίτευξη του στόχου της αναγνώρισης αντικειμένων σε παλαιούς χάρτες και την αντιμετώπιση των δυσκολιών που διέπουν το συγκεκριμένο εγχείρημα, απαιτείται μία εύρωστη προσέγγιση, υλοποιήσιμη μέσω κάποιου υπολογιστικού μοντέλου, το οποίο είναι ικανό να μαθαίνει να αναγνωρίζει αντικείμενα ενδιαφέροντος, αφού προηγουμένως έχει εκπαιδευτεί με έναν ικανοποιητικό όγκο δεδομένων. Τα Συνελικτικά Νευρωνικά Δίκτυα, έχουν πρόσφατα συγκεντρώσει μεγάλο ενδιαφέρον όσον αφορά εργασίες αναγνώρισης και εντοπισμού αντικειμένων και ταξινόμησης, αφού είναι σε θέση να εντοπίζουν πολύπλοκες δομές και μοτίβα, εντός των δεδομένων, μεταφέροντας χαρακτηριστικά μεταξύ πολλαπλών κρυφών επιπέδων, με ένα μη γραμμικό τρόπο.

Στην παρούσα εργασία, προτείνεται μία προσέγγιση για τον εντοπισμό κτιρίων, σε ιστορικούς τοπογραφικούς χάρτες, βάσει μίας από εικόνα σε εικόνα παλινδρόμησης, στηριζόμενης σε ένα Συνελικτικό Νευρωνικό Δίκτυο. Το δίκτυο εκπαιδεύεται με πλήθος ζευγαριών δειγμάτων εικόνων. Ως εικόνες εισόδου, χρησιμοποιούνται δείγματα από σαρωμένο τοπογραφικό χάρτη, ο οποίος απεικονίζει την περιοχή της Πετρούπολης και παρήχθη το 1970. Ως εικόνες εξόδου, χρησιμοποιούνται δείγματα από εικόνα επαληθευμένων δεδομένων (ground truth), της αντίστοιχης περιοχής, η οποία έχει επεξεργαστεί από ειδικό και απεικονίζει μόνο τα κτίρια. Η αρχιτεκτονική του Συνελικτικού Νευρωνικού Δικτύου είναι τέτοια, ώστε εκπαιδεύεται πραγματοποιώντας “πυκνές προβλέψεις”, επιπέδου εικονοστοιχείου και δοκιμάζεται στη συνέχεια, αντίστοιχα, ως προς τις προβλέψεις του, χρησιμοποιώντας σετ δεδομένων ελέγχου. Πλήθος πειραμάτων πραγματοποιούνται, τα οποία διακρίνονται βάσει του τρόπου με τον οποία παρήχθησαν τα σετ δεδομένων με τα δείγματα εικόνων και του μεγέθους των δειγμάτων εικόνων. Τα αποτελέσματα των διαφορετικών αυτών πειραμάτων, αξιολογούνται με τον υπολογισμό πλήθους μετρικών, τα οποία αξιολογούν τις προβλέψεις των διαφορετικών, εκπαιδευμένων Συνελικτικών Νευρωνικών Δικτύων, σε επίπεδο εικονοστοιχείων.

Η δομή της εργασίας έχει ως εξής: στο δεύτερο κεφάλαιο πραγματοποιείται παρουσίαση του θεωρητικού υποβάθρου των Τεχνητών Νευρωνικών Δικτύων και παρουσιάζονται μέθοδοι και τεχνικές οι οποίες έχουν εφαρμογή, γενικότερα, στον χώρο της Μηχανικής Μάθησης. Στο τρίτο κεφάλαιο παρουσιάζονται βασικά θέματα της θεωρίας που διέπει τα Συνελικτικά Νευρωνικά Δίκτυα. Στο τέταρτο κεφάλαιο παρουσιάζονται οι μέθοδοι και τα βήματα που ακολουθήθηκαν για την υλοποίηση της προτεινόμενης μεθόδου καθώς και τα πειραματικά αποτελέσματα. Τέλος, στο κεφάλαιο 5 παρουσιάζονται τα συμπεράσματα που προέκυψαν σχετικά με την

αποτελεσματικότητα της προτεινόμενης μεθόδου και προτείνονται μελλοντικοί τρόποι βελτίωσής της.

2 Τεχνητά Νευρωνικά Δίκτυα

2.1 Εισαγωγή

Η Τεχνητή Νοημοσύνη-ΤΝ (Artificial Intelligence-AI) είναι σήμερα μία από τις νεότερες επιστήμες, η οποία συγκεντρώνει ένα από τα μεγαλύτερα, αν όχι το μεγαλύτερο, ποσοστά της παγκόσμιας ερευνητικής δραστηριότητας. Σχετικά με το τι είναι ΤΝ, ποικίλλουν οι ορισμοί που έχουν δοθεί από επιστήμονες του χώρου. Ένας σύντομος και καλά διατυπωμένος ορισμός, είναι ο ακόλουθος: «Τεχνητή Νοημοσύνη είναι η μελέτη των νοητικών ικανοτήτων με τη χρήση υπολογιστικών μοντέλων» (Charniak και McDermott, 1985). Για πρώτη φορά, ο όρος ΤΝ διατυπώθηκε επίσημα στο Dartmouth College, το 1956, από τον John McCarthy, όπου σε μία συνάντηση εργασίας που διήρκεσε δύο μήνες και έλαβαν μέρος ερευνητές του χώρου της θεωρίας των υπολογισμών, των αυτόματων συστημάτων και της μελέτης της νοημοσύνης, συμφώνησαν ότι ο συγκεκριμένος όρος ήταν ο καταλληλότερος για το πεδίο αυτό.

Πριν, όμως, από την επίσημη διατύπωση του όρου ΤΝ, ο επιστήμονας των υπολογιστών Άλαν Τούρινγκ (1913-1954), σε εργασία (Turing 1950) του το 1950, πρότεινε την εξέταση του ερωτήματος: «Μπορούν οι μηχανές να σκεφτούν;». Στη συγκεκριμένη εργασία επιχειρηματολόγησε στη βάση της μη ύπαρξης οποιασδήποτε αξιόπιστης απόδειξης ότι οι μηχανές δε μπορούν να σκεφτούν όπως οι άνθρωποι, παρουσίασε το Τεστ Τούρινγκ και ουσιαστικά διατύπωσε μία πλήρη προοπτική για την ΤΝ, στη βάση της οποίας εργάστηκαν μετέπειτα αρκετοί επιστήμονες του χώρου αυτού. Στο Τεστ Τούρινγκ λαμβάνουν μέρος τρεις οντότητες, ένα φυσικό πρόσωπο-«ανακριτής», ένα δεύτερο φυσικό πρόσωπο και μία υπολογιστική μηχανή. Το πρώτο φυσικό πρόσωπο-«ανακριτής» υποβάλλει κάποιες ερωτήσεις και για κάθε ερώτηση λαμβάνει μία απάντηση από το δεύτερο φυσικό πρόσωπο και μία από την υπολογιστική μηχανή. Ο «ανακριτής» βρίσκεται σε διαφορετικό χώρο από τις άλλες δύο οντότητες, χωρίς να γνωρίζει κάθε φορά ποιος απαντά, και επιχειρεί να ξεχωρίσει αν δόθηκε η απάντηση από τον άνθρωπο ή την υπολογιστική μηχανή.

Σήμερα, η ΤΝ χρησιμοποιείται σε πλήθος εφαρμογών, είτε της καθημερινότητας ενός απλού ανθρώπου, είτε για πιο εξειδικευμένη χρήση, επαγγελματικού και ακαδημαϊκού-ερευνητικού επιπέδου. Το περιεχόμενο που εμφανίζει στον καθένα, στην αρχική σελίδα της, μία πλατφόρμα κοινωνικής δικτύωσης, ο τρόπος με τον οποίο ένα έξυπνο κινητό τηλέφωνο εντοπίζει αντικείμενα σε εικόνες, τα αυτόνομα οχήματα, ο τρόπος με τον οποίο η ανεπιθύμητη αλληλογραφία εντοπίζεται και απομονώνεται, και άλλα πάρα πολλά στοιχεία της καθημερινότητας ενός μέσου ανθρώπου, είναι όλα ΤΝ.

Η ΤΝ αποτελεί μία από τις επιστήμες που συνθέτουν αυτό που ονομάζεται Γνωστική ή Γνωσιακή Επιστήμη και χωρίζεται σε κάποιους κλάδους , όπως αυτοί της Επίλυσης Προβλημάτων, της Αναπαράστασης Γνώσης, των Συστημάτων Βασισμένων στη Γνώση, των Νοημόνων Πρακτόρων και της Μηχανικής Μάθησης. Ανάλογα το πεδίο εφαρμογών και ερευνητικής δραστηριότητας, συχνά κάποιοι από αυτούς τους κλάδους αλληλεπικαλύπτονται. Κάποιοι από τους ερευνητικούς χώρους, στους οποίους εστιάζουν πολλοί επιστήμονες παγκοσμίως, είναι η Μάθηση Μηχανής, η Όραση Υπολογιστών, η Ρομποτική, η Επεξεργασία Φυσικής Γλώσσας και τα Έμπειρα Συστήματα.

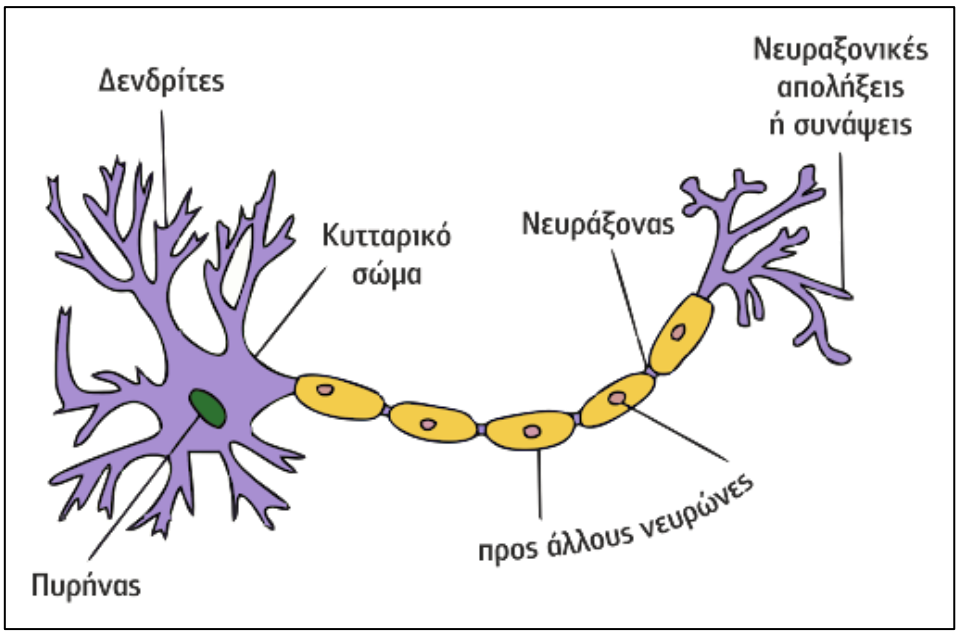
Τα Τεχνητά Νευρωνικά Δίκτυα είναι ένα υποσύνολο της Μάθησης Μηχανής. Η Μάθηση Μηχανής είναι ένας κλάδος της ΤΝ, που σήμερα εξελίσσεται ταχύτατα. Ως όρος διατυπώθηκε για πρώτη φορά το 1959, από τον Arthur Samuel, ένα πρωτοπόρο του κλάδου αυτού αλλά και της ΤΝ γενικότερα, ο οποίος εργαζόταν τότε στην IBM. Ο ίδιος, της απέδωσε τον ακόλουθο ορισμό: «*Μάθηση Μηχανής είναι το πεδίο μελέτης που προσδίδει στους υπολογιστές την ικανότητα να μαθαίνουν χωρίς να έχουν προγραμματιστεί ρητά για αυτό*» (Arthur Samuel, 1959). Ένας δεύτερος ορισμός, πιο τεχνικός, αλλά και αρκετά σαφής, είναι ο ακόλουθος: «*Ένα πρόγραμμα ηλεκτρονικού υπολογιστή λέγεται ότι μαθαίνει από μία εμπειρία Ε, σε σχέση με κάποια τάξη εργασιών Τ και ένα μέτρο απόδοσης Ρ, αν η απόδοση του στις εργασίες της Τ, όπως μετριέται από το Ρ, βελτιώνεται με την εμπειρία Ε*» (Tom Mitchell, 1998).

Η Μάθηση Μηχανής, ανάλογα με το είδος των αλγορίθμων της, χωρίζεται σε τρεις διαφορετικές κατηγορίες:

- Επιβλεπόμενη Μάθηση (Supervised Learning): Ο αλγόριθμος μαθαίνει μία συνάρτηση, από ένα σύνολο γνωστών παραδειγμάτων εκπαίδευσης (ζευγαριών δεδομένων εισόδου-εξόδου), που για νέες τιμές εισόδου, να υπολογίζει τις αντίστοιχες άγνωστες τιμές εξόδου τους. Εφαρμογές στις οποίες χρησιμοποιούνται αλγόριθμοι Επιβλεπόμενης Μάθησης, είναι αυτές της πρόβλεψης και της ταξινόμησης.
- Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning): Ο αλγόριθμος μαθαίνει ένα μοντέλο, με την έννοια των μοτίβων, για ένα σύνολο δεδομένων εισόδου, χωρίς να γνωρίζει τις αντίστοιχες εξόδους. Εφαρμογές στις οποίες χρησιμοποιούνται αλγόριθμοι Μη Επιβλεπόμενης Μάθησης, είναι αυτές της ομαδοποίησης και της ανάλυσης συσχετισμών.
- Ενισχυτική Μάθηση (Reinforcement Learning): Ο αλγόριθμος μαθαίνει μία στρατηγική ενεργειών, μέσα από άμεση αλληλεπίδραση με το περιβάλλον του, βάσει της επιβράβευσης επιθυμητών συμπεριφορών ή/και της τιμώρησης ανεπιθύμητων συμπεριφορών. Οι αλγόριθμοι Ενισχυτικής Μάθησης χρησιμοποιούνται κυρίως σε εφαρμογές ρομποτικής, βιομηχανικές, αυτόνομων οχημάτων και σχεδιασμού διεργασιών.

2.2 Ιστορική Αναδρομή

Το 1943, ο νευροφυσιολόγος Warren McCulloch και ο μαθηματικός Walter Pitts εκπόνησαν μία εργασία (Warren S. McCulloch 1943), η οποία θεωρείται σήμερα η πρώτη εργασία Τεχνητής Νοημοσύνης, αλλά και η απαρχή των Τεχνητών Νευρωνικών Δικτύων. Ο πυρήνας της ιδέας τους για τη συγκεκριμένη εργασία ήταν η προτυποποίηση του πως μπορεί να λειτουργούν οι βιολογικοί νευρώνες στον ανθρώπινο εγκέφαλο.



Εικόνα 1: Βιολογικός νευρώνας.

Ο νευρώνας είναι το κύριο δομικό στοιχείο του εγκεφάλου και η βασική μονάδα μέσω της οποίας πραγματοποιεί υπολογισμούς. Το ανθρώπινο νευρικό σύστημα περιέχει γύρω στους 85 δισεκατομμύρια νευρώνες, με τους περίπου 10 δισεκατομμύρια από αυτούς να βρίσκονται στον εγκέφαλο, οργανωμένοι σε ομάδες. Κάθε τέτοια ομάδα νευρώνων, συνιστά ένα νευρωνικό δίκτυο. Οι νευρώνες συνδέονται μεταξύ τους στις συνάψεις, οι οποίες είναι τα σημεία ένωσης των δενδριτών τους, με δενδρίτες άλλων νευρώνων και μέσω των οποίων λαμβάνει ηλεκτρικά σήματα εισόδου. Ο κάθε νευρώνας, από τα ηλεκτρικά αυτά σήματα εισόδου, παράγει σήματα εξόδου, τα οποία μεταβιβάζει μέσω του μοναδικού νευροάξονα του, ο οποίος στο τέλος του διακλαδίζεται και συνδέεται στις συνάψεις με δενδρίτες άλλων νευρώνων. Τα σήματα που λαμβάνει ένας νευρώνας, μέσω των συνάψεων, από πολλαπλούς άλλους νευρώνες, σταθμίζονται και αθροίζονται, και μεταδίδονται σε άλλους νευρώνες, βάσει των συναπτικών δυνάμεων. Το δυναμικό του κάθε νευρώνα μεταβάλλεται, ανάλογα με τα σήματα που δέχεται, με

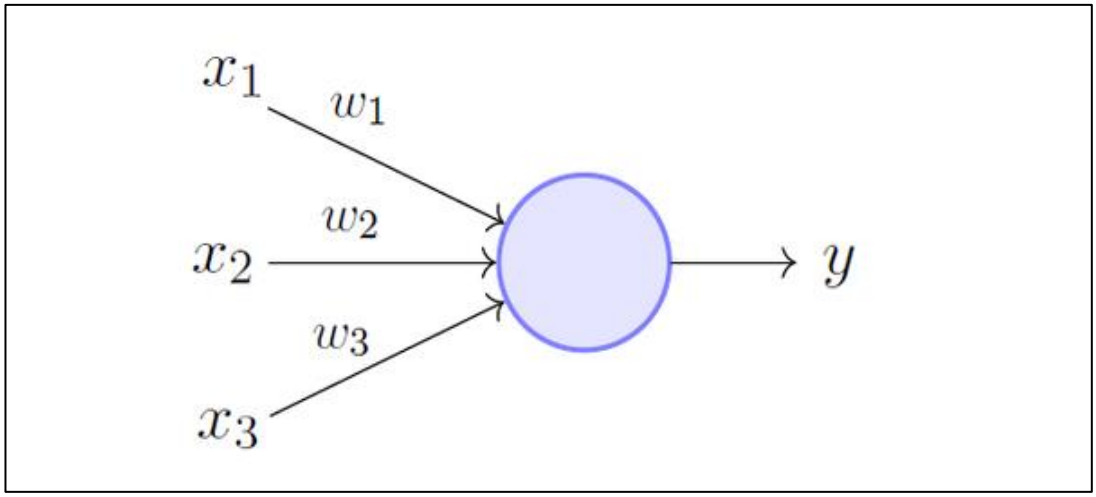
αποτέλεσμα αν ξεπεράσει μία τιμή κατωφλίου, να διεγείρεται-ενεργοποιείται και να μεταδίδει σήματα στους νευρώνες με τους οποίους συνδέεται.

Οι McCulloch και Pitts, λοιπόν, βάσει της παραπάνω γνώσης, εισήγαγαν το πρώτο υπολογιστικό μοντέλο ενός νευρώνα. Κατά το συγκεκριμένο μοντέλο, κάθε νευρώνας μπορεί να δεχτεί μόνο δυαδικές τιμές εισόδου (0 ή 1). Εντός του νευρώνα, γίνεται ουσιαστικά υπολογισμός δύο συναρτήσεων. Η πρώτη συνάρτηση υπολογίζει το άθροισμα όλων των τιμών εισόδου και η δεύτερη υπολογίζει αν αυτό το άθροισμα είναι μεγαλύτερο ή όχι από κάποια ορισμένη τιμή κατωφλίωσης˙ αν είναι, ο νευρώνας εξάγει την τιμή 1, και αν δεν είναι, ο νευρώνας εξάγει την τιμή 0. Έτσι, κάθε νευρώνας είναι είτε ενεργός (1), είτε ανενεργός (0), και εξάγει την αντίστοιχη τιμή σε άλλον ή άλλους νευρώνες, στην περίπτωση όπου υπάρχει δικτυακή δομή. Από τη συγκεκριμένη εργασία, αποδεικνύεται ότι κάθε υπολογίσιμη συνάρτηση μπορεί να υπολογιστεί από ένα δίκτυο τέτοιων νευρώνων, και ότι όλες οι λογικές πράξεις (και, ή, όχι, κ.λπ.) μπορούν να μοντελοποιηθούν τοποθετώντας την κατάλληλη τιμή κατωφλίωσης.

Το βασικό μειονέκτημα του μοντέλου νευρώνα των McCulloch και Pitts, ήταν ότι δεν υπήρχε η δυνατότητα ενσωμάτωσης βαρών για τις διαφορετικές τιμές εισόδου. Ο Donald Hebb, στο βιβλίο (Hebb 1949) του που εξέδωσε το 1949, διατύπωσε έναν απλό κανόνα βάσει του οποίου ενισχύονται ή αποδυναμώνονται οι συνδέσεις μεταξύ των νευρώνων. Στη σελίδα 62 του συγκεκριμένου βιβλίου αναφέρει: «Όταν ο άξονας ενός νευρώνα Α, είναι τόσο κοντά ώστε να διεγείρει ένα νευρώνα Β, και επαναλαμβανόμενα ή επιμόνως συμμετέχει στην ενεργοποίηση του, κάποια διεργασία εξέλιξης ή μεταβολική τροποποίηση λαμβάνει χώρα στον έναν ή και στους δύο νευρώνες, τέτοια ώστε η ικανότητα του Α, ως ενός από τους νευρώνες που ενεργοποιεί τον Β, να αυξάνεται». Η διατύπωση αυτή είναι ο πυρήνας αυτού που ονομάζεται μάθηση Hebb, αλλά περιγράφει και μία θεμελιώδη και απαραίτητη διαδικασία για τη μάθηση και τη μνήμη των ανθρώπων.

Το 1958, ο ψυχολόγος Frank Rosenblatt, δημοσιεύει μία εργασία (Rosenblatt 1958), στην οποία παρουσιάζει ένα είδος τεχνητού νευρώνα, το Perceptron. Η ανάπτυξη του Perceptron είναι εμφανώς εμπνευσμένη από την προγενέστερη δουλειά των McCulloch και Pitts, αλλά το στοιχείο που την καθιστά έως και σήμερα ένα από τα πιο κομβικά σημεία στην εξέλιξη των Τεχνητών Νευρωνικών Δικτύων, είναι η αξιοποίηση του κανόνα της μάθησης Hebb. Σχετικά με τη δομή του, ένα Perceptron λαμβάνει κάποιες δυαδικές τιμές εισόδου και παράγει μία δυαδική τιμή εξόδου. Αυτό είναι κάτι που έως τότε προϋπήρχε. Η καινοτομία του τεχνητού νευρώνα του Rosenblatt, ήταν η εισαγωγή των βαρών, πραγματικών αριθμών οι οποίοι εκφράζουν τη βαρύτητα της αντίστοιχης τιμής εισόδου, για την τιμή εξόδου. Η τιμή εξόδου του κάθε νευρώνα, καθορίζεται από το εάν το άθροισμα των σταθμισμένων τιμών εισόδου είναι μεγαλύτερο ή μικρότερο από μία τιμή κατωφλίου, η οποία είναι ένας πραγματικός αριθμός και παράμετρος του νευρώνα. Γίνεται αντιληπτό ότι

διαφοροποιώντας τα βάρη και την τιμή κατωφλίωσης, το μοντέλο του νευρώνα αλλάζει εντελώς. Το Perceptron, λοιπόν, σταθμίζοντας διαφορετικά στοιχεία, καταλήγει σε κάποια απόφαση.



Εικόνα 2 : Οπτικοποίηση του Perceptron του Rosenblatt.

Βάσει του μοντέλου του Rosenblatt, εντός του μπλε κύκλου της Εικόνας 2, πραγματοποιούνται δύο διαφορετικοί υπολογισμοί. Ο πρώτος υπολογισμός είναι αυτός του αθροίσματος $\sum_{i=1}^{3} w_i x_i$. Ο δεύτερος υπολογισμός είναι ουσιαστικά η εφαρμογή μίας συνάρτησης βήματος στο παραπάνω άθροισμα. Έτσι, λοιπόν, ισχύει:

$$y = \begin{cases} 0 \ \alpha\nu \ \sum_{i=1}^{3} w_i x_i \leq \tau\iota\mu\acute{\eta} \ \kappa\alpha\tau\omega\varphi\lambda\acute{\iota}\omega\sigma\eta\varsigma \\ 1 \ \alpha\nu \ \sum_{i=1}^{3} w_i x_i > \tau\iota\mu\acute{\eta} \ \kappa\alpha\tau\omega\varphi\lambda\acute{\iota}\omega\sigma\eta\varsigma \end{cases} \quad \textbf{2.2.1}$$

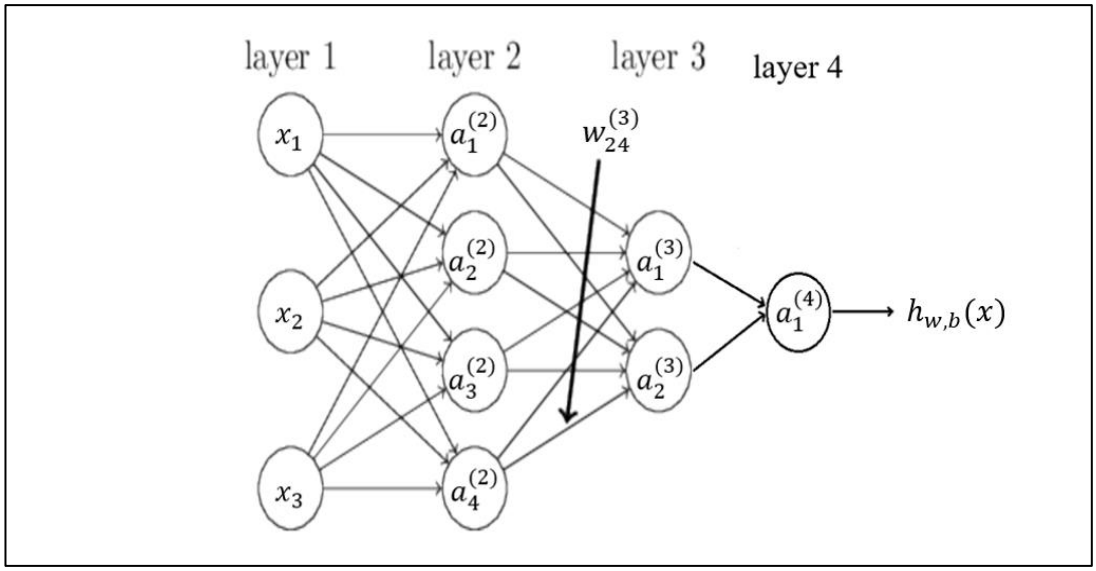
Μετακινώντας την τιμή κατωφλίωσης στην άλλη πλευρά της ανισότητας και αντικαθιστώντας την με κάτι γνωστό ως πόλωση του Perceptron, για την οποία ισχύει $b \equiv -\tau\iota\mu\acute{\eta} \ \kappa\alpha\tau\omega\varphi\lambda\acute{\iota}\omega\sigma\eta\varsigma$, ισχύει:

$$y = \begin{cases} 0 \ \alpha\nu \ \sum_{i=1}^{3} w_i x_i + b \leq 0 \\ 1 \ \alpha\nu \ \sum_{i=1}^{3} w_i x_i + b > 0 \end{cases} \quad \textbf{2.2.2}$$

Γίνεται αντιληπτό, διαισθητικά, ότι στην περίπτωση του Perceptron του Rosenblatt, η πόλωση είναι μία παράμετρος του νευρώνα η οποία ορίζει ένα μέτρο για το πόσο εύκολο ή δύσκολο είναι να εξάγει τιμή ίση με τη μονάδα, δηλαδή να ενεργοποιηθεί.

2.3 Μοντέλο Τεχνητού Νευρωνικού Δικτύου

Ένα νευρωνικό δίκτυο αποτελείται από κόμβους ή μονάδες. Κάθε μονάδα του νευρωνικού δικτύου, είναι ένας τεχνητός νευρώνας, του οποίου η έξοδος, χρησιμοποιείται ως είσοδος, σε κάποιο άλλο νευρώνα, με τον οποίο συνδέεται.



Εικόνα 3: Παράδειγμα τεχνητού νευρωνικού δικτύου πολλών επιπέδων. (Nielsen 2019)

Κάθε στήλη νευρώνων, συνιστά ένα επίπεδο. Το πρώτο επίπεδο, ονομάζεται επίπεδο εισόδου, τα δύο ενδιάμεσα επίπεδα, ονομάζονται κρυφά επίπεδα και το τελευταίο επίπεδο, ονομάζεται επίπεδο εξόδου. Από την Εικόνα 3, δίνεται η εντύπωση ότι από κάθε νευρώνα εξάγονται διαφορετικές τιμές, προς διαφορετικές κατευθύνεις˙ στην πραγματικότητα είναι η ίδια τιμή, η οποία χρησιμοποιείται ως τιμή εισόδου, από πολλούς άλλους νευρώνες. Για τα κλασσικά νευρωνικά δίκτυα, το πιο σύνηθες είδος επιπέδου είναι το πλήρως-συνδεδεμένο επίπεδο, στο οποίο νευρώνες μεταξύ δύο παρακειμένων επιπέδων είναι πλήρως κατά-ζεύγη συνδεδεμένοι, αλλά νευρώνες εντός κάποιου επιπέδου δε μοιράζονται καθόλου συνδέσεις.

Στην παραπάνω εικόνα, έχει επισημανθεί μία από τις συνδέσεις και φαίνεται σαν μέσω της σύνδεσης αυτής να μεταφέρεται το βάρος $w_{24}^{(3)}$, όπως και συμβαίνει. Αυτός είναι ένας συμβολισμός, λοιπόν ,με σκοπό να υποδηλώσουμε το βάρος, σε μία σύνδεση από τον τέταρτο (4ο) νευρώνα, του δεύτερου (2ου) επιπέδου, στο δεύτερο (2ο) νευρώνα, του τρίτου (3ου) επιπέδου του δικτύου. Γενικεύοντας το συμβολισμό

αυτό, ισχύει ότι w_{jk}^{l} είναι το βάρος από τον k νευρώνα, του (l-1) επιπέδου, στο j νευρώνα, του l επιπέδου. Με παρόμοιο τρόπο συμβολίζονται και οι πολώσεις του δικτύου, έτσι η πόλωση του j νευρώνα, του l επιπέδου, συμβολίζεται με $b_j^{(l)}$, η οποία είναι μία πόλωση η οποία έχει μεταφερθεί από ένα νοητό, θα μπορούσε κάποιος να πει, νευρώνα, του (l-1) επιπέδου. Έτσι, λοιπόν, στην Εικόνα 3, εντός των κύκλων που συμβολίζουν τους τεχνητούς νευρώνες, θα μπορούσε να υπάρχει, ανάλογα το επίπεδο και το νευρώνα, η αντίστοιχη πόλωση $b_j^{(l)}$. Με $a_j^{(l)}$ συμβολίζεται η ενεργοποίηση του νευρώνα j, του επιπέδου l, η οποία είναι το αποτέλεσμα της εφαρμογής μία συνάρτησης και για την οποία παρουσιάζονται περισσότερα παρακάτω. Για το επίπεδο εισόδου, με $a_i^{(1)}$=x_i υποδηλώνονται οι τιμές εισόδου. Τα βάρη και οι πολώσεις είναι παράμετροι του δικτύου και όταν ξεκινάει η διαδικασία που ονομάζεται προς τα εμπρός τροφοδότηση (ή προς τα εμπρός διάδοση) και αναλύεται παρακάτω, αρχικοποιούνται ώστε να είναι δυνατό να ξεκινήσουν οι υπολογισμοί. Υπάρχουν διάφορες τεχνικές για να πραγματοποιηθεί η συγκεκριμένη αρχικοποίηση. Οι υπολογισμοί, λοιπόν, που πραγματοποιεί το δίκτυο, κατά τη διαδικασία της προς τα εμπρός διάδοσης, είναι οι παρακάτω:

$$
\begin{aligned}
a_1^{(2)} &= f(w_{11}^{(2)} * x_1 + w_{12}^{(2)} * x_2 + w_{13}^{(2)} * x_3 + b_1^{(2)}) \\
a_2^{(2)} &= f(w_{21}^{(2)} * x_1 + w_{22}^{(2)} * x_2 + w_{23}^{(2)} * x_3 + b_2^{(2)}) \\
a_3^{(2)} &= f(w_{31}^{(2)} * x_1 + w_{32}^{(2)} * x_2 + w_{33}^{(2)} * x_3 + b_3^{(2)}) \\
a_4^{(2)} &= f(w_{41}^{(2)} * x_1 + w_{42}^{(2)} * x_2 + w_{43}^{(2)} * x_3 + b_4^{(2)}) \\
a_1^{(3)} &= f(w_{11}^{(3)} * a_1^{(2)} + w_{12}^{(3)} * a_2^{(2)} + w_{13}^{(3)} * a_3^{(2)} + w_{14}^{(3)} * a_4^{(2)} + b_1^{(3)}) \\
a_2^{(3)} &= f(w_{21}^{(3)} * a_1^{(2)} + w_{22}^{(3)} * a_2^{(2)} + w_{23}^{(3)} * a_3^{(2)} + w_{24}^{(3)} * a_4^{(2)} + b_2^{(3)}) \\
h_{w,b}(x) &= a_1^{(4)} = f(w_{11}^{(4)} * a_1^{(3)} + w_{12}^{(4)} * a_2^{(3)} + b_1^{(4)})
\end{aligned}
\qquad \textbf{2.3.1}
$$

Η $h_{w,b}(x)$ είναι η συνάρτηση υπόθεσης, περισσότερα για την οποία και για το πως λειτουργεί, παρουσιάζονται στη συνέχεια. Από τα παραπάνω, φαίνεται ότι η ενεργοποίηση $a_j^{(l)}$, του j νευρώνα, του l επιπέδου, σχετίζεται με τις ενεργοποιήσεις στο (l-1) επίπεδο. Γενικεύοντας τις Εξισώσεις 2.3.1, ισχύει:

$$
a_j^l = f\left(\sum_k w_{jk}^l \, a_k^{l-1} + b_j^l\right) \qquad \textbf{2.3.2}
$$

όπου το συγκεκριμένο άθροισμα εμπεριέχει όλους τους k νευρώνες, στο (l-1) επίπεδο.

Η Εξίσωση 2.3.2 μπορεί να γραφτεί σε μορφή πινάκων. Αρχικά, ορίζεται ένας πίνακας βαρών w^l, για κάθε επίπεδο l. Τα στοιχεία του πίνακα βαρών w^l, είναι τα βάρη που συνδέονται στο l επίπεδο νευρώνων του δικτύου, δηλαδή, το στοιχείο στη j γραμμή και k στήλη, είναι το βάρος w^l_{jk}. Ομοίως, ορίζεται ένα διάνυσμα b^l για τις πολώσεις του κάθε επιπέδου l. Τα στοιχεία του διανύσματος b^l, είναι οι τιμές b^l_j· ένα στοιχείο για κάθε νευρώνα στο l επίπεδο. Τέλος, ορίζεται ένα διάνυσμα ενεργοποιήσεων a^l, το οποίο περιέχει της ενεργοποιήσεις a^l_j, του επιπέδου l. Η συγκεκριμένη εξίσωση, περιέχει όπως φαίνεται μία συνάρτηση, τη συνάρτηση ενεργοποίησης, η οποία διανυσματοποιείται. Τελικά, η Εξίσωση 2.3.2, γράφεται με τη διανυσματοποιημένη μορφή:

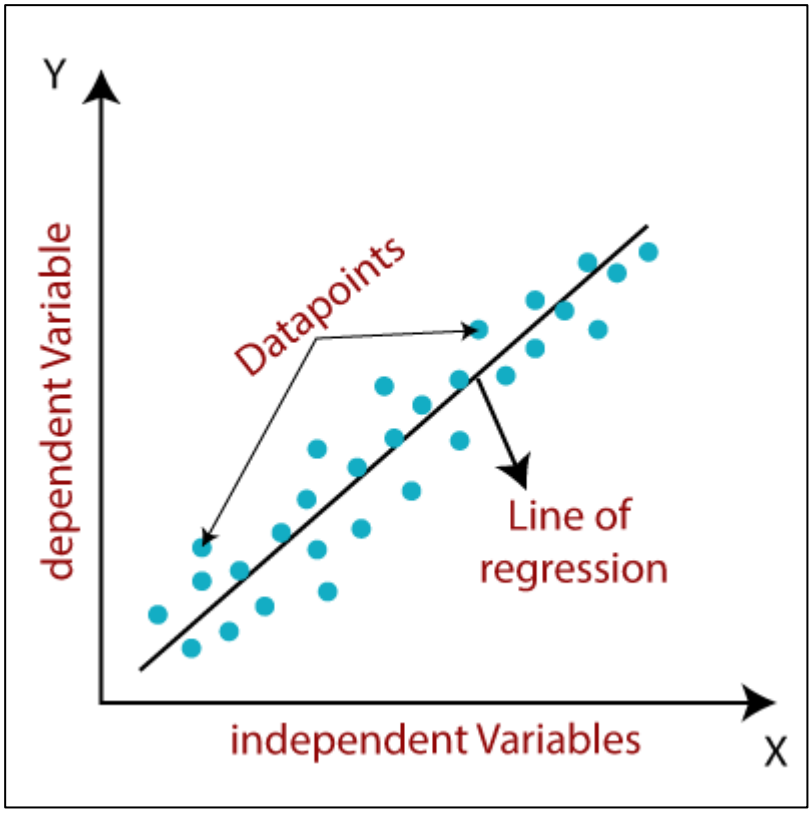
$$a^l = f(w^l * a^{l-1} + b^l) \qquad \textbf{2.3.3}$$

Η έκφραση της Εξίσωσης 2.3.3 προσφέρει μία σφαιρική εικόνα, η οποία είναι αρκετά πιο εύκολη και περιεκτική, από την οπτική νευρώνα με νευρώνα αφού περιέχει πολύ λιγότερους δείκτες, αλλά δεν υστερεί και από πλευράς σαφήνειας, αφού προσδιορίζει μία πλήρη και ολοκληρωμένη δομή σκέψης όσον αφορά το πως οι ενεργοποιήσεις σε ένα επίπεδο, σχετίζονται με τις ενεργοποιήσεις στο προηγούμενο επίπεδο· απλά εφαρμόζεται ο πίνακας βαρών στις ενεργοποιήσεις, μετά προστίθεται το διάνυσμα πολώσεων και τέλος εφαρμόζεται η συνάρτηση ενεργοποίησης.

Όταν χρησιμοποιείται η Εξίσωση 2.3.3 για τον υπολογισμό της a^l, υπολογίζεται η ενδιάμεση ποσότητα $z^l = w^l * a^{l-1} + b^l$ καθοδόν. Η συγκεκριμένη ποσότητα καλείται σταθμισμένη είσοδος στους νευρώνες στο επιπέδου l. Συμπεραίνεται, λοιπόν, ότι η Εξίσωση 2.3.3 μπορεί να γραφτεί περιέχοντας τον όρο της σταθμισμένης εισόδου, δηλαδή $a^l = f(z^l)$. Το z^l έχει συνιστώσες $z^l_j = \sum_k w^l_{jk} a^{l-1}_k + b^l_j$, δηλαδή το z^l_j είναι το σταθμισμένο άθροισμα που εισάγεται στη συνάρτηση ενεργοποίησης, για το νευρώνα j, στο επίπεδο l. (Nielsen 2019)

2.4 Γραμμική Παλινδρόμηση

Η γραμμική παλινδρόμηση, με τη μέθοδο που αναπτύσσεται στη συνέχεια, είναι ένα από τα κλασσικότερα παραδείγματα επιβλεπόμενης μάθησης και ουσιαστικά προσδιορίζει μια προσέγγιση για τη μοντελοποίηση της σχέσης μεταξύ μιας ή περισσοτέρων μεταβλητών/στοιχείων εισόδου και μιας μεταβλητής/στοιχείου εξόδου. Όπως φαίνεται και από το όνομα της (γραμμική), στη γραμμική παλινδρόμηση θεωρείται δεδομένο ότι η σχέση μεταξύ των μεταβλητών εισόδου και εξόδου περιγράφεται από μια γραμμική σχέση. Η συνάρτηση που μοντελοποιεί αυτή τη σχέση, είναι η συνάρτηση υπόθεσης, η οποία έχει κάποιες παραμέτρους και χρησιμοποιείται με σκοπό λαμβάνει τιμές δεδομένων εισόδου και να υπολογίζει εκτιμώμενες τιμές δεδομένων εξόδου.

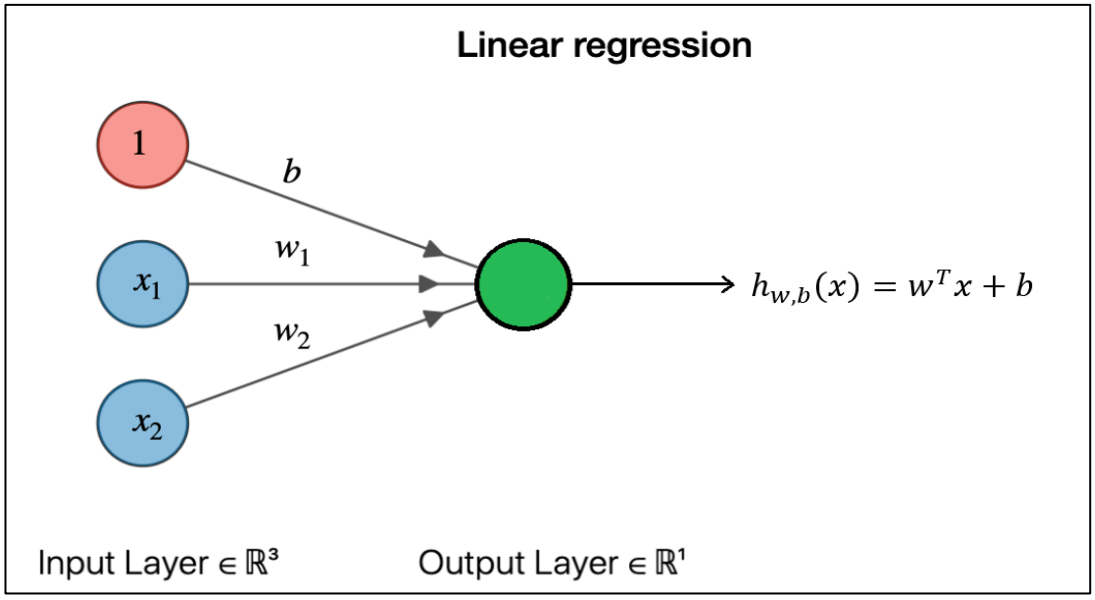


Εικόνα 4: Γραμμική Παλινδρόμηση

Έστω ότι διατίθενται κάποια δεδομένα, ένα σύνολο δειγμάτων, σε μορφή ζευγαριών, όπου κάθε ζευγάρι έχει μία ή περισσότερες μεταβλητές εισόδου και μία μεταβλητή εξόδου. Το κάθε ζευγάρι καλείται στιγμιότυπο του συνόλου των δειγμάτων. Τέτοιου είδους δεδομένα, παραδείγματος χάριν, μπορούν να είναι χαρακτηριστικά για κάποια ακίνητα, όπως το εμβαδόν σε τ.μ. και ο αριθμός δωματίων ως μεταβλητές-δεδομένα εισόδου και η τιμή κάθε ακινήτου ως μεταβλητή εξόδου. Στόχος της γραμμικής παλινδρόμησης είναι να βρεθούν εκείνες οι παράμετροι, που μπαίνοντας στη συνάρτηση υπόθεσης, η συνάρτηση αυτή θα αποτελεί τη βέλτιστη εκδοχή της γραμμικής σχέσης μεταξύ των δεδομένων εισόδου και εξόδου.

Οι μεταβλητές εισόδου, είναι ένα διάνυσμα $x \in R^n$, όπου κάθε στοιχείο x_j του x, αναπαριστά ένα χαρακτηριστικό-γνώρισμα. Οι μεταβλητές εισόδου, δηλαδή, ενός στιγμιότυπου του συνόλου δειγμάτων, αναπαρίστανται με ένα διάνυσμα x. Θεωρώντας ότι το σύνολο των δειγμάτων περιέχει αρκετά στιγμιότυπα, τα χαρακτηριστικά-γνωρίσματα του i-οστού στιγμιότυπου συμβολίζονται ως $x^{(i)}$. Η μεταβλητή εξόδου είναι για κάθε στιγμιότυπο μία μοναδική τιμή και συμβολίζεται ως $y^{(i)}$. Το σύνολο των δειγμάτων, δηλαδή, είναι $\{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$. Στόχος της γραμμικής παλινδρόμησης, είναι να βρεθούν εκείνες οι παράμετροι w, b, ώστε οι τιμές της συνάρτησης υπόθεσης $h_{w,b}(x^{(i)})$ να πλησιάζουν συνολικά όσο το δυνατόν περισσότερο, τις τιμές $y^{(i)}$. Αν βρεθεί μία τέτοια συνάρτηση, και βάσει του

παραπάνω παραδείγματος με τα χαρακτηριστικά και τις τιμές ακινήτων, θα μπορεί να τροφοδοτηθεί με χαρακτηριστικά για κάποιο νέο ακίνητο, και να προβλέψει την τιμή πώλησης του σχετικά καλά.



Εικόνα 5:Γραμμική Παλινδρόμηση με Τεχνητό Νευρωνικό Δίκτυο.

Για την εύρεση των βέλτιστων παραμέτρων w, b, χρησιμοποιείται ένα μέτρο ποσοτικοποίησης του πόσο καλά οι τιμές της συνάρτησης υπόθεσης $h_{w,b}(x^{(i)})$, παραμετροποιημένης με κάποια w, b, προσεγγίζουν τις μεταβλητές εξόδου $y^{(i)}$, για όλες τις μεταβλητές εισόδου $x^{(i)}$. Το μέτρο ποσοτικοποίησης, αυτό, δημιουργείται ορίζοντας μία "συνάρτηση κόστους" (cost function). Υπάρχουν αρκετές συναρτήσεις κόστους, η συνηθέστερη από αυτές για προβλήματα παλινδρόμησης-πρόβλεψης είναι το "Μέσο Τετραγωνικό Σφάλμα" ή "Τετραγωνικό Κόστος" ή "L2 απώλεια" (Mean Squared Error, Quadratic Cost, L2 Loss):

$$C(w,b) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_{w,b}(x^{(i)}) - y^{(i)}\right)^2 \qquad 2.4.1$$

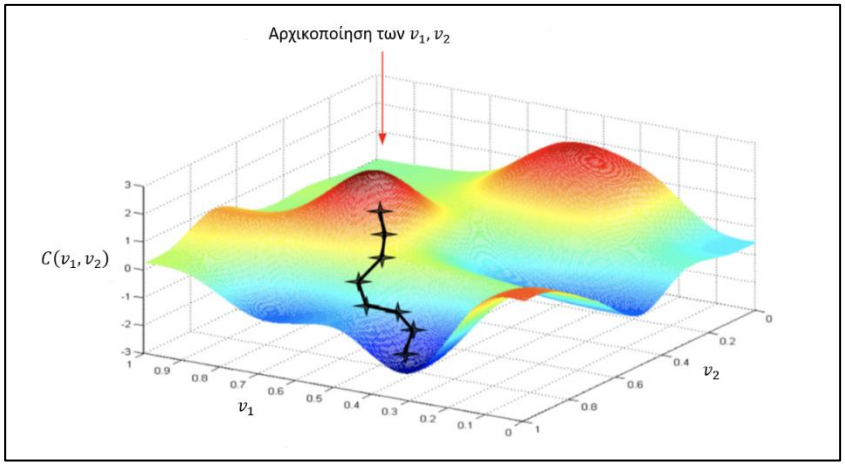
όπου w είναι το σύνολο όλων των βαρών, b είναι όλες οι πολώσεις, m είναι ο αριθμός των στιγμιότυπων του συνόλου δειγμάτων, $h_{w,b}(x^{(i)})$ είναι το διάνυσμα όλων των τιμών της συνάρτησης υπόθεσης, όταν οι $x^{(i)}$ είναι οι μεταβλητές εισόδου και $y^{(i)}$ είναι οι μεταβλητές εξόδου του συνόλου των δειγμάτων.

Εξετάζοντας τη μορφή της συνάρτησης 2.4.1, φαίνεται ότι το κόστος γίνεται μικρό, δηλαδή $C(w,b) \approx 0$, όταν η $h_{w,b}(x^{(i)})$ είναι περίπου ίδια με την $y^{(i)}$, για όλες τις

μεταβλητές εισόδου $x^{(i)}$. Φαίνεται, λοιπόν, ότι έχουν βρεθεί καλές παράμετροι για τη συνάρτηση υπόθεσης αν $C(w,b) \approx 0$. Έτσι, ορίζεται ένα κριτήριο για την εύρεση των βέλτιστων παραμέτρων w, b της συνάρτησης υπόθεσης, το οποίο είναι η ελαχιστοποίηση της συνάρτησης κόστους $C(w,b)$. (Fei-Fei Li 2020)

2.5 Ελαχιστοποίηση Συνάρτησης – Αλγόριθμος Gradient Descent

Ας υποθέσουμε ότι θέλουμε να ελαχιστοποιήσουμε κάποια συνάρτηση $C(v)$. Αυτή θα μπορούσε να είναι οποιαδήποτε συνάρτηση πραγματικών αριθμών, πολλών μεταβλητών $v = v_1, v_2, \ldots$. Για την ελαχιστοποίηση της $C(v)$, ας θεωρήσουμε ότι η C είναι συνάρτηση δύο μεταβλητών, των v_1, v_2. Στόχος, λοιπόν, είναι να βρεθεί το ολικό ελάχιστο της συγκεκριμένης συνάρτησης.



Εικόνα 6: Οπτικοποίηση ελαχιστοποίησης συνάρτησης δύο μεταβλητών.

Όπως φαίνεται στην Εικόνα 6, η συνάρτηση κόστους μπορεί να οπτικοποιηθεί ως μία επιφάνεια στον τρισδιάστατο χώρο. Πραγματοποιείται αρχικοποίηση των παραμέτρων v_1, v_2, ορίζοντας έτσι ουσιαστικά την αρχική θέση στον τρισδιάστατο αυτό χώρο. Η προσπάθεια εύρεσης του ολικού ελαχίστου της συνάρτησης, απαιτεί κινήσεις στον χώρο, τις οποίες υποδεικνύει ο υπολογισμός μερικών παραγώγων της C. Ο υπολογισμός των παραγώγων αυτών, στην κάθε θέση του τρισδιάστατου χώρου της C, ουσιαστικά δίνει μία εικόνα του τοπικού ανάγλυφου, ώστε να είναι δυνατόν η κίνηση να γίνει προς τη σωστή κατεύθυνση, αυτή δηλαδή του ολικού ελαχίστου.

Ας θεωρήσουμε ότι Δv_1 είναι μία κίνηση κατά ένα μικρό μήκος στην κατεύθυνση v_1 και Δv_2 είναι μία κίνηση κατά ένα μικρό μήκος στην κατεύθυνση v_2. Τότε η C αλλάζει ως εξής:

$$\Delta C \approx \frac{\partial C}{\partial v_1}\Delta v_1 + \frac{\partial C}{\partial v_2}\Delta v_2 \qquad 2.5.1$$

Πρέπει να επιλεγούν, λοιπόν, τα κατάλληλα $\Delta v_1, \Delta v_2$, ώστε η τιμή της ΔC να γίνει αρνητική. Ας ορίσουμε ως Δv το διάνυσμα των αλλαγών στο v, $\Delta v \equiv (\Delta v_1, \Delta v_2)^T$. Ακόμα, ας ορίσουμε την κλίση της C να είναι το διάνυσμα των μερικών παραγώγων $\left(\frac{\partial C}{\partial v_1}, \frac{\partial C}{\partial v_2}\right)^T$. Το διάνυσμα κλίσης (gradient vector) συμβολίζεται ως ∇C, δηλαδή:

$$\nabla C \equiv \left(\frac{\partial C}{\partial v_1}, \frac{\partial C}{\partial v_2}\right)^T \qquad \textbf{2.5.2}$$

Με αυτούς τους ορισμούς, η παράσταση 2.5.1 μπορεί να ξαναγραφτεί ως:

$$\Delta C \approx \nabla C * \Delta v \qquad \textbf{2.5.3}$$

Το διάνυσμα κλίσης ∇C συσχετίζει αλλαγές στο v, με αλλαγές στην C, όπως ακριβώς θα ήταν αναμενόμενο να συμβαίνει για κάτι που καλείται κλίση. Το πραγματικά χρηστικό, όμως, με την εξίσωση 2.5.3, είναι ότι καθιστά δυνατή την εξαγωγή συμπεράσματος σχετικά με την κατάλληλη επιλογή του Δv, ώστε να γίνει η ΔC αρνητική. Ας υποτεθεί, λοιπόν, ότι επιλέγεται:

$$\Delta v = -\eta \nabla C \qquad \textbf{2.5.4}$$

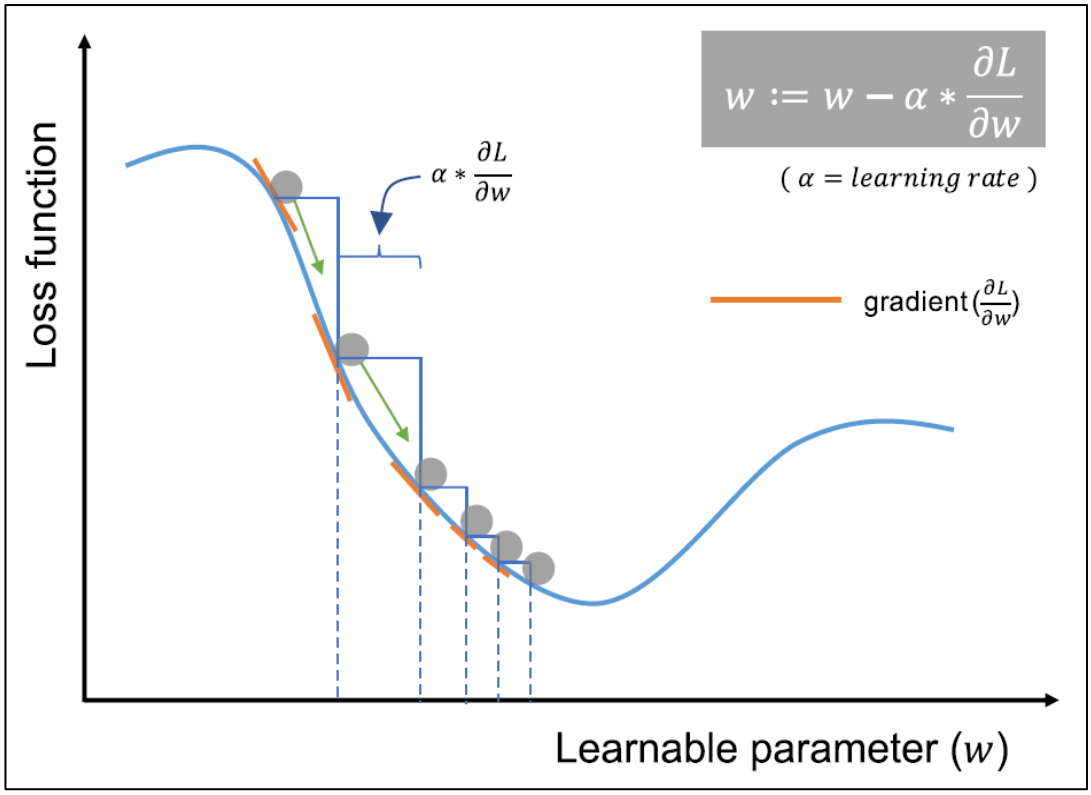
όπου η είναι μία μικρή, θετική παράμετρος η οποία καλείται βαθμός-ρυθμός μάθησης (learning rate). Τότε, από την εξίσωση 2.5.3 συμπεραίνεται ότι $\Delta C \approx -\eta \nabla C * \nabla C = -\eta ||\nabla C||^2$. Επειδή $||\nabla C||^2 \geq 0$, διασφαλίζεται ότι $\Delta C \leq 0$, δηλαδή η C, βάσει των παραπάνω, πάντα θα μειώνεται, ποτέ δεν θα αυξάνεται, αν το v μεταβάλλεται σύμφωνα με την εξίσωση 2.5.4 (εντός, φυσικά, των περιορισμών της προσέγγισης στην εξίσωση 2.5.3). Έτσι, λοιπόν, προσδιορίζεται ένας κανόνας ανανέωσης-κίνησης από την Εξίσωση 2.5.4, η οποία χρησιμοποιείται για να υπολογιστεί μία τιμή για το Δv και στη συνέχεια να μεταβληθεί το v (να υπάρξει μετακίνηση από την προηγούμενη θέση, κατά την οπτικοποίηση της Εικόνας 6) κατά αυτή την ποσότητα:

$$v \rightarrow v' = v - \eta \nabla C \qquad \textbf{2.5.5}$$

Χρησιμοποιώντας τον κανόνα ανανέωσης της Εξίσωσης 2.5.5 επαναληπτικά, η τιμή της C θα μειώνεται όλο και περισσότερο, έως ότου φτάσει σε ολικό ελάχιστο. Αυτό

είναι το επιθυμητό, αλλά μπορούν να εμφανιστούν προβλήματα κατά τη συγκεκριμένη διαδικασία. Συνοψίζοντας, ο τρόπος με τον οποίο λειτουργεί η παραπάνω διαδικασία, είναι με το να υπολογίζει επαναλαμβανόμενα το διάνυσμα κλίσης ∇C και στη συνέχεια να πραγματοποιεί μεταβολές προς την αντίθετη κατεύθυνση. Η διαδικασία αυτή καλείται αλγόριθμος gradient descent. (Nielsen 2019)

Σχετικά με το παράδειγμα της Εικόνας 5, ο αλγόριθμος gradient descent λειτουργεί ως πυρήνας της μάθησης των βέλτιστων παραμέτρων w, b. Όπως ειπώθηκε και παραπάνω, πραγματοποιείται μία επαναληπτική διαδικασία, με σκοπό τη μείωση και τελικά την ελαχιστοποίηση της συνάρτησης κόστους $C(w, b)$. Κατά τη διαδικασία αυτή, η κατεύθυνση προς την οποία μειώνεται και τελικά ελαχιστοποιείται η τιμή του κόστους C, δίνεται από την παράγωγο της συνάρτησης κόστους, συναρτήσει των παραμέτρων w_1, w_2 και b. Ουσιαστικά, εντός του αλγόριθμου πραγματοποιείται η παρακάτω διαδικασία. Αρχικά δίνονται κάποιες τιμές εκκίνησης-αρχικοποίησης στις παραμέτρους w_1, w_2 και b. Δημιουργείται έτσι η συνάρτηση υπόθεσης $h_{w,b}(x) = w_1 x_1 + w_2 x_2 + b$ και υπολογίζεται το άθροισμα των τετραγώνων των διαφορών των μεταβλητών εξόδου και των τιμών που υπολογίζονται από τη συνάρτηση υπόθεσης βάσει των μεταβλητών εισόδου, μια ποσότητα δηλαδή ίση με $\sum_{i=1}^{n}\left(h_{w,b}\left(x^{(i)}\right) - y^{(i)}\right)^2$. Αυτό συμβαίνει σε μια διαδικασία, η οποία λαμβάνει υπόψη τα δεδομένα συνολικά, κάθε στιγμιότυπο τους δηλαδή, και για να το πραγματοποιήσει αυτό, επαναλαμβάνεται τόσες φορές όσα και τα στιγμιότυπα του συνολικού δείγματος. Στη συνέχεια υπολογίζεται για κάθε παράμετρο (w_1, w_2 και b), ποσότητα ίση με την πρώτη παράγωγο της παραμέτρου ως προς το κόστος C (δηλαδή οι ποσότητες $\frac{\partial C}{\partial w_1}, \frac{\partial C}{\partial w_2}, \frac{\partial C}{\partial b}$). Έπειτα, αφαιρείται από την κάθε παράμετρο w_1, w_2 και b, η ποσότητα που υπολογίστηκε πριν, πολλαπλασιασμένη με την τιμή του ρυθμού μάθησης η και διαιρεμένη με τον αριθμό των στιγμιότυπων. Έτσι, υπολογίζονται οι νέες παράμετροι w_1, w_2 και b, εισέρχονται ξανά στην ίδια επαναληπτική διαδικασία και με βάση αυτές υπολογίζονται οι νέες ποσότητες (πρώτες παράγωγοι των παραμέτρων ως προς το νέο κόστος C) που αφαιρούνται ξανά από τις παραμέτρους. Η διαδικασία αυτή επαναλαμβάνεται είτε για κάποιο προκαθορισμένο αριθμό επαναλήψεων, είτε μέχρι η τιμή της συνάρτησης κόστους να φτάσει σε κάποια προκαθορισμένη ελάχιστη τιμή. Με αυτή τη λογική ανανέωσης των παραμέτρων και των κοστών, καταλήγουμε στην ελαχιστοποίηση του κόστους, που οδηγεί στην εύρεση των βέλτιστων, βάσει του συνόλου δειγμάτων, παραμέτρων w_1, w_2 και b.



Εικόνα 7: Η κλίση της συνάρτησης κόστους (L εδώ), παρέχει την κατεύθυνση στην οποία η συνάρτηση έχει τον πιο απότομο ρυθμό αύξησης, και όλες οι παράμετροι ανανεώνονται στην αντίθετη κατεύθυνση αυτής της κλίσης, με ένα μέγεθος βήματος που ορίζεται από το βαθμό μάθησης. (Rikiya Yamashita 2018)

Συνοψίζοντας, οι ανανεώσεις των βαρών w_1, w_2 και της πόλωσης b, βάσει του αλγόριθμου gradient descent, στην περίπτωση του παραδείγματος της Εικόνας 5, είναι:

$$w_1 \to w_1' = w_1 - \eta \frac{\partial C}{\partial w_1}$$
$$w_2 \to w_2' = w_2 - \eta \frac{\partial C}{\partial w_2} \qquad \textbf{2.5.6}$$
$$b \to b' = b - \eta \frac{\partial C}{\partial b}$$

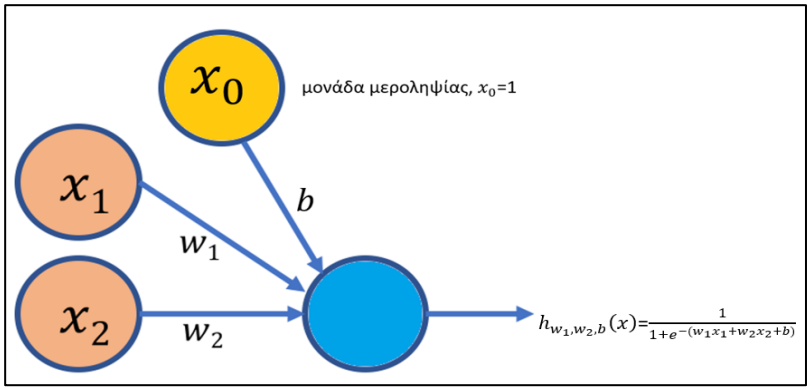
Για να λειτουργήσει ο gradient descent σωστά, χρειάζεται να επιλεγεί ένας βαθμός μάθησης η, τόσο μικρός, ώστε η Εξίσωση 2.5.3, να είναι μία καλή προσέγγιση. Αν αυτό δε συμβεί, είναι πιθανό να προκύψει $\Delta C > 0$, κάτι το οποίο προφανώς θα δημιουργούσε πρόβλημα στην διαδικασία. Ταυτόχρονα, η υπέρ-παράμετρος η δεν πρέπει να είναι υπερβολικά μικρή, γιατί αυτό θα έχει σαν αποτέλεσμα οι μεταβολές Δv να είναι πολύ μικρές και έτσι ο αλγόριθμος να γίνει πολύ αργός.

2.6 Λογιστική Παλινδρόμηση

Η λογιστική παλινδρόμηση (logistic regression) είναι ένα μοντέλο ταξινόμησης δεδομένων σε μία ή και περισσότερες κατηγορίες, το οποίο εντάσσεται στη λογική της επιβλεπόμενης μάθησης. Στη συνέχεια, παρουσιάζεται η δυαδική λογιστική παλινδρόμηση (binary logistic regression). Η ταξινόμηση αυτή ονομάζεται δυαδική διότι τα δεδομένα του δείγματος, μετά από μια διαδικασία, κατατάσσονται μεταξύ δύο κατηγοριών.

Η λογιστική παλινδρόμηση μπορεί να υλοποιηθεί από ένα σιγμοειδή νευρώνα. Αυτό πραγματοποιείται με μια συνάρτηση υπόθεσης που καλείται σιγμοειδής συνάρτηση, η οποία δύναται να λαμβάνει τιμές δεδομένων εισόδου και να υπολογίζει κάποιες εκτιμώμενες τιμές εξόδου, οι οποίες ερμηνεύονται ως πιθανότητες να ανήκουν τα δεδομένα εισόδου ή όχι σε μια από τις δύο κατηγορίες (0 ή 1).

Ο λογιστικός νευρώνας λαμβάνει τις τιμές των μεταβλητών εισόδου $x_0, x_1, \dots, x_n$ (όπου x_0=1 γιατί αντιστοιχεί στην παράμετρο της πόλωσης b) και τις παραμέτρους (βάρος για κάθε μεταβλητή εισόδου συν μία συνολική πόλωση) $b, w_1, \dots, w_n$. Εντός του νευρώνα, υπολογίζονται δύο ποσότητες, μία ενδιάμεση θα μπορούσε κανείς να πει και αυτή που τελικά εξάγεται. Η ενδιάμεση ποσότητα είναι η z=$\sum_{i=1}^{n} w_i x_i + b$, η οποία καλείται και σταθμισμένη είσοδος. Η δεύτερη ποσότητα που υπολογίζεται και τελικά εξάγει ο λογιστικός νευρώνας, είναι η $h(z) = \sigma(z) = \frac{1}{1+e^{-z}}$, όπου σ καλείται η σιγμοειδής συνάρτηση. Η συνάρτηση υπόθεσης, λοιπόν, στη συγκεκριμένη περίπτωση είναι η σιγμοειδής συνάρτηση.

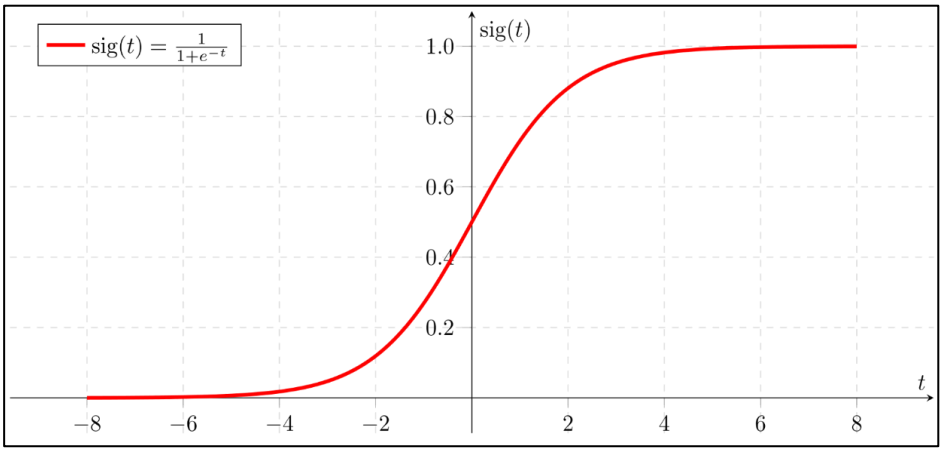


Εικόνα 8: Λογιστικός τεχνητός νευρώνας.

Ας υποτεθεί, λοιπόν, ότι έχουμε ένα σετ δεδομένων-σύνολο δειγμάτων $\{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$, με κάποιο αριθμό στιγμιότυπων m, όπου κάθε στιγμιότυπο έχει κάποιες τιμές μεταβλητών εισόδου x_1, …, και μία τιμή μεταβλητής εξόδου $y \in \{0,1\}$. Η τιμή της κάθε μεταβλητής εξόδου μπορεί να είναι είτε 0, είτε 1, ανάλογα την κατηγορία στην οποία ανήκει. Τα δεδομένα αυτά, χρησιμοποιούνται

σαν δεδομένα εκμάθησης ενός αλγορίθμου, με σκοπό να υπολογιστούν οι βέλτιστες παράμετροι b, $w_1, \ldots$, που θα ενταχθούν στη συνάρτηση υπόθεσης-σιγμοειδή συνάρτηση με αποτέλεσμα να υπολογίζονται τιμές ερμηνεύσιμες ως πιθανότητες για νέα δεδομένα εισόδου να ανήκουν σε μια από τις δύο κατηγορίες (0,1). Η συνάρτηση υπόθεσης υπολογίζει κάποιες τιμές, οι οποίες ανάλογα αν είναι μεγαλύτερες ή μικρότερες από ένα όριο που τίθεται (πχ 0,5), αποφασίζεται το σε ποια από τις δύο κατηγορίες (0,1) κατατάσσονται τα δεδομένα.

Η συνάρτηση υπόθεσης $h(z) = \sigma(z) = \frac{1}{1+e^{-z}}$ είναι αποτέλεσμα την τοποθέτησης της εξίσωσης της γραμμικής παλινδρόμησης $z=\sum_{i=1}^{n} w_i x_i + b$ στη σιγμοειδή συνάρτηση, κάτι το οποίο γεωμετρικά συνιστά μια απλή γραμμική μετατροπή. Η σιγμοειδής, έχει μια πολύ χρήσιμη ιδιότητα σχετικά με τον υπολογισμό ερμηνεύσιμων ως πιθανοτήτων τιμών, η οποία είναι ότι το πεδίο τιμών της είναι από 0 έως 1. Η ιδιότητα αυτή είναι πολύ χρήσιμη διότι εξαιτίας της μορφής της συγκεκριμένης συνάρτησης, είναι δυνατό για μεγάλα εύρη τιμών z,να δίνει τιμές από 0 έως 1.Γραφικά, βασικό της χαρακτηριστικό είναι ότι έχει ένα διάστημα στο οποίο αναπτύσσεται εκθετικά και με σταδιακή επιβράδυνση του ρυθμού της ανάπτυξης αυτής και στη συνέχεια περατώνεται συμπωτικά και παραλληλίζεται με τον οριζόντιο άξονα. Σχετικά με το κομμάτι z (εξίσωσης της γραμμικής παλινδρόμησης) της συνάρτησης υπόθεσης, αυτό είναι που ορίζει μια διαχωριστική γραμμή μεταξύ των δεδομένων και το ποια μορφή θα έχει αυτή η διαχωριστική γραμμή(σχήμα, αριθμός διαστάσεων), οφείλεται στη μορφή του z αυτού (γραμμικό, μη γραμμικό, αριθμός μεταβλητών εισόδου).



Εικόνα 9: Σιγμοειδής συνάρτηση.

Για να εκπαιδευτεί ο αλγόριθμος και να μάθει τις βέλτιστες παραμέτρους για τη συνάρτηση υπόθεσης, χρειάζεται κάποιο κριτήριο. Το κριτήριο αυτό, είναι η ελαχιστοποίηση της τιμής μίας συνάρτησης κόστους. Η συνάρτηση κόστους που χρησιμοποιείται σε τέτοιες περιπτώσεις, είναι η *cross-entropy*, η οποία είναι της παρακάτω μορφής:

$$C(w,b) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(h_{w,b}\left(x^{(i)}\right)\right) + \left(1-y^{(i)}\right)log\left(1-h_{w,b}\left(x^{(i)}\right)\right)\right] \quad \textbf{2.6.1}$$

όπου m είναι ο αριθμός των στιγμιότυπων του συνόλου των δεδομένων εκπαίδευσης, $x^{(i)}$ είναι το σύνολο (διάνυσμα) των τιμών όλων των μεταβλητών εισόδου του κάθε στιγμιότυπου και $y^{(i)}$ είναι η (γνωστή) τιμή της μεταβλητής εξόδου του κάθε στιγμιότυπου.

Δύο ιδιότητες καθιστούν λογική τη χρήση της συνάρτησης cross-entropy, ως συνάρτηση κόστους. Πρώτον, είναι μη-αρνητική, δηλαδή $C > 0$, κάτι το οποίο φαίνεται από το ότι όλοι οι διακριτοί όροι στο άθροισμα της εξίσωσης 2.6.1 είναι αρνητικοί, αφού και οι δύο λογάριθμοι είναι αριθμών μεταξύ 0 και 1 και υπάρχει αρνητικό πρόσημο στην αρχή, εκτός του αθροίσματος. Δεύτερον, αν η έξοδος του νευρώνα είναι κοντά στην επιθυμητή, γνωστή από τα δεδομένα, τιμή της μεταβλητής εξόδου, τότε η τιμή της cross-entropy θα είναι κοντά στο μηδέν. Για να γίνει αυτό αντιληπτό, ας υποτεθεί ότι $y^{(i)} = 0$ και $h_{w,b}\left(x^{(i)}\right) \approx 0$, για ένα σύνολο τιμών των μεταβλητών εισόδου $x^{(i)}$. Αυτή είναι μία περίπτωση όπου ο λογιστικός νευρώνας λειτουργεί καλά, για αυτές τις τιμές εισόδου. Φαίνεται ότι ο πρώτος όρος στην Εξίσωση 2.6.1 του κόστους, εξαφανίζεται, αφού $y^{(i)} = 0$, ενόσω ο δεύτερος όρος είναι - $log\left(1-h_{w,b}\left(x^{(i)}\right)\right) \approx 0$. Κάτι παρόμοιο συμβαίνει και στην περίπτωση που $y^{(i)} = 1$ και $h_{w,b}\left(x^{(i)}\right) \approx 1$. Έτσι, φαίνεται ότι σε περιπτώσεις όπου $h_{w,b}\left(x^{(i)}\right)$ και $y^{(i)}$ είναι αρκετά κοντά, η τιμή του cross-entropy κόστους είναι μικρή.

Εκτός των παραπάνω δύο ιδιοτήτων, οι οποίες είναι πραγματικά πολύ χρηστικές για μία συνάρτηση κόστους, η συνάρτηση κόστους cross-entropy έχει άλλο ένα πλεονέκτημα, το οποίο καθιστά ταχύτερη τη διαδικασία μάθησης των βέλτιστων παραμέτρων της σιγμοειδούς συνάρτησης υπόθεσης. Για την εύρεση των βέλτιστων αυτών παραμέτρων, χρησιμοποιείται η επαναληπτική διαδικασία του αλγόριθμου gradient descent, όπου σε κάθε επανάληψη υπολογίζεται μία διόρθωση και ανανεώνεται κάθε παράμετρος. Η διόρθωση της κάθε παραμέτρου προσδιορίζεται από τη μερική παράγωγο της συνάρτησης κόστους ως προς την κάθε παράμετρο. Οι ανανεώσεις των παραμέτρων, στο παράδειγμα λογιστικού νευρώνα της Εικόνας 8, γίνονται βάσει των παρακάτω κανόνων:

$$w_1 \to w_1' = w_1 - \eta\frac{\partial C}{\partial w_1} = w_1 - \eta\frac{1}{m}\sum_{i=1}^{m}\left(h_{w,b}\left(x^{(i)}\right) - y^{(i)}\right)x_1^{(i)} \quad \textbf{2.6.2}$$

$$w_2 \to w_2' = w_2 - \eta \frac{\partial C}{\partial w_2} = w_2 - \eta \frac{1}{m} \sum_{i=1}^{m} \left(h_{w,b}\left(x^{(i)}\right) - y^{(i)}\right) x_2^{(i)}$$

$$b \to b' = b - \eta \frac{\partial C}{\partial b} = b - \eta \frac{1}{m} \sum_{i=1}^{m} \left(h_{w,b}\left(x^{(i)}\right) - y^{(i)}\right) x_o^{(i)}$$

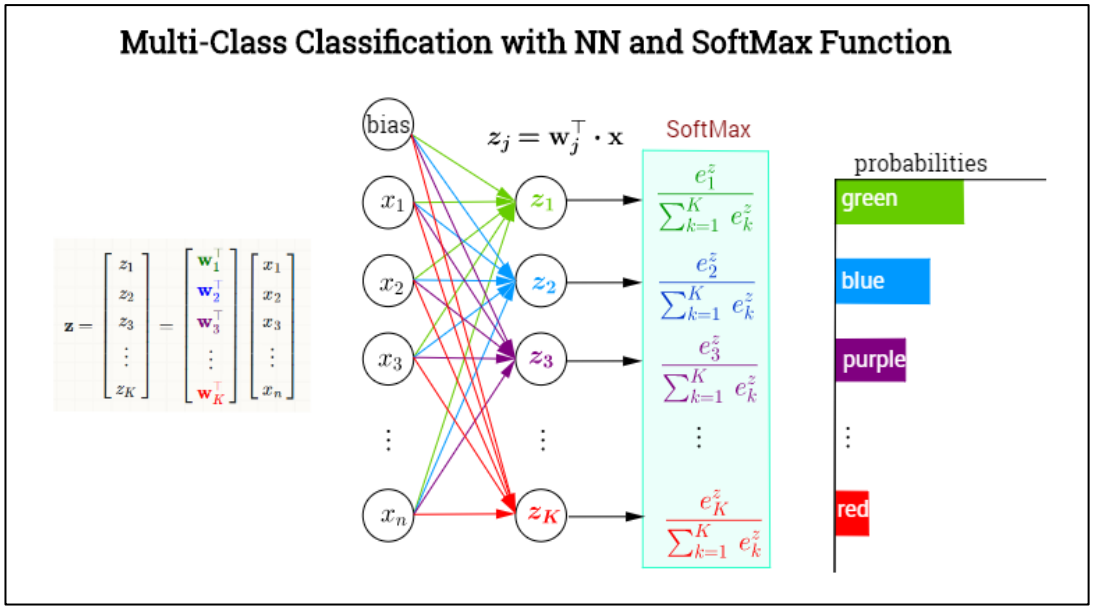
$$= b - \eta \frac{1}{m} \sum_{i=1}^{m} \left(h_{w,b}\left(x^{(i)}\right) - y^{(i)}\right)$$

Από τις Εξισώσεις 2.6.2 φαίνεται ότι το επίπεδο μάθησης των παραμέτρων της σιγμοειδούς συνάρτησης υπόθεσης ρυθμίζεται από την τιμή της αφαίρεσης $\left(h_{w,b}\left(x^{(i)}\right) - y^{(i)}\right)$, η οποία ουσιαστικά είναι το σφάλμα της εκτίμησης του δικτύου και της πραγματικής, γνωστής από τα δεδομένα, τιμής της μεταβλητής εξόδου. Η συγκεκριμένη ιδιότητα αποσοβεί το πρόβλημα της επιβράδυνσης της μάθησης των παραμέτρων, κάτι το οποίο συμβαίνει αρκετές φορές χρησιμοποιώντας της συνάρτηση κόστους του μέσου τετραγωνικού σφάλματος. (Andrew Ng n.d.)

2.7 Softmax Παλινδρόμηση

Η softmax παλινδρόμηση (ή αλλιώς πολυωνυμική λογιστική παλινδρόμηση) είναι μία γενίκευση της λογιστικής παλινδρόμησης και χρησιμοποιείται σε περιπτώσεις όπου τα δεδομένα πρόκειται να ταξινομηθούν σε πολλαπλές κατηγορίες. Στη (δυαδική) λογιστική παλινδρόμηση που παρουσιάστηκε προηγούμενα, οι τιμή της μεταβλητής εξόδου ήταν είτε 0, είτε 1, δηλαδή $y^{(i)} \in \{0,1\}$. Στην περίπτωση της Softmax παλινδρόμησης, για την κάθε μεταβλητή εξόδου ισχύει $y^{(i)} \in \{1,2,\dots,K\}$, όπου K είναι ο αριθμός των διαφορετικών κατηγοριών-τάξεων στις οποίες πρόκειται να ταξινομηθεί το σύνολο των δεδομένων.

Για την πραγματοποίηση softmax παλινδρόμησης, αντί της σιγμοειδούς συνάρτησης, ως συνάρτησης υπόθεσης, χρησιμοποιείται η συνάρτηση softmax. Στην περίπτωση ενός τεχνητού νευρωνικού δικτύου ταξινόμησης, που πραγματοποιεί παλινδρόμηση softmax, συνήθως η συνάρτηση softmax χρησιμοποιείται ως συνάρτηση ενεργοποίησης του κάθε νευρώνα του επιπέδου εξόδου, ορίζοντας έτσι το softmax επίπεδο εξόδου, το οποίο στις πλείστες των περιπτώσεων είναι ένα πλήρως συνδεδεμένο επίπεδο. Ανάλογα του σε πόσες κατηγορίες θέλουμε να ταξινομηθούν τα δεδομένα, τόσοι είναι και οι νευρώνες του softmax επιπέδου εξόδου.



Εικόνα 10: Softmax παλινδρόμηση.

Ας υποθέσουμε ότι έχουμε ένα σύνολο δεδομένων-δειγμάτων εκπαίδευσης, με m πλήθος στιγμιότυπων $\{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$ όπου $y^{(i)} \in \{1,2, \dots, K\}$ είναι η τιμής της μεταβλητής εξόδου του στιγμιότυπου i και $x^{(i)} = [x_0^{(i)}; x_1^{(i)}; \dots]$ το διάνυσμα στήλης των τιμών των μεταβλητών εισόδου του στιγμιότυπου i (πάντα $x_0^{(i)} = 1$ γιατί αντιστοιχεί στην παράμετρο πόλωσης). Σε σχέση με τις παραμέτρους (βάρη και πόλωση) της συνάρτησης υπόθεσης, ας δούμε ένα διαφορετικό συμβολισμό από ότι προηγούμενα. Το διάνυσμα στήλης των παραμέτρων είναι $W^{(k)} = [W_0^{(k)}; W_1^{(k)}; \dots], k = 1, \dots, K$ και $W_0^{(k)} = b$. Το διάνυσμα στήλης $W^{(k)}$ είναι οι παράμετροι που μεταφέρονται από τους νευρώνες του αμέσως προηγούμενου, του softmax επιπέδου, προς το softmax επίπεδο· δηλαδή $W^{(1)}$ είναι οι παράμετροι που μεταφέρονται στον πρώτο νευρώνα του softmax επιπέδου, από τους νευρώνες του αμέσως προηγούμενου επιπέδου με το οποίο συνδέεται. Εντός του κάθε νευρώνα του softmax επιπέδου, αρχικά υπολογίζεται η σταθμισμένη είσοδος:

$$z^{(k)} = W^{(k)^T} x^{(i)} \qquad 2.7.1$$

Στην περίπτωση όπου το αμέσως προηγούμενο επίπεδο του softmax επιπέδου δεν είναι το επίπεδο εισόδου, αλλά ένα ενδιάμεσο κρυφό επίπεδο, η οποία είναι και η συνηθέστερη περίπτωση, στη θέση του $x^{(i)}$ της παραπάνω εξίσωσης, θα μπορούσαν να είναι οι τιμές των ενεργοποιήσεων α (δείτε Κεφάλαιο 2.3) των νευρώνων, οι οποίοι συνδέονται με το νευρώνα στον οποίο υπολογίζεται η σταθμισμένη είσοδος.

Αφού έχει υπολογιστεί η σταθμισμένη είσοδος του κάθε νευρώνα, έπειτα, χρησιμοποιείται η συνάρτηση softmax, και εντός του κάθε νευρώνα υπολογίζεται μία

τιμή η οποία ερμηνεύεται ως πιθανότητα, οι τιμές εισόδου $x^{(i)}$ να ανήκουν στην κατηγορία-τάξη k που αντιστοιχεί στο νευρώνα αυτό, δηλαδή $P(y^{(i)} = k|x^{(i)})$:

$$h_{W^{(k)}}(x^{(i)}) = \frac{e^{z^{(k)}}}{\sum_{j=1}^{K} e^{z^{(j)}}} \qquad \textbf{2.7.2}$$

Βλέποντας το softmax επίπεδο συνολικά και όχι ως ξεχωριστούς νευρώνες, η συνάρτηση υπόθεσης για το συνολικό δίκτυο, είναι της μορφής:

$$h_W(x^{(i)}) = \begin{bmatrix} P(y^{(i)} = 1|x^{(i)}, W) \\ P(y^{(i)} = 2|x^{(i)}, W) \\ \vdots \\ P(y^{(i)} = K|x^{(i)}, W) \end{bmatrix} = \frac{1}{\sum_{j=1}^{K} e^{z^{(j)}}} \begin{bmatrix} e^{z^{(1)}} \\ e^{z^{(2)}} \\ \vdots \\ e^{z^{(K)}} \end{bmatrix} \qquad \textbf{2.7.3}$$

Το W, στην $h_W(x^{(i)})$, υποδηλώνει όλες τις παραμέτρους (βάρη w και πολώσεις b) που μεταφέρονται στους νευρώνες του επιπέδου softmax. Ο όρος $\frac{1}{\sum_{j=1}^{K} e^{z^{(j)}}}$ κανονικοποιεί την κατανομή, ώστε το άθροισμα των τιμών των πιθανοτήτων, να ισούται με ένα. Ας υποθέσουμε ότι έχουμε ένα δίκτυο με τέσσερις softmax νευρώνες εξόδου, ο καθένας εκ των οποίων λαμβάνει μία σταθμισμένη είσοδο $z^{(k)}$. Στην περίπτωση όπου αυξηθεί, παραδείγματος χάριν, η σταθμισμένη είσοδος $z^{(4)}$, θα αυξηθεί αντίστοιχα η τιμή της συνάρτησης softmax-της ενεργοποίησης του αντίστοιχου νευρώνα, h_{W^4}, και θα μειωθούν ταυτόχρονα οι ενεργοποιήσεις εξόδου των άλλων τριών softmax νευρώνων. Παρομοίως, αν μειωθεί η σταθμισμένη είσοδος $z^{(4)}$, η αντίστοιχη ενεργοποίηση εξόδου-τιμή της συνάρτησης softmax του νευρώνα, h_{W^4}, θα μειωθεί και οι ενεργοποιήσεις εξόδου των άλλων τριών νευρώνων θα αυξηθούν.

Στη συνέχεια περιγράφεται η συνάρτηση κόστους που χρησιμοποιείται στη softmax παλινδρόμηση και με την ελαχιστοποίηση της οποίας υπολογίζονται οι βέλτιστες παράμετροι της συνάρτησης υπόθεσης. Στην Εξίσωση 2.7.4, $1\{\cdot\}$ είναι η "δείκτρια συνάρτηση", έτσι ώστε 1{αληθής δήλωση}=1, και 1{ψευδής δήλωση}=0. Στην Εξίσωση 2.7.4, η "δείκτρια συνάρτηση" χρησιμοποιείται ώστε να συμπεριλαμβάνεται στο κόστος μόνο η έξοδος του ταξινομητή που αντιστοιχεί στη σωστή (γνωστή) τιμή της $y^{(i)}$.

$$C(W) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{k=1}^{K} 1\{y^{(i)} = k\} log \frac{e^{z^{(k)}}}{\sum_{j=1}^{K} e^{z^{(j)}}}\right] \qquad \textbf{2.7.4}$$

Για να ελαχιστοποιηθεί η συνάρτηση κόστους $C(W)$, χρησιμοποιείται ένας επαναληπτικός αλγόριθμος βελτιστοποίησης, όπως ο gradient descent. Το διάνυσμα κλίσης της συνάρτησης κόστους συναρτήσει του διανύσματος παραμέτρων (βαρών και πόλωσης) $W^{(k)}$, είναι:

$$\nabla_{W^{(k)}} C(W) = -\frac{1}{m}\sum_{i=1}^{m}\left[x^{(i)}\left(1\{y^{(i)} = k\} - P(y^{(i)} = k|x^{(i)}, W)\right)\right] \qquad \textbf{2.7.5}$$

Το $\nabla_{W^{(k)}} C(W)$ είναι ένα διάνυσμα, του οποίου το κάθε στοιχείο είναι η μερική παράγωγος της συνάρτησης κόστους $C(W)$ ως προς το αντίστοιχο στοιχείο-παράμετρο του διανύσματος $W^{(k)}$. Η συνάρτηση 2.7.5 υπολογίζει τις κλίσεις για κάποια συγκεκριμένη κατηγορία-τάξη k. Η έκφραση εντός της παρένθεσης παίρνει μία τιμή μεταξύ 0 και 1. (Andrew Ng n.d.)

2.8 Οπισθοδιάδοση Σφάλματος – Backpropagation

Όταν χρησιμοποιείται ένα τεχνητό νευρωνικό δίκτυο εμπρόσθιας τροφοδότησης, πολλαπλών επιπέδων, το οποίο λαμβάνει κάποιες τιμές εισόδου και υπολογίζει κάποια έξοδο, η πληροφορία ρέει προς τα μπροστά εντός του δικτύου. Αυτή η διαδικασία ονομάζεται προς τα εμπρός διάδοση (forward propagation). Όπως περιεγράφηκε στο Κεφάλαιο 2.3, το επίπεδο εισόδου (που κάποιες φορές δε λαμβάνεται υπόψη ως επίπεδο) περιέχει τις τιμές των μεταβλητών εισόδου. Ο κάθε νευρώνας, του κάθε ενδιάμεσου, κρυφού επιπέδου λαμβάνει παραμέτρους (βάρη και πόλωση) και ενεργοποιήσεις-τιμές συναρτήσεων από τους νευρώνες του προηγούμενου επιπέδου με τους οποίους συνδέεται και παράγει μία νέα ενεργοποίηση-υπολογίζει μία συνάρτηση. Τέλος, το επίπεδο εξόδου παράγει μία τιμή, η οποία συγκρίνεται με την πραγματική-γνωστή από τα δεδομένα τιμή της μεταβλητής εξόδου (επιβλεπόμενη μάθηση) και υπολογίζεται ένα κόστος. Βάσει της μερικής παραγώγου της συνάρτησης κόστους ως προς κάθε παράμετρο του δικτύου, διορθώνονται οι παράμετροι αυτές. Γίνεται έτσι μία επαναληπτική διαδικασία, εκπαίδευση ενός αλγόριθμου, όπου κάθε φορά διορθώνονται οι παράμετροι του δικτύου. Ο αλγόριθμος backpropagation (David E. Rumelhart 1986) επιτρέπει στην πληροφορία του κόστους να διαδοθεί προς τα πίσω εντός του δικτύου, ώστε να υπολογιστούν οι μερικές παράγωγοι της συνάρτησης κόστους ως προς κάθε βάρος και πόλωση του. Ο υπολογισμός των μερικών παραγώγων αναλυτικά είναι ευθύς, αλλά μπορεί να είναι υπολογιστικά πολύ κοστοβόρος. Αντίθετα, ο αλγόριθμος backpropagation πραγματοποιεί αυτούς τους υπολογισμούς, με μία απλή και πολύ λιγότερο υπολογιστικά κοστοβόρα διεργασία.

Ο όρος backpropagation συχνά παρερμηνεύεται ωσάν να είναι ο συνολικός αλγόριθμος μάθησης για νευρωνικά δίκτυα πολλών επιπέδων. Στην πραγματικότητα,

ο backpropagation αναφέρεται μόνο στη μέθοδο για τον υπολογισμό των κλίσεων, ενόσω κάποιος άλλος αλγόριθμος, προφανώς όπως ο gradient descent, χρησιμοποιείται για τη μάθηση, εκμεταλλευόμενος αυτές τις κλίσεις. Επιπρόσθετα, ο αλγόριθμος backpropagation συχνά παρεξηγείται ωσάν να λειτουργεί μόνο σε νευρωνικά δίκτυα πολλών επιπέδων. Στην πραγματικότητα, μπορεί να υπολογίσει παραγώγους οποιασδήποτε συνάρτησης.

Για την περιγραφή του αλγόριθμου οπισθοδιάδοσης σφάλματος backpropagation, που πραγματοποιείται στη συνέχεια, χρειάζεται να χρησιμοποιηθεί ένα παράδειγμα συνάρτησης κόστους. Η συνάρτηση κόστους, λοιπόν, η οποία χρησιμοποιείται, είναι το μέσο τετραγωνικό σφάλμα (τετραγωνική συνάρτηση κόστους), της μορφής $C(w,b)=\frac{1}{2m}\sum_{i=1}^{m}\left(h_{w,b}\left(x^{(i)}\right)-y^{(i)}\right)^2=\frac{1}{2m}\sum_{i=1}^{m}\left(\alpha^L\left(x^{(i)}\right)-y^{(i)}\right)^2$, όπου m είναι ο συνολικός αριθμός των στιγμιότυπων εκπαίδευσης, το άθροισμα εμπεριέχει όλα τα διαφορετικά στιγμιότυπα $\left\{\left(x^{(1)},y^{(1)}\right),\ldots,\left(x^{(m)},y^{(m)}\right)\right\}$, $y^{(i)}$ είναι η επιθυμητή-γνωστή τιμή της μεταβλητής εξόδου του κάθε στιγμιότυπου, ο δείκτης L υποδηλώνει τον αριθμό επιπέδων του δικτύου και $\alpha^L\left(x^{(i)}\right)$ είναι το διάνυσμα ενεργοποιήσεων εξόδου του δικτύου, όταν $x^{(i)}$ είναι η είσοδος.

Για να μπορέσει να εφαρμοστεί ο αλγόριθμος backpropagation, πρέπει να γίνουν δύο βασικές υποθέσεις. Η πρώτη είναι ότι η συνάρτηση κόστους μπορεί να γραφτεί ως ένας μέσος όρος των συναρτήσεων κόστους των διαφορετικών στιγμιότυπων $\left(x^{(i)},y^{(i)}\right)$, δηλαδή $C(w,b)=\frac{1}{m}\sum_{i=1}^{m}C(w,b,x^{(i)},y^{(i)})$. Η υπόθεση αυτή γίνεται εξαιτίας του ότι ο backpropagation επιτρέπει τον υπολογισμό των μερικών παραγώγων $\frac{\partial C(w,b,x^{(i)},y^{(i)})}{\partial w}$ και $\frac{\partial C(w,b,x^{(i)},y^{(i)})}{\partial b}$ για το κάθε στιγμιότυπο εκπαίδευσης ξεχωριστά. Επίσης, σχετικά με το κόστος, υποτίθεται ότι μπορεί να γραφτεί συναρτήσει του διανύσματος ενεργοποιήσεων εξόδου του δικτύου, δηλαδή $C(w,b;x^{(i)},y^{(i)})=C(a^L)$. Η τετραγωνική συνάρτηση κόστους ικανοποιεί τη συγκεκριμένη συνθήκη, αφού το τετραγωνικό κόστος για ένα συγκεκριμένο στιγμιότυπο εκπαίδευσης, μπορεί να γραφτεί ως:

$$C(w,b,x,y)=\frac{1}{2}\sum_{j}\left(a_j^L-y_j\right)^2 \qquad \textbf{2.8.1}$$

όπου j είναι ο αριθμός νευρώνων του επιπέδου εξόδου. Συνεπώς, το κόστος είναι μία συνάρτηση των ενεργοποιήσεων εξόδου και εξαρτάται επίσης από την επιθυμητή-γνωστή τιμή της μεταβλητής εξόδου, η οποία ουσιαστικά βοηθά στον ορισμό της συνάρτησης αυτής.

Δεδομένου ενός συγκεκριμένου στιγμιότυπου εκπαίδευσης (x,y), πρώτα πραγματοποιείται ένα προς τα εμπρός πέρασμα στο δίκτυο, υπολογίζονται έτσι όλες

οι ενεργοποιήσεις εντός του δικτύου, ακόμα και η τιμή-τιμές της ενεργοποίησης-ενεργοποιήσεων εξόδου. Έπειτα, υπολογίζεται η τιμή της συνάρτησης κόστους 2.8.1 για το συγκεκριμένο στιγμιότυπο. Στη συνέχεια, εισάγεται μία ενδιάμεση ποσότητα δ_j^l, η οποία καλείται το σφάλμα στο j-οστό κόμβο, του l-οστού επιπέδου. Αυτός ο όρος σφάλματος, ουσιαστικά, μετρά κατά πόσο αυτός ο κόμβος ήταν υπεύθυνος για τα όποια σφάλματα στην έξοδο του δικτύου. Για να γίνει κατανοητό, το πως ορίζεται αυτό το σφάλμα, ας υποτεθεί ότι κάθε φορά που μία σταθμισμένη είσοδος z_j^l εισέρχεται σε ένα κόμβο, προστίθεται μία μικρή αλλαγή Δz_j^l, με αποτέλεσμα, τελικά, ο κόμβος αυτός να εξάγει $f(z_j^l + \Delta z_j^l)$, όπου f είναι προφανώς κάποια συνάρτηση ενεργοποίησης, όπως η σιγμοειδής. Η αλλαγή αυτή διαδίδεται από επίπεδο σε επίπεδο εντός του δικτύου, προκαλώντας τελικά αλλαγή στο συνολικό κόστος κατά ποσότητα $\frac{\partial C(w,b,x,y)}{\partial z_j^l}\Delta z_j^l$. Φαίνεται, λοιπόν, ότι αν η ποσότητα Δz_j^l, έχει αντίθετο πρόσημο από την $\frac{\partial C(w,b,x,y)}{\partial z_j^l}$, μειώνεται το κόστος. Αντίθετα, αν η ποσότητα $\frac{\partial C(w,b,x,y)}{\partial z_j^l}$ είναι κοντά στο μηδέν, το κόστος δε βελτιώνεται αλλάζοντας τη σταθμισμένη είσοδο z_j^l. Διαισθητικά, λοιπόν, η ποσότητα $\frac{\partial C(w,b,x,y)}{\partial z_j^l}$ είναι μετρικό σφάλματος του κάθε κόμβου. Έτσι, ορίζεται το σφάλμα δ_j^l, του νευρώνα j, στο επίπεδο l, ως:

$$\delta_j^l = \frac{\partial C(w,b,x,y)}{\partial z_j^l} \qquad \textbf{2.8.2}$$

Ο backpropagation στηρίζεται σε τέσσερις βασικές εξισώσεις. Η πρώτη εξίσωση υπολογίζει τα στοιχεία του δ^L, δηλαδή του διανύσματος των σφαλμάτων που στο επίπεδο εξόδου L. Ισχύει:

$$\delta_j^L = \frac{\partial C(w,b,x,y)}{\partial a_j^L} f'(z_j^L) \qquad \textbf{2.8.3}$$

Ο πρώτος όρος, στα δεξιά της Εξίσωσης 2.8.3, δείχνει το ρυθμό μεταβολής του κόστους, συναρτήσει της j-οστής ενεργοποίησης εξόδου. Αν ένας κόμβος εξόδου δεν έχει μεγάλη επίδραση στο κόστος, τότε το αντίστοιχο δ_j^L θα είναι μικρό. Ο δεύτερος όρος, στα δεξιά, δείχνει το ρυθμό μεταβολής της όποιας συνάρτησης ενεργοποίησης f, στο z_j^L, στην σταθμισμένη είσοδο, δηλαδή, του j-οστού κόμβου, του επιπέδου εξόδου L. Βάσει του στοιχειώδους πολλαπλασιασμού πινάκων (προϊόν Hadamard), η Εξίσωση 2.8.3, μπορεί να ξαναγραφτεί στη διανυσματοποιημένη μορφή:

$$\delta^L = \nabla_{a^L} C(w,b,x,y) \odot f'(z^L) \qquad \textbf{2.8.4}$$

Επειδή, στην περίπτωση της τετραγωνικής συνάρτησης κόστους, ισχύει $\nabla_{a^L} C(w,b,x,y) = a^L - y$, η Εξίσωση 2.8.4 γίνεται:

$$\delta^L = (a^L - y) \odot f'(z^L) \qquad \textbf{2.8.5}$$

Η δεύτερη βασική εξίσωση του backpropagation συσχετίζει ουσιαστικά το διάνυσμα σφαλμάτων σε ένα επίπεδο l, δ^l, με το διάνυσμα σφαλμάτων του αμέσως επόμενου επιπέδου l+1, δ^{l+1} και ισχύει:

$$\delta^l = ((w^{l+1})^T \delta^{l+1} \odot f'(z^l)) \qquad \textbf{2.8.6}$$

όπου $(w^{l+1})^T$ είναι ο ανάστροφος του πίνακα βαρών του (l+1)-οστού επιπέδου (τα βάρη που συνδέονται στους κόμβους του (l+1)-οστού επιπέδου). Γνωρίζοντας το σφάλμα δ^{l+1}, στο (l+1)-οστό επίπεδο και εφαρμόζοντας τον ανάστροφο του πίνακα βαρών, $(w^{l+1})^T$, το σφάλμα διαδίδεται προς τα πίσω μέσα στο δίκτυο, δίνοντας ένα μετρικό για το σφάλμα εξόδου του προηγούμενο επιπέδου, δ^l. Ο πολλαπλασιασμός ανά στοιχείο, του μετρικού αυτού, με το ρυθμό μεταβολής της συνάρτησης ενεργοποίησης στη σταθμισμένη είσοδο του επιπέδου l, δίνει το σφάλμα δ^l, στη σταθμισμένη είσοδο, στο επίπεδο l. Με μία αλληλουχία υπολογισμών, συνδυάζοντας τις Εξισώσεις 2.8.4 και 2.8.6, μπορεί να υπολογιστεί το σφάλμα δ^l κάθε επιπέδου στο δίκτυο.

Η τρίτη βασική εξίσωση του backpropagation ορίζει ότι ο ρυθμός μεταβολής της συνάρτησης κόστους ως προς κάποια πόλωση στο δίκτυο, ισούται με το σφάλμα στον αντίστοιχο κόμβο του δικτύου, ισχύει:

$$\frac{\partial C(w,b,x,y)}{\partial b_j^l} = \delta_j^l \qquad \textbf{2.8.7}$$

Η τέταρτη βασική εξίσωση, παρέχει ένα τρόπο υπολογισμού των μερικών παραγώγων της συνάρτησης κόστους ως προς οποιοδήποτε βάρος του δικτύου, με όρους των ποσοτήτων της ενεργοποίησης που μεταβαίνει σε ένα κόμβο, από κόμβο με τον οποίο συνδέεται, και χρησιμοποιείται ως είσοδος για να υπολογιστεί η αντίστοιχη δική του ενεργοποίηση και του σφάλματος στον κόμβο αυτό. Ισχύει:

$$\frac{\partial C(w,b,x,y)}{\partial w_{jk}^{l}} = a_k^{l-1}\delta_j^l \qquad \textbf{2.8.8}$$

Στην Εξίσωση 2.8.8, η παράμετρος w_{jk}^{l} είναι το βάρος από τον k νευρώνα, του (l-1) επιπέδου, στο j νευρώνα, του l επιπέδου˙ θεωρείται ότι είναι ένα από τα βάρη του j νευρώνα, του l επιπέδου. Εντός του j νευρώνα, του l επιπέδου, υπολογίζεται η σταθμισμένη είσοδος z_j^l όπου ένα από τα ζευγάρια "βάρος επί ενεργοποίηση", για να υπολογιστεί η σταθμισμένη αυτή είσοδος, είναι το $w_{jk}^{l} * a_k^{l-1}$.

Οι τέσσερις βασικές εξισώσεις του backpropagation αποδεικνύονται βάσει του κανόνα της αλυσίδας (chain rule), ο οποίος χρησιμοποιείται για τον υπολογισμό παραγώγων συναρτήσεων, οι οποίες σχηματίζονται από το συνδυασμό άλλων συναρτήσεων, των οποίων οι παράγωγοι είναι γνωστές.

Ο αλγόριθμος backpropagation, πραγματοποιεί τα παρακάτω βήματα:

1. Τιμές εισόδου x: Θέσε το αντίστοιχο διάνυσμα ενεργοποιήσεων a^1, για το επίπεδο εισόδου.
2. Προς τα εμπρός τροφοδότηση-πέρασμα: Για κάθε $l = 2,3,\dots,L$, υπολόγισε $z^l = w^l a^{l-1} + b^l$ και $a^l = f(z^l)$.
3. Σφάλμα εξόδου δ^L: Υπολόγισε το διάνυσμα $\delta^L = \nabla_{a^L} C(w,b,x,y) \odot f'(z^L)$.
4. Οπισθοδιάδοσε το σφάλμα εξόδου δ^L: Για κάθε $l = L-1, L-2, \dots, 2$, υπολόγισε $\delta^l = ((w^{l+1})^T \delta^{l+1} \odot f'(z^l))$.
5. Έξοδος: Η κλίση της συνάρτησης κόστους υπολογίζεται από: $\frac{\partial C(w,b,x,y)}{\partial w_{jk}^{l}} = a_k^{l-1}$ και $\frac{\partial C(w,b,x,y)}{\partial b_j^{l}} = \delta_j^l$.

Το προς τα εμπρός πέρασμα-εμπρόσθια τροφοδότηση του δικτύου, υπολογίζει τιμές, ξεκινώντας από κάποιες τιμές εισόδου και καταλήγοντας σε κάποια έξοδο. Ο αλγόριθμος backpropagation πραγματοποιεί το προς τα πίσω πέρασμα (backward pass) του δικτύου, ξεκινώντας από το τέλος και εφαρμόζοντας αναδρομικά τον κανόνα της αλυσίδας, ώστε να υπολογίσει τις κλίσεις, σε όλη τη διαδρομή, έως την αρχή του δικτύου. (Nielsen 2019)

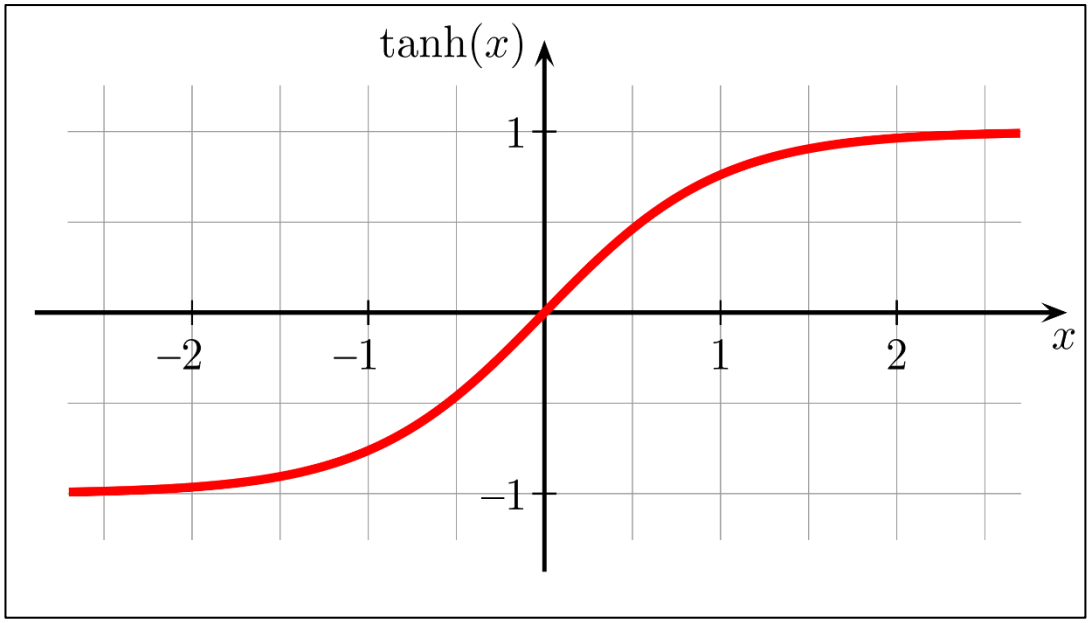
2.9 Συναρτήσεις Ενεργοποίησης

Μία από τις βασικότερες συναρτήσεις ενεργοποίησης, η οποία περιεγράφηκε στο Κεφάλαιο 2.6, είναι η σιγμοειδής συνάρτηση. Όπως γράφτηκε, λαμβάνει έναν πραγματικό αριθμό και τον "συμπιέζει" σε εύρος μεταξύ 0 και 1. Συγκεκριμένα, μεγάλοι αρνητικοί αριθμοί γίνονται 0 και μεγάλοι θετικοί αριθμοί γίνονται 1. Η σιγμοειδής συνάρτηση έχει συναντήσει συχνή χρήση ιστορικά, αφού έχει μία καλή ερμηνεία ως ο βαθμός ενεργοποίησης ενός νευρώνα. Πρακτικά, η σιγμοειδής μη-

γραμμικότητα έχει πέσει σε δυσμένεια τελευταία και χρησιμοποιείται σπάνια. Έχει δύο σημαντικά μειονεκτήματα. Πρώτον, οι σιγμοειδείς νευρώνες φτάνουν σε κορεσμό και "σκοτώνουν" τις κλίσεις. Από την γραφική παράσταση της σιγμοειδούς, φαίνεται ότι η συνάρτηση επιπεδώνεται όταν $\sigma\left(z_j^l\right)$ είναι περίπου 0 ή 1, τότε ο νευρώνας έχει φτάσει σε κορεσμό και ισχύει $\sigma'\left(z_j^l\right) \approx 0$. Αυτό το φαινόμενο καλείται "πρόβλημα εξαφανιζόμενης κλίσης" (vanishing gradient problem) και εμφανίζεται όταν το σφάλμα που διαδίδεται προς τα πίσω, έχει αρχίσει να φτάνει κοντά στο μηδέν. Ανακαλώντας την Εξίσωση 2.8.6, φαίνεται ότι αν $\sigma'\left(z_j^l\right) \approx 0$, τότε το σφάλμα δ_j^l θα είναι μικρό. Ανακαλώντας την Εξίσωση 2.8.8, φαίνεται ότι αν το σφάλμα δ_j^l είναι μικρό, τα όποια βάρη εισάγονται σε αυτό το νευρώνα, θα "μαθαίνουν" αργά, αφού η μερική παράγωγος της συνάρτησης κόστους, ως προς καθένα από αυτά τα βάρη, άρα και οι τιμές των ανανεώσεων τους, θα έχουν μικρές τιμές. Το δεύτερο μειονέκτημα της σιγμοειδούς, είναι ότι δεν είναι κεντραρισμένη ως προς το μηδέν. Μία συνάρτηση καλείται κεντραρισμένη ως προς το μηδέν όταν το εύρος των τιμών της περιλαμβάνει αρνητικές και θετικές τιμές. Αυτό είναι ανεπιθύμητο αφού οι νευρώνες στα μετέπειτα επίπεδα των διεργασιών σε ένα δίκτυο, θα λαμβάνουν τιμές οι οποίες δεν είναι κεντραρισμένες ως προς το μηδέν. Αυτό έχει επιπτώσεις στη δυναμική κατά τη διάρκεια του gradient descent, γιατί αν τα δεδομένα που έρχονται σε ένα νευρώνα είναι πάντα θετικά, τότε οι κλίσεις στα βάρη w, κατά τη διάρκεια της οπισθοδιάδοσης, θα γίνουν είτε όλα θετικές, είτε όλες αρνητικές (εξαρτάται από την παράγωγο της συνάρτησης ενεργοποίησης). Αυτό θα μπορούσε να κάνει τις ανανεώσεις από τις κλίσεις, να διαδίδονται σε διαφορετικές κατευθύνσεις. Παρ 'όλα αυτά, μόλις αυτές οι κλίσεις αθροίζονται κατά το εύρος ενός συνόλου δεδομένων, η τελική ανανέωση των βαρών μπορεί να έχει μεταβλητά πρόσημα, κάτι το οποίο μετριάζει αυτό το πρόβλημα. Συνεπώς, αυτό είναι ένα πρόβλημα, αλλά έχει λιγότερο σοβαρές συνέπειες συγκριτικά με το πρόβλημα κορεσμού των ενεργοποιήσεων.

Μία άλλη συχνά χρησιμοποιούμενη συνάρτηση ενεργοποίησης είναι η συνάρτηση υπερβολικής εφαπτομένης (hyperbolic tangent function ή tanh). Η συνάρτηση tanh είναι μη-γραμμική, κεντραρισμένη ως προς το μηδέν, της οποίας το εύρος τιμών είναι [-1,1] και είναι της μορφής:

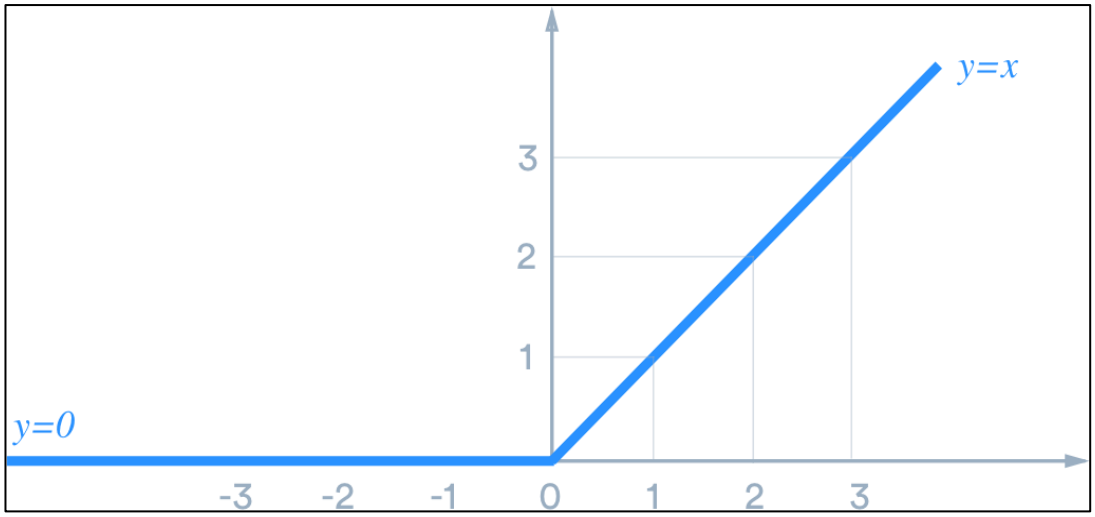
$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \qquad \textbf{2.9.1}$$



Εικόνα 11: Γραφική παράσταση συνάρτησης tanh.

Στην πράξη, η tanh προτιμάται πάντα, συγκριτικά με τη σιγμοειδή αφού αποδεδειγμένα αποδίδει καλύτερα κατά την εκπαίδευση νευρωνικών δικτύων πολλών επιπέδων. Παρ'όλα αυτά, η tanh δεν κατάφερε να δώσει λύση στο σημαντικό πρόβλημα της "κλίσης που εξαφανίζεται", από το οποίο όπως γράφτηκε παραπάνω, πάσχει και η σιγμοειδής συνάρτηση. Το βασικό πλεονέκτημα που παρέχει η tanh, είναι ότι παράγει κεντραρισμένες ως προς το μηδέν τιμές εξόδου, υποστηρίζοντας έτσι τη διαδικασία του backpropagation. Η σημαντική αρνητική ιδιότητα της tanh, είναι ότι $tanh'(z_j^l) = 1$, μόνο όταν z_j^l=0 και γενικά έχει τιμές κλίσης κοντά στο μηδέν, με αποτέλεσμα να παράγει "νεκρούς νευρώνες" κατά τη διαδικασία των υπολογισμών και τα βάρη που εισάγονται σε αυτούς τους νευρώνες είτε να μη "μαθαίνουν" καθόλου, είτε να "μαθαίνουν" πολύ αργά.

Η tanh είναι κεντραρισμένη ως προς το μηδέν, δίνοντας λύση σε ένα περιορισμό της σιγμοειδούς συνάρτησης· δεν κατάφερε, όμως, να δώσει λύση στο πρόβλημα των "νεκρών νευρώνων", από το οποίο ταλανίζεται. Η προσπάθεια λύσης σε αυτό το πρόβλημα, προκάλεσε περαιτέρω έρευνα σχετικά με τις συναρτήσεις ενεργοποίησης, έρευνα η οποία είχε σαν αποτέλεσμα τη γέννηση της συνάρτησης ReLU (Rectified Linear Unit). Η ReLU προτάθηκε για πρώτη φορά από τους Nair και Hinton, σε εργασία (Vinod Nair 2010) τους, το 2010, και έκτοτε είναι η πιο ευρέως χρησιμοποιούμενη συνάρτηση ενεργοποίησης, με κορυφαία απόδοση σε εφαρμογές νευρωνικών δικτύων, έως και σήμερα.



Εικόνα 12: Συνάρτηση ReLU.

Η συνάρτηση ReLU είναι συνεχής και μη κεντραρισμένη ως προς το μηδέν. Πραγματοποιεί μία κατωφλίωση σε κάθε τιμή εισόδου, όπου κάθε τιμή μικρότερη του μηδενός, τίθεται μηδέν και τιμές μεγαλύτερες ή ίσες του μηδενός, μένουν ως έχουν. Η μορφή της είναι:

$$f(x) = \max(0, x) = \begin{cases} x_i, \alpha\nu\ x_i \geq 0 \\ 0, \alpha\nu\ x_i < 0 \end{cases} \qquad \textbf{2.9.2}$$

Από την εργασία των Krizhevsky et al. (2012), φάνηκε ότι οι ReLU νευρώνες, συγκριτικά με τους σιγμοειδείς και τους tanh, επιτάχυναν πολύ της διαδικασία σύγκλισης του αλγόριθμου stochastic gradient descent (περισσότερα σχετικά με το συγκεκριμένο αλγόριθμο στη συνέχεια), συγκεκριμένα κατά ένα συντελεστή της τάξης του έξι (Alex Krizhevsky 2012). Ακόμα, ένα σημαντικό πλεονέκτημα της συνάρτησης ReLU, είναι ότι πραγματοποιεί απλές πράξεις, υπολογιστικά μη κοστοβόρες, αντίθετα με την tanh και τη σιγμοειδή, οι οποίες περιέχουν πολύ περισσότερο κοστοβόρες πράξεις (εκθετικά κ.λπ.). Ένα σημαντικό πλεονέκτημα της ReLU, είναι ότι αντιμετωπίζει το πρόβλημα της "εξαφανιζόμενης κλίσης". Για τη συγκεκριμένη συνάρτηση ισχύει:

$$f'(x) = \begin{cases} 1, \gamma\iota\alpha\ x > 0 \\ 0, \alpha\lambda\lambda\omicron\acute{\upsilon} \end{cases} \qquad \textbf{2.9.3}$$

Φαίνεται, λοιπόν, ότι όταν ένας νευρώνας ενεργοποιείται με είσοδο μεγαλύτερη του μηδενός, η μερική παράγωγος της συνάρτησης ενεργοποίησης είναι 1. Αφού η κλίση είναι είτε ακριβώς μηδέν, είτε ένα, το πρόβλημα της "εξαφανιζόμενης κλίσης" δεν

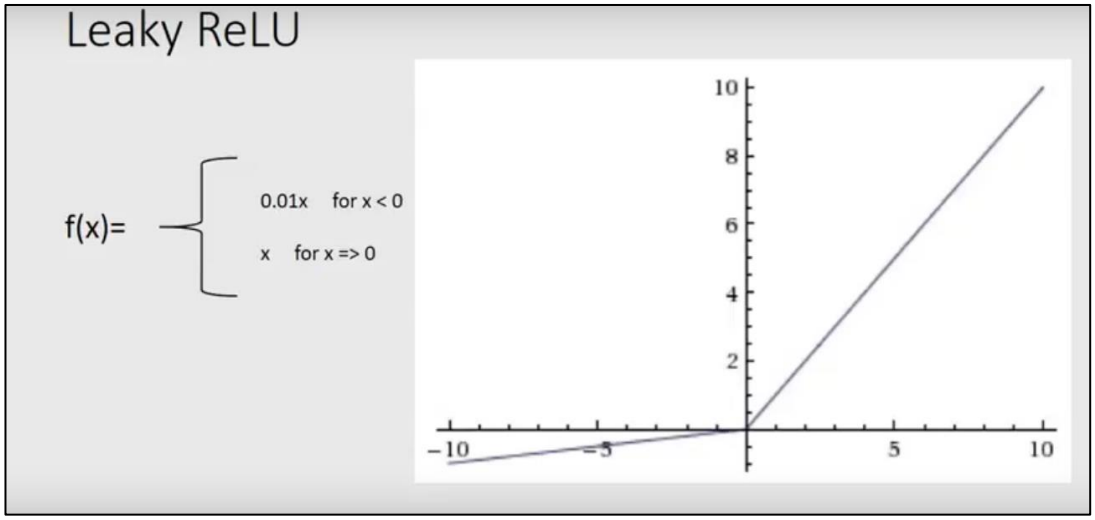
υπάρχει κατά το μήκος μονοπατιών με ενεργούς κρυφούς νευρώνες εντός του δικτύου (Andrew L. Maas 2013). Η ReLU, όμως, έχει ένα σημαντικό μειονέκτημα. Όπως φαίνεται από τη μορφή της συνάρτησης, κάθε τιμή εισόδου στη ReLU, που είναι μικρότερη του μηδενός, παράγει τιμή εξόδου μηδέν. Αυτό έχει σαν αποτέλεσμα νευρώνες που λαμβάνουν σταθμισμένες εισόδους με τιμή μικρότερη του μηδενός, να εξάγουν τιμή ενεργοποίησης μηδέν. Όταν αρκετοί νευρώνες παράγουν έξοδο μηδέν, το δίκτυο γίνεται εύθραυστο, δυσχεραίνει η ροή των κλίσεων προς τα πίσω κατά τη διάρκεια του backpropagation και τα βάρη δεν ανανεώνονται. Ουσιαστικά, ένα μεγάλο τμήμα του δικτύου γίνεται ανενεργό και δεν "μαθαίνει" περαιτέρω. Οι νευρώνες που πάσχουν από αυτό το πρόβλημα καλούνται "νεκροί νευρώνες" (dead neurons) και το πρόβλημα αυτό γενικά καλείται "πρόβλημα της ReLU που πεθαίνει" (dying ReLU) Επειδή η κλίση της ReLU στο εύρος αρνητικών τιμών, είναι επίσης μηδέν, όταν ένας νευρώνας λάβει επαναλαμβανόμενα αρνητικές σταθμισμένες εισόδους και εξάγει μηδέν, είναι πολύ δύσκολο να επανακάμψει (Datta 2020). Με άλλα λόγια, οι νευρώνες ReLU μπορούν μη-αναστρέψιμα να γίνουν ανενεργοί κατά τη διάρκεια της εκπαίδευσης αφού ίσως "χτυπηθούν" από τις πολλές πτυχές των δεδομένων. Ένας σημαντικός παράγοντας, ο οποίος σε πολλές περιπτώσεις παίζει καθοριστικό ρόλο ως αιτία του συγκεκριμένου προβλήματος, είναι ο υψηλός βαθμός μάθησης. Ανακαλώντας τον κανόνα ανανέωσης των παραμέτρων, $w_{jk}^{l} \rightarrow w_{jk}^{l'} = w_{jk}^{l} - \eta \frac{\partial C(w,b,x,y)}{\partial w_{jk}^{l}}$, φαίνεται ότι αν ο βαθμός μάθησης η, είναι πολύ υψηλός, είναι πολύ πιθανό ότι τα ανανεωμένο βάρος θα καταλήξει σε ένα εύρος υψηλών αρνητικών τιμών, αφότου το παλιό βάρος θα αφαιρεθεί από ένα μεγάλο αριθμό. Τα αρνητικά βάρη, καταλήγουν σε αρνητική σταθμισμένη είσοδο στη ReLU, προκαλώντας το πρόβλημα της "ReLU που πεθαίνει". Ακόμα, πολώσεις με υψηλές αρνητικές τιμές, μπορούν να επιφέρουν σημαντικά προβλήματα. Η πόλωση είναι μία σταθερή τιμή της σταθμισμένης εισόδου. Αν η πόλωση είναι υψηλή αρνητική τιμή, μπορεί να κάνει τη σταθμισμένη είσοδο στη συνάρτηση ReLU, επίσης αρνητική. Έτσι, λοιπόν, η πόλωση έχει το δικό της ρόλο σχετικά με το πρόβλημα της "ReLU που πεθαίνει". Για να αντιμετωπιστεί το συγκεκριμένο πρόβλημα των νευρώνων ReLU, προτάθηκε η συνάρτηση ενεργοποίησης της διαρρέουσας (Leaky) ReLU.

Η Leaky ReLU προτάθηκε για πρώτη φορά από τους Maas et al. (2013) (Andrew L. Maas 2013) και ορίζεται ως:

$$f(x) = \begin{cases} x, \alpha\nu\ x > 0 \\ 0{,}01x, \alpha\nu\ x \leq 0 \end{cases} \qquad \textbf{2.9.4}$$

Όπως φαίνεται και από τη μορφή της, η Leaky ReLU είναι μία παραλλαγή της ReLU, όπου αντίθετα με τη ReLU, έχει μία μικρή κλίση στο εύρος των αρνητικών τιμών εισόδου, ουσιαστικά επιτρέποντας ένα μικρό τμήμα αρνητικών σταθμισμένων

εισόδων, αντί να τις “συμπιέζει” στο μηδέν. Η συγκεκριμένη συνάρτηση είναι συνεχής, κεντραρισμένη ως προς το μηδέν και υπολογιστικά μη κοστοβόρα.



Εικόνα 13: Συνάρτηση ενεργοποίησης leaky ReLU.

Η παράγωγος της συνάρτησης Leaky ReLU, είναι της μορφής:

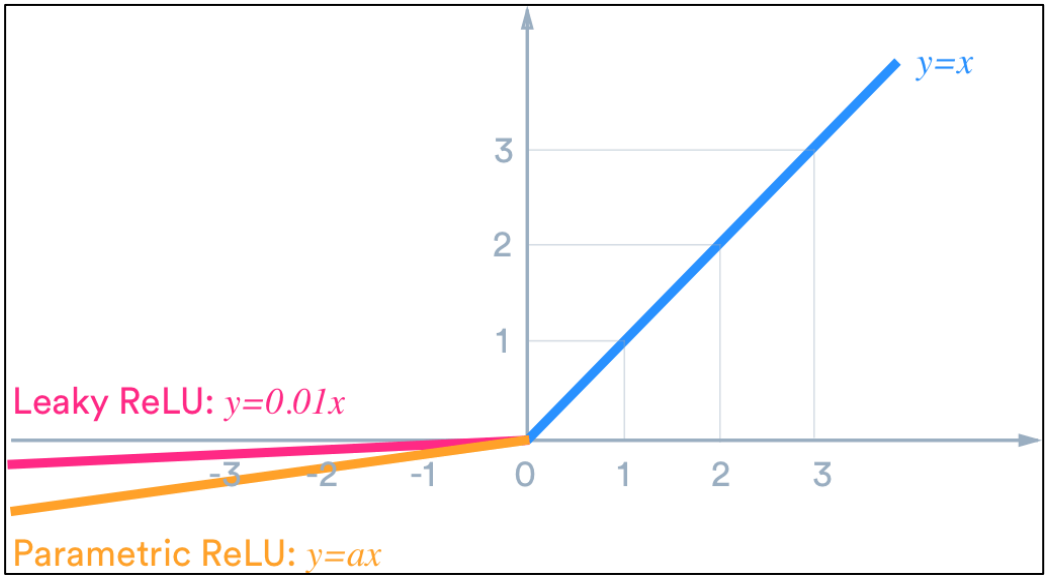
$$f'(x) = \begin{cases} 1, \alpha\nu\ x > 0 \\ 0{,}01, \alpha\nu\ x < 0 \end{cases} \qquad 2.9.5$$

Φαίνεται, λοιπόν, ότι στο εύρος των θετικών τιμών, η κλίση της συνάρτησης είναι πάντα 1, άρα το πρόβλημα της “εξαφανιζόμενης κλίσης”, δεν υπάρχει. Αντίθετα, στο εύρος των αρνητικών τιμών, η κλίση είναι πάντα 0,01. Αυτή είναι μία τιμή αρκετά κοντά στο 0, η οποία επιφέρει ρίσκο. Οι Maas et al., στην εργασία τους (Andrew L. Maas 2013), υποστηρίζουν ότι *“Η Leaky ReLU επιτρέπει μία μικρή, μη-μηδενική κλίση, όταν ο νευρώνας είναι κορεσμένος και μη ενεργός”,* υποδηλώνοντας ότι έτσι δίνεται στο νευρώνα και στα συνδεδεμένα βάρη η ευκαιρία να επανακάμψουν, με αποτέλεσμα να βελτιωθεί η συνολική απόδοση μάθησης.

Μία παραλλαγή της leaky ReLU, η οποία εισήχθη για πρώτη φορά από τους Kaiming He et al. (Kaiming He 2015), το 2015, είναι η Parametric ReLU (PReLU). Η διαφορά με τη Leaky ReLU, είναι ότι η κλίση στο εύρος των αρνητικών τιμών εισόδου, γίνεται μία παράμετρος του νευρώνα, η οποία μαθαίνεται προσαρμοστικά (στην περίπτωση της Leaky ReLU, θα μπορούσαμε να πούμε ότι το 0,01 είναι μία υπέρ-παράμετρος, αφού καθορίζεται εξαρχής και δε μπορεί να αλλάξει κατά τη διάρκεια της εκπαίδευσης) κατά τη διάρκεια του backpropagation, ενόσω στο εύρος των θετικών τιμών, η συνάρτηση είναι γραμμική. Η συνάρτηση PReLU, είναι της μορφής:

$$f(x) = \begin{cases} x, \alpha\nu\ x > 0 \\ \alpha_i x, \alpha\nu\ x \leq 0 \end{cases} \qquad \textbf{2.9.6}$$

Η συνάρτηση PReLU είναι συνεχής και κεντραρισμένη ως προς το μηδέν. Η παράγωγος της συνάρτησης, στο εύρος των θετικών τιμών, είναι πάντα 1, άρα δεν υπάρχει θέμα "εξαφανιζόμενης κλίσης". Στο εύρος των αρνητικών τιμών, όμως, η παράγωγος είναι α, η οποία συνήθως είναι μία τιμή κοντά στο 0, οπότε επιφέρει ρίσκο σχετικά με το πρόβλημα "εξαφανιζόμενης κλίσης". Στην Εξίσωση 2.9.6, x είναι η σταθμισμένη είσοδος στο νευρώνα και a_i είναι ο συντελεστής που ελέγχει την κλίση στο εύρος των αρνητικών τιμών. Οι He et al., στην εργασία τους (Kaiming He 2015),υποστηρίζουν ότι "*Η μέθοδος μας προσαρμοστικά μαθαίνει τις παραμέτρους PReLU, ενιαία με το συνολικό μοντέλο. Ευελπιστούμε σε μία από άκρη σε άκρη εκπαίδευση, που θα οδηγήσει σε πιο εξειδικευμένες ενεργοποιήσεις*".



Εικόνα 14: Σύγκριση ReLU και PReLU.

Άλλα είδη συναρτήσεων έχουν προταθεί, τα οποία δεν έχουν τη μορφή $f(w^T x + b)$, όπου μία μη-γραμμικότητα εφαρμόζεται στο εσωτερικό γινόμενο των βαρών και των δεδομένων (ή ενεργοποιήσεων του προηγούμενου επιπέδου). Μία σχετικά διάσημη επιλογή είναι ο νευρώνας Maxout, ο οποίος προτάθηκε για πρώτη φορά το 2013, από τους Goodfellow et al. (Ian J. Goodfellow 2013). Η συνάρτηση Maxout, ουσιαστικά γενικεύει τη ReLU και τη Leaky εκδοχή της, "κληρονομώντας", θα μπορούσε κάποιος να πει, την ιδιότητα να αντιμετωπίζει το πρόβλημα των "νευρώνων που πεθαίνουν" και του κορεσμού. Η Maxout, αντί να εφαρμόζει μία στοιχειώδη συνάρτηση $g(z)$, διαιρεί τη z σε k τιμές˙ δηλαδή περισσότεροι του ενός Maxout νευρώνες, διαιρούν τη z σε ομάδες των k τιμών. Κάθε νευρώνας Maxout, έτσι, εξάγει το μέγιστο των k

τιμών, μίας από αυτές τις ομάδες. Δεδομένου ενός διανύσματος $x \in \mathbb{R}^d$ (μπορεί να είναι η κατάσταση ενός κρυφού επιπέδου, οι ενεργοποιήσεις του προηγούμενου του Maxout επιπέδου δηλαδή):

$$h_i(x) = \max z_{ij} \qquad \textbf{2.9.7}$$

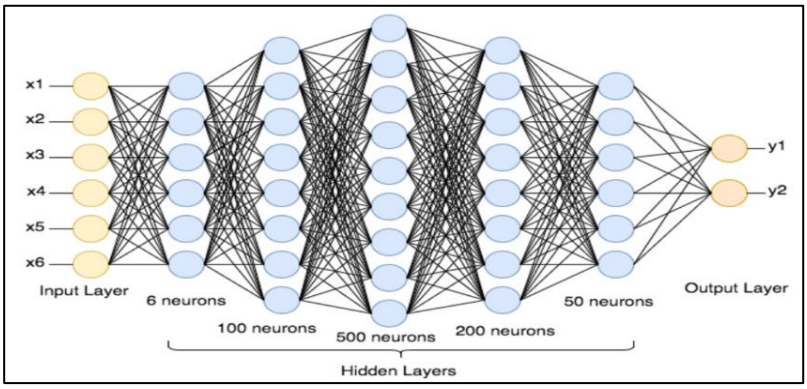
όπου $j \in [1, k]$, $z_{ij} = w_{\dots ij}^T x + b_{ij}$ και $w \in \mathbb{R}^{d \times m \times k}$ και $b \in \mathbb{R}^{m \times k}$. Η Maxout, λοιπόν, εφαρμόζει k εσωτερικά γινόμενα στο x, συν k μετατοπίσεις $(b_{i1}, \dots, b_{ik})$ και τελικά κρατάει τη μέγιστη των k αυτών τιμών. Κάθε νευρώνας Maxout, λοιπόν, έχει k αφινικούς συνδυασμούς του προηγούμενου επιπέδου, και εξάγει το μέγιστο αυτών των k συνδυασμών. Το υπολογιστικό κόστος αυξάνεται, αφού κάθε Maxout νευρώνας παραμετροποιείται με k διανύσματα βαρών, αντί απλά ενός, όπου k είναι μία υπέρ-παράμετρος, η οποία δε μαθαίνεται κατά την εκπαίδευση, αλλά προκαθορίζεται για κάθε Maxout νευρώνα εξαρχής. (Chigozie Enyinna Nwankpa 2018)

Ένας νευρώνας Maxout μπορεί να μάθει μία τμηματική, γραμμική, κυρτή συνάρτηση, με έως k τμήματα, που αποκρίνεται σε διάφορες κατευθύνσεις στο χώρο τιμών εισόδου x. Ένα επίπεδο με Maxout νευρώνες μπορεί να προσαρμόσει τις τιμές εισόδου του σε διάφορα γεωμετρικά πολύτοπα στο d-διάστατο χώρο και να υπολογίσει μία γραμμική συνάρτηση σε κάθε πολύτοπο. Με ένα επαρκώς μεγάλο αριθμό k, ο νευρώνας Maxout μπορεί να μάθει να προσεγγίζει οποιαδήποτε κυρτή συνάρτηση αρκετά ικανοποιητικά. Από αυτή την οπτική, οι Maxout νευρώνες καλούνται "καθολικοί προσεγγιστές" (universal approximators). Η Maxout, στην περίπτωση όπου $k = 2$, μπορεί να μάθει να υλοποιεί την ίδια συνάρτηση των τιμών εισόδου x, όπως ένας νευρώνας που χρησιμοποιεί ως συνάρτηση ενεργοποίησης τη ReLU, τη Leaky ReLU ή την Parametric ReLU. Ο Maxout νευρώνας θα μάθει την ίδια συνάρτηση π.χ. με τη ReLU, αλλά με μεγαλύτερο υπολογιστικό κόστος, αφού θα πρέπει να παραμετροποιηθεί διαφορετικά. Όσων αφορά τη σχέση της Maxout, με τη ReLU, αν υποθέσουμε ότι για τη συνάρτηση Maxout, ισχύει $k = 2$, δηλαδή $h_1(x) = \max(z_{11}, z_{12})$= $\max(w_{11}^T x + b_{11}, w_{12}^T x + b_{12})$, τότε η ReLU είναι μία ειδική περίπτωση αυτής της μορφής, όταν $w_{11}, b_{11} = 0$. (Ian Goodfellow 2016)

2.10 Κανονικοποίηση (Regularization)

Όπως περιεγράφηκε προηγουμένως, τα τεχνητά νευρωνικά δίκτυα τροφοδοτούνται με στιγμιότυπα δεδομένων εκπαίδευσης, τα οποία έχουν κάποια χαρακτηριστικά. Έτσι, λοιπόν, σε διαδικασίες επιβλεπόμενης μάθησης, το δίκτυο εκπαιδεύεται ώστε να μάθει πως και σε ποιο βαθμό τα χαρακτηριστικά του κάθε στιγμιότυπου εκπαίδευσης σχετίζονται με την αντίστοιχη τιμή κάποιας μεταβλητής εξόδου, η οποία σε περιπτώσεις ταξινόμησης, όπως γράφτηκε προηγουμένως, είναι μία τιμή-ετικέτα

η οποία προσδιορίζει σε ποια κατηγορία ανήκει κάποιο στιγμιότυπο. Στην επιβλεπόμενη μάθηση, ο αλγόριθμος εκπαιδεύεται να εντοπίζει μοτίβα στα δεδομένα και να αντιλαμβάνεται το πως αυτά τα μοτίβα σχετίζονται με τιμές εξόδου. Ουσιαστικά, ο αλγόριθμος ανταποκρίνεται σε μία πολύπλοκη ερώτηση όπως “σε ποια κατηγορία ανήκει αυτό το στιγμιότυπο;”, κατακερματίζοντας την ερώτηση αυτή σε απλούστερες ερωτήσεις που να μπορούν να απαντηθούν βάσει των μοτίβων των χαρακτηριστικών του στιγμιότυπου. Οι νευρώνες του κάθε κρυφού επιπέδου μπορούν να παρομοιαστούν με τέτοιες ερωτήσεις σχετικά με τα χαρακτηριστικά του κάθε στιγμιότυπου. Το κάθε κρυφό επίπεδο μπορεί να παρομοιαστεί σαν να προσδιορίζει κάποια βαθμίδα ιεραρχίας, όπου τα πρώτα επίπεδα αντιστοιχούν σε μοτίβα πιο γενικών χαρακτηριστικών, ενώ τα επίπεδα πιο μέσα, πιο “βαθιά” στο δίκτυο, αντιστοιχούν σε πιο εξειδικευμένα, λεπτομερή χαρακτηριστικά. Γίνεται, λοιπόν, αντιληπτό ότι ένα νευρωνικό δίκτυο με πολλά κρυφά επίπεδα, που τροφοδοτείται με μεγάλο όγκο δεδομένων εκπαίδευσης, μπορεί να μάθει ιεραρχίες χαρακτηριστικών, όπου τα χαρακτηριστικά από τις υψηλότερες βαθμίδες της ιεραρχίας, συντίθενται από χαρακτηριστικά κατώτερων βαθμίδων, με αποτέλεσμα το συνολικό σύστημα να δημιουργεί πολύπλοκες συναρτήσεις συσχέτισης δεδομένων εισόδου και εξόδου, κατευθείαν από τα δεδομένα. Η διαδικασία αυτή καλείται Βαθιά Μάθηση (Deep Learning) και σήμερα, με την πρόοδο της τεχνολογίας και των δυνατοτήτων των ηλεκτρονικών υπολογιστών, δίνει λύσεις σε πολύπλοκα προβλήματα.



Εικόνα 15: Βαθύ Νευρωνικό Δίκτυο (Deep Neural Network).

Οι αλγόριθμοι, τα βαθιά νευρωνικά δίκτυα, εκπαιδεύονται με δεδομένα, με σκοπό, στη συνέχεια, να δώσουν αποτελέσματα για νέα δεδομένα εισόδου. Ένα νευρωνικό δίκτυο, λοιπόν, προσπαθεί να μάθει τις βέλτιστες παραμέτρους του, βάρη και πολώσεις, μέσα από τη συστηματική βελτίωση της απόδοσης του, σε δεδομένα εκπαίδευσης, με σκοπό να μπορεί να αποδώσει καλά και σε νέα δεδομένα εισόδου, που δεν έχει “ξαναδεί” ποτέ του. Αυτό είναι ένα βασικό πρόβλημα στη μάθηση μηχανής, δηλαδή το πως ο αλγόριθμος θα μάθει να αποδίδει καλά όχι μόνο στα

δεδομένα εκπαίδευσης, αλλά και σε νέα δεδομένα εισόδου. Όταν αυτό πραγματοποιείται, αυτό σημαίνει ότι αλγόριθμος γενικεύει, δηλαδή μέσα από τα δεδομένα εκπαίδευσης έχει ανακαλύψει γενικά μοτίβα και δεν απομνημονεύει απλώς τα ειδικά μοτίβα των δεδομένων εκπαίδευσης. Όταν αυτό δεν πραγματοποιείται, αυτό σημαίνει ότι υπάρχει υπέρ-προσαρμογή ή υπέρ-εκπαίδευση (overfitting or overtraining). Για την αντιμετώπιση της υπέρ-προσαρμογής, έχουν αναπτυχθεί αρκετές μέθοδοι, οι οποίες συνολικά ανήκουν σε μία ομάδα μεθόδων που ονομάζεται κανονικοποίηση (regularization).

Υπάρχουν αρκετές μέθοδοι κανονικοποίησης. Κάποιες εξ αυτών, τοποθετούν πρόσθετους περιορισμούς σε ένα μοντέλο μηχανικής μάθησης, όπως η προσθήκη περιστολής στις τιμές των παραμέτρων. Κάποιες τοποθετούν πρόσθετους όρους στη συνάρτηση κόστους, κάτι το οποίο μπορεί να θεωρηθεί ως ένα είδος ήπιου περιορισμού των τιμών των παραμέτρων, υπό την έννοια ότι αυτοί οι όροι λειτουργούν σαν ποινές εντός του κόστους. Κάποιες φορές αυτοί οι περιορισμοί και οι όροι ποινής τοποθετούνται με σκοπό να κωδικοποιηθεί και να εισαχθεί στο μοντέλο κάποια διαθέσιμη, υπάρχουσα γνώση. Άλλες φορές, επιλέγονται και τοποθετούνται ως προσπάθεια έκφρασης μίας γενικής προτίμησης απλούστερων, λιγότερο πολύπλοκων μοντέλων, με σκοπό να προαχθεί η γενίκευση.

Οι δύο εκ των βασικότερων μεθόδων κανονικοποίησης, ανήκουν στην ομάδα των ποινών των παραμέτρων (parameter norm penalties). Η μία εκ των δύο αυτών μεθόδων, ονομάζεται κανονικοποίηση L2 (L2 regularization) ή αποσάθρωση-ελάττωση βάρους (weight decay). Στην εργασία (Anders Krogh 1992) τους, οι Krogh και Hertz, υποστηρίζουν ότι "*ένας τρόπος να περιοριστεί ένα δίκτυο, και συνεπώς να ελαττωθεί η πολυπλοκότητα του, είναι με το να ελεγχθεί η μεγέθυνση των βαρών του, μέσω κάποιου είδους ελάττωσης βαρών. Αυτή η διαδικασία, θα πρέπει να αποτρέψει τα βάρη από το να γίνουν πολύ μεγάλα, εκτός αν είναι απολύτως απαραίτητο. Αυτό μπορεί να γίνει αντιληπτό, τοποθετώντας έναν όρο στη συνάρτηση κόστους, που να τιμωρεί τα μεγάλα βάρη* ". Η L2 κανονικοποίηση έχει τη μορφή:

$$C(w,b,x,y) = C_0(w,b,x,y) + \frac{\lambda}{2m}\sum_w w^2 \qquad \textbf{2.10.1}$$

όπου ο όρος $C_0(w,b,x,y)$ είναι η κλασσική, μη-κανονικοποιημένη συνάρτηση κόστους. Έχει προστεθεί, όμως, ένας ακόμα όρος, ο $\sum_w w^2$, ο οποίος είναι το άθροισμα των τετραγώνων όλων των βαρών του δικτύου. Ο όρος αυτός πολλαπλασιάζεται με ένα συντελεστή $\lambda/2m$, όπου λ είναι η παράμετρος κανονικοποίησης (regularization parameter) και m, κλασσικά, είναι ο αριθμός των στιγμιότυπων εκπαίδευσης.

Η επίδραση της κανονικοποίησης, ουσιαστικά, είναι να αναγκάζει το δίκτυο να προτιμά να μαθαίνει μικρά βάρη, ενώ όλα τα άλλα παραμένουν ίδια. Τα μεγάλα βάρη επιτρέπονται μόνο όταν βελτιώνουν σημαντικά το πρώτο μέρος της συνάρτησης κόστους. Επιπρόσθετα, η L2 κανονικοποίηση μπορεί να εκληφθεί ως ένας συμβιβασμός ανάμεσα στον υπολογισμό μικρών βαρών και στην ελαχιστοποίηση της αρχικής συνάρτησης κόστους. Η τιμή της παραμέτρου κανονικοποίησης λ είναι που ρυθμίζει τις ισορροπίες του συμβιβασμού αυτού˙ όταν το λ είναι μικρή τιμή, σημαίνει ότι προτιμάται η ελαχιστοποίηση της συνάρτησης κόστους και όταν το λ είναι μεγάλη τιμή, αυτό σημαίνει ότι προτιμώνται τα μικρά βάρη.

Η υλοποίηση του gradient descent, σε ένα L2 κανονικοποιημένο δίκτυο, είναι αρκετά απλή. Όπως περιεγράφηκε προηγουμένως, στόχος είναι να υπολογιστούν οι μερικές παράγωγοι της συνάρτησης κόστους ως προς κάθε βάρος και κάθε πόλωση στο δίκτυο. Παίρνοντας τις μερικές παραγώγους της Εξίσωσης 2.10.1, ισχύει:

$$\frac{\partial C(w,b,x,y)}{\partial w_{jk}^{l}} = \frac{\partial C_0(w,b,x,y)}{\partial w_{jk}^{l}} + \frac{\lambda}{m} w_{jk}^{l}$$
$$\frac{\partial C(w,b,x,y)}{\partial b_{j}^{l}} = \frac{\partial C_0(w,b,x,y)}{\partial b_{j}^{l}} \qquad \textbf{2.10.2}$$

Οι όροι $\frac{\partial C_0(w,b,x,y)}{\partial w_{jk}^{l}}$ και $\frac{\partial C_0(w,b,x,y)}{\partial b_{j}^{l}}$, υπολογίζονται με τον αλγόριθμο backpropagation, βάσει της διαδικασίας που περιεγράφηκε στο Κεφάλαιο 2.8. Όπως φαίνεται από τις Εξισώσεις 2.10.2, ο gradient descent, με την L2 κανονικοποίηση, εφαρμόζεται χωρίς τον υπολογισμό πολύπλοκων σχέσεων˙ μάλιστα οι μερικές παράγωγοι ως προς τις πολώσεις b_j^l, παραμένουν ακριβώς ίδιες, όπως στην περίπτωση που δεν υφίσταται κανονικοποίηση L2. Ο κανόνας ανανέωσης βαρών και πολώσεων με τον gradient descent, γίνεται:

$$w_{jk}^{l} \to w_{jk}^{l\,\prime} = w_{jk}^{l} - \eta \frac{\partial C_0(w,b,x,y)}{\partial w_{jk}^{l}} - \frac{\eta\lambda}{m} w_{jk}^{l} =$$
$$= \left(1 - \frac{\eta\lambda}{m}\right) w_{jk}^{l} - \eta \frac{\partial C_0(w,b,x,y)}{\partial w_{jk}^{l}} \qquad \textbf{2.10.3}$$
$$b_{j}^{l} \to b_{j}^{l\,\prime} = b_{j}^{l} - \eta \frac{\partial C_0(w,b,x,y)}{\partial b_{j}^{l}}$$

Από τις Εξισώσεις 2.10.3, φαίνεται ότι ο κανόνας ανανέωσης των βαρών είναι απαράλλαχτος, με τη διαφορά ότι πρώτα γίνεται μία προσαρμογή του βάρους, βάσει της ποσότητας $\left(1 - \frac{\eta\lambda}{m}\right)$. Από τη σχέση αυτή, δίνεται η εντύπωση ότι το βάρος w_{jk}^{l}

συρρικνώνεται συνεχώς και τείνει προς το μηδέν˙ όντως αυτό θα συνέβαινε, αλλά στη συγκεκριμένη περίπτωση ο όρος $\left(-\eta \frac{\partial C_0(w,b,x,y)}{\partial w_{jk}^l}\right)$ τείνει να παρέχει μία εξισορροπητική αύξηση στην τιμή του βάρους w_{jk}^l, εάν κάτι τέτοιο προκαλεί μείωση της τιμής της μη-κανονικοποιημένης συνάρτησης κόστους.

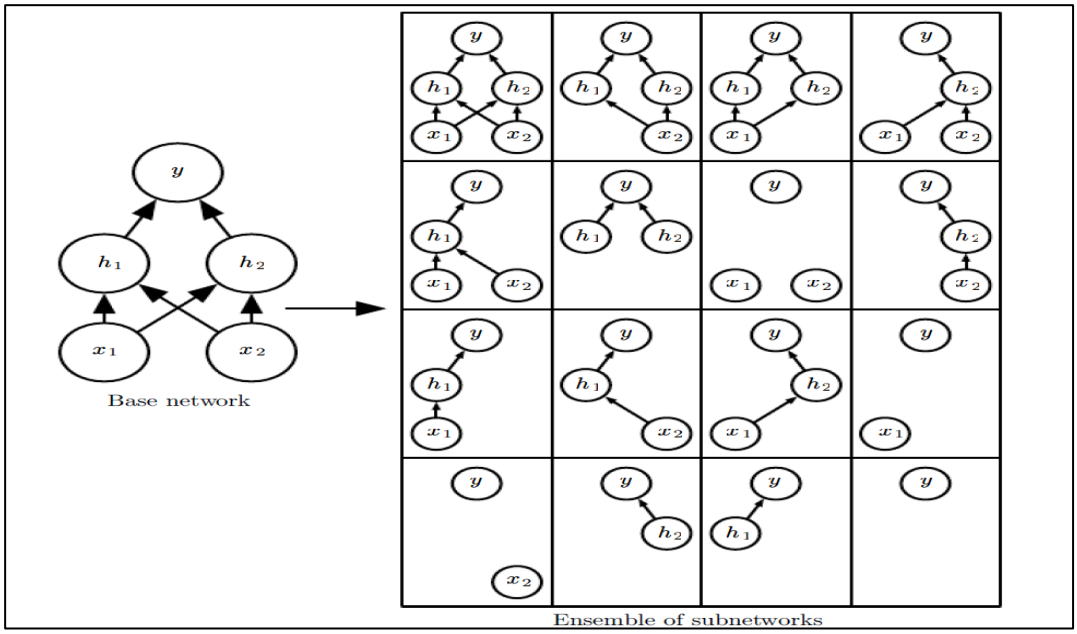
Η δεύτερη εκ των βασικότερων μεθόδων κανονικοποίησης, η οποία ανήκει επίσης στην ομάδα των ποινών των παραμέτρων (parameter norm penalties), είναι η L1 κανονικοποίηση (L1 regularization), η οποία τροποποιεί τη μη-κανονικοποιημένη συνάρτηση κόστους προσθέτοντας της το άθροισμα των απόλυτων τιμών των βαρών. Η μορφή της L1 κανονικοποίησης είναι:

$$C(w,b,x,y) = C_0(w,b,x,y) + \frac{\lambda}{m}\sum_w |w| \qquad \textbf{2.10.4}$$

Η L1 κανονικοποίηση είναι παρόμοια με την L2, αφού επίσης "τιμωρεί" τα μεγάλα βάρη και τείνει να εξαναγκάζει το συνολικό δίκτυο να προτιμά μικρά βάρη. Η L1 κανονικοποίηση έχει σαν αποτέλεσμα μοντέλα τα οποία συμπυκνώνουν βάρη τα που εστιάζουν κυρίως σε ένα μικρό τμήμα των χαρακτηριστικών των δεδομένων εκπαίδευσης, αφού οδηγεί, κατά κάποιον τρόπο, τα βάρη να γίνουν μηδενικά. Αυτή της η ιδιότητα έχει σαν αποτέλεσμα πιο αραιά (sparse) βάρη, σε σχέση με την L1 κανονικοποίηση. Η ιδιότητα αυτή, της L1, καλείται συχνά επιλογή χαρακτηριστικών, αφού το μοντέλο τείνει να σταματά να λαμβάνει υπόψη του συγκεκριμένες περιοχές του χώρου των χαρακτηριστικών, δίνοντας στα βάρη τους μηδενική τιμή.

Μία απλή, και σχετικά πρόσφατα εισηγμένη μέθοδος κανονικοποίησης, είναι η Dropout (Nitish Srivastava 2014). Η συγκεκριμένη μέθοδος, δεν τροποποιεί τη συνάρτηση κόστους, όπως οι L1 και L2, αλλά τροποποιεί το ίδιο το νευρωνικό δίκτυο. Η μέθοδος Dropout είναι εμπνευσμένη από την παρατήρηση του ότι σχεδόν πάντα, ο συνδυασμός διαφορετικών μοντέλων νευρωνικών δικτύων, βελτιώνει την απόδοση όσον αφορά τα εξαγόμενα αποτελέσματα. Αυτό αποδεικνύεται και από την εργασία (Yehuda Koren 2009) των Koren et al., η οποία κέρδισε το διαγωνισμό Netflix Grand Prize του 2009. Η εκπαίδευση διαφορετικών μοντέλων, όμως, είναι αρκετά χρονοβόρα αφού προϋποθέτει πολλαπλές υπέρ-παραμετροποιήσεις και επίσης απαιτεί μεγάλες ποσότητες δεδομένων, ώστε τα αποτελέσματα των διακριτών μοντέλων να είναι αξιόπιστα. Οι Srivastava et al., λοιπόν, υποστηρίζουν "Η Dropout είναι μία τεχνική που αντιμετωπίζει και τα δύο θέματα. Προλαμβάνει την υπέρ-προσαρμογή και παρέχει ένα τρόπο προσεγγιστικού συνδυασμού εκθετικά πολλών διαφορετικών αρχιτεκτονικών νευρωνικών δικτύων, αποδοτικά". Όπως φαίνεται και από την ονομασία της μεθόδου, Dropout-εγκατάλειψη, η συγκεκριμένη μέθοδος απενεργοποιεί-εγκαταλείπει νευρώνες, κατά τη διαδικασία της εκπαίδευσης. Αυτό

έχει σαν αποτέλεσμα ο νευρώνας αυτός να αφαιρείται, προσωρινά, από το δίκτυο, μαζί με όλες τις συνδέσεις από και προς αυτόν. Η Dropout μπορεί να εφαρμοστεί σε οποιοδήποτε επίπεδο, ακόμα και στο επίπεδο εισόδου, εκτός, όμως, από το επίπεδο εξόδου. Η επιλογή του ποιοι νευρώνες θα αφαιρεθούν είναι τυχαία. Στην απλούστερη περίπτωση, κάθε μονάδα διατηρείται με μία σταθερή πιθανότητα p ανεξάρτητη από τις άλλες μονάδες, όπου η p συνήθως τίθεται ως 0,5, μία τιμή δείχνει να είναι αρκετά κοντά στο βέλτιστο βάσει πειραματικών αποτελεσμάτων από ένα μεγάλο εύρος δικτύων και εργασιών. Για τους νευρώνες εισόδου, παρ'όλα αυτά, η βέλτιστη πιθανότητα διατήρησης είναι συνήθως κάποια τιμή πιο κοντά στη μονάδα, παρά στην τιμή 0,5. Η Dropout, λοιπόν, εκπαιδεύει ένα σύνολο, που αποτελείται από όλα τα υπό-δίκτυα που μπορούν να σχηματιστούν, αφαιρώντας νευρώνες από το βασικό, υποκείμενο δίκτυο. Η διαδικασία αυτή, δηλαδή, ισοδυναμεί με τη λήψη δείγματος-"αραιωμένου" δικτύου από το βασικό δίκτυο, με το "αραιωμένο" δίκτυο, κάθε φορά, να αποτελείται από όλες τις μονάδες που "επιβίωσαν" από τη Dropout.



Εικόνα 16: Αριστερά το βασικό δίκτυο, δεξιά όλα τα 16 πιθανά υπό-δίκτυα που μπορούν να σχηματιστούν, ως αποτέλεσμα της Dropout, αφαιρώντας διαφορετικά υποσύνολα νευρώνων. (Ian Goodfellow 2016)

Για την εκπαίδευση με Dropout, χρησιμοποιείται ένας αλγόριθμος που να κάνει μικρά βήματα, όπως ο gradient descent. Κάθε φορά που πρόκειται να εισαχθεί ένα στιγμιότυπο εκπαίδευσης, επιλέγεται τυχαία μία δυαδική μάσκα, ώστε να εφαρμοστεί σε όλους τους νευρώνες εισόδου και τους κρυφούς νευρώνες. Η πιθανότητα να επιλεγεί μία τιμή μάσκας της τάξης του ένα (κάτι που είναι αιτία να συμπεριληφθεί ο νευρώνας στην εμπρόσθια διάδοση για το συγκεκριμένο στιγμιότυπο), είναι μία υπέρ-παράμετρος η οποία ορίζεται πριν ξεκινήσει η εκπαίδευση, και όπως γράφτηκε παραπάνω, για τους κρυφούς νευρώνες είναι

συνήθως 0,5 και για τους νευρώνες εισόδου, 0,8. Έτσι γίνεται η εμπρόσθια διάδοση, στη συνέχεια η οπισθοδιάδοση σφάλματος και ανανεώνονται οι παράμετροι, όπως συνήθως.

Όπως περιεγράφηκε στο Κεφάλαιο 2.3, οι ενεργοποιήσεις που εξάγουν οι νευρώνες ενός επιπέδου l, του δικτύου, συνδέονται με τις ενεργοποιήσεις του προηγούμενου επιπέδου βάσει της σχέσης $a^l = f(w^l * a^{l-1} + b^l)$, όπου f είναι οποιαδήποτε συνάρτηση ενεργοποίησης. Με τη Dropout, οι νευρώνες του επιπέδου l, θα εξάγουν το διάνυσμα ενεργοποιήσεων a^l:

$$\begin{aligned} r^{l-1} &\sim Bernoulli\ (p) \\ \tilde{a}^{(l-1)} &= r^{(l-1)} \circ a^{(l-1)} \\ a^l &= f(w^l \tilde{a}^{(l-1)} + b^l) \end{aligned} \qquad \textbf{2.10.5}$$

Εδώ το σύμβολο ∘ υποδηλώνει το προϊόν Hadamard. Για κάθε επίπεδο l, $r^{(l)}$ είναι ένα διάνυσμα από ανεξάρτητες Bernoulli τυχαίες μεταβλητές, κάθε μία από τις οποίες έχει πιθανότητα p να είναι 1. Αυτό το διάνυσμα πολλαπλασιάζεται, ανά στοιχείο, με τις ενεργοποιήσεις εξόδου αυτού του επιπέδου, $a^{(l)}$, για να δημιουργηθούν οι "αραιωμένες" έξοδοι $\tilde{a}^{(l)}$. Οι "αραιωμένες" ενεργοποιήσεις εξόδου χρησιμοποιούνται μετά ως είσοδοι για να παραχθούν οι ενεργοποιήσεις του επόμενου επιπέδου. Αυτή η διαδικασία εφαρμόζεται σε κάθε επίπεδο. Αυτό ισοδυναμεί με δειγματοληψία ενός υπό-δικτύου από το βασικό δίκτυο. Για τις ανανεώσεις των παραμέτρων, οι παράγωγοι της συνάρτησης κόστους οπισθοδιαδίδονται μέσω του υπό-δικτύου. Κατά τη φάση εφαρμογής του δικτύου σε νέες τιμές εισόδου, τα μεγέθη των βαρών τροποποιούνται, δηλαδή $W^{(l)}_{new\_data} = pW^{(l)}$. Το νευρωνικό δίκτυο που προκύπτει χρησιμοποιείται χωρίς Dropout. (Aston Zhang 2021)

2.11 Αλγόριθμοι Βελτιστοποίησης

Στο συγκεκριμένο κεφάλαιο, η οποιαδήποτε παράμετρος, είτε βάρος (συμβολιζόταν προηγουμένως με w), είτε πόλωση (συμβολιζόταν προηγουμένως με b), συμβολίζεται ως θ και το σύνολο όλων των παραμέτρων σε μία κατάσταση του δικτύου συμβολίζεται ως Θ. Στο Κεφάλαιο 2.5 παρουσιάστηκε ο αλγόριθμος βελτιστοποίησης gradient descent. Όπως περιεγράφηκε, ο gradient descent είναι μία μέθοδος ελαχιστοποίησης μίας συνάρτησης κόστους $C(\Theta)$, παραμετροποιημένης από τα βάρη και τις πολώσεις, ανανεώνοντας την κάθε παράμετρο στην αντίθετη κατεύθυνση της κλίσης (προς την κατεύθυνση της αρνητικής κλίσης, αφότου η κλίση υποδεικνύει την κατεύθυνση της αύξησης) της συνάρτησης κόστους $\nabla_\theta C(\Theta)$ ως προς

την αντίστοιχη παράμετρο. Ο βαθμός μάθησης η, προσδιορίζει το μέγεθος του βήματος που πραγματοποιείται, με σκοπό το φθάσιμο στο ολικό ελάχιστο. Η διαδικασία αυτή μπορεί να οπτικοποιηθεί ως ακολούθηση κατηφορικής κατεύθυνσης, στη $(n+1)$-διάστατη επιφάνεια που δημιουργείται από τη συνάρτηση κόστους και τις παραμέτρους, όπου n είναι ο αριθμός των παραμέτρων, συν μία διάσταση για τη συνάρτηση κόστους.

Ο gradient descent (ή αλλιώς vanilla gradient descent ή batch gradient descent), έχει ένα χαρακτηριστικό, εξαιτίας του οποίου επιλέγεται πολύ σπάνια πιά. Όπως φαίνεται, η τετραγωνική συνάρτηση κόστους $C(\Theta)=\frac{1}{2m}\sum_{i=1}^{m}\left(h_\Theta\left(x^{(i)}\right)-y^{(i)}\right)^2$ υπολογίζει ένα μέσο όρο των κοστών $C(\Theta)=\frac{\left(h_\Theta\left(x^{(i)}\right)-y^{(i)}\right)^2}{2}$ όλων των στιγμιότυπων εκπαίδευσης $(x^{(i)},y^{(i)})$. Για να υπολογιστεί, λοιπόν, η συνολική κλίση $\nabla_\theta C(\Theta)$, πρέπει να υπολογιστούν πρώτα ξεχωριστά όλα τα $\nabla_\theta C(\Theta,x(i),y(i))$ και στη συνέχεια να υπολογιστεί ο μέσος όρος τους, $\nabla_\theta C(\Theta)$=$\frac{1}{m}\sum_{i=1}^{m}\nabla_\theta C(\Theta,x(i),y(i))$, όπου m ο αριθμός των στιγμιότυπων εκπαίδευσης. Για να γίνει μία ανανέωση των παραμέτρων, πρέπει να υπολογιστούν οι κλίσεις για το συνολικό σετ δεδομένων εκπαίδευσης. Φαίνεται, λοιπόν, ότι ο αλγόριθμος αυτός μπορεί να γίνει πολύ αργός, ειδικά όταν ο αριθμός των δεδομένων εισόδου είναι μεγάλος, και έτσι η διαδικασία της μάθησης να γίνει πολύ χρονοβόρα.

Για να αντιμετωπιστεί το πρόβλημα της αργής σύγκλισης και να επιταχυνθεί η μάθηση, με αποδοτικότερη ανανέωση παραμέτρων, επινοήθηκε μία παραλλαγή του αλγόριθμου gradient descent, που καλείται αλγόριθμος stochastic gradient descent (SGD) (Herbert Robbins 1951). Ο συγκεκριμένος αλγόριθμος στηρίζεται πάνω στην ιδέα ότι τα χαρακτηριστικά ενός συνόλου n στιγμιότυπων εκπαίδευσης, πιθανότατα περιγράφουν-αντιπροσωπεύουν επαρκώς τα χαρακτηριστικά του συνόλου των στιγμιότυπων εκπαίδευσης, όλου του σετ των δεδομένων εκπαίδευσης, δηλαδή. Ο αλγόριθμος SGD, λοιπόν, επιλέγει μία μικρή ομάδα (mini-batch), n τυχαίων στιγμιότυπων εκπαίδευσης, και εκτιμά τη συνολική κλίση της συνάρτησης κόστους $\nabla_\theta C(\Theta)$, υπολογίζοντας την κλίση της συνάρτησης κόστους $\nabla_\theta C(\Theta,x(q),y(q))$ (όπου $q=1:n$), για τις τιμές εισόδου και εξόδου αυτού του mini batch στιγμιότυπων εκπαίδευσης:

$$\nabla_\theta C(\Theta)=\frac{\sum_{i=1}^{m}\nabla_\theta C(\Theta,x(i),y(i))}{m}\approx\frac{\sum_{q=1}^{n}\nabla_\theta C(\Theta,x(q),y(q))}{n} \qquad 2.11.1$$

Ο SGD, λοιπόν, κάνει μία ανανέωση για κάθε παράμετρο, υπολογιζόμενη ως προς ένα mini-batch στιγμιότυπων εκπαίδευσης, και όχι το συνολικό σετ δεδομένων. Ο κανόνας ανανέωσης βαρών και πολώσεων, γίνεται:

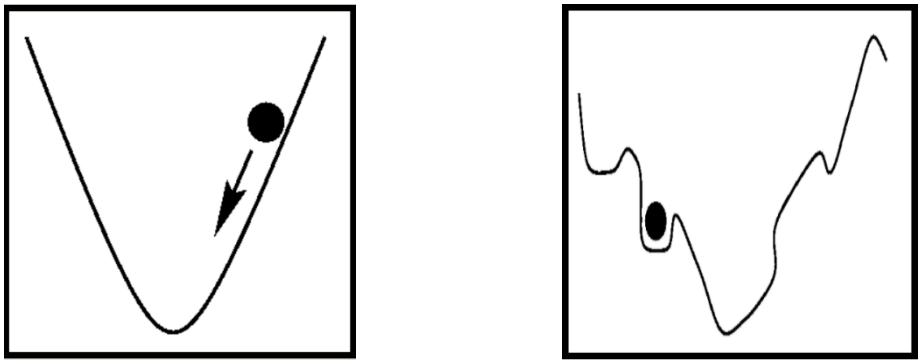
$$\theta \rightarrow \theta' = \theta - \frac{\eta}{n}\sum_{q=1}^{n}\frac{\partial C(\Theta, x(q), y(q))}{\partial \theta} \qquad \textbf{2.11.2}$$

Εδώ, το άθροισμα περιλαμβάνει όλα τα n στιγμιότυπα εκπαίδευσης του mini batch. Ο αλγόριθμος, λοιπόν, κάνει την ίδια διαδικασία για κάθε mini-batch και όταν φτάσει να έχει κάνει τόσες ανανεώσεις για κάθε παράμετρο, ώστε ανά mini-batch, να έχει χρησιμοποιηθεί το συνολικό σετ δεδομένων, τότε έχει ολοκληρωθεί μία εποχή εκπαίδευσης (training epoch). Σχετικά με τον αριθμό στιγμιότυπων του mini-batch, συνήθως επιλέγονται δυνάμεις του δύο, κάτι το οποίο σχετίζεται με την αντιστοίχιση των εικονικών επεξεργαστών, στους φυσικούς επεξεργαστές.

Ο SGD παραμένει έως και σήμερα πολύ δημοφιλής, αλλά η μάθηση με αυτόν, συχνά μπορεί να γίνει αρκετά αργή. Όπως περιεγράφηκε προηγουμένως, κατά τη διαδικασία της βελτιστοποίησης, ορίζεται μία επιφάνεια σε ένα (n+1)-διάστατο χώρο, από τη συνάρτηση κόστους και τις n παραμέτρους του μοντέλου. Ξεκινώντας από μία θέση βάσει της αρχικοποίησης των παραμέτρων, επαναληπτικά γίνονται κινήσεις ανανεώνοντας τις παραμέτρους, με στόχο να βρεθεί ολικό ελάχιστο. Οι επιφάνειες αυτές, όμως, είναι αρκετά πολύπλοκες, συχνά με αρκετά τοπικά ελάχιστα, στα οποία ο SGD, φαίνεται επιρρεπής στο να παγιδευτεί. Ακόμα, σε περιοχές που ομοιάζουν με "μεγάλες ρηχές χαράδρες, με απότομα τοιχώματα στα πλάγια", ο κλασσικός SGD θα τείνει να ταλαντεύεται κατά μήκος της στενής χαράδρας, αφού η αρνητική κλίση θα τον κατευθύνει σε μια από τις απότομες πλευρές και όχι κατά μήκος της χαράδρας προς το βέλτιστο. Οι συναρτήσεις κόστους βαθιών αρχιτεκτονικών, έχουν τέτοια μορφή γύρω από τοπικά βέλτιστα, δηλαδή περιοχές όπου η επιφάνεια καμπυλώνει πολύ πιο απότομα σε μια διάσταση παρά σε μια άλλη, με αποτέλεσμα ο SGD να συγκλείνει πολύ αργά. Όπως φαίνεται στην Εικόνα 17, στην περίπτωση της (α), ο αλγόριθμος θα κινηθεί στην αντίθετη κατεύθυνση από αυτή στην οποία ο ρυθμός μεταβολής της συνάρτησης κόστους είναι μέγιστος, ως προς κάθε παράμετρο, και θα πραγματοποιήσει ένα βήμα προς τη σύγκλειση. Στην περίπτωση της (β), όμως, ο SGD, έχει παγιδευτεί, και αυτό που θα κάνει είναι να ταλαντευτεί από τη μία πλευρά του κοίλου, στην άλλη, και τελικά να κινηθεί προς το βαθύτερο σημείο του τοπικού ελαχίστου. (Genevieve Orr n.d.)

Βγαίνει λοιπόν το ασφαλές συμπέρασμα, ότι στη θέση που βρίσκεται παγιδευμένος ο αλγόριθμος, στην περίπτωση (β) της Εικόνας 17, βελτίωση είναι δυνατόν να υπάρξει μόνο αν "ανηφορίσει" υψηλότερα, πριν κατευθυνθεί προς το ολικό ελάχιστο. Η μέθοδος του momentum (B.T.Polyak 1964) εισήχθη ώστε να επιταχυνθεί η μάθηση, ειδικά σε περιπτώσεις υψηλής καμπυλότητας, μικρών αλλά σταθερών κλίσεων και κλίσεων με θόρυβο, της συνάρτησης κόστους. Η μέθοδος momentum, αθροίζει έναν εκθετικά μειούμενο κινούμενο μέσο των προηγούμενων κλίσεων και συνεχίζει να κινείται σε αυτή την κατεύθυνση. Συγκεκριμένα, η μέθοδος αυτή, εισάγει μία μεταβλητή v, η οποία εκπροσωπεί τη διανυσματική ποσότητα του ρυθμού

μεταβολής της θέσης, στη μονάδα του χρόνου, δηλαδή την ταχύτητα (και τη διεύθυνση) που κινούνται οι παράμετροι εντός του χώρου των παραμέτρων, κατά τη διαδικασία βελτιστοποίησης.



(α) (β)

Εικόνα 17: Περιπτώσεις κατά τη διαδικασία ελαχιστοποίησης της συνάρτησης κόστους με αλγόριθμο βελτιστοποίησης, όπως ο SGD. (Genevieve Orr n.d.)

Η συγκεκριμένη μέθοδος, λοιπόν, συσσωρεύει ένα διάνυσμα ταχύτητας, στις κατευθύνσεις συνεχούς μείωσης της συνάρτησης κόστους, κατά τις επαναλήψεις. Η κλίση της συνάρτησης κόστους, ενεργεί σαν να αλλάζει τον ρυθμό μεταβολής της θέσης - την ταχύτητα και όχι απευθείας τη θέση, όπως και οι φυσικές δυνάμεις αλλάζουν την ταχύτητα και εμμέσως επιδρούν στην θέση. Ακόμα, η μέθοδος momentum εισάγει έναν όρο που λειτουργεί σαν κάποιου είδους τριβή, ο οποίος τείνει βαθμιαία να μειώνει την ταχύτητα. Εισηγμένων κάποιων μεταβλητών ταχύτητας $v = v_1, v_2, \dots, v_j$, μία για κάθε αντίστοιχη παράμετρο θ (είτε βάρος, είτε πόλωση) του δικτύου, ο κανόνας ανανέωσης των παραμέτρων, βάσει του αλγόριθμου SGD με momentum, είναι:

$$v \to v' = \mu v - \frac{\eta}{n}\sum_{q=1}^{n} \nabla_\theta C(\Theta, x(q), y(q)) \qquad \textbf{2.11.3}$$

$$\theta \to \theta' = \theta + v' \qquad \textbf{2.11.4}$$

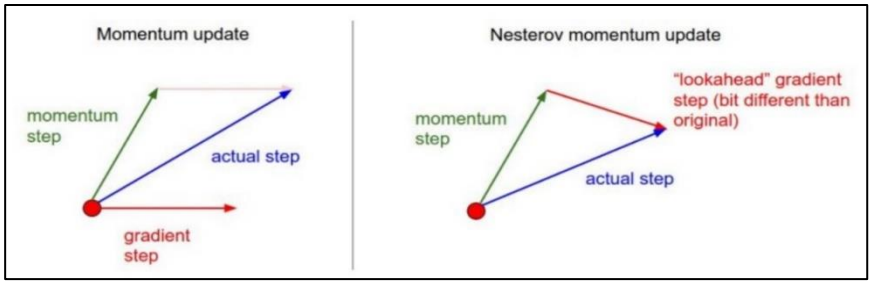
"Ο όρος momentum αυξάνει για διαστάσεις, των οποίων οι κλίσεις δείχνουν προς τις ίδιες κατευθύνσεις και μειώνει τις ανανεώσεις για διαστάσεις των οποίων οι κλίσεις αλλάζουν κατευθύνσεις" (Ruder 2017). Στην Εξίσωση 2.11.4, ο όρος μ είναι μία υπέρ-παράμετρος, η οποία καλείται όρος momentum. Ο όρος momentum ελέγχει την ποσότητα της τριβής στο σύστημα, υπό την έννοια ότι προσδιορίζει πόσο γρήγορα οι συνεισφορές των προτέρων κλίσεων μειώνονται εκθετικά, έως ότου να

εξαφανιστούν, δηλαδή για πόσες επαναλήψεις, οι πρότερες κλίσεις λαμβάνονται υπόψη στην τρέχουσα ανανέωση. Στην περίπτωση όπου μ=0, η Εξίσωση 2.11.4 παίρνει τη μορφή του κλασσικού SGD.

Μία άλλη μέθοδος η οποία πρόσφατα έχει δείξει ενθαρρυντικά αποτελέσματα, είναι η επιταχυμένη κλίση Nesterov (Nesterov's Accelerated Gradient – NAG) ή αλλιώς Nesterov momentum (Nesterov 1983).*"Όπως η μέθοδος momentum, η Nesterov momentum είναι μία πρώτης τάξης μέθοδος βελτιστοποίησης, η οποία εξασφαλίζει καλύτερο ρυθμό σύγκλισης συγκριτικά με το gradient descent, σε συγκεκριμένες καταστάσεις. Ενόσω η μέθοδος Nesterov momentum δε μπορεί να θεωρηθεί τυπικά ως ένα είδος momentum, στην πραγματικότητα αποδεικνύεται ότι σχετίζεται άμεσα με την κλασσική μέθοδο momentum, διαφέροντας μόνο στην ανανέωση του διανύσματος ταχύτητας v."* (Ilya Sutskever 2013). Ο κανόνας ανανέωσης παραμέτρων του SGD, με τη μέθοδο Nesterov momentum, είναι:

$$v \to v' = \mu v - \frac{\eta}{n}\sum_{q=1}^{n} \nabla_\theta C(\Theta + \eta v, x(q), y(q)) \qquad \textbf{2.11.5}$$
$$\theta \to \theta' = \theta + v'$$

Όπου n είναι ο αριθμός των στιγμιότυπων του mini-batch και οι υπέρ-παράμετροι μ και η, έχουν την ίδια λειτουργία όπως στην κλασσική μέθοδο momentum. Η διαφορά μεταξύ της κλασσικής momentum και της Nesterov momentum, είναι στο που υπολογίζεται η κλίση της συνάρτησης κόστους. Με τη μέθοδο Nesterov, η κλίση υπολογίζεται αφότου έχει εφαρμοστεί η τρέχουσα ταχύτητα. Το χαρακτηριστικό αυτό, επιτρέπει στη Nesterov, να αλλάζει το v με ένα ταχύτερο και πιο "ευαίσθητο" τρόπο, αφήνοντας το να συμπεριφέρεται πιο ευσταθώς, συγκριτικά με την περίπτωση της κλασσικής momentum, σε πολλές καταστάσεις, ειδικά για υψηλότερες τιμές του μ. Έτσι, η Nesterov momentum, μπορεί να θεωρηθεί ως προσπάθεια να εισαχθεί ένας διορθωτικός συντελεστής στην κλασσική momentum.



Εικόνα 18: Οπτικοποίηση ανανεώσεων κλασσικού momentum και Nesterov momentum. Με τη Nesterov momentum, αντί να υπολογιστεί η κλίση στην τρέχουσα θέση (κόκκινος κύκλος), γνωρίζοντας ότι το momentum πρόκειται να μας μεταφέρει στην αιχμή του πράσινου βέλους, γίνεται υπολογισμός της κλίσης σε αυτή τη θέση, στη θέση που φαίνεται μπροστά δηλαδή. (Fei-Fei Li 2020)

Ο πιο δύσκολος, από τους ορισμούς των υπέρ-παραμέτρων, είναι, ίσως, αυτός του βαθμού μάθησης, αφού έχει τεράστια επίδραση στη συνολική απόδοση του μοντέλου. Όπως περιεγράφηκε παραπάνω, η συνάρτηση κόστους είναι αρκετά "ευαίσθητη" σε κάποιες κατευθύνσεις στον χώρο των παραμέτρων, ενώ σε κάποιες άλλες δεν είναι σχεδόν καθόλου. Ουσιαστικά, οι "κατευθύνσεις ευαισθησίας" αυτές είναι ευθυγραμμισμένες αξονικά, με τους άξονες που ορίζονται από την κάθε παράμετρο, στον χώρο των παραμέτρων. Έτσι, λοιπόν, βγαίνει το συμπέρασμα ότι η χρήση διαφορετικού βαθμού μάθησης για κάθε παράμετρο και η προσαρμογή αυτών των βαθμών μάθησης μέσω της εκπαίδευσης του μοντέλου, μπορεί να βελτιώσει την απόδοση συνολικά. Αυτό πραγματοποιούν οι αλγόριθμοι βελτιστοποίησης προσαρμοστικών βαθμών μάθησης (adaptive learning rate optimization algorithms), μεταχειρίζονται δηλαδή την κάθε παράμετρο ξεχωριστά και δε χρησιμοποιούν ένα καθολικό βαθμό μάθησης, όπως οι μέθοδοι που παρουσιάστηκαν παραπάνω.

Ένας αλγόριθμος τέτοιου είδους, που προτάθηκε για πρώτη φορά, και ο οποίος μπορεί να εφαρμοστεί σε mini-batches δεδομένων, είναι ο AdaGrad (John Duchi 2011). Ο συγκεκριμένος αλγόριθμος ανήκει και αυτός στην οικογένεια των gradient descent, αλλά προσαρμόζει ξεχωριστά το βαθμό μάθησης κάθε παραμέτρου του μοντέλου, αυξάνοντας τον ή μειώνοντας τον, αντιστρόφως ανάλογα με την τετραγωνική ρίζα του αθροίσματος όλων των παλαιότερων τετραγωνισμένων τιμών της αντίστοιχης παραμέτρου. Οι παράμετροι με τις μεγαλύτερες μερικές παράγωγους της συνάρτησης κόστους, έχουν αντίστοιχα μεγάλη μείωση του ρυθμού μάθησης τους, ενώ παράμετροι με μικρές μερικές παράγωγους, έχουν συγκριτικά μικρότερη μείωση του ρυθμού μάθησης τους. Ας θεωρηθεί, λοιπόν, ότι $g_{t,i}$ είναι η μερική παράγωγος της συνάρτησης κόστους, ως προς την όποια παράμετρο θ_i, σε χρονικό βήμα t, δηλαδή:

$$g_{t,i} = \nabla_{\theta_{t,i}} C(\Theta_t) \qquad \textbf{2.11.6}$$

Η κανόνας ανανέωσης του SGD, για κάθε παράμετρο θ_i, σε χρονικό βήμα t, είναι:

$$\theta_{t+1,i} = \theta_{t,i} - \eta * g_{t,i} \qquad \textbf{2.11.7}$$

Βάσει του κανόνα ανανέωσης του, ο AdaGrad τροποποιεί το γενικό βαθμό εκμάθησης η, σε κάθε χρονικό βήμα t, για κάθε παράμετρο θ_i, βασιζόμενος στις προηγούμενες κλίσεις που είχαν υπολογιστεί για την παράμετρο θ_i με αποτέλεσμα να ισχύει:

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} * g_{t,i} \qquad \textbf{2.11.8}$$

Όπου $G_t \in \mathbb{R}^{d \times d}$ είναι ένας διαγώνιος πίνακας όπου κάθε στοιχείο της διαγώνιού του, i, i είναι το άθροισμα των τετραγώνων των κλίσεων ως προς θ_i, σε χρονικό βήμα t, ενόσω ο ϵ είναι ένας εξομαλυντικός όρος που αποσοβεί τη διαίρεση με το μηδέν (συνήθως της τάξης του 1e-8). Οι Duchi et al., παρέχουν αυτόν τον πίνακα ως εναλλακτική του πλήρη πίνακα που περιέχει τα εξωτερικά γινόμενα όλων των προηγούμενων κλίσεων, αφού ο υπολογισμός της τετραγωνικής ρίζας πίνακα είναι ανέφικτος ακόμα και για ένα μικρό αριθμό παραμέτρων. Έχει ενδιαφέρον ότι χωρίς την τετραγωνική ρίζα, ο αλγόριθμος αποδίδει πολύ χειρότερα. (Ruder 2017)

Ένας άλλος αλγόριθμος προσαρμοστικών βαθμών μάθησης, και ίσως ο πλέον διαδεδομένος σήμερα, είναι ο Adam (Adaptive Moment Estimation) (Diederik P. Kingma 2015). Το βασικό χαρακτηριστικό του συγκεκριμένου αλγόριθμου, είναι ότι όχι μόνο αποθηκεύει έναν εκθετικά μειούμενο μέσο των παλαιότερων κλίσεων στο τετράγωνο (όπως κάνουν άλλοι αλγόριθμοι, π.χ. ο Adadelta, ο RMSProp), u_t, αλλά αποθηκεύει και έναν εκθετικά μειούμενο μέσο των παλαιότερων κλίσεων, m_t, παρόμοια με τη μέθοδο momentum, που περιεγράφηκε παραπάνω:

$$g_t = \nabla_\theta C_t(\theta)$$
$$m_t \leftarrow \beta_1 * m_{t-1} + (1 - \beta_1) * g_t \qquad \textbf{2.11.9}$$
$$u_t \leftarrow \beta_2 * u_{t-1} + (1 - \beta_2) * g_t^2$$

Όπου g_t, υποδηλώνει την κλίση, δηλαδή το διάνυσμα των μερικών παραγώγων της συνάρτησης κόστους C_t, ως προς θ, εκτιμημένη σε χρονικό βήμα t. Ο αλγόριθμος ανανεώνει εκθετικούς κινούμενους μέσους της κλίσης (m_t) και της τετραγωνισμένης κλίσης (u_t), όπου οι υπέρ-παράμετροι $\beta_1, \beta_2 \in [0,1)$ ελέγχουν τους ρυθμούς εκθετικής μείωσης αυτών των κινούμενων μέσων. Οι κινούμενοι μέσοι αυτοί, είναι εκτιμήσεις της πρώτης βασικής ποσότητας(ο μέσος όρος) και της δεύτερης βασικής ποσότητας (μη κεντραρισμένη μεταβλητότητα). *"Παρ'όλα αυτά, αυτοί οι κινητοί μέσοι είναι αρχικοποιημένοι (ως διανύσματα) με 0, οδηγώντας σε εκτιμήσεις των βασικών ποσοτήτων που έχουν πόλωση προς το μηδέν, ειδικά κατά τη διάρκεια των αρχικών βημάτων, και ειδικά όταν οι ρυθμοί μείωσης είναι μικροί (π.χ. τα β_s είναι κοντά στο 1). Τα καλά νέα είναι ότι αυτή η πόλωση αρχικοποίησης μπορεί εύκολα να αντιμετωπιστεί, οδηγώντας σε διορθωμένες από πλευράς πόλωσης εκτιμήσεις $\hat{m}_t$ και $\hat{u}_t$"* (Diederik P. Kingma 2015):

$$\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$$
$$\hat{u}_t \leftarrow u_t / (1 - \beta_2^t) \qquad \textbf{2.11.10}$$

Ο κανόνας ανανέωσης των παραμέτρων, βάσει του αλγόριθμου Adam, είναι:

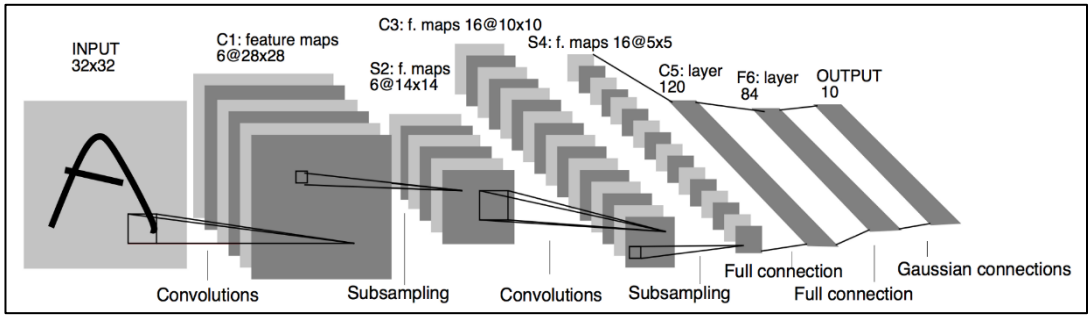
$$\theta_t \leftarrow \theta_{t-1} - \alpha * \hat{m}_t / (\sqrt{\hat{u}_t} + \epsilon) \qquad \textbf{2.11.11}$$

3 Συνελικτικά Νευρωνικά Δίκτυα

3.1 Εισαγωγή

Τα συνελικτικά νευρωνικά δίκτυα (convolutional neural networks – CNNs) είναι ένα εξειδικευμένο είδος τεχνητών νευρωνικών δικτύων. Χρησιμοποιούνται για επεξεργασία δεδομένων τα οποία έχουν πλεγματική δομή. Τέτοιου είδους δεδομένα, π.χ., είναι οι χρονολογικές σειρές, οι οποίες μπορούν να θεωρηθούν ως μονοδιάστατα πλέγματα φατνίων, τα οποία περιέχουν τιμές που λαμβάνοντα δειγματοληπτικά, ανά τακτά χρονικά διαστήματα και οι εικόνες, οι οποίες είναι δυσδιάστατα πλέγματα εικονοστοιχείων. Η ονομασία τους, "Συνελικτικά Νευρωνικά Δίκτυα", προέρχεται από το όνομα της βασικής μαθηματικής πράξης, την οποία εφαρμόζουν, τη συνέλιξη.

Η απαρχή των CNNs έγινε με το Neocognitron (Fukushima 1980), το πρώτο, ίσως, τεχνητό νευρωνικό δίκτυο το οποίο άξιζε το χαρακτηρισμό "βαθύ". Το Neocognitron, το οποίο ουσιαστικά εισήγαγε τα CNNs, είναι ένα ιεραρχικό, πολλών επιπέδων, τεχνητό νευρωνικό δίκτυο το οποίο κατέχει τη δυνατότητα αναγνώρισης οπτικών μοτίβων. Χρησιμοποιήθηκε για την αναγνώριση χειρόγραφων χαρακτήρων, αλλά και σε άλλες εργασίες αναγνώρισης μοτίβων. Η δυνατότητα γενίκευσης, της αναγνώρισης των μοτίβων, επετεύχθη βάσει των γεωμετρικών ομοιοτήτων των σχημάτων τους, χωρίς το σύστημα να επηρεάζεται από τις θέσεις τους. Η "επανάσταση", όμως, έγινε με δύο



Εικόνα 19:Αρχιτεκτονική LeNet. (Y. Lecun 1998)

εργασίες των LeCun et al. Στη μία εργασία (Y. LeCun 1989), για πρώτη φορά παρουσιάστηκε η χρήση του αλγόριθμου οπισθοδιάδοσης σφάλματος (backpropagation), σε πρακτική εφαρμογή, και υποστηρίχτηκε ότι η ικανότητα ενός δικτύου να μάθει να γενικεύει, μπορεί να βελτιωθεί σημαντικά βάζοντας περιορισμούς από τον τομέα εργασιών. Σχεδίασε ένα CNN, και το εκπαίδευσε με αλγόριθμους backpropagation, ώστε να αναγνωρίζει χειρόγραφους αριθμούς και το

εφάρμοσε, με επιτυχία, για αναγνώριση χειρόγραφων ταχυδρομικών κωδίκων, με δεδομένα που του παρείχε η ταχυδρομική υπηρεσία των ΗΠΑ. Αυτή η εργασία, ήταν ο προπομπός αυτού που κατόπιν θα ονομαζόταν LeNet (Y. Lecun 1998).Η συγκεκριμένη εργασία ήταν η πρώτη πάνω σε σύγχρονα CNNs. Οι συγγραφείς της, υποστήριξαν ότι μία αρχιτεκτονική η οποία συγκεντρώνει απλά χαρακτηριστικά, σε προοδευτικά πιο περίπλοκα χαρακτηριστικά, και η οποία προφανώς δε θα απαρτίζονταν από μόνο ένα κρυφό επίπεδο, θα μπορούσε να χρησιμοποιηθεί με επιτυχία για αναγνώριση χειρόγραφων ψηφίων, όπως και έγινε τελικά. Το μοντέλο τους εκπαιδεύτηκε στο σετ δεδομένων MNIST, το οποίο περιέχει 60000 εικόνες χειρόγραφων ψηφίων, με την κάθε μία από αυτές να έχει ετικέτα ανάλογα με τον αριθμό που περιέχει. Επίσης, υποστήριξαν ότι η μείωση των παραμέτρων ενός μοντέλου, θα περιόριζε την πολυπλοκότητα του, άρα θα βελτίωνε την ικανότητα του να γενικεύει. Η υλοποίηση αυτή των LeCun et al., έθεσε τις βάσεις για τις εφαρμογές όρασης υπολογιστών και επεξεργασίας εικόνας του σήμερα.

3.2 Συνέλιξη

Όπως μαρτυρά και η ονομασία τους, “CNNs”, η πράξη της συνέλιξης (convolution), είναι ο θεμέλιος λίθος της λειτουργίας τους. Γενικά η συνέλιξη είναι μία πράξη μεταξύ δύο συναρτήσεων, πραγματικών τιμών. Ας θεωρήσουμε ότι ένας λέιζερ αισθητήρας καταγράφει τη θέση ενός διαστημόπλοιου, και σε κάθε χρονική στιγμή t, αποκρίνεται $\mathrm{x}(t)$, δηλαδή τη θέση του διαστημοπλοίου, όπου x, t, είναι δύο πραγματικές τιμές. Όπως συνήθως συμβαίνει με τους αισθητήρες, το σήμα εμπεριέχει θόρυβο, οπότε για να εκτιμηθεί η θέση, σε κάθε χρονική στιγμή, εμπεριέχοντας λιγότερο θόρυβο, μπορεί να υπολογιστεί ένας μέσος όρος από πολλαπλές αποκρίσεις. Επειδή οι πιο πρόσφατες μετρήσεις, σε κάθε υπολογισμό μέσου όρου, είναι πιο σχετικές, θα υπολογίζεται ένας σταθμισμένος μέσος όρος, ώστε να δίνεται μεγαλύτερο βάρος, σε αυτές κάθε φορά. Αυτό μπορεί να πραγματοποιηθεί χρησιμοποιώντας μία συνάρτηση στάθμισης $w(a)$, όπου a είναι η “ηλικία” της μέτρησης. Αν εφαρμοστεί μία πράξη σταθμισμένου μέσου όρου, σε κάθε χρονική στιγμή, προσδιορίζεται μία νέα συνάρτηση s, η οποία παρέχει μία ομαλοποιημένη εκτίμηση της θέσης του διαστημοπλοίου, σε κάθε χρονική στιγμή t:

$$s(t) = \int x(a)w(t-a)da \qquad \textbf{3.2.1}$$

Η πράξη αυτή καλείται συνέλιξη και συνήθως συμβολίζεται με ένα αστερίσκο:

$$s(t) = (x * w)(t) \qquad \textbf{3.2.2}$$

Στο παραπάνω εξειδικευμένο παράδειγμα, η w θα έπρεπε να είναι μία κατάλληλη συνάρτηση πυκνότητας πιθανότητας, αλλιώς οι τιμές εξόδου δε θα ήταν σταθμισμένοι μέσοι όροι. Επίσης, η w θα έπρεπε να γίνεται μηδέν, για οποιαδήποτε αρνητική παράμετρο, γιατί σε αντίθετη περίπτωση θα σήμαινε ότι "βλέπει" στο μέλλον. Αυτοί οι περιορισμοί ισχύουν μόνο για το συγκεκριμένο παράδειγμα, στην πραγματικότητα η συνέλιξη εφαρμόζεται σε οποιαδήποτε περίπτωση ορίζεται το παραπάνω ολοκλήρωμα και συνήθως χρησιμοποιείται και για άλλους σκοπούς, εκτός του υπολογισμού σταθμισμένων μέσων όρων.

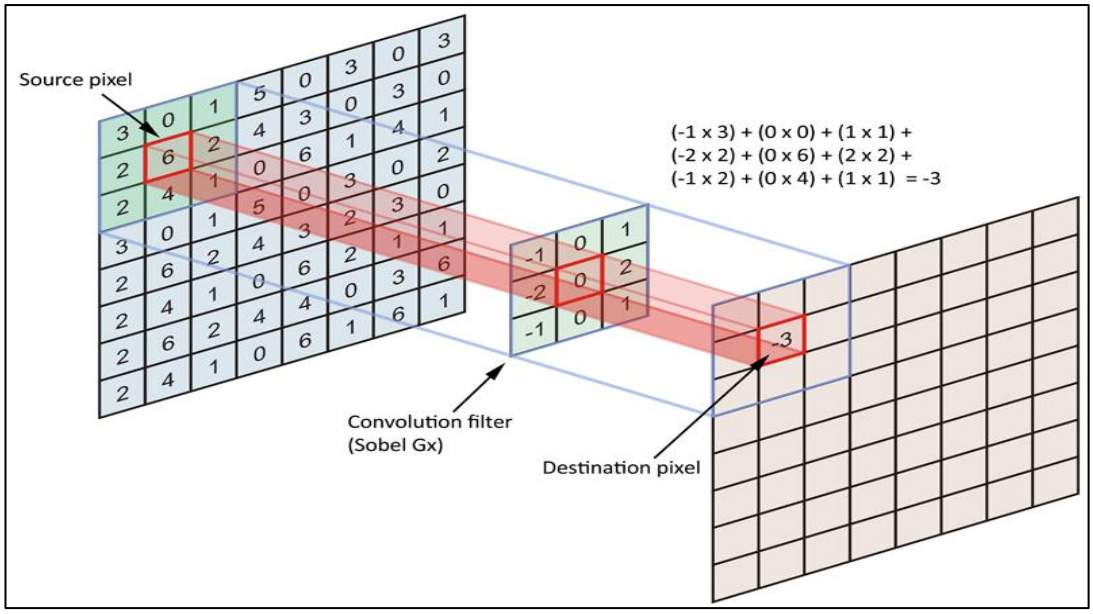
Η συνέλιξη χρησιμοποιείται για πλήθος εφαρμογών στην ψηφιακή επεξεργασία εικόνας. Αυτό σημαίνει ότι πολύ συχνά εφαρμόζεται σε περισσότερες από μία διαστάσεις την κάθε φορά, αφού η ψηφιακή εικόνα είναι ένας πολυδιάστατος πίνακας δεδομένων. Σε τέτοιες εφαρμογές, ένας τοπικός τελεστής (local operator) χρησιμοποιεί μία ομάδα τιμών εικονοστοιχείων, τις γειτονικές τιμές ενός συγκεκριμένου εικονοστοιχείου, για να υπολογίσει μία τιμή εξόδου, για τη θέση εφαρμογής αυτή. Ο τοπικός τελεστής, αυτός, είναι συνήθως ένας πίνακας, πολύ μικρότερος προφανώς σε διαστάσεις από την εικόνα, στης οποίας τις θέσεις εφαρμόζεται, ο οποίος καλείται φίλτρο ή kernel. Ο πιο συχνά χρησιμοποιούμενος τοπικός τελεστής είναι το γραμμικό φίλτρο, όπου η τιμή εξόδου του εικονοστοιχείου στο οποίο εφαρμόζεται, είναι ένα σταθμισμένο άθροισμα τιμών εικονοστοιχείων. Για παράδειγμα, στην περίπτωση δυσδιάστατης εικόνας I, στην οποία εφαρμόζεται δυσδιάστατο φίλτρο K, ισχύει:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n) \qquad \textbf{3.2.3}$$

Η συνέλιξη είναι αντιμεταθετική πράξη, ισχύει δηλαδή:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n) \qquad \textbf{3.2.4}$$

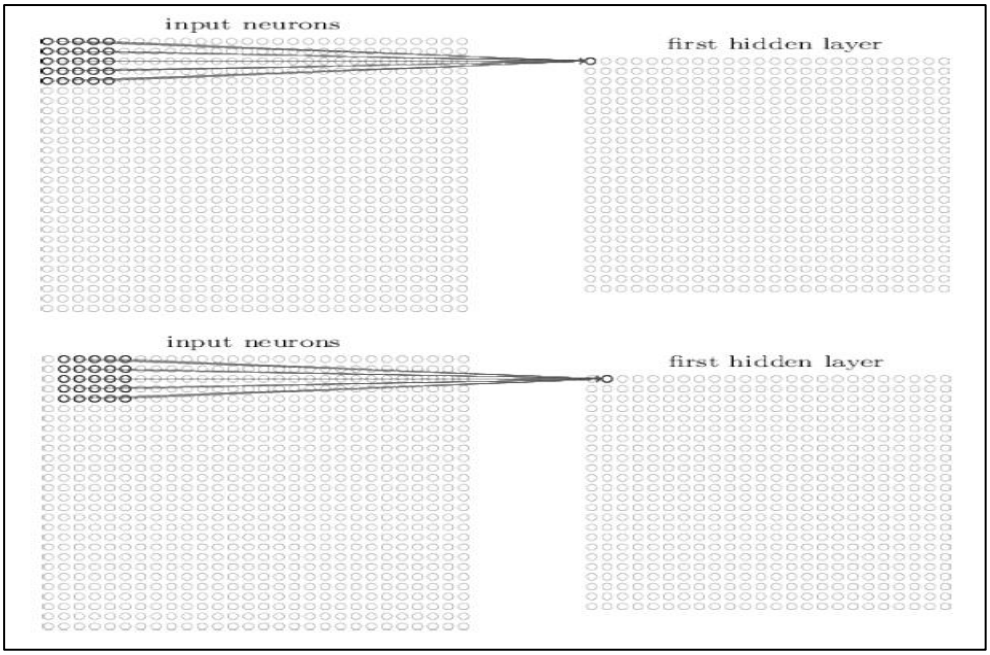
Το φίλτρο εφαρμόζεται στις διαφορετικές θέσεις της εικόνας, με μία διαδικασία η οποία ομοιάζει με την κίνηση ενός "παραθύρου που ολισθαίνει", κατά το εύρος του μήκους και του πλάτους της εικόνας. Η αρχική εικόνα στην οποία εφαρμόζεται το φίλτρο συνέλιξης, συχνά καλείται "χάρτης χαρακτηριστικών εισόδου" (input feature map) ή τανυστής εισόδου (input tensor) . Το αποτέλεσμα που προκύπτει μετά την εφαρμογή της συνέλιξης στις διαφορετικές θέσεις, συχνά καλείται "χάρτης χαρακτηριστικών εξόδου" (output feature map).



Εικόνα 20: Παράδειγμα εφαρμογής φίλτρου συνέλιξης. Ενώ ο χάρτης χαρακτηριστικών εισόδου, η αρχική εικόνα δηλαδή, είναι διαστάσεων 8x8, ο χάρτης χαρακτηριστικών εξόδου που προκύπτει, είναι διαστάσεων 6x6. Αυτό συμβαίνει γιατί μόνο στις αντίστοιχες θέσεις, της αρχικής εικόνας, "χώρεσε" και εφαρμόστηκε το φίλτρο συνέλιξης. Το συγκεκριμένο φίλτρο ονομάζεται φίλτρο Sobel, χρησιμοποιείται για την ανάδειξη ακμών και στη περίπτωση του συγκεκριμένου παραδείγματος, καθέτων ακμών.

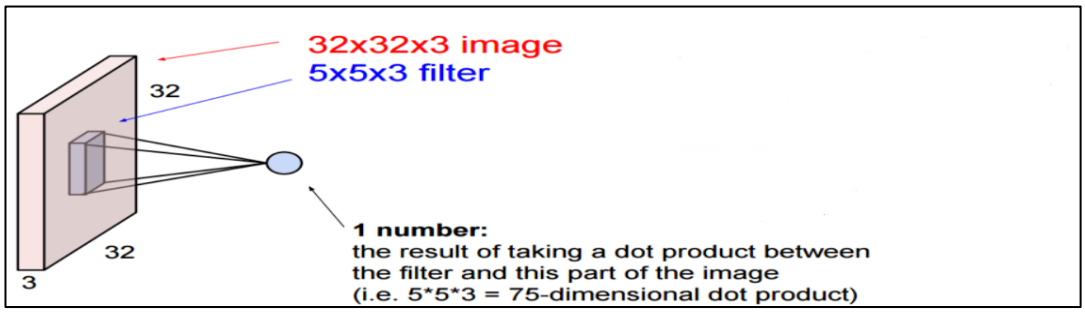
3.3 Θεμελιώδεις Επινοήσεις

Τα CNNs αξιοποιούν τρεις βασικές επινοήσεις, ως απόρροια της εφαρμογής συνελίξεων, οι οποίες μπορούν να κάνουν αποδοτικότερο ένα σύστημα μηχανικής μάθησης. Η πρώτη βασική επινόηση, είναι αυτή της τοπικής συνδεσιμότητας (local connectivity) ή αλλιώς των τοπικών δεκτικών πεδίων (local receptive fields). Όταν διαχειριζόμαστε υψηλών διαστάσεων εισόδους, όπως εικόνες, είναι μη πρακτικό να συνδέουμε νευρώνες ενός κρυφού επιπέδου, με όλους τους νευρώνες στο προηγούμενο επίπεδο, το οποίο πιθανώς να είναι το επίπεδο εισόδου, με τις ακατέργαστες τιμές εικονοστοιχείων. Αυτό που συμβαίνει, λοιπόν, είναι ότι ο κάθε νευρώνας του π.χ. του πρώτου κρυφού επιπέδου, "εστιάζει" σε ένα τοπικό τμήμα της εικόνας εισόδου. Το τοπικό τμήμα, αυτό, ονομάζεται τοπικό δεκτικό πεδίο (local receptive field) (ισοδύναμα αυτό είναι το μέγεθος του φίλτρου). Κάθε σύνδεση μαθαίνει ένα βάρος και κάθε κρυφός νευρώνας, μαθαίνει μία συνολική πόλωση. Κάθε κρυφός νευρώνας, κατά τη διαδικασία της μάθησης, μαθαίνει να αναλύει όλο και καλύτερα, το τοπικό δεκτικό πεδίο του. Για κάθε τοπικό δεκτικό πεδίο, λοιπόν, υπάρχει ένας διαφορετικός νευρώνας, στο πρώτο κρυφό επίπεδο.



Εικόνα 21: Τοπική συνδεσιμότητα. Παράδειγμα με επίπεδο εισόδου (εικόνα εισόδου) και πρώτο κρυφό επίπεδο. (Nielsen 2019)

Οι συνδεσιμότητες είναι τοπικές στο χώρο, κατά μήκος του πλάτους και του ύψους πχ της εικόνας εισόδου, αλλά πάντα πλήρεις κατά μήκος ολόκληρου του βάθους του "όγκου" (volume) εισόδου. Αυτό αναφέρεται στην περίπτωση όπου η εικόνα εισόδου είναι μία πχ RGB εικόνα, δηλαδή τρεις πίνακες, ένας για κάθε χρώμα, οι οποίοι δημιουργούν έναν όγκο. Ο όγκος αυτός έχει διαστάσεις, προφανώς, ίδιες με την εικόνα και βάθος τρία, ένα για κάθε χρώμα. Ο νευρώνας του πρώτου κρυφού επιπέδου θα εστιάσει σε μία τοπική περιοχή, σε όλος το εύρος του βάθους όμως, δηλαδή και στους τρεις πίνακες, στις ίδιες θέσεις.



Εικόνα 22: Παράδειγμα τοπικής συνδεσιμότητας, στην περίπτωση όπου η εικόνα εισόδου έχει βάθος μεγαλύτερο του ένα.

Η δεύτερη επινόηση, την οποία αξιοποιούν τα CNNs, είναι αυτό που καλείται "διαμοιρασμός παραμέτρων" (parameter sharing). Ο διαμοιρασμός παραμέτρων αναφέρεται στη χρήση της ίδια παραμέτρου, για περισσότερες, από μία,

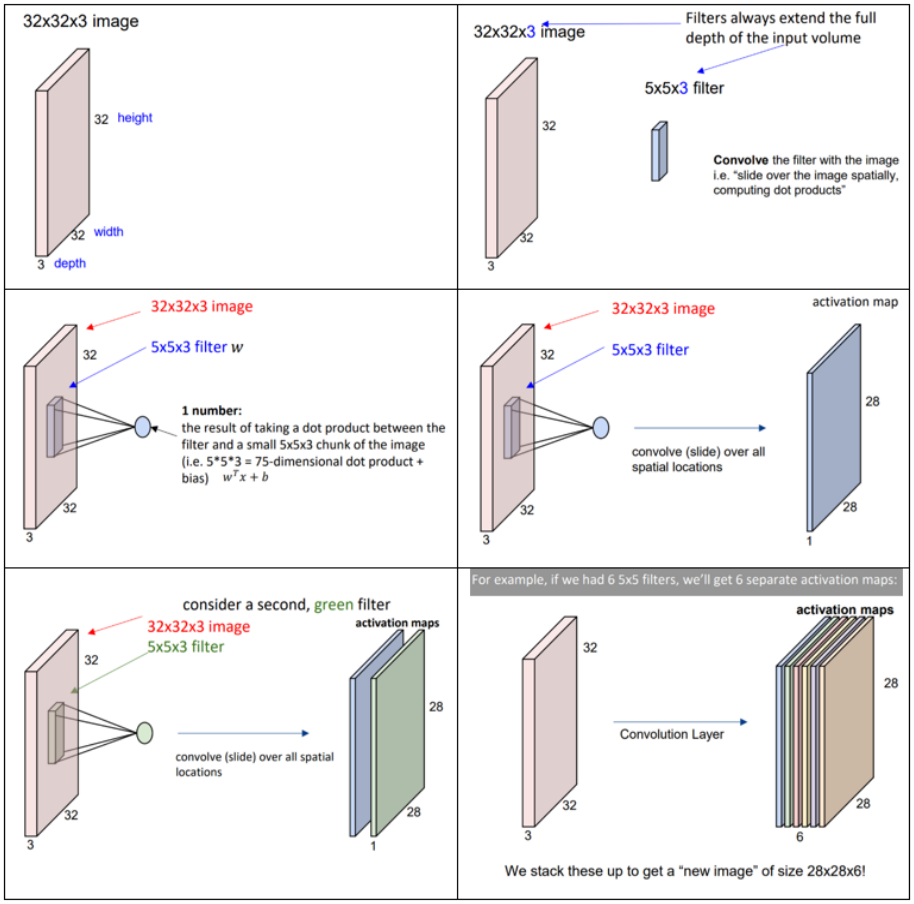
συναρτήσεις, στο ίδιο μοντέλο. Με αυτό τον τρόπο μειώνονται κατά πολύ οι παράμετροι του μοντέλου, και όσον αφορά την περίπτωση των CNNs, αυτό γίνεται απλά κάνοντας μία λογική υπόθεση, ότι, δηλαδή, αν ένα χαρακτηριστικό, είναι χρήσιμο να υπολογιστεί σε κάποια χωρική θέση (x_1, y_1), τότε θα είναι χρήσιμο να υπολογιστεί και σε μία διαφορετική θέση (x_2, y_2). Πρακτικά, αυτό σημαίνει ότι οι νευρώνες του πρώτου π.χ. κρυφού επιπέδου, που εστιάζουν ο καθένας σε ένα τοπικό τμήμα της εικόνας εισόδου, χρησιμοποιούν όλοι τις ίδιες παραμέτρους (βάρη και ο καθένας μία συνολική πόλωση), κάτι το οποίο έχει σαν αποτέλεσμα οι διαδοχικές εστιάσεις των νευρώνων, να ομοιάζουν με ένα παράθυρο που "σαρώνει" την εικόνα και εντοπίζει το ίδιο χαρακτηριστικό, απλά σε διαφορετικές θέσεις στην εικόνα εισόδου.

Ο διαμοιρασμός παραμέτρων, δηλαδή η εφαρμογή φίλτρων τα οποία είναι αμετάβλητα στη μετατόπιση (translation invariant), έχει σαν αποτέλεσμα ένα επίπεδο να είναι συμμετρικά μεταβαλλόμενο στη μετατόπιση (translation equivariant), κάτι το οποίο είναι η τρίτη βασική ιδέα πίσω από τα CNNs. Μία συνάρτηση καλείται συμμετρικά μεταβαλλόμενη στη μετατόπιση, αν σε περίπτωση που αλλάξει κατά μία ποσότητα η τιμή της μεταβλητής εισόδου, τότε αλλάξει κατά την ίδια ποσότητα και η τιμή της μεταβλητής εξόδου. Συγκεκριμένα, μία συνάρτηση $f(x)$, καλείται συμμετρικά μεταβαλλόμενη σε μία συνάρτηση g, αν ισχύει $f\big(g(x)\big) = g(f(x))$. Παραδείγματος χάριν, ας υποτεθεί συνάρτηση g, η οποία μετατοπίζει κάθε εικονοστοιχείο σε μία εικόνα I, κατά μία θέση δεξιότερα, δηλαδή $I' = g(I)$, με $I'(x, y) = I(x - 1, y)$. Αν εφαρμοστεί ο μετασχηματισμός g στην εικόνα I και στη συνέχεια συνέλιξη, το αποτέλεσμα θα είναι το ίδιο με την περίπτωση που εφαρμοζόταν συνέλιξη στην I' και στη συνέχεια εφαρμοζόταν ο μετασχηματισμός g, στην έξοδο της. Έτσι, λοιπόν, με τις εικόνες, όπου η συνέλιξη δημιουργεί ένα χάρτη χαρακτηριστικών εξόδου, βάσει του που εμφανίζονται συγκεκριμένα χαρακτηριστικά στον χάρτη χαρακτηριστικών εισόδου, στην εικόνα, δηλαδή, αν μετακινηθούν συνολικά τα εικονοστοιχεία της εικόνας, και άρα ένα οποιοδήποτε αντικείμενο που εμφανίζεται σε αυτό, κατά την ίδια ποσότητα θα μετακινηθεί η αντίστοιχη "αναπαράσταση" του, στον χάρτη χαρακτηριστικών εξόδου.

3.4 Βάθος - Βήμα (Stride) – Πρόσθεση Μηδενικών (Zero Padding)

Παραπάνω επεξηγήθηκε η συνδεσιμότητα του κάθε νευρώνα του συνελικτικού επιπέδου, στον όγκο (π.χ. RGB εικόνα με μήκος=αριθμός εικονοστοιχείων, πλάτος=αριθμός εικονοστοιχείων, βάθος=3 χρωματικά κανάλια) εισόδου. Σχετικά με τις διαστάσεις του όγκου εξόδου, από το συνελικτικό επίπεδο, τις "ελέγχουν" τρεις υπέρ-παράμετροι: το βάθος, το βήμα (stride) και η πρόσθεση μηδενικών (zero padding). (Keiron O'Shea 2015)

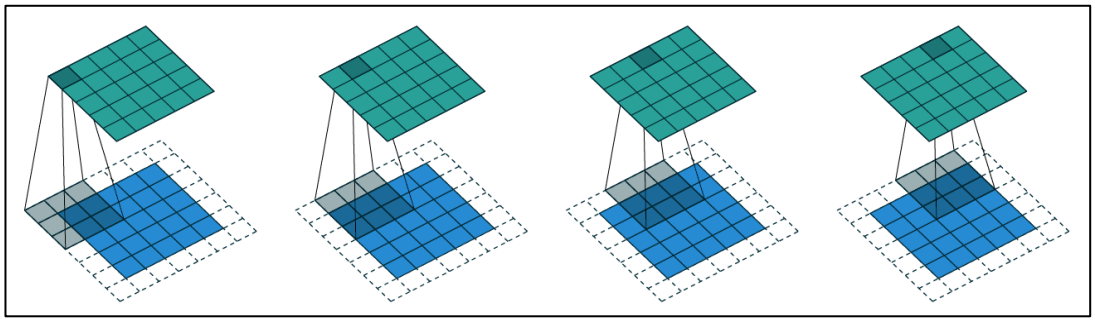
Το βάθος του όγκου εξόδου, που παράγεται από ένα συνελικτικό επίπεδο, αντιστοιχεί στον αριθμό των διαφορετικών φίλτρων που χρησιμοποιεί. Η μείωση αυτής της υπέρ-παραμέτρου, μπορεί να μειώσει δραστικά τον συνολικό αριθμό νευρώνων του δικτύου, αλλά αυτό μπορεί να έχει σαν αποτέλεσμα τη δραστική μείωση των δυνατοτήτων αναγνώρισης προτύπων του μοντέλου. Επίσης, αν το πρώτο συνελικτικό επίπεδο λαμβάνει ως είσοδο την ανεπεξέργαστη αρχική εικόνα, τότε διαφορετικοί νευρώνες κατά το εύρος της διάστασης του βάθους, ίσως ενεργοποιηθούν, λόγω της παρουσίας διαφόρων χαρακτηριστικών. Ένα σετ νευρώνων που εστιάζουν στην ίδια περιοχή της εισόδου καλούνται “στήλη βάθους” (depth column).



Εικόνα 23:Παράδειγμα δημιουργίας όγκου εξόδου (σύνολο χαρτών χαρακτηριστικών εξόδου), συνελικτικού επιπέδου. (Διαβάζεται από αριστερά προς δεξιά και από πάνω προς τα κάτω). (Sarkar 2018)

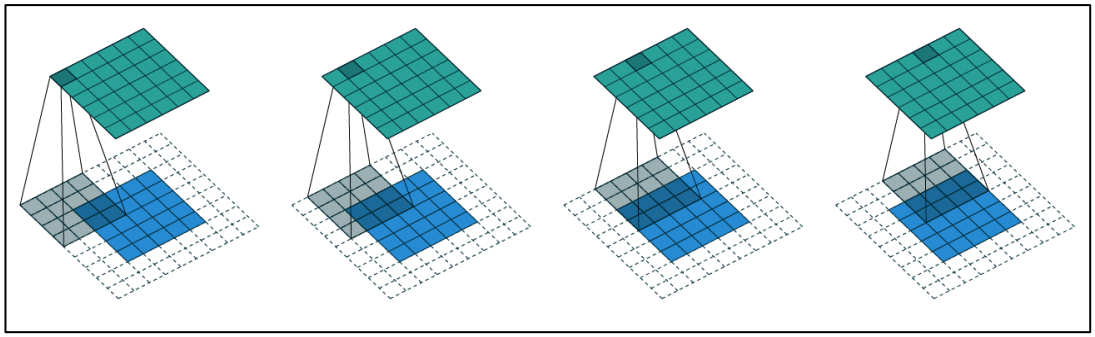
Η δεύτερη υπέρ-παράμετρος ενός συνελικτικού επιπέδου είναι το βήμα (stride), βάσει του οποίου ορίζεται η απόσταση, σε εικονοστοιχεία, στη χωρική διάσταση της εισόδου, κατά την οποία κινείται κάθε φορά το τοπικό δεκτικό πεδίο. Ουσιαστικά, το βήμα ορίζει ανά πόσα εικονοστοιχεία εφαρμόζεται κάθε φορά το φίλτρο συνέλιξης. Όταν το βήμα είναι ένα, το φίλτρο κινείται και εφαρμόζεται ανά ένα εικονοστοιχείο τη φορά. Στην περίπτωση που είναι ένα, αυτό έχει σαν αποτέλεσμα ένα υπερβολικά

αλληλεπικαλυπτόμενο δεκτικό πεδίο, που παράγει ένα πολύ μεγάλο χάρτη χαρακτηριστικών εξόδου. Αντίθετα, ένα βήμα που έχει οριστεί με ένα μεγάλο αριθμό, θα μειώσει την αλληλοεπικάλυψη, και αυτό θα έχει σαν αποτέλεσμα εξόδους μικρότερων διαστάσεων.



Εικόνα 24: Συνέλιξη ενός 3x3 φίλτρου, με ένα χάρτη χαρακτηριστικών εισόδου 5x5, με 1x1 πρόσθεση μηδενικών επί των ορίων και βήμα 1. (Vincent Dumoulin 2018)

Η τρίτη υπέρ-παράμετρος του συνελικτικού επιπέδου, είναι το μέγεθος της πρόσθεση μηδενικών (zero padding). Συνήθως, είναι χρήσιμο να συμπληρώνεται ο όγκος εισόδου με μηδενικά περιμετρικά των ορίων, γιατί επιτρέπει τον έλεγχο των χωρικών διαστάσεων του όγκου εξόδου. Συχνά, μάλιστα, χρησιμοποιείται ώστε να προκαθοριστούν οι χωρικές διαστάσεις του όγκου εξόδου και να είναι ακριβώς εισόδου, με τον όγκου εισόδου. Όπως γίνεται αντιληπτό, προσθέτοντας μηδενικά γύρω από τα όρια του όγκου εισόδου, αλλάζουν οι πιθανές θέσεις που μπορεί να εφαρμοστεί το φίλτρο.

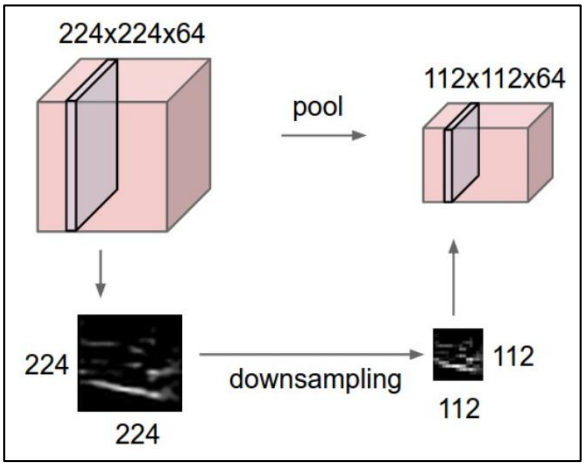


Εικόνα 25: Συνέλιξη φίλτρου 4x4, με χάρτη χαρακτηριστικών εισόδου 5x5, με 2x2 πρόσθεση μηδενικών επί των ορίων και βήμα 1. (Vincent Dumoulin 2018)

3.5 Συγκέντρωση (Pooling)

Μία τυπική αλληλουχία επιπέδων ενός CNN, αποτελείται από τρία διαφορετικά επίπεδα. Πρώτα, ένα συνελικτικό επίπεδο παράγει ένα σύνολο (όγκο) χαρτών χαρακτηριστικών εξόδου. Στη συνέχεια, κάθε τιμή του συνόλου των χαρτών χαρακτηριστικών εξόδου, που εξήγαγε το συνελικτικό επίπεδο, εισάγεται σε μία συνάρτηση ενεργοποίησης, όπως η ReLU, κατωφλιώνοντας στο μηδέν. Το επίπεδο που πραγματοποιεί τη συγκεκριμένη διεργασία, καλείται επίπεδο ReLU και θεωρείται ως στάδιο εντοπισμού χαρακτηριστικών. Το επίπεδο ReLU, αφήνει το μέγεθος του όγκου που λαμβάνει, απαράλλαχτο. Τέλος, το τρίτο επίπεδο μίας τυπικής αλληλουχίας, καλείται επίπεδο συγκέντρωσης (pooling).

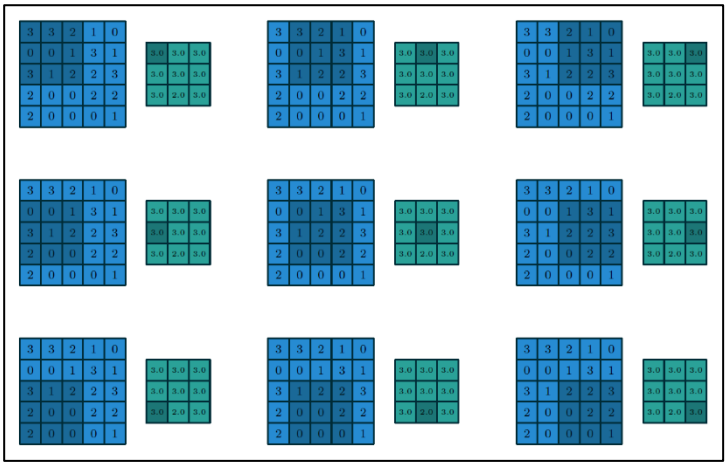
"*Οι σύγχρονες αρχιτεκτονικές υπολογιστικής όρασης, συνήθως περιλαμβάνουν ένα στάδιο χωρικής συγκέντρωσης, το οποίο συνδυάζει, τις αποκρίσεις των ανιχνευτών χαρακτηριστικών (επίπεδο ReLU) σε παρακείμενες τοποθεσίες, σε κάποια στατιστική η οποία συνοψίζει την από κοινού κατανομή των χαρακτηριστικών σε κάποια περιοχή ενδιαφέροντος. Η συγκέντρωση χαρακτηριστικών μίας τοπικής περιοχής, ώστε να γίνει η αναπαράσταση σχεδόν αμετάβλητη σε μικρές μετατοπίσεις των τιμών εισόδου, χρησιμοποιείται σε μεγάλο αριθμό μοντέλων οπτικής αναγνώρισης*" (Y-Lan Boureau 2010).



Εικόνα 26: Το επίπεδο συγκέντρωσης μειώνει το μέγεθος του όγκου χωρικά, επιδρώντας ανεξάρτητα στην κάθε χάρτη ενεργοποιήσεων, του όγκου χαρτών ενεργοποιήσεων που εισάγονται από το επίπεδο ReLU. Στο συγκεκριμένο παράδειγμα, ο όγκος εισόδου στο επίπεδο συγκέντρωσης είναι διαστάσεων 224x224x64, εφαρμόζεται συγκέντρωση με φίλτρο διαστάσεων 2x2, βήμα 2. Το επίπεδο συγκέντρωσης εξάγει τελικά όγκο διαστάσεων 112x112x64. (Fei-Fei Li 2020)

"*Ο σκοπός των επιπέδων συγκέντρωσης, είναι να επιτευχθεί αμεταβλητότητα σε μικρές μετατοπίσεις των τιμών εισόδου, μειώνοντας τη χωρική ανάλυση των χαρτών χαρακτηριστικών που λαμβάνουν ως είσοδο και επεξεργάζονται. Κάθε χάρτης χαρακτηριστικών στον οποίο έχει εφαρμοστεί συγκέντρωση, αντιστοιχεί σε ένα χάρτη χαρακτηριστικών του προηγούμενου επιπέδου*" (Dominik Scherer 2010). Άρα, το

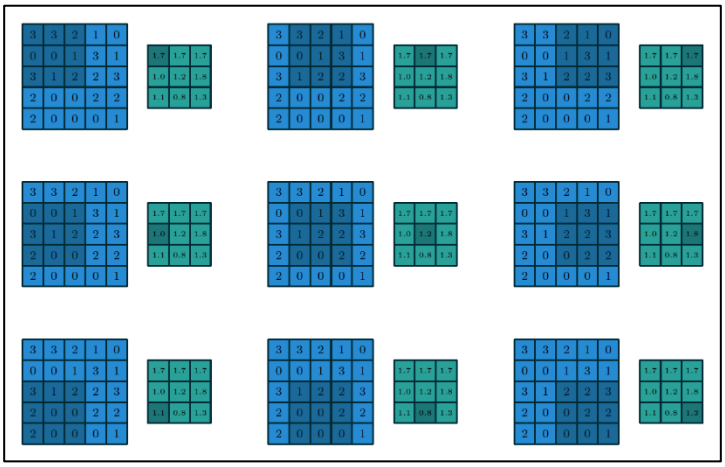
επίπεδο συγκέντρωσης λαμβάνει ένας όγκο ενεργοποιήσεων (στις περιπτώσεις όπου προηγείται ένα επίπεδο ReLU, αν προηγείται συνελικτικό επίπεδο λαμβάνει έναν όγκο με τους χάρτες χαρακτηριστικών εξόδου αυτού του συνελικτικού επιπέδου που προηγείται) και επιδρά με τέτοιο τρόπο, ώστε τροποποιεί τις χωρικές διαστάσεις του μήκους και του πλάτους του όγκου αυτού, αλλά όχι τη διάσταση του βάθους. Σε κάποιες περιπτώσεις, δε χρησιμοποιείται επίπεδο εφαρμογής συνάρτησης ενεργοποίησης, π.χ. ReLU, και έτσι το επίπεδο συγκέντρωσης επιδρά στον όγκο χαρτών χαρακτηριστικών εξόδου, του συνελικτικού επιπέδου. Και στη συγκεκριμένη περίπτωση, το επίπεδο συγκέντρωσης μειώνει τις διαστάσεις μήκους και πλάτους του όγκου αυτού, αλλά αφήνει απαράλλαχτη τη διάσταση του βάθους. Ακόμα, αφού ένα επίπεδο συγκέντρωσης μειώνει τις χωρικές διαστάσεις της αναπαράστασης, με αποτέλεσμα να μειώνεται η ποσότητα των παραμέτρων και των υπολογισμών στο δίκτυο, συνεπώς συμβάλει και στον έλεγχο της υπερπροσαρμογής. (Fei-Fei Li 2020)



Εικόνα 27: Υπολογισμός τιμών εξόδου μίας 3x3 πράξης μέγιστης συγκέντρωσης (max pooling), επί μίας 5x5 εισόδου, χρησιμοποιώντας βήμα 1x1. (Vincent Dumoulin 2018)

Υπάρχουν αρκετά είδη συγκέντρωσης. Το πιο συνηθισμένο από αυτά, είναι αυτό της μέγιστης συγκέντρωσης (max pooling) (Zhou n.d.), όπου κάθε νευρώνας του επιπέδου, υπολογίζει τη συνάρτηση μεγίστου, για μία τοπική περιοχή των χαρτών ενεργοποιήσεων εισόδου. Θεωρώντας ότι το μπλε πλέγμα τιμών, της Εικόνα 27 είναι είτε ένας από τους χάρτες χαρακτηριστικών, ενός όγκου που εξάχθηκε από ένα συνελικτικό επίπεδο, είτε ένας από τους χάρτες ενεργοποιήσεων, ενός όγκου που εξάχθηκε από ένα επίπεδο ReLU, η μέγιστη συγκέντρωση υπολογίζει τη μέγιστη τιμή, για μία τοπική περιοχή, ως ένα είδος υπό-δειγματοληψίας, όπου κάθε τοπική περιοχή συνοψίζεται από το στατιστικό στοιχείο της μέγιστης του τιμής. *"Κατά κάποιο τρόπο, η συγκέντρωση λειτουργεί αρκετά όμοια με μία διακριτή συνέλιξη, αλλά αντικαθιστά τον γραμμικό συνδυασμό που περιγράφεται από το φίλτρο, με κάποια άλλη συνάρτηση, όπως π.χ. η συνάρτηση μέγιστης τιμής"* (Vincent Dumoulin 2018).

Εκτός της μέγιστης συγκέντρωσης, οι νευρώνες ενός επιπέδου συγκέντρωσης, μπορούν επίσης να εκτελέσουν άλλες λειτουργίες, υπολογίζοντας κάποια άλλη συνάρτηση, όπως αυτή της συγκέντρωσης μέσου όρου (average pooling). Ένα επίπεδο συγκέντρωσης μέσου όρου, πραγματοποιεί υπό-δειγματοληψία, διαχωρίζοντας επίσης τις τιμές εισόδου σε τετραγωνικές τοπικές περιοχές και υπολογίζοντας την τιμή του μέσου όρου, σε κάθε μία από αυτές τις περιοχές.



Εικόνα 28: Υπολογισμός των τιμών εξόδου μίας 3x3 πράξης συγκέντρωσης μέσου όρου, επί μίας 5x5 εισόδου, χρησιμοποιώντας βήματα 1x1. (Vincent Dumoulin 2018)

"*Συνοψίζοντας, ένα επίπεδο συγκέντρωσης:*

- *Λαμβάνει έναν όγκο διαστάσεων $W_1 \times H_1 \times D_1$*
- *Απαιτεί δύο υπέρ-παραμέτρους:*
  Τη χωρική του έκταση F,
  Το βήμα S
- *Παράγει ένα volume διαστάσεων $W_2 \times H_2 \times D_2$ όπου:*
  $W_2 = (W_1 – F)/S + 1$
  $H_2 = (H_1 – F)/S + 1$
  $D_2 = D_1$
- *Εισάγει μηδενικές παραμέτρους αφού υπολογίζει μία σταθερή συνάρτηση των τιμών εισόδου.*
- *Για τα επίπεδα συγκέντρωσης, δεν είναι σύνηθες να συμπληρώνεται η είσοδος χρησιμοποιώντας μηδενικά."* (Fei-Fei Li 2020)

3.6 Ανεστραμμένη Συνέλιξη (Transposed Convolution)

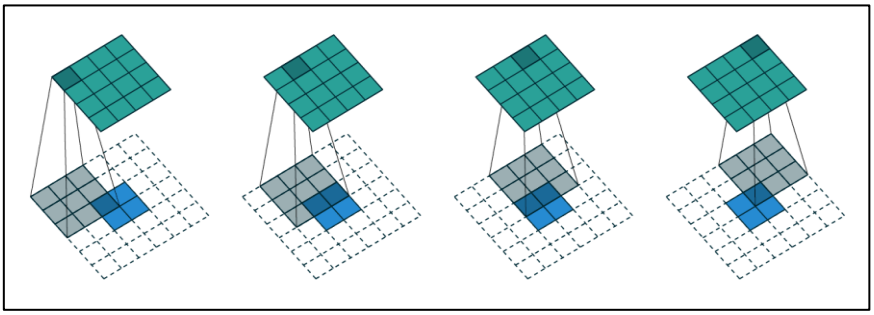
Η ανεστραμμένη συνέλιξη είναι μία διεργασία η οποία αυξάνει τις χωρικές διαστάσεις (up-sample) των χαρτών χαρακτηριστικών και με τις κατάλληλες προσαρμογές, διατηρεί το μοτίβο συνδεσιμότητας .Ως επίπεδο το οποίο να πραγματοποιεί ανεστραμμένες συνελίξεις, προτάθηκε για πρώτη φορά στην εργασία (Matthew D. Zeiler 2010), όπου οι Zeiler et al. υποστηρίζουν ότι "*κάθε επίπεδο στο*

δίκτυο ανεστραμμένων συνελίξεων μας είναι εκ των άνω προς τα κάτω˙ επιδιώκει να δημιουργήσει το σήμα εισόδου από ένα άθροισμα συνελίξεων επί των χαρτών χαρακτηριστικών (σε αντίθεση με το σήμα εισόδου), με φίλτρα που το δίκτυο μαθαίνει. Δεδομένου ενός σήματος εισόδου και ενός συνόλου φίλτρων, το να εξαχθούν οι ενεργοποιήσεις χαρτών χαρακτηριστικών, απαιτεί την επίλυση ενός προβλήματος ανεστραμμένων συνελίξεων πολλών συνιστωσών, το οποίο είναι υπολογιστικά απαιτητικό". Βγαίνει λοιπόν το συμπέρασμα, ότι σε αντίθεση με τις κλασσικές μεθόδους αύξησης του μεγέθους μίας εικόνας (π.χ. μέθοδοι παρεμβολής), οι παράμετροι, δηλαδή οι τιμές των φίλτρων, των ανεστραμμένων συνελίξεων, δεν είναι σταθερές, δεδομένες τιμές, αλλά μαθαίνονται από το δίκτυο και ανανεώνονται κατά τη διαδικασία της εκπαίδευσης. *"Αυτό επιτυγχάνεται τοποθετώντας μηδενικά μεταξύ των διαδοχικών νευρώνων στο δεκτικό πεδίο της εισόδου, έπειτα ολισθαίνοντας το φίλτρο συνέλιξης με μοναδιαία βήματα"* (Thangarajah Akilan 2019).

Σε αντίθεση με την κλασσική συνέλιξη, η οποία συνδέει περισσότερες από μία ενεργοποιήσεις εισόδου, με μία μεμονωμένη ενεργοποίηση, η ανεστραμμένη συνέλιξη συσχετίζει μόνο μία ενεργοποίηση, με περισσότερες από μία ενεργοποιήσεις εξόδου. Μία ανεστραμμένη συνέλιξη, σε μία δεδομένη είσοδο, πραγματοποιείται ώστε να αναπαραστήσει μία τέτοια είσοδο, σα να είναι η έξοδος μίας διακριτής συνέλιξης, εφαρμοσμένης σε οποιονδήποτε χάρτη χαρακτηριστικών. Η ανεστραμμένη συνέλιξη, μπορεί τότε να θεωρηθεί ως η διαδικασία που επιτρέπει την αποκατάσταση της μορφής αυτού του χάρτη χαρακτηριστικών. "*Προσοχή, όμως, η ανεστραμμένη συνέλιξη δεν εξασφαλίζει την αποκατάσταση του ακριβώς ίδιου χάρτη χαρακτηριστικών εισόδου, αφού δεν ορίζεται ως η αντίστροφη διαδικασία της συνέλιξης, αλλά επιστρέφει ένα χάρτη χαρακτηριστικών με τις ίδιες χωρικές διαστάσεις."* (Vincent Dumoulin 2018)

Ας υποτεθεί μία συνέλιξη, ενός 3x3 φίλτρου, με μία 4x4 είσοδο, με μοναδιαίο βήμα και καθόλου συμπλήρωμα μηδενικών (δηλ., $i = 4,\ k = 3,\ s = 1,\ p = 0$). Η συνέλιξη αυτή παράγει μία έξοδο 2x2. Η αντίστοιχη ανεστραμμένη συνέλιξη, θα έχει ως αποτέλεσμα μία έξοδο 4x4, στην περίπτωση που εφαρμοστεί σε μία είσοδο 2x2. Ένας τρόπος να υπολογιστεί το αποτέλεσμα μίας ανεστραμμένης συνέλιξης είναι να εφαρμοστεί μία ανάλογη (αλλά αρκετά λιγότερο αποδοτική) ευθεία συνέλιξη. Το παραπάνω παράδειγμα, θα μπορούσε να αντιμετωπιστεί με μία συνέλιξη ενός 3x3 φίλτρου, με μία 2x2 είσοδο, με ένα 2x2 συμπλήρωμα μηδενικών στα όρια, χρησιμοποιώντας μοναδιαία βήματα (δηλ., $i' = 2,\ k' = k,\ s' = 1,\ p' = 2$). Όπως φαίνεται, το μέγεθος του φίλτρου και του βήματος παραμένει το ίδιο, αλλά έχει εφαρμοστεί συμπλήρωμα μηδενικών στις τιμές εισόδου της ανεστραμμένης συνέλιξης. Η συμπλήρωση μηδενικών επιδρά στο μοτίβο συνδεσιμότητας της ανεστραμμένης συνέλιξης και μπορεί να χρησιμοποιηθεί για να καθοδηγηθεί ο σχεδιασμός της ανάλογης συνέλιξης. Παραδείγματος χάριν, το πάνω αριστερά εικονοστοιχείο της εισόδου της απευθείας συνέλιξης, συνεισφέρει μόνο στο πάνω

αριστερά εικονοστοιχείο της εξόδου, το πάνω δεξιά εικονοστοιχείο της εισόδου, συνδέεται μόνο με το πάνω δεξιά εικονοστοιχείο της εξόδου, και πάει λέγοντας. Για να διατηρηθεί το ίδιο μοτίβο συνδεσιμότητας κατά τη διαδικασία της ανάλογης συνέλιξης, είναι απαραίτητο να συμπληρωθούν οι τιμές εισόδου με μηδενικά, με τέτοιο τρόπο ώστε η πρώτη (πάνω αριστερά) εφαρμογή του φίλτρου να "ακουμπά" μόνο το πάνω αριστερά εικονοστοιχείο, δηλαδή, το μέγεθος του συμπληρώματος μηδενικών πρέπει να ισούται με το μέγεθος του φίλτρου μείον ένα. Εδώ πρέπει να τονιστεί ότι μόνο όταν τα βήματα είναι μοναδιαία, η ανεστραμμένη συνέλιξη είναι ουσιαστικά μία κλασσική συνέλιξη, αλλά με διαφορετικούς κανόνες συμπλήρωσης μηδενικών.



Εικόνα 29: Ανεστραμμένη συνέλιξη ενός 3x3 φίλτρου, με μία 4x4 είσοδο, με μοναδιαία βήματα και καθόλου συμπλήρωμα μηδενικών. Είναι ανάλογη της συνέλιξης ενός 3x3 φίλτρου, με μία 2x2 είσοδο, με συμπλήρωμα μηδενικών 2x2 στα όρια και μοναδιαία βήματα. (Vincent Dumoulin 2018)

Οι ανεστραμμένες συνελίξεις λειτουργούν εναλλάσσοντας τα προς τα εμπρός και προς τα πίσω περάσματα μίας συνέλιξης. Με άλλα λόγια, το φίλτρο ορίζει μία συνέλιξη, αλλά το αν είναι μία ευθεία ή ανεστραμμένη συνέλιξη, καθορίζεται από το πως υπολογίζονται τα προς τα εμπρός και τα προς τα πίσω περάσματα. Έτσι, λοιπόν, η ανεστραμμένη συνέλιξη είναι, ουσιαστικά, είναι ένα προς τα πίσω πέρασμα μίας αντίστοιχης, κλασσικής συνέλιξης. Θεωρώντας ότι η υλοποίηση μίας συνέλιξης πραγματοποιείται με πολλαπλασιασμό πινάκων δεδομένου ενός διανύσματος τιμών εισόδου x και ενός πίνακα βαρών W, η συνάρτηση συνέλιξης κατά την προς τα εμπρός διάδοση υλοποιείται πολλαπλασιάζοντας τις τιμές εισόδου με τον πίνακα βαρών, εξάγοντας τελικά $y = Wx$. Αφού η οπισθοδιάδοση σφάλματος υλοποιείται βάσει του κανόνα της αλυσίδας και $\nabla_x y = W^T$, η συνάρτηση συνέλιξης κατά την οπισθοδιάδοση σφάλματος μπορεί να υλοποιηθεί πολλαπλασιάζοντας την είσοδο της με τον ανάστροφο πίνακα βαρών W^T. Συνεπώς, ένα επίπεδο ανεστραμμένης συνέλιξης μπορεί απλά να εναλλάξει τη συνάρτηση προς τα εμπρός διάδοσης με αυτή της οπισθοδιάδοσης σφάλματος του συνελικτικού επιπέδου, αφού οι συναρτήσεις προς τα εμπρός διάδοσης και προς τα πίσω διάδοσης πολλαπλασιάζουν το διάνυσμα εισόδου τους με W^T και W, αντίστοιχα. (Aston Zhang 2021)

Η συνέλιξη μπορεί να εκφραστεί ως πολλαπλασιασμός πινάκων. Ένα φίλτρο συνέλιξης, είναι δυνατόν να αναδιαταχθεί με τέτοιο τρόπο ώστε να περιέλθει σε μορφή πίνακα, όπου αν πολλαπλασιαστεί με τον αντίστοιχο πίνακα κάποιας εισόδου, θα υλοποιηθούν, με τη μία αυτή πράξη πολλαπλασιασμού πινάκων, όλες οι διαφορετικές πράξεις συνέλιξης από την εφαρμογή του φίλτρου στις διαφορετικές θέσεις της εισόδου. Παραδείγματος χάριν, ας υποτεθεί φίλτρο 3x3 και είσοδος 4x4:

$$K = \begin{bmatrix} w_{0,0} & w_{0,1} & w_{0,2} \\ w_{1,0} & w_{1,1} & w_{1,2} \\ w_{2,0} & w_{2,1} & w_{2,2} \end{bmatrix} \qquad I = \begin{bmatrix} x_{0,0} & x_{0,1} & x_{0,2} & x_{0,3} \\ x_{1,0} & x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,0} & x_{2,1} & x_{2,2} & x_{2,3} \\ x_{3,0} & x_{3,1} & x_{3,2} & x_{3,3} \end{bmatrix} \tag{3.6.1}$$

Η συνέλιξη μεταξύ του φίλτρου K και του πίνακα τιμών εισόδου I, με μοναδιαία βήματα και καθόλου συμπλήρωμα μηδενικών, μπορεί να πραγματοποιηθεί ως πολλαπλασιασμός πινάκων. Ο πίνακας I θα αναδιαταχθεί σε ένα διάνυσμα 16 διαστάσεων και θα δημιουργηθεί ο πίνακας συνέλιξης C, από το φίλτρο K:

$$I = \begin{bmatrix} x_{0,0} \\ x_{0,1} \\ x_{0,2} \\ x_{0,3} \\ \vdots \\ x_{3,0} \\ x_{3,1} \\ x_{3,2} \\ x_{3,3} \end{bmatrix} \tag{3.6.2}$$

$$C = \begin{bmatrix} w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 & 0 \\ 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 \\ 0 & 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} \end{bmatrix}$$

Ο πολλαπλασιασμός $C \times I$, έχει σαν αποτέλεσμα ένα διάνυσμα 4 διαστάσεων, το οποίο μπορεί να αναδιαταχθεί σε ένα πίνακα 2x2, τον πίνακα τιμών εξόδου, δηλαδή, από τη διαδικασία της συνέλιξης:

$$G = C \times I = \begin{bmatrix} y_{0,0} \\ y_{1,0} \\ y_{2,0} \\ y_{3,0} \end{bmatrix} = \begin{bmatrix} y_{0,0} & y_{1,0} \\ y_{2,0} & y_{3,0} \end{bmatrix} \tag{3.6.3}$$

Αν, τώρα, χρησιμοποιηθεί το 4-διάστατο διάνυσμα G, της Εξίσωσης 3.6.3, και πολλαπλασιαστεί με τον ανάστροφο του πίνακα C, τον C^T, θα παραχθεί ένα διάνυσμα 16 διαστάσεων, το οποίο μπορεί να αναδιαταχτεί σε ένα πίνακα 4x4, ίδιων διαστάσεων, δηλαδή, με τον αρχικό πίνακα τιμών εισόδου. Όπως, όμως, επισημάνθηκε πριν, δεν εξασφαλίζεται ότι θα είναι και ίδιων στοιχείων:

$$C^T \times G = \begin{bmatrix} z_{0,0} \\ \vdots \\ z_{15,0} \end{bmatrix} = \begin{bmatrix} z_{0,0} & \cdots & z_{3,0} \\ \vdots & \ddots & \vdots \\ z_{12,0} & \cdots & z_{15,0} \end{bmatrix} \qquad \textbf{3.6.4}$$

4 Αναγνώριση Γεωγραφικών Χαρακτηριστικών

4.1 Εισαγωγή

Οι ιστορικοί χάρτες είναι πηγές πολύτιμης πληροφορίας, από την οποία ένα πλήθος επιστημονικών πεδίων, όπως η αρχαιολογία, η γεωγραφία, η κοινωνιολογία, η πολεοδομία, η αρχιτεκτονική, κ.λπ., μπορούν να επωφεληθούν. Η μελέτη του αστικού τοπίου του παρελθόντος χρησιμοποιώντας χάρτες, μπορεί να αποτελέσει σημαντικό επιστημονικό "εργαλείο", εξαιτίας των συμπερασμάτων που μπορούν να εξαχθούν, σε ιστορικό, οικολογικό και κοινωνικό-οικονομικό επίπεδο. Η εξαγωγή πληροφορίας από παλαιούς χάρτες, είναι συνήθως μία απαιτητική διαδικασία, εξαιτίας κυρίως του ότι στις πλείστες των περιπτώσεων χαρακτηρίζονται από χαμηλή γραφική ποιότητα.

Η αυτοματοποίηση του εντοπισμού και της εξαγωγής συγκεκριμένου περιεχομένου από πολλαπλές εικόνες, γρήγορα και με ικανοποιητική ακρίβεια, είναι μία σημαντική εργασία. Στην περίπτωση όπου οι εικόνες είναι αποτέλεσμα σάρωσης χαρτών, τέτοιου είδους περιεχόμενο μπορεί να είναι σύμβολα σχετικά με το ανάγλυφο μίας περιοχής, τοπωνύμια˙ και όσον αφορά το αστικό περιβάλλον, οδοί, όρια οικοδομικών τετραγώνων ή κτιρίων. Όταν τέτοιου είδους πληροφορία αναφέρεται σε μία παλαιότερη χρονική περίοδο, αποτελεί υψηλής χρησιμότητας υπόβαθρο για τη μέτρηση, ανάλυση και ταυτοποίηση του αστικού περιβάλλοντος του παρελθόντος. Πληθώρα χαρτών με τέτοιου είδους πληροφορία, όπως τοπογραφικοί ή ρυμοτομικά σχέδια, υπάρχει σε δημόσιες υπηρεσίες. Συχνά, καθίσταται απαραίτητο να εξαχθεί περιεχόμενο από τέτοιους χάρτες, με ένα συστηματικό τρόπο και βάσει συγκεκριμένων χαρακτηριστικών, για κάποια μελέτη, υποστήριξη λήψης αποφάσεων ή χάραξη στρατηγικής, και τότε η συγκεκριμένη διαδικασία μπορεί να αποδειχτεί πολύ χρονοβόρα, και απαιτητική, αφού μπορεί να πραγματοποιηθεί μόνο από κάποιον ειδικό.

Υπάρχουν αρκετά λογισμικά συστημάτων γεωγραφικών πληροφοριών, και όχι μόνο, τα οποία παρέχουν τη δυνατότητα διανυσματοποίησης μίας εικόνας. Η διαδικασία αυτή μπορεί να πραγματοποιηθεί είτε (ημί)-χειρωνακτικά, υπό την επίβλεψη κάποιου ειδικού επί του αντικειμένου, είτε αυτόματα. Το παραγόμενο προϊόν, παρ'όλα αυτά, δεν ξεχωρίζει τα αντικείμενα που παρουσιάζονται σε ένα χάρτη. Με άλλα λόγια, γραμμικά στοιχεία δεν διαχωρίζονται από πολύγωνα, κείμενο (γράμματα ή αριθμούς) και σημειακά δεδομένα. Για αυτό το λόγο, η επίτευξη διακριτής απόδοσης του κάθε διαφορετικού αντικειμένου, καθιστά αναγκαία την υλοποίηση

κάποιας άλλης μεθόδου, η οποία να είναι σε θέση να αναγνωρίζει το κάθε αντικείμενο ξεχωριστά, με βάσει κάποια χαρακτηριστικά.

4.2 Εντοπισμός κτιρίων σε ιστορικούς τοπογραφικούς χάρτες

Στόχος της παρούσας εργασίας, είναι ο αυτόματος εντοπισμός και η εξαγωγή των κτιρίων, που αναπαρίστανται σε μία εικόνα, ενός σαρωμένου ιστορικού χάρτη. Ο συγκεκριμένος σαρωμένος χάρτης, ο οποίος παρουσιάζεται στην Εικόνα 30, διατέθηκε από την Ελληνική Στατιστική Αρχή (ΕΛΣΤΑΤ) και είχε χρησιμοποιηθεί για λόγους απογραφής κατοικιών και πληθυσμού, το 1971. Η κλίμακα του είναι 1:5000, οι διαστάσεις του είναι 70×50 εκατοστά και απεικονίζει τον οικισμό της Πετρούπολης, του ομώνυμου δήμου, ο οποίος ανήκει σήμερα στον Δυτικό Τομέα Αθηνών, της Περιφέρειας Αττικής.

Εικόνα 30: Τοπογραφικός χάρτης περιοχής Πετρούπολης.

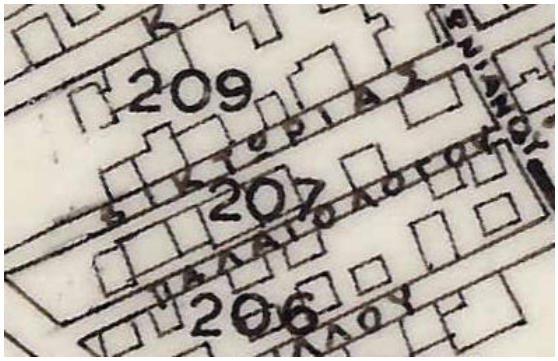
(α)

(β)

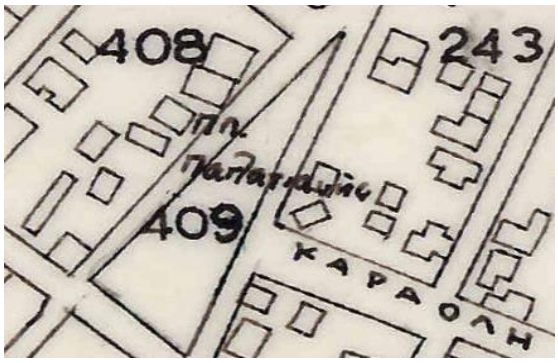
Εικόνα 31: (α) Υπό-περιοχή δυαδικοποιημένου αρχικού χάρτη (β) Εντοπισμός κτιρίων.

Η Εικόνα 31 δείχνει ένα παράδειγμα εντοπισμού κτιρίων στο τοπογραφικό χάρτη. Συγκεκριμένα, η Εικόνα 31α δείχνει μία υπό-περιοχή του αρχικού χάρτη της Εικόνα 30, σε δυαδική μορφή και η Εικόνα 31β, παρουσιάζει τα κτίρια που εντοπίζονται, στην ίδια υπό-περιοχή. Στην προτεινόμενη μεθοδολογία χρησιμοποιείται η αρχική έγχρωμη μορφή του χάρτη όπως φαίνεται στην Εικόνα 30, και όχι σε δυαδική μορφή. Όπως φαίνεται στην Εικόνα 31β, τα όρια, οι αριθμοί των οικοδομικών τετραγώνων και οι ονομασίες των οδών, έχουν απομονωθεί με επιτυχία.

Ο εντοπισμός κτιρίων, συχνά αποδεικνύεται αρκετά δύσκολη διαδικασία, αφού διαφόρων ειδών προκλήσεις μπορούν να προκύψουν, οι οποίες κυρίως σχετίζονται με την ύπαρξη θορύβου, ασχέτων γραφικών στοιχείων, κειμένου, κ.λπ.

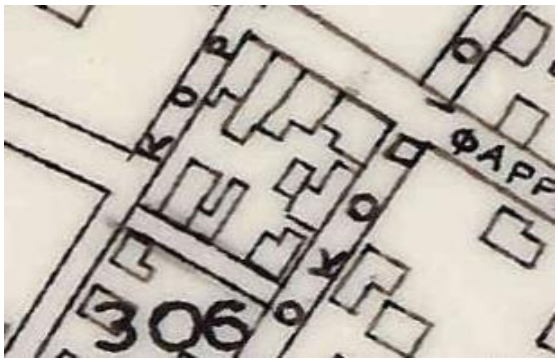


(α)



(β)

(γ)



(δ)

Εικόνα 32: Προκλήσεις κατά τον εντοπισμό κτιρίων (α) Πυκνό κείμενο που αλληλεπικαλύπτεται με το περιεχόμενο (β) Κείμενο σε αυθαίρετη θέση (γ) Κτίρια που ποικίλουν σε σχήμα, μέγεθος και προσανατολισμό (δ) Επικάλυψη ορίου οικοδομικού τετραγώνου, με όριο κτιρίου.

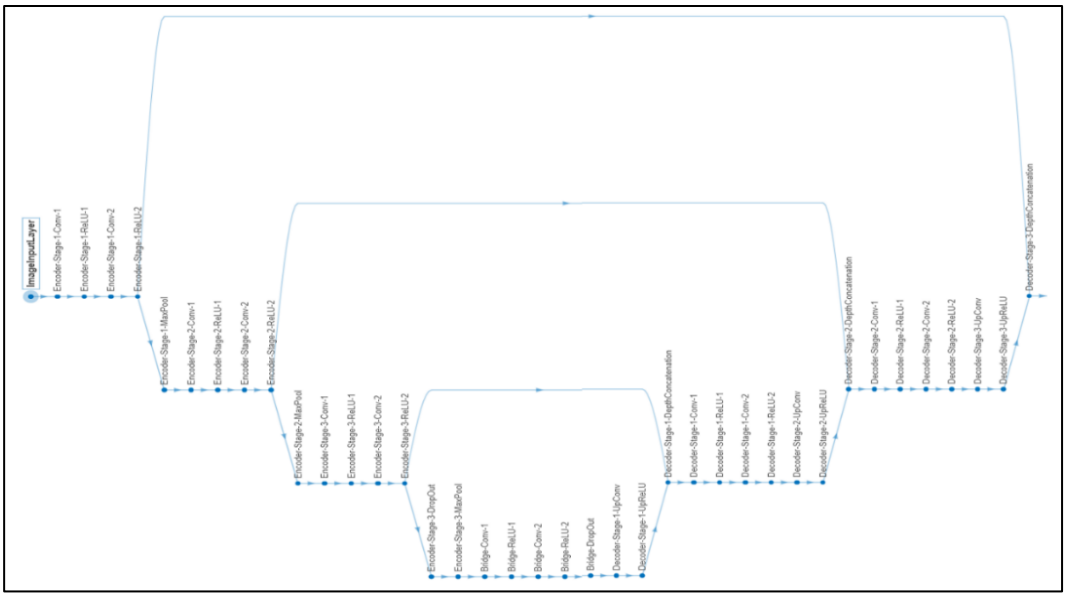
Για παράδειγμα, τα κτίρια συχνά αλληλεπικαλύπτεται με αριθμούς οικοδομικών τετραγώνων, όπως φαίνεται στην Εικόνα 32α. Ακόμα, σε ορισμένες περιπτώσεις, κείμενο τοποθετημένο σε αυθαίρετες θέσεις, επικαλύπτει τα κτίρια, όπως φαίνεται στην Εικόνα 32β. Επιπρόσθετα, η ποικιλία που χαρακτηρίζει το μέγεθος, σχήμα και προσανατολισμό των κτιρίων του συνόλου της περιοχής (Εικόνα 32γ), δυσχεραίνει την αναγνώριση τους. Τέλος, υπάρχουν κτίρια που καμία τους πλευρά δεν αλληλεπικαλύπτεται με το όριο οικοδομικού τετραγώνου και υπάρχουν άλλα, των οποίων μία ή και δύο πλευρές αλληλεπικαλύπτονται με το όριο οικοδομικού τετραγώνου (Εικόνα 32δ).

Στην παρούσα εργασία, προτείνεται μία μέθοδος η οποία αντιμετωπίζει το πρόβλημα αναγνώρισης και εξαγωγής των κτιρίων από ψηφιακές εικόνες, με μία προσέγγιση βαθέως CNN (Deep Convolutional Neural Network), η οποία βασίζεται στην αρχιτεκτονική U-Net (Olaf Ronneberger 2015). Το CNN εκπαιδεύεται χρησιμοποιώντας ένα μεγάλο αριθμό δειγμάτων εικόνων (image patches), ως μοντέλο βαθιάς, από εικόνα σε εικόνα, παλινδρόμησης, ώστε να απομονωθούν τα κτίρια επιτυχώς, από το χάρτη, και να αγνοηθεί το υπόλοιπο περιεχόμενο. Η αποτίμηση της προτεινόμενης μεθόδου ως προς την αποτελεσματικότητας της, γίνεται με βάση ένα τμήμα του χάρτη της Εικόνα 30, το οποίου έχει προ-σημανθεί από ειδικούς και χρησιμοποιείται ως σετ επαληθευμένων δεδομένων (ground truth data). Το σετ των δεδομένων αυτών χρησιμοποιείται τόσο για την εκπαίδευση του συστήματος όσο και για τον έλεγχο της αποτελεσματικότητάς του καθώς για κάθε δεδομένο εισόδου παρέχεται και η επιθυμητή απόκριση του CNN. Η αποτελεσματικότητα της μεθόδου αξιολογείται μέσω ενός συνόλου πειραμάτων, όπου η διαφοροποίηση τους σχετίζεται με τον διαφορετικό τρόπο συλλογής των δειγμάτων των εικόνων καθώς και στα διαφορετικά μεγέθη αυτών των δειγμάτων. Η αποδοτικότητα της μεθόδου αποτιμάται ποσοτικά μέσω κατάλληλων μετρικών, οι οποίες συγκρίνουν τις τιμές εξόδου του CNN, με τις αντίστοιχες τιμές του σετ επαληθευμένων δεδομένων για το σύνολο των δεδομένων ελέγχου.

4.3 Προτεινόμενη αρχιτεκτονική βαθιάς μάθησης

Το βαθύ CNN εκπαιδεύεται με δείγματα εικόνας “κομμένα” από μέρος του αρχικού χάρτη της Εικόνα 30, ως δεδομένα εισόδου, και με προσημασμένα δείγματα εικόνας, “κομμένα” από το αντίστοιχο μέρος, ως επαληθευμένα-επιθυμητά δεδομένα εξόδου. Το CNN, αρχιτεκτονικής U-Net, υλοποιεί μία βαθιά, επιπέδου εικονοστοιχείων, παλινδρόμηση. Η αρχιτεκτονική του δικτύου, αποτελείται από δύο βασικά μέρη, τα οποία συνδέονται από μία “γέφυρα”, εξού και η ονομασία “U-Net”. Το πρώτο μέρος, είναι ο κωδικοποιητής, ο οποίος επεξεργάζεται την εικόνα εισόδου “περνώντας” την μέσα από μία συστολική διαδρομή, όπου πραγματοποιείται μείωση των χωρικών της διαστάσεων και αύξηση της διάστασης του βάθους, από διαδοχικές συνελίξεις, ReLUs

και μέγιστες συγκεντρώσεις. Το δεύτερο μέρος, είναι ο αποκωδικοποιητής, ο οποίος επεξεργάζεται το προϊόν του κωδικοποιητή, “περνώντας” το μέσα από μία διαστολική διαδρομή, όπου πραγματοποιείται αύξηση των χωρικών του διαστάσεων και μείωση σταδιακά της διάστασης του βάθους, από διαδοχικές συνελίξεις, ReLUs και ανεστραμμένες συνελίξεις. Το χαρακτηριστικό πλεονέκτημα της αρχιτεκτονικής U-Net είναι η ικανότητας της να βρίσκει τη βέλτιστη ισορροπία μεταξύ τοποθεσίας και περιεχομένου. Όσο οι διαστάσεις πλάτους και μήκους της εικόνας εισόδου μειώνονται, εντός της συστολικής διαδρομής, τα φίλτρα στα βαθύτερα επίπεδα εστιάζουν σε μεγαλύτερο δεκτικό πεδίο (μειώνεται η πληροφορία θέσεων στην εικόνα), αντίστοιχα όμως αυξάνονται βαθμιαία τα κανάλια, δηλαδή η διάσταση του βάθους, κάθε φορά που ένας όγκος χαρτών χαρακτηριστικών, περνάει από ένα συνελικτικό επίπεδο (αυξάνεται η πληροφορία περιεχομένου στην εικόνα), με αποτέλεσμα την εκμάθηση πολύπλοκων χαρακτηριστικών, εντός αυτού του μέρους του δικτύου. Αντίθετα, κατά τη διαστολική διαδρομή, οι ανεστραμμένες συνελίξεις αυξάνουν βαθμιαία τις χωρικές διαστάσεις της εικόνας (αυξάνεται η πληροφορία θέσεων στην εικόνα), αλλά μειώνονται σταδιακά η διάσταση του βάθους (μειώνεται η πληροφορία περιεχομένου στην εικόνα).



Εικόνα 33: Αρχιτεκτονική U-Net του προτεινόμενου CNN.

Για μεγαλύτερη ακρίβεια στον εντοπισμό θέσης, σε κάθε βήμα του αποκωδικοποιητή χρησιμοποιούνται συνδέσεις παράλειψης (skip connections), συνενώνοντας την έξοδο των επιπέδων ανεστραμμένης συνέλιξης με τους χάρτες χαρακτηριστικών, από τον κωδικοποιητή, στο ίδιο επίπεδο. Τα δύο υπό-δίκτυα, όπως θα μπορούσε να χαρακτηρίσει κανείς τον κωδικοποιητή και τον αποκωδικοποιητή, αποτελούνται από πολλαπλά στάδια. Οι περισσότερες υλοποιήσεις αρχιτεκτονικών U-Net, δέχονται ως

υπέρ-παράμετρο τον αριθμό αυτών των σταδίων (βάθος κωδικοποιητή). Στην περίπτωση της αρχιτεκτονικής της Εικόνα 33, ο αριθμός σταδίων ισούται με 3. Κάθε στάδιο κωδικοποιητή αποτελείται από δύο σετ με συνελικτικά και ReLU επίπεδα, ακολουθούμενα από ένα επίπεδο 2x2 μέγιστης συγκέντρωσης. Κάθε στάδιο αποκωδικοποιητή, αποτελείται από ένα επίπεδο ανεστραμμένης συνέλιξης, ακολουθούμενο από δύο σετ συνελικτικών και ReLU επιπέδων.

1	'ImageInputLayer'	Image Input	224x224x3 images with 'zerocenter' normalization
2	'Encoder-Stage-1-Conv-1'	Convolution	64 3x3 convolutions with stride [1 1] and padding 'same'
3	'Encoder-Stage-1-ReLU-1'	ReLU	ReLU
4	'Encoder-Stage-1-Conv-2'	Convolution	64 3x3 convolutions with stride [1 1] and padding 'same'
5	'Encoder-Stage-1-ReLU-2'	ReLU	ReLU
6	'Encoder-Stage-1-MaxPool'	Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0 0 0]
7	'Encoder-Stage-2-Conv-1'	Convolution	128 3x3 convolutions with stride [1 1] and padding 'same'
8	'Encoder-Stage-2-ReLU-1'	ReLU	ReLU
9	'Encoder-Stage-2-Conv-2'	Convolution	128 3x3 convolutions with stride [1 1] and padding 'same'
10	'Encoder-Stage-2-ReLU-2'	ReLU	ReLU
11	'Encoder-Stage-2-MaxPool'	Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0 0 0]
12	'Encoder-Stage-3-Conv-1'	Convolution	256 3x3 convolutions with stride [1 1] and padding 'same'
13	'Encoder-Stage-3-ReLU-1'	ReLU	ReLU
14	'Encoder-Stage-3-Conv-2'	Convolution	256 3x3 convolutions with stride [1 1] and padding 'same'
15	'Encoder-Stage-3-ReLU-2'	ReLU	ReLU
16	'Encoder-Stage-3-DropOut'	Dropout	50% dropout
17	'Encoder-Stage-3-MaxPool'	Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0 0 0]
18	'Bridge-Conv-1'	Convolution	512 3x3 convolutions with stride [1 1] and padding 'same'
19	'Bridge-ReLU-1'	ReLU	ReLU
20	'Bridge-Conv-2'	Convolution	512 3x3 convolutions with stride [1 1] and padding 'same'
21	'Bridge-ReLU-2'	ReLU	ReLU
22	'Bridge-DropOut'	Dropout	50% dropout
23	'Decoder-Stage-1-UpConv'	Transposed Convolution	256 2x2 transposed convolutions with stride [2 2] and cropping [0 0 0 0]
24	'Decoder-Stage-1-UpReLU'	ReLU	ReLU
25	'Decoder-Stage-1-DepthConcatenation'	Depth concatenation	Depth concatenation of 2 inputs
26	'Decoder-Stage-1-Conv-1'	Convolution	256 3x3 convolutions with stride [1 1] and padding 'same'
27	'Decoder-Stage-1-ReLU-1'	ReLU	ReLU
28	'Decoder-Stage-1-Conv-2'	Convolution	256 3x3 convolutions with stride [1 1] and padding 'same'
29	'Decoder-Stage-1-ReLU-2'	ReLU	ReLU
30	'Decoder-Stage-2-UpConv'	Transposed Convolution	128 2x2 transposed convolutions with stride [2 2] and cropping [0 0 0 0]
31	'Decoder-Stage-2-UpReLU'	ReLU	ReLU
32	'Decoder-Stage-2-DepthConcatenation'	Depth concatenation	Depth concatenation of 2 inputs
33	'Decoder-Stage-2-Conv-1'	Convolution	128 3x3 convolutions with stride [1 1] and padding 'same'
34	'Decoder-Stage-2-ReLU-1'	ReLU	ReLU
35	'Decoder-Stage-2-Conv-2'	Convolution	128 3x3 convolutions with stride [1 1] and padding 'same'
36	'Decoder-Stage-2-ReLU-2'	ReLU	ReLU
37	'Decoder-Stage-3-UpConv'	Transposed Convolution	64 2x2 transposed convolutions with stride [2 2] and cropping [0 0 0 0]
38	'Decoder-Stage-3-UpReLU'	ReLU	ReLU
39	'Decoder-Stage-3-DepthConcatenation'	Depth concatenation	Depth concatenation of 2 inputs
40	'Decoder-Stage-3-Conv-1'	Convolution	64 3x3 convolutions with stride [1 1] and padding 'same'
41	'Decoder-Stage-3-ReLU-1'	ReLU	ReLU
42	'Decoder-Stage-3-Conv-2'	Convolution	64 3x3 convolutions with stride [1 1] and padding 'same'
43	'Decoder-Stage-3-ReLU-2'	ReLU	ReLU
44	'Final-ConvolutionLayer'	Convolution	1 1x1 convolutions with stride [1 1] and padding [0 0 0 0]

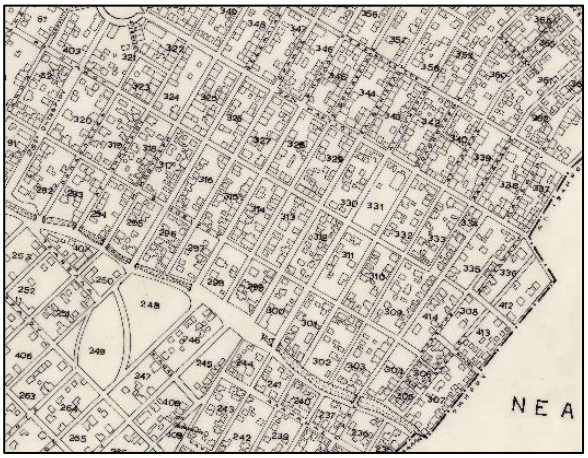
Εικόνα 34: Επίπεδα προκαθορισμένης αρχιτεκτονικής U-Net, βάθους κωδικοποιητή ίσου με 3 (εικόνα εισόδου 224x224x3).

Στόχος του προτεινόμενου CNN, είναι να προβλέψει, για κάθε εικονοστοιχείο της εικόνας εισόδου, μία τιμή, η οποία να είναι όσο το δυνατόν πιο κοντά στην τιμή του αντίστοιχου εικονοστοιχείου, του τμήματος εικόνας των επαληθευμένων δεδομένων. Όσον αφορά τη συνάρτηση κόστους, η οποία υλοποιείται για τη βελτιστοποίηση του μοντέλου, όπως είναι λογικό για ένα πρόβλημα παλινδρόμησης, χρησιμοποιείται το μέσο τετραγωνικό σφάλμα, όχι διαιρεμένο με τον αριθμό εικονοστοιχείων του κάθε τμήματος εικόνας, δηλαδή:

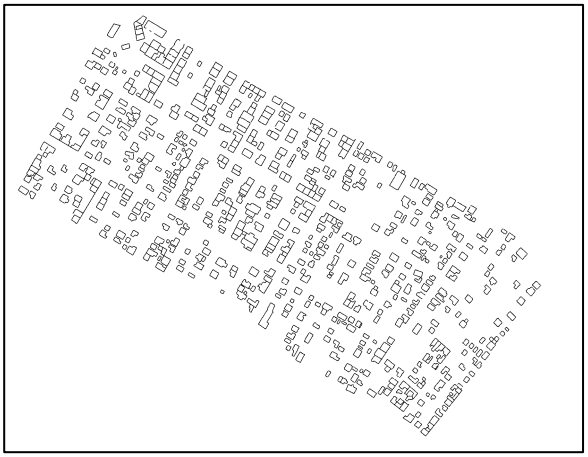
$$C = \frac{1}{2}\sum_{i=1}^{HW}\left(y_{gt} - y_p\right) \qquad 4.3.1$$

όπου οι H, W υποδηλώνουν το μήκος και πλάτος του τμήματος εικόνας επαληθευμένων δεδομένων, y_{gt} είναι η δυαδική τιμή του κάθε εικονοστοιχείου, της εικόνας επαληθευμένων δεδομένων, και y_p είναι η τιμή της απόκρισης του CNN, δεδομένου του αντίστοιχου τμήματος εικόνας εισόδου. Το κόστος της Εξίσωσης 4.3.1

οπισθοδιαδίδεται σε όλα τα κρυφά επίπεδα του CNN, βάσει του κανόνα αλυσίδα του αλγόριθμου οπισθοδιάδοσης σφάλματος backpropagation και οι παράμετροι του CNN ανανεώνονται επαναληπτικά, χρησιμοποιώντας τον αλγόριθμο SGD, με την βελτιστοποίηση του Adam. Λαμβάνοντας υπόψη, ότι η έξοδος του CNN είναι ένα σύνολο "πυκνών" προβλέψεων, όπου κάθε εικονοστοιχείο αντιστοιχεί σε μία συνεχή τιμή μεταξύ 0 και 1, η έξοδος του δικτύου μετατρέπεται σε δυαδική, ώστε να είναι συγκρίσιμη με την αντίστοιχη τιμή του εικονιστικού τμήματος των επαληθευμένων δεδομένων.



(α)



(β)

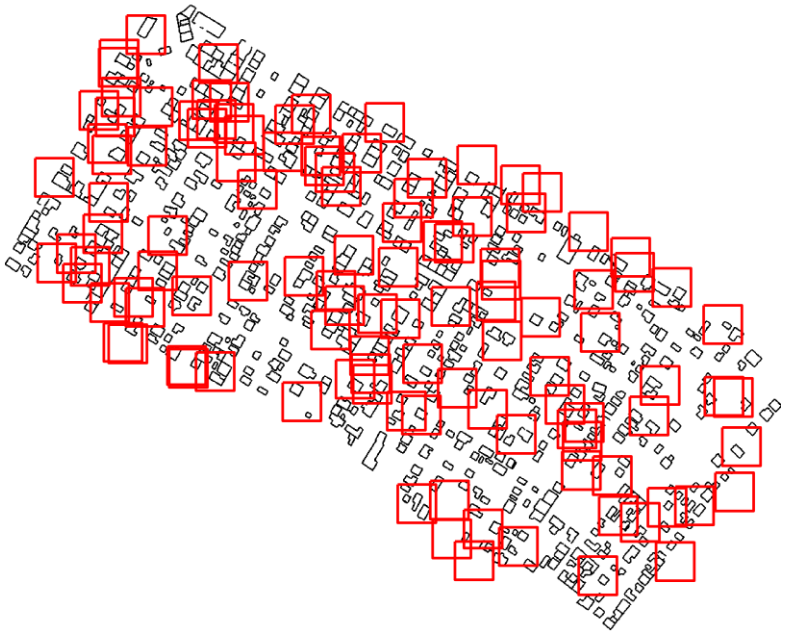
Εικόνα 35: (α) Αρχική εικόνα σαρωμένου τοπογραφικού χάρτη (β) Εικόνα επαληθευμένων δεδομένων (ground truth).

Η εκπαίδευση του U-Net βασίζεται σε ένα μεγάλο αριθμό ζευγαριών δειγμάτων εικόνας, τα οποία αντλούνται από την αρχική εικόνα και την εικόνα επαληθευμένων δεδομένων, αντίστοιχα. Η αρχική εικόνα (Εικόνα 35α) είναι κλίμακας του γκρι και απεικονίζει τμήμα του χάρτη της περιοχής μελέτης, ενώ η εικόνα επαληθευμένων δεδομένων (Εικόνα 35β), είναι μία δυαδική εικόνα, η οποία για την αντίστοιχη περιοχή, απεικονίζει μόνο τα κτίρια. Ένα τμήμα της αρχικής εικόνας χρησιμοποιείται

ως είσοδος στο CNN, ένα τμήμα εικόνας των επαληθευμένων δεδομένων, χρησιμοποιείται ως επιθυμητή έξοδος του CNN.

4.4 Τεχνικές λήψης των δειγμάτων της εικόνας

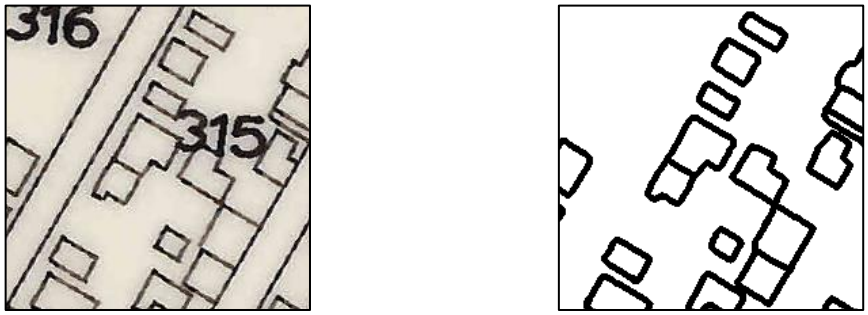
Τα δείγματα εικόνας δημιουργούνται χρησιμοποιώντας τρεις διαφορετικές προσεγγίσεις, οι οποίες ονομάζονται, “Τυχαία” (Random), “Πλεγματική-Τυχαία” (Grid-Random) και “Πλεγματική-Πλεγματική” (Grid-Grid). Οι τρεις αυτές προσεγγίσεις, διαφοροποιούνται μεταξύ τους, όσον αφορά τον τρόπο με τον οποίο τα δείγματα εικόνας καλύπτουν τη συνολική περιοχή της αρχικής εικόνας και της εικόνας επαληθευμένων δεδομένων.



Εικόνα 36: Στιγμιότυπο κατά την υλοποίηση της “Random” διεργασίας. Ο συνολικός αριθμός των ζητούμενων δειγμάτων εικόνας, δίνεται από τον χρήστη.

Στην “Random” περίπτωση, δείγματα εικόνας από την αρχική εικόνα και την εικόνα επαληθευμένων δεδομένων, επιλέγονται με τυχαίο τρόπο. Το μέγεθος του τμήματος εικόνας ορίζεται από το χρήστη και είναι μία παράμετρος του συστήματος. Για την αποφυγή αλληλοεπικαλύψεων μεταξύ διαφορετικών δειγμάτων εικόνας, η διεργασία καταγράφει τις εικονοσυντεταγμένες των ήδη δημιουργημένων δειγμάτων εικόνας και επιτρέπει τη δημιουργία νέων με την προϋπόθεση ότι οι εικονοσυντεταγμένες τους διαφέρουν από των προηγουμένων τουλάχιστον κατά ένα ελάχιστο αριθμό εικονοστοιχείων (minimum pixel difference). Επιπρόσθετα, μία παράμετρος ποσοστού κάλυψης (cover percentage), απορρίπτει ζευγάρια δειγμάτων εικόνας, των οποίων το τμήμα εικόνας επαληθευμένων δεδομένων περιέχει

ανεπαρκή αριθμό ενεργών (μαύρων) εικονοστοιχείων. Αυτός ο έλεγχος διασφαλίζει ότι τα δείγματα εικόνας επαληθευμένων δεδομένων που δημιουργούνται, περιέχουν έναν αποδεκτό, ελάχιστο αριθμό χρήσιμης πληροφορίας, δηλαδή εικονοστοιχεία που απαρτίζουν κτίρια. Η Εικόνα 36 δείχνει ένα στιγμιότυπο της "Random" διεργασίας δημιουργίας ζευγαριών δειγμάτων εικόνας. Οι ακριβώς ίδιες εικονοσυντεταγμένες εφαρμόζονται στην αρχική εικόνα και στην εικόνα επαληθευμένων δεδομένων, αντίστοιχα.



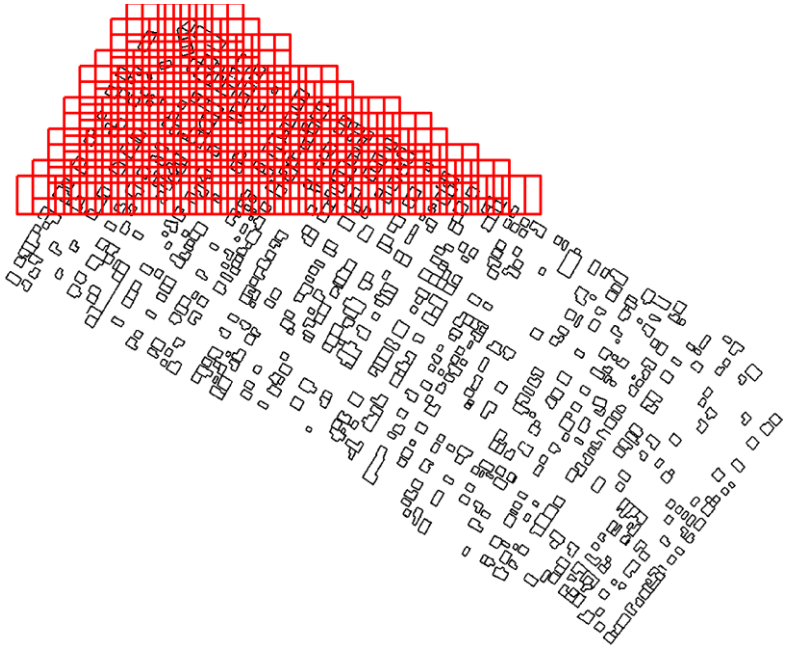
(α) (β)

Εικόνα 37: Παράδειγμα ζευγαριού δειγμάτων εικόνων μεγέθους 224x224. (α) Τμήμα της αρχικής εικόνας (β) Τμήμα εικόνας επαληθευμένων δεδομένων (ground truth).

Η Εικόνα 37 δείχνει ένα ζευγάρι δειγμάτων εικόνων. Το τμήμα στα αριστερά, έχει εξαχθεί από την αρχική εικόνα, ενώ το τμήμα στα δεξιά, έχει εξαχθεί από την εικόνα επαληθευμένων δεδομένων. Κατά τη διαδικασία της εκπαίδευσης του U-Net, το τμήμα στα δεξιά, χρησιμοποιείται ως είσοδος του δικτύου, ενώ το τμήμα στα αριστερά, χρησιμοποιείται ως η αντίστοιχη επιθυμητή έξοδος. Βάσει αυτής της επιθυμητής εξόδου, και της πραγματικής, δηλαδή της απόκρισης του δικτύου, υπολογίζεται η τιμή της συνάρτησης κόστους (Εξίσωση 4.3.1) και επαναληπτικά ανανεώνονται οι παράμετροι (βάρη και πολώσεις) του δικτύου.

Στην "Grid-Random" περίπτωση, τα δείγματα εικόνων από την αρχική εικόνα και την εικόνα επαληθευμένων δεδομένων, δημιουργούνται με μία σειριακή, "ολισθαίνοντος παραθύρου", προσέγγιση, βασιζόμενη σε ένα πλεγματικό βήμα (grid step), το οποίο ορίζει ο χρήστης. Η Εικόνα 38 απεικονίζει ένα παράδειγμα, μίας τέτοιας διεργασίας, για τη δημιουργία ζευγαριών δειγμάτων εικόνων μεγέθους 128x128. Πάλι ορίζεται από τον χρήστη η τιμή της παραμέτρου ποσοστού κάλυψης, ώστε να αποφευχθεί η συμπερίληψη λευκών δειγμάτων εικόνας ή δειγμάτων με πολύ λίγα μαύρα εικονοστοιχεία. Ενόσω τα δείγματα εικόνας δημιουργούνται με ένα σειριακό τρόπο, ανακατεύονται, στη συνέχεια, μέσω μιας διαδικασίας τυχαίας δεικτοδότησης. Η συγκεκριμένη προσέγγιση δειγματοληψίας, επιτρέπει στο δίκτυο να εκπαιδευτεί με δείγματα εικόνας τα οποία συστηματικά καλύπτουν, με μία πλεγματική δομή, το συνολικό εύρος της αρχικής εικόνας και της εικόνας

επαληθευμένων δεδομένων, και ταυτόχρονα διασφαλίζει ένα συγκεκριμένο ποσοστό διαφορετικότητας μεταξύ των δειγμάτων.

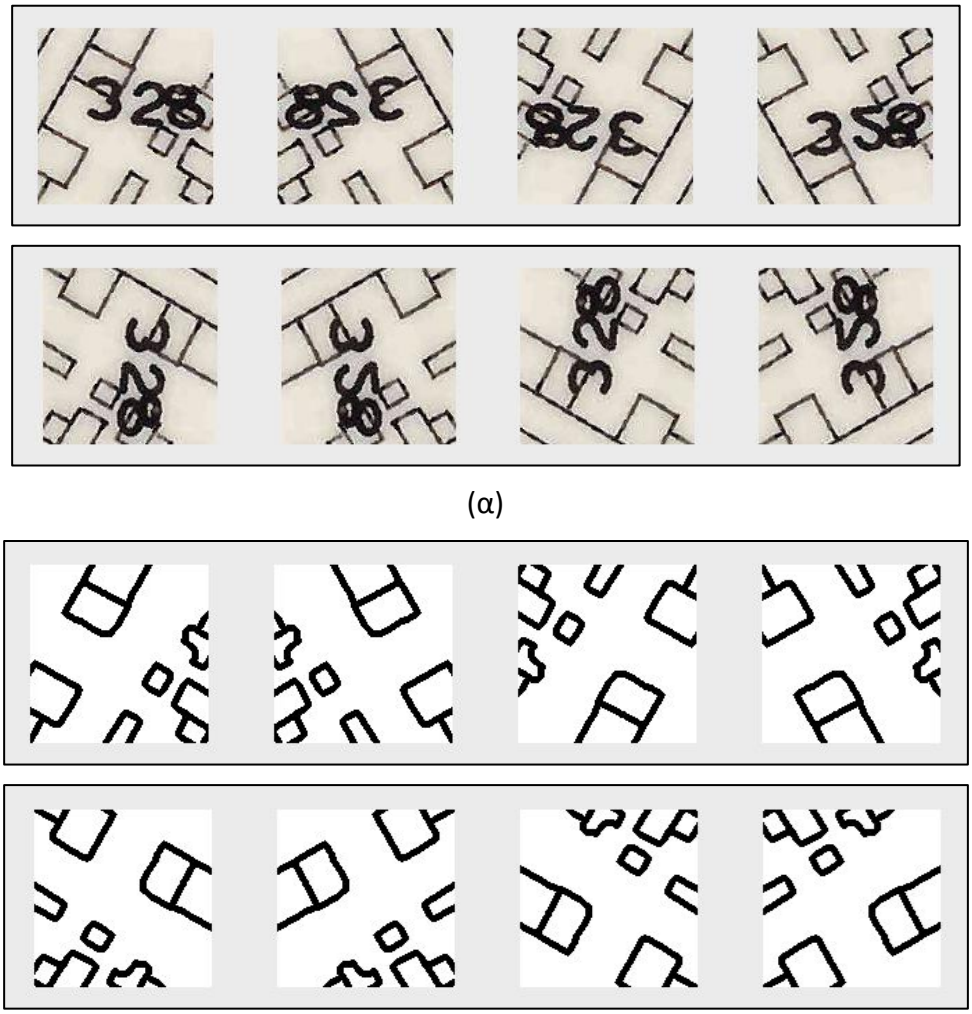


Εικόνα 38: Στιγμιότυπο κατά την υλοποίηση της "Grid-Random" και της "Grid-Grid" διεργασίας. Η σειριακή παραγωγή δειγμάτων εικόνας συνεχίζεται έως ότου να καλυφθεί το σύνολο της περιοχής.

Τέλος, στη "Grid-Grid" προσέγγιση, τα δείγματα εικόνας εξάγονται με παρόμοιο τρόπο με αυτό της "Grid-Random" προσέγγισης, αλλά διατηρούν την αρχική, σειριακή δεικτοδότηση τους, δηλαδή δεν εφαρμόζεται κάποια τυχαία δεικτοδότηση. Η συγκεκριμένη προσέγγιση, επιδρά στη διαδικασία της εκπαίδευσης, αφού δεν επιτρέπει στο δίκτυο να εκπαιδευτεί με ζευγάρια δειγμάτων εικόνων από όλο το εύρος της αρχικής εικόνας, και αυτής των επαληθευμένων δεδομένων. Συγκεκριμένα, τα διαφορετικά σετ δεδομένων που δημιουργούνται, διαχωρίζονται το κάθε ένα, σε: 50% για εκπαίδευση, 25% για επικύρωση και 25% για έλεγχο. Είναι σημαντικό να τονιστεί ότι η διαδικασία διαχωρισμού, πραγματοποιείται σειριακά, βάσει του δείκτη του κάθε τμήματος εικόνας. Συνεπώς, στα δύο πρώτα σενάρια, "Random" και "Grid-Random", τα διαφορετικά τμήμα καλύπτουν όλο το εύρος της αρχικής και της επαληθευμένων δεδομένων εικόνας. Αντίθετα, στην τρίτη μέθοδο δειγματοληψίας, αφού μόνο το 50% των παραγμένων δειγμάτων χρησιμοποιείται για εκπαίδευση, το δίκτυο κατά την εκπαίδευση "τροφοδοτείται" μόνο με δείγματα της περιοχής που καλύπτεται από το πρώτο 50% των δειγμάτων εικόνας.

Η παράμετρος της ελάχιστης διαφοράς εικονοστοιχείων, για την "Random" προσέγγιση, η παράμετρος πλεγματικού βήματος, για την "Grid-Random" και "Grid" προσέγγιση και η παράμετρος ελάχιστου ποσοστού κάλυψης, και για τις τρεις περιπτώσεις, επηρεάζουν το συνολικό αριθμό εξαγμένων ζευγαριών δειγμάτων

εικόνων. Για παράδειγμα, όσο μικρότερη είναι η τιμή του πλεγματικού βήματος ή η τιμή του ποσοστού κάλυψης, τόσο περισσότερα ζευγάρια δειγμάτων εικόνων μπορούν να εξαχθούν. Αντίθετα, η αύξηση της τιμής της ελάχιστης διαφοράς εικονοστοιχείων, μειώνει τον αριθμό των ζευγαριών δειγμάτων που μπορούν να εξαχθούν, από την “Random” διεργασία.



(α)

(β)

Εικόνα 39: Παράδειγμα επαύξησης δεδομένων (α) ένα τμήμα της αρχικής εικόνας (β) το αντίστοιχο τμήμα από εικόνα επαληθευμένων δεδομένων. Και στις δύο σειρές, η πρώτη εικόνα είναι από αυτές που παράχθηκαν από κάποια από τις τρεις διαφορετικές μεθόδους δειγματοληψίας, ενώ οι υπόλοιπες εφτά, είναι αποτέλεσμα της διαδικασίας επαύξησης δεδομένων.

Αφού δημιουργούνται τα δείγματα εικόνας, εφαρμόζονται επιπλέον τεχνικές επαύξησης τους, με σκοπό την αύξηση του μεγέθους των σετ δεδομένων, επιτρέποντας στο δίκτυο να “χτίσει” καλύτερα και πιο αξιόπιστα μοντέλα κατά την διαδικασία της εκπαίδευσης. Συγκεκριμένα, για κάθε ζευγάρι δειγμάτων από την αρχική, και την εικόνα επαληθευμένων δεδομένων, δημιουργούνται άλλες εφτά

εκδοχές του, μέσω τεχνητής επαύξησης, όπου πραγματοποιείται στροφή κατά 90, 180, 270 μοίρες, μαζί με μία οριζόντια ανάκλαση (horizontal mirroring). Αποτέλεσμα της διαδικασίας αυτής είναι ένα επαυξημένο σετ οκτώ ζευγαριών δειγμάτων, όπως φαίνεται στο παράδειγμα της Εικόνα 39.

4.5 Πειραματικά αποτελέσματα

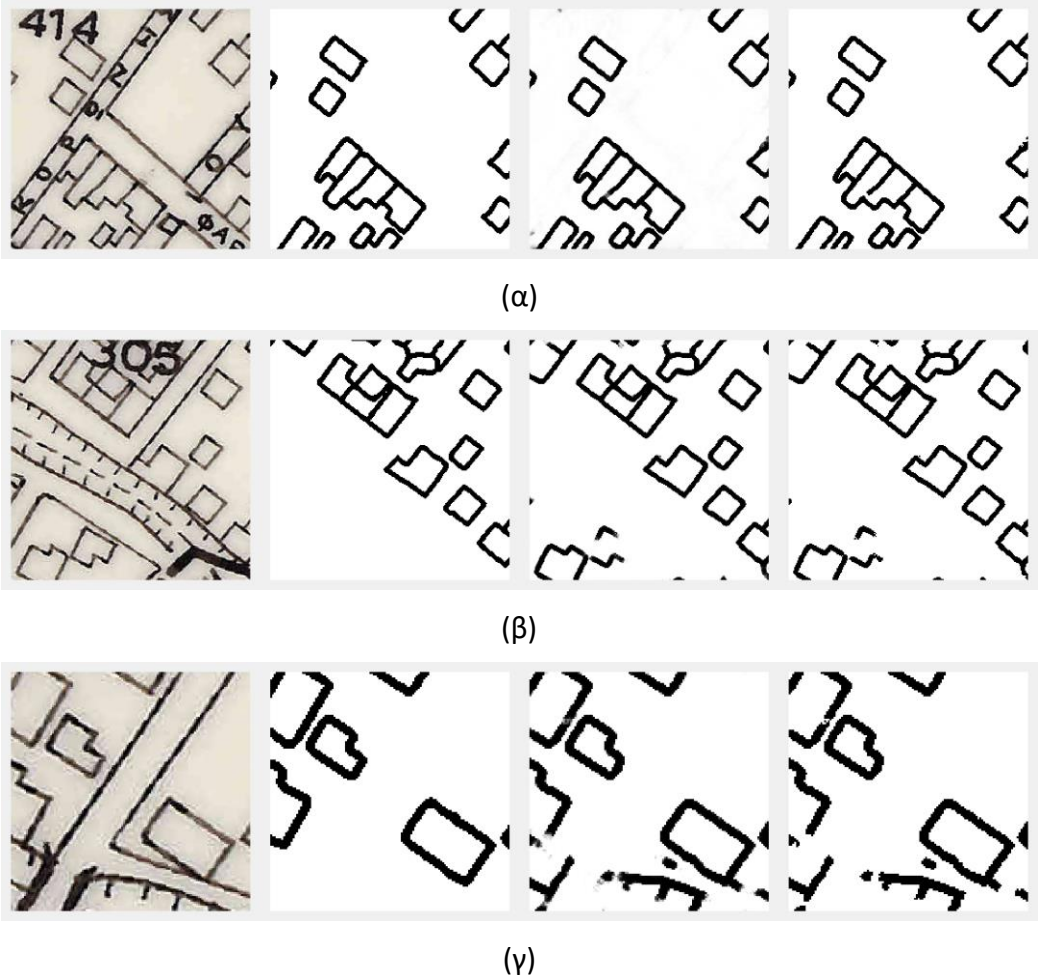
Η προτεινόμενη μέθοδος για την εξαγωγή των κτιρίων από τοπογραφικό χάρτη, δοκιμάστηκε μέσω διαφόρων πειραμάτων τα οποία μπορούν να ομαδοποιηθούν σε τρεις διαφορετικές προσεγγίσεις, ανάλογα με την μεθοδολογία δημιουργίας δειγμάτων εικόνων που χρησιμοποιήθηκε. Επιπρόσθετα, για κάθε διαφορετική προσέγγιση δημιουργίας δειγμάτων εικόνων, πραγματοποιήθηκαν τρία υπό-πειράματα, χρησιμοποιώντας διαφορετικό μέγεθος δειγμάτων εικόνων. Συγκεκριμένα, το πρώτο σετ δεδομένων περιλαμβάνει 16000 δείγματα εικόνας εισόδου και 16000 δείγματα εικόνας επαληθευμένων δεδομένων, μεγέθους 64x64 εικονοστοιχείων. Οι 2000 εξ αυτών εξάχθηκαν από την αρχική και την εικόνα επαληθευμένων δεδομένων, ενώ οι υπόλοιπες 14000 είναι αποτέλεσμα της τεχνητής επαύξησης δεδομένων. Το δεύτερο σετ δεδομένων περιλαμβάνει 8000 ζευγάρια δειγμάτων εικόνων, μεγέθους 128x128 εικονοστοιχείων. Αντίστοιχα, τα 1000 ζευγάρια είναι πρωτότυπα και τα υπόλοιπα 7000 ζευγάρια επαυξημένα. Τέλος, το τρίτο σετ περιλαμβάνει 4000 ζευγάρια δειγμάτων εικόνων, μεγέθους 224x224 εικονοστοιχείων με, αντίστοιχα, 500 ζευγάρια πρωτότυπα και 3500 ζευγάρια επαυξημένα. Ο Πίνακας 1 συνοψίζει τον αριθμό δειγμάτων που περιλαμβάνει το κάθε σετ δεδομένων πειράματος, όπως και τις τιμές των παραμέτρων που εφαρμόστηκαν για τη δημιουργία του.

		Random		Grid-Random & Grid-Grid	
Patch size	Patch pairs	Minimum cover percentage	Minimum pixel difference	Minimum cover percentage	Grid step
64x64	16000 (14000 augmented)	3%	3	3,60%	34
128x128	8000 (7000 augmented)	3%	3	2,30%	52
224x224	4000 (3500 augmented)	3%	3	3%	76

Πίνακας 1: Σετ δεδομένων πειραμάτων και παράμετροι δημιουργίας τους.

Με τα εννιά διαφορετικά σετ δεδομένων, πραγματοποιήθηκαν εννιά διαφορετικές διαδικασίες εκπαίδευσης. Σε όλες τις διαφορετικές εκπαιδεύσεις, χρησιμοποιήθηκε ο SGD, με τον αλγόριθμο βελτιστοποίησης Adam. Ο αριθμός εποχών ορίστηκε σε 100, χρησιμοποιώντας ένα mini-batch, ίσο με 8, με ένα βαθμό μάθησης ίσο με 0.001, ο

οποίος παρέμεινε σταθερός σε όλη τη διάρκεια των εννιά διαφορετικών εκπαιδεύσεων (δε χρησιμοποιήθηκε προγραμματισμός βαθμού μάθησης). Σχετικά με τις υπέρ-παραμέτρους του αλγόριθμου Adam, χρησιμοποιήθηκαν οι τιμές τις οποίες προτείνουν ως βέλτιστες στην εργασία τους (Diederik P. Kingma 2015), οι Kingma et al., εκτός της τιμής του συντελεστή μείωσης των παλαιότερων τετραγωνισμένων κλίσεων (β_2), όπου αντί για 0,999, εφαρμόστηκε τιμή ίση με 0,99.



(α)

(β)

(γ)

Εικόνα 40: Οπτικά αποτελέσματα της προτεινόμενης μεθόδου. Σε κάθε σειρά, η πρώτη εικόνα είναι τμήμα εικόνας από το σετ δεδομένων ελέγχου, η δεύτερη εικόνα είναι το αντίστοιχο τμήμα εικόνας επαληθευμένων δεδομένων (ground truth), η τρίτη εικόνα είναι η πρόβλεψη του δικτύου και η τέταρτη εικόνα αντιστοιχεί στην πρόβλεψη του δικτύου, σε δυαδική (binarized) μορφή. (α) Παράδειγμα 224×224 "Grid-Random" (β) Παράδειγμα 224×224 "Random" (c) Παράδειγμα 128×128 "Grid-Grid".

Στην Εικόνα 40, παρουσιάζονται τρία διαφορετικά παραδείγματα προβλέψεων του προτεινόμενου CNN. Σε κάθε μία από τις περιπτώσεις, η πρώτη εικόνα είναι το τμήμα εικόνας εισόδου από το σετ δεδομένων ελέγχου, η δεύτερη εικόνα είναι το αντίστοιχο τμήμα εικόνας επαληθευμένων δεδομένων, η τρίτη εικόνα είναι η ανεπεξέργαστη απόκριση-σύνολο "πυκνών προβλέψεων" του δικτύου και η τέταρτη εικόνα η επεξεργασμένη, σε δυαδική μορφή, πρόβλεψη του δικτύου. Η Εικόνα 40α

απεικονίζει ένα παράδειγμα δεδομένων μεγέθους 224x224, προερχόμενων από την “Grid-Random” μέθοδο. Όπως φαίνεται το εκπαιδευμένο CNN, αφαίρεσε αποτελεσματικά τους αριθμούς και τα όρια των οικοδομικών τετραγώνων, όπως και τις ονομασίες οδών.

Η Εικόνα 40β απεικονίζει ένα παράδειγμα “Random” δειγματοληψίας, όπου φαίνεται ότι το δίκτυο έχει αφαιρέσει τον αριθμό οικοδομικού τετραγώνου, ο οποίος αλληλεπικαλύπτεται με διάφορα κτίρια. Ακόμα έχει αφαιρεθεί η βαθιά γραμμή υδατορεύματος και τα φρύδια πρανών και στις δύο πλευρές του. Το ενδιαφέρον εδώ, είναι ότι το δίκτυο έχει εξαγάγει και δύο κτίρια, κάτω αριστερά, το ένα όχι τόσο ορθά, παρόλο που, εσφαλμένα, δεν υπάρχουν στο αντίστοιχο τμήμα εικόνας επαληθευμένων δεδομένων, εξαιτίας του ότι το τμήμα αυτό έχει “κοπεί” στα όρια της περιοχής του ground truth. Η Εικόνα 40γ αναφέρεται σε ένα “Grid-Grid” παράδειγμα δειγματοληψίας. Φαίνεται ότι το δίκτυο έχει ορθώς αναγνωρίσει τα κτίρια, αλλά δεν έχει αφαιρέσει αρκετά αποτελεσματικά το ένα τμήμα της γέφυρας και το φρύδι πρανούς της μίας πλευράς. Αυτό συμβαίνει εξαιτίας του τρόπου που δημιουργούνται τα δεδομένα με την “Grid-Grid” μέθοδο, ο οποίος έχει σαν αποτέλεσμα το δίκτυο να μην έχει εκπαιδευτεί με δείγματα εικόνας που να περιλαμβάνουν κάποιον παρόμοιο συμβολισμό, ώστε κατά τη διαδικασία ελέγχου να είναι σε θέση να τον αφαιρέσει.

Οι μετρικές που χρησιμοποιήθηκαν για την αξιολόγηση των εννιά διαφορετικά εκπαιδευμένων δικτύων, των πειραμάτων, βασίζονται σε συγκρίσεις επιπέδου εικονοστοιχείου, μεταξύ του κάθε τμήματος εικόνας επαληθευμένων δεδομένων του σετ ελέγχου και της κάθε αντίστοιχης πρόβλεψης δυαδικής μορφής του δικτύου, καταμετρώντας τα αληθώς θετικά (true positives-TP), τα εσφαλμένα θετικά (false positives-FP) και τα εσφαλμένα αρνητικά (false negatives-FN) εικονοστοιχεία, αντίστοιχα.

Για τα δείγματα εικόνας επαληθευμένων δεδομένων, όπως και για την αντίστοιχη πρόβλεψη δυαδικής μορφής, τα λευκά εικονοστοιχεία λαμβάνονται υπόψη ως τάξη παρασκηνίου και τα μαύρα εικονοστοιχεία λαμβάνονται υπόψη ως τάξη προσκηνίου. Οι μετρικές που χρησιμοποιήθηκαν είναι:

- *Καθολική Ακρίβεια (Global Accuracy):* Ο λόγος των εικονοστοιχείων που προβλέφθηκαν ορθά, ανεξαρτήτου τάξης, ως προς τον συνολικό αριθμό εικονοστοιχείων.
- *Μέση Ακρίβεια (Mean Accuracy):* Η μέση ακρίβεια όλων των τάξεων, σε όλα τα δείγματα εικόνας, όπου για κάθε τάξη, η ακρίβεια ορίζεται ως ο λόγος των ορθά ταξινομημένων εικονοστοιχείων, προς τον συνολικό αριθμό εικονοστοιχείων στην συγκεκριμένη τάξη, βάσει του τμήματος εικόνας επαληθευμένων δεδομένων, δηλαδή, το σκορ ακρίβειας (accuracy score) = TP/(TP+FN).

- *Μέση Τομή προς Ένωση (Mean Intersection over Union – Mean IoU):* Το μέσο σκορ Τομής προς Ένωση, όλων των τάξεων, σε όλες τις εικόνες, όπου για κάθε τάξη, το σκορ Τομής προς Ένωση είναι ο λόγος των ορθά ταξινομημένων εικονοστοιχείων, προς το συνολικό αριθμό των εικονοστοιχείων επαληθευμένων δεδομένων (ground truth pixels) και αυτών που έχουν προβλεφθεί, για την αντίστοιχη τάξη, δηλαδή σκορ Τομής προς Ένωση (IoU score) = TP / (TP + FP + FN).
- *Σταθμισμένη Τομή προς Ένωση (Weighted Intersection over Union):* Ο σταθμισμένος μέσος του σκορ Τομής προς Ένωση, όλων των τάξεων, σε όλες τις εικόνες, όπου για κάθε τάξη, ο σταθμισμένος μέσος Τομής προς Ένωση, είναι το μέσο σκορ Τομής προς Ένωση, σταθμισμένο βάσει του αριθμού εικονοστοιχείων που ανήκουν στην αντίστοιχη τάξη.
- *Μέσο F1 Σκορ Περιγράμματος (Mean Boundary F1 (BF) Score):* Το σκορ ταιριάσματος περιγράμματος, το οποίο υποδεικνύει κατά πόσο το περίγραμμα που προβλέφθηκε, για κάθε τάξη, συμφωνεί με το αντίστοιχο πραγματικό περίγραμμα. Για το συνολικό σετ δεδομένων, το Μέσο F1 Σκορ Περιγράμματος, ισούται με τον μέσο όρο του αντίστοιχου σκορ, όλων των τάξεων, σε όλες τις εικόνες και για κάθε τάξη, ισούται με το αντίστοιχο σκορ, υπολογισμένο για κάποια τάξη, για όλες τις εικόνες.

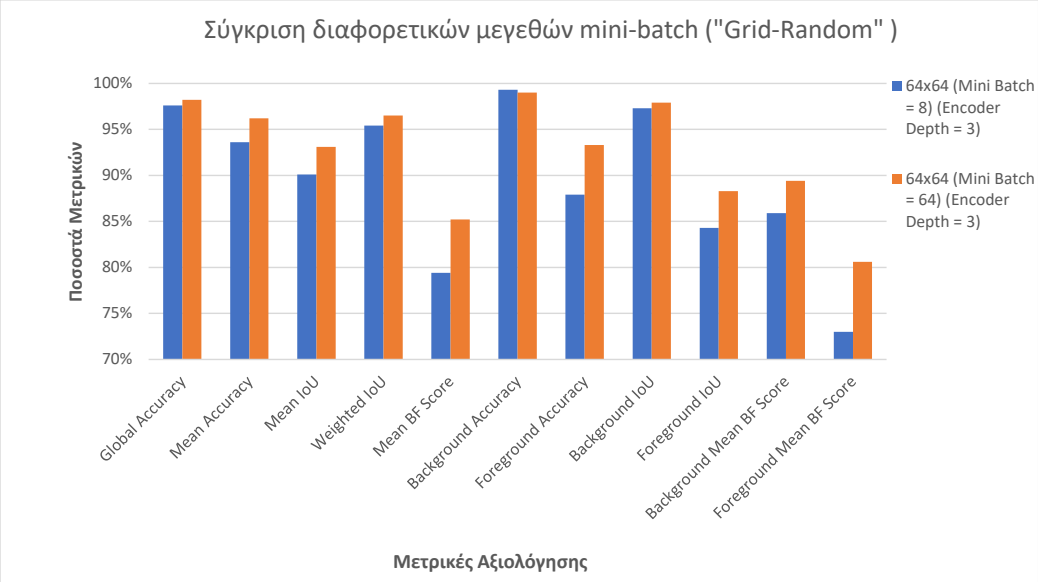
Τα αποτελέσματα των παραπάνω μετρικών, για όλες, τις εννιά, περιπτώσεις πειραμάτων, συνοψίζονται στον Πίνακας 2. Αναφορικά με τα συγκεκριμένα μετρικά, τα πιο χαρακτηριστικά για την αξιολόγηση του κάθε πειράματος, είναι αυτά που αναφέρονται στην τάξη προσκηνίου (foreground class), τα μαύρα εικονοστοιχεία, δηλαδή, αφού η συγκεκριμένη τάξη περιλαμβάνει τα γεωγραφικά χαρακτηριστικά ενδιαφέροντος, δηλαδή, τα κτίρια. Από τον Πίνακας 2 μπορεί να παρατηρηθεί, ότι και η “Random”, και η “Grid-Random” μέθοδος, βελτιώνουν την απόδοση τους σύμφωνα με το μέγεθος του τμήματος εικόνας, κάτι το οποίο φαίνεται να μην συμβαίνει στις περισσότερες περιπτώσεις της “Grid-Grid” μεθόδου. Επιπρόσθετα, οι περιπτώσεις “Random” και “Grid-Random”, φαίνεται να παρέχουν υψηλή ακρίβεια αναγνώρισης, ειδικά για τα μεγέθη δειγμάτων εικόνας 128x128 και 244x224.

Η προσέγγιση δειγματοληψίας στις δύο αυτές περιπτώσεις, δίνει τη δυνατότητα στο CNN να εκπαιδευτεί με δείγματα εικόνας από διάφορες περιοχές της αρχικής εικόνας και της εικόνας επαληθευμένων δεδομένων, με αποτέλεσμα να είναι πιο εύρωστο όσον αφορά τις προβλέψεις που πραγματοποιεί. Στην περίπτωση της “Grid-Grid” προσέγγισης, τα μετρικά δεν είναι τόσο υψηλά, εξαιτίας του ότι δεν έχουμε επιτρέψει στο δίκτυο να εκπαιδευτεί με δείγματα που να καλύπτουν όλο το χωρικό εύρος της αρχικής εικόνας και αυτής των επαληθευμένων δεδομένων. Η μέγιστη καθολική ακρίβεια, 99.1%, επιτυγχάνεται με την “Grid-Random” μέθοδο δειγματοληψίας, για μεγέθη δειγμάτων 128×128 και 224×224.

Evaluation Metrics		Random			Grid-Random			Grid-Grid		
		64x64	128x128	224x224	64x64	128x128	224x224	64x64	128x128	224x224
Global Accuracy		97.5%	98.6%	98.9%	97.6%	99.1%	99.1%	95.4%	95.9%	95.1%
Mean Accuracy		93.2%	96.9%	97.8%	93.6%	97.9%	98.2%	91%	89.8%	80.7%
Mean IoU		90.2%	93.8%	94.7%	90.1%	95.9%	95.8%	83.5%	82.4%	75.7%
Weighted IoU		95.1%	97.2%	97.8%	95.4%	98.2%	98.3%	91.6%	92.5%	90.9%
Mean BF Score		78.3%	96.2%	96.8%	79.4%	97.2%	97.3%	70.4%	85.6%	82.5%
Accuracy	Background	99.2%	99.1%	99.2%	99.3%	99.5%	99.4%	97.1%	97.7%	98.8%
	Foreground	87.2%	94.7%	96.5%	87.9%	96.4%	97%	84.8%	81.8%	62.5%
IoU	Background	97.1%	98.4%	98.7%	97.3%	99%	99%	94.8%	95.4%	94.8%
	Foreground	83.2%	89.1%	90.6%	84.3%	92.8%	92.6%	72.2%	69.3%	56.7%
Mean BF Score	Background	85.3%	97.3%	97.7%	85.9%	98.1%	98%	79.4%	89.4%	86.8%
	Foreground	71.2%	95.1%	95.9%	73%	96.4%	96.6%	61.4%	81.8%	78.1%

Πίνακας 2: Μετρικά αξιολόγησης των εννιά διαφορετικών πειραμάτων.

Στην συνέχεια, πραγματοποιήθηκε ένα πείραμα, με σκοπό την εξαγωγή συμπερασμάτων σχετικά με την επίδραση του μεγέθους του mini-batch, στην απόδοση των εκπαιδευμένων συνελικτικών δικτύων. Πραγματοποιήθηκε εκπαίδευση με δεδομένα “Grid-Random” δειγμάτων εικόνων μεγέθους 64×64 εικονοστοιχείων, με τις ακριβώς ίδιες παραμέτρους για τον αλγόριθμο Adam, για ίδιο αριθμό εποχών (100), με μοναδική διαφορά το μέγεθος του mini-batch, το οποίο αντί για 8, που είχε οριστεί για τα παραπάνω πειράματα, ορίστηκε σε 64. Τα συγκριτικά αποτελέσματα φαίνονται στο γράφημα της Εικόνα 41 .



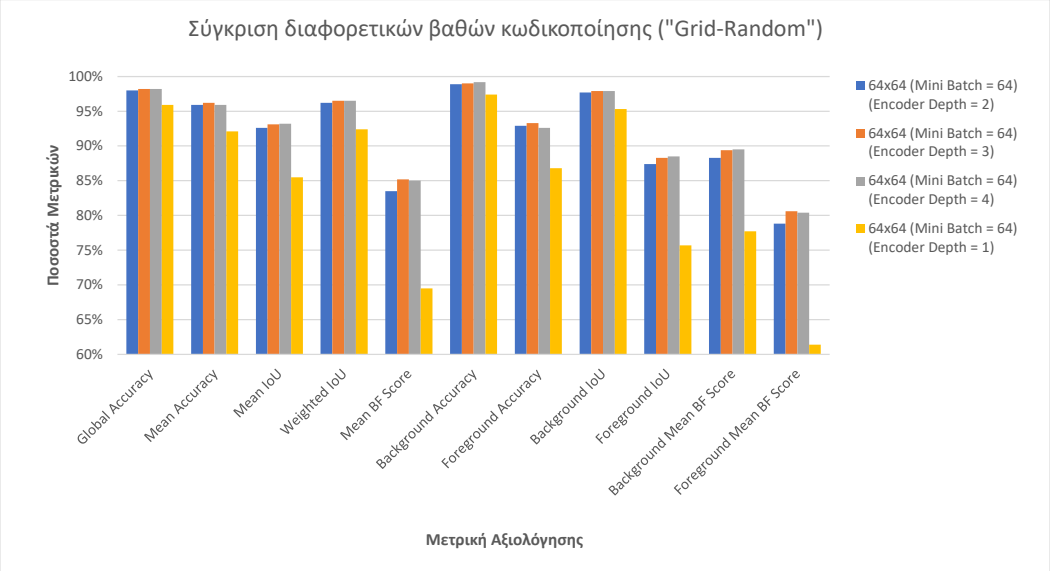
Εικόνα 41: Σύγκριση διαφορετικών μεγεθών mini-batch.

Στο γράφημα της Εικόνα 41, φαίνεται ότι σχεδόν όλες οι μετρικές, είναι υψηλότερα στην περίπτωση όπου η εκπαίδευση έχει πραγματοποιηθεί με mini-batch μεγέθους 64, και όχι 8. Αυτό είναι κάτι το οποίο, γενικώς, δε συμβαίνει και τόσο συχνά. Το

μέγεθος του mini-batch είναι μία από τις σημαντικότερες υπέρ-παραμέτρους, κατά την βέλτιστη ρύθμιση (fine-tuning) συστημάτων βαθιάς μάθησης.

Τα μεγαλύτερα μεγέθη mini-batch, κατά την εκπαίδευση, έχουν σαν αποτέλεσμα, η όλη διαδικασία να γίνεται λιγότερο υπολογιστικά χρονοβόρα, αλλά στον αντίποδα, θεωρείται ότι λειτουργούν ως αρνητικός παράγοντας ως προς τη γενίκευση του αλγόριθμου. Μάλιστα, τα μικρά mini-batch εισάγουν ένα είδος θορύβου κατά την διαδικασία της εκπαίδευσης, ο οποίος καθιστά το μοντέλο πιο εύρωστο και λιγότερο επιρρεπές σε προβλήματα υπέρ-προσαρμογής. Στη συγκεκριμένη περίπτωση, τα παραπάνω αποτελέσματα, είναι πιθανό να σχετίζονται με το ότι το μέγεθος των δειγμάτων εικόνων (64×64 εικονοστοιχεία) είναι πολύ μικρό, με αποτέλεσμα ένα μεγαλύτερο mini-batch να βοηθά τον αλγόριθμο να καταλήξει σε βελτιωμένες παραμέτρους και άρα σε υψηλότερη, συνολικά, απόδοση.

Στην συνέχεια, πραγματοποιήθηκε ένα ακόμη πείραμα, σε σχέση με τον αντίκτυπο του βάθους κωδικοποίησης (encoder depth) της αρχιτεκτονικής U-Net στην απόδοση των συνελικτικών δικτύων. Για τις τέσσερεις διαφορετικές εκπαιδεύσεις, των οποίων τα αποτελέσματα παρουσιάζονται στη συνέχεια, χρησιμοποιήθηκαν δεδομένα δειγματοληψίας “Grid=Random”, δείγματα εικόνων 64×64 εικονοστοιχείων, οι ίδιες παράμετροι για τον αλγόριθμο βελτιστοποίησης Adam, ο ίδιος αριθμός εποχών (100) και μέγεθος mini-batch, ίσο με 64. Τα αποτελέσματα παρουσιάζονται στο γράφημα της Εικόνα 42.



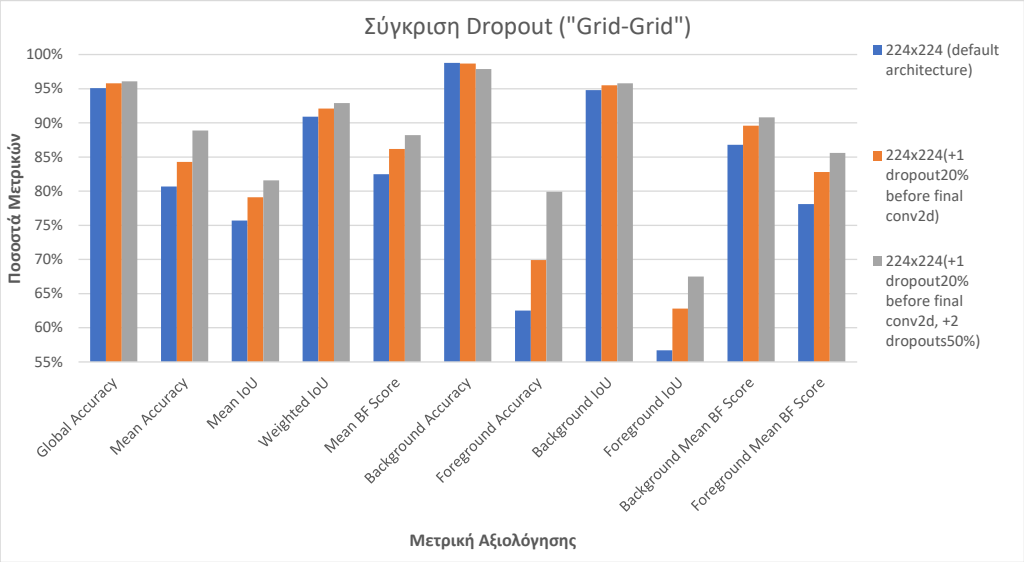
Εικόνα 42: Σύγκριση διαφορετικών βαθών κωδικοποίησης (encoder depth).

Το συμπέρασμα που προκύπτει είναι ότι η αρχιτεκτονική U-Net, με βάθος κωδικοποίησης ίσο με 1, δίνει χαμηλά αποτέλεσμα σε αντίθεση με τις άλλες τρεις περιπτώσεις. Αυτό είναι κάτι αναμενόμενο, αφού εφαρμόζοντας μοναδιαίο βάθος

κωδικοποίησης, αυτομάτως σταματά να υφίσταται διαδικασία βαθιάς μάθησης, αφού η αρχιτεκτονική του U-Net στην συγκεκριμένη περίπτωση, είναι “αβαθής” (shallow architecture). Όπως περιεγράφηκε παραπάνω, το βάθος κωδικοποιητή προσδιορίζει των αριθμών των σταδίων επεξεργασίας των δειγμάτων εικόνας εισόδου, εντός του δικτύου, για τον κωδικοποιητή και αποκωδικοποιητή, αντίστοιχα. Κάθε στάδιο κωδικοποιητή αποτελείται από δύο σετ με συνελικτικά και ReLU επίπεδα, ακολουθούμενα από ένα επίπεδο 2x2 μέγιστης συγκέντρωσης. Κάθε στάδιο αποκωδικοποιητή, αποτελείται από ένα επίπεδο ανεστραμμένης συνέλιξης, ακολουθούμενο από δύο σετ συνελικτικών και ReLU επιπέδων. Όσο λιγότερα είναι τα στάδια επεξεργασίας από τα οποία αποτελείται ένα δίκτυο, τόσο πιο ανίσχυρο καθίσταται όσον αφορά τον εντοπισμό πολύπλοκων μοτίβων, στα δεδομένα με τα οποία τροφοδοτείται. Από το γράφημα της Εικόνα 42 φαίνεται επίσης ότι οι αποδόσεις των δικτύων για βάθη κωδικοποίησης 2,3 και 4, είναι αρκετά κοντά. Λαμβάνοντας υπόψη i) την έστω μικρή υπεροχή της απόδοσης με βάθος 3, συγκριτικά με την απόδοση με βάθος 2 που είναι σημαντική για τέτοιου είδους εργασίες και ii) ότι η υψηλή πολυπλοκότητα ενός δικτύου βάθους 4 συχνά οδηγεί σε προβλήματα υπέρ-προσαρμογής καθώς και ότι η αρχιτεκτονική βάθους 4 είναι προφανώς πιο υπολογιστικά χρονοβόρα, φαίνεται ότι η βέλτιστη επιλογή είναι αυτή του βάθους 3.

Το τελευταίο πείραμα, σχετίζεται με την παρατήρηση ότι σε αρκετές από τις διαδικασίες εκπαίδευσης των παραπάνω πειραμάτων εμφανίζεται το πρόβλημα της υπέρ-προσαρμογής. Το συγκεκριμένο πρόβλημα παρατηρήθηκε κυρίως κατά τις εκπαιδεύσεις με δεδομένα δειγματοληψίας “Grid-Grid”, όπου, όπως έχει ήδη αναφερθεί, ο αλγόριθμος “βλέπει” δεδομένα κατά την διάρκεια της εκπαίδευσης, όχι πλήρως αντιπροσωπευτικά του συνόλου των δεδομένων, με αποτέλεσμα τελικά να ελέγχεται ως προς την απόδοση του σε περιεχόμενο δειγμάτων εικόνας, για τα οποία δεν έχει εκπαιδευτεί καθόλου ή ελάχιστα. Θα μπορούσε κανείς να πει ότι η υπέρ-προσαρμογή στην συγκεκριμένη περίπτωση είναι φαινομενική και ότι το δίκτυο στην πραγματικότητα τροφοδοτείται με δεδομένα κακής ποιότητας και ανάλογα αποδίδει. Για την διαλεύκανση του συγκεκριμένου θέματος πραγματοποιήθηκε ένα πείραμα όπου τροποποιήθηκε η συνολική αρχιτεκτονική του U-Net, η οποία χρησιμοποιήθηκε σε όλες τις προηγούμενες περιπτώσεις, προσθέτοντας διαφορετικό αριθμό Dropout επιπέδων. Πραγματοποιήθηκαν δύο διαφορετικές τροποποιήσεις στην προκαθορισμένη αρχιτεκτονική, για την οποία πρέπει να επισημανθεί, πως όπως φαίνεται και στην Εικόνα 33, περιλαμβάνει ένα επίπεδο Dropout, με $p = 50\%$, στο τμήμα “γέφυρας”, μεταξύ κωδικοποιητή και αποκωδικοποιητή και ένα επίπεδο Dropout, επίσης με $p = 50\%$, σε στάδιο του κωδικοποιητή. Η πρώτη τροποποίηση αφορά την πρόσθεση ενός Dropout επιπέδου, με $p = 20\%$, πριν το τελευταίο συνελικτικό επίπεδο του δίκτυο. Η δεύτερη τροποποίηση αφορά επίσης την πρόσθεση ενός Dropout επιπέδου, πριν το τελευταίο συνελικτικό επίπεδο, με p=20%, αλλά αφορά επίσης και την πρόσθεση δύο άλλων Dropout επιπέδων, με $p = 50\%$ το

καθένα, εκ των οποίων το ένα τοποθετήθηκε ενδιάμεσα του πρώτου σταδίου του κωδικοποιητή και το άλλο ενδιάμεσα του πρώτου σταδίου του αποκωδικοποιητή. Εφαρμόστηκαν παράμετροι για τον Adam, ίδιες με όλες τις προηγούμενες περιπτώσεις, αριθμός εποχών ίσος με 100, μέγεθος mini-batch ίσο με 8 και χρησιμοποιήθηκαν δεδομένα εικόνων 224×224, δειγματοληψίας “Grid-Grid”.



Εικόνα 43: Σύγκριση εφαρμογής διαφορετικού αριθμού Dropout επιπέδων, εντός του δικτύου.

Όπως φαίνεται από το γράφημα της Εικόνα 43, η εφαρμογή περισσοτέρων επιπέδων Dropout, από αυτά της προκαθορισμένης αρχιτεκτονικής, φαίνεται να αντιμετωπίζει επιτυχημένα το πρόβλημα της υπέρ-προσαρμογής, με αποτέλεσμα για το ίδιο ακριβώς πειραματικό σενάριο, η απόδοση του δικτύου, βάσει των μετρικών αξιολόγησης, να βελτιώνεται σε σημαντικό βαθμό στην πλειοψηφία των μετρικών αξιολόγησης.

4.6 Υλοποίηση

Η υλοποίηση όλων των παραπάνω πραγματοποιήθηκε σε υπολογιστικό σύστημα με Intel Xeon X5690 στα 3.47GHz (2 processors) με 96 GB RAM χρησιμοποιώντας το προγραμματιστικό περιβάλλον MATLAB, έκδοσης R2020b. Η επιλογή του MATLAB, πραγματοποιήθηκε εξαιτίας εξειδικευμένων εργαλειοθηκών τις οποίες διαθέτει, οι οποίες παρέχουν τη δυνατότητα σχεδιασμού και εκπαίδευσης μοντέλων βαθιών νευρωνικών δικτύων, μέσω της χρήσης κατάλληλων εντολών. Ακόμα, υπάρχει πλούσιο υλικό, σχετικά με το συγκεκριμένο περιβάλλον, σε διαδικτυακούς τόπους δημόσιας συζήτησης, κάτι το οποίο καθιστά την αντιμετώπιση προβλημάτων, τα οποία εμφανίζονται κατά την προσπάθεια πραγματοποίησης τέτοιου είδους εγχειρημάτων, πιο εύκολη.

Κατά την υλοποίηση του συνόλου των διαδικασιών, δημιουργήθηκαν τρία διαφορετικά αρχεία MATLAB. Κατά την εκτέλεση του πρώτου αρχείου, πραγματοποιείται η δημιουργία του σετ δεδομένων με τα ζευγάρια τμημάτων εικόνων, ανάλογα την επιλογή του χρήστη σχετικά με το μέγεθος αυτών, τον αριθμό τους, τη μέθοδο δειγματοληψίας και τις διάφορες παραμέτρους κάθε μίας εξ αυτών. Τα ζευγάρια τμημάτων εικόνων που δημιουργούνται αποθηκεύονται, τα μεν εισόδου, σε ένα ξεχωριστό φάκελο, και τα δε επαληθευμένων δεδομένων, σε έναν άλλο φάκελο. Το δεύτερο αρχείο το οποίο δημιουργήθηκε, είναι το βασικό και κατά την εκτέλεση του, αρχικά δημιουργούνται δύο διαφορετικές αποθήκες δεδομένων εικόνας (συνάρτηση imageDatastore), μία για τα τμήματα εικόνας εισόδου και μία για τα τμήματα εικόνας επαληθευμένων δεδομένων. Οι δύο αυτές αποθήκες δεδομένων εικόνας χωρίζονται ποσοστιαία σε εκπαίδευσης, επικύρωσης και ελέγχου. Στη συνέχεια δημιουργείται η αρχιτεκτονική του συνελικτικού νευρωνικού δικτύου (συνάρτηση unetLayers), ορίζονται η παράμετροι εκπαίδευσης του (συνάρτηση trainingOptions), όπως ο αλγόριθμος βελτιστοποίησης, ο αριθμός εποχών εκπαίδευσης και το μέγεθος mini-batch, και πραγματοποιείται η εκπαίδευση του (συνάρτηση trainNetwork, παρέχεται από την εργαλειοθήκη Deep Learning). Στα πειράματα κατά τα οποία πραγματοποιήθηκε παρέμβαση στην αρχιτεκτονική του δικτύου, προσθέτοντας επίπεδα dropout, χρησιμοποιήθηκε η εφαρμογή Deep Network Designer, της εργαλειοθήκης Deep Learning. Από το δεύτερο αρχείο, τελικά, εκτός των άλλων, εξάγονται και αποθηκεύονται πολλαπλά εκπαιδευμένα CNNs, ανάλογα με την πάροδο της διαδικασίας της εκπαίδευσης. Κατά την εκτέλεση του τρίτου αρχείου, το οποίο δημιουργήθηκε, επιλέγεται από το χρήστη και “φορτώνεται” το εκπαιδευμένο CNN, το οποίο προέκυψε από το δεύτερο αρχείο. Το εκπαιδευμένο αυτό CNN, “τροφοδοτείται” με τα τμήματα εικόνας εισόδου του σετ δεδομένων ελέγχου και εξάγονται τα εκτιμώμενα τμήματα εικόνας (predicted images). Αυτά στην συνέχεια μετατρέπονται σε δυαδική μορφή (συνάρτηση imbinarize, χρησιμοποιήθηκε BinarizeThreshold=0.3). Στη συνέχεια ορίζονται δύο διαφορετικές τάξεις (παρασκήνιο, προσκήνιο), με αναγνωριστικά ετικέτας 1 και 0, αντίστοιχα, ανάλογα το χρώμα του κάθε εικονοστοιχείου και δημιουργούνται δύο διαφορετικές αποθήκες ετικετών εικονοστοιχείων (συνάρτηση pixelLabelDatastore), μία για τα εκτιμώμενα τμήματα εικόνας και μία για τα αντίστοιχα τμήματα εικόνας επαληθευμένων δεδομένων (ground truth). Τέλος, πραγματοποιείται ο υπολογισμός των μετρικών αξιολόγησης, συγκρίνοντας τις δύο αυτές αποθήκες ετικετών εικονοστοιχείων (συνάρτηση evaluateSemanticSegmentation).

5 Συμπεράσματα

Στο πλαίσιο της παρούσας διπλωματικής εργασίας, παρουσιάζεται μία προσέγγιση βαθιάς μάθησης, για την επίλυση του προβλήματος του εντοπισμού κτιρίων, σε τοπογραφικούς χάρτες παλαιότερων περιόδων. Για την επίτευξη του στόχου αυτού, εκπαιδεύτηκε ένα βαθύ συνελικτικό νευρωνικό δίκτυο, αρχιτεκτονικής U-Net, με τη μέθοδο της από εικόνα σε εικόνα, παλινδρόμησης. Τα πειράματα, τα οποία πραγματοποιήθηκαν με δεδομένα τα οποία αντλήθηκαν από ιστορικό τοπογραφικό χάρτη, δείχνουν ότι η προτεινόμενη μέθοδος εντοπίζει τα κτίρια με σημαντική ακρίβεια, ακόμα και σε περιπτώσεις όπου η περιβάλλουσα γραφική πληροφορία είναι πυκνή και υφίστανται αλληλοεπικαλύψεις με άλλου είδος περιεχόμενο, όπως κείμενο ή άλλα γραφικά στοιχεία.

5.1 Γενικά συμπεράσματα

Η αξιολόγηση της απόδοσης της προτεινόμενης μεθόδου μέσω κατάλληλων μετρικών παρέχει πολύ ικανοποιητικά αποτελέσματα, ειδικά στην περίπτωση όπου τα δείγματα εικόνας, που χρησιμοποιούνται για την εκπαίδευση του δικτύου, συλλέγονται τυχαία. Αυτή είναι και η ενδεδειγμένη τακτική, κατά την εκπαίδευση αλγόριθμων επιβλεπόμενης μάθησης, δηλαδή, ο αλγόριθμος να τροφοδοτηθεί με δεδομένα εκπαίδευσης, τα οποία να είναι όσο το δυνατόν πιο αντιπροσωπευτικά του συνολικού σετ δεδομένων, αφού η ποιότητα του σετ δεδομένων εκπαίδευσης, είναι ίσως ο σημαντικότερος από τους παράγοντες οι οποίοι επηρεάζουν την απόδοση συστημάτων μηχανικής μάθησης.

Η αρχιτεκτονική U-Net, δείχνει να είναι κατάλληλη για αυτό το πρόβλημα και αποδίδει καλύτερα όταν χρησιμοποιείται κατάλληλο βάθος κωδικοποιητή, ο οποίος προσδιορίζει τον αριθμό σταδίων, με διαδοχικά επίπεδα, σε κωδικοποιητή και αποκωδικοποιητή. Η ισχύς της συγκεκριμένης αρχιτεκτονικής, φαίνεται και από την απόδοση της κατά την εξάλειψη από την τελική έξοδο του δικτύου τόσο του κειμένου όσο και άλλων γραφικών στοιχείων που δεν είναι επιθυμητά κατά την διαδικασία εκπαίδευσης.

Τα πειράματα που πραγματοποιήθηκαν αφορούσαν τόσο το μέγεθος των δειγμάτων των εικόνων όσο και τον τρόπο λήψης τους από την αρχική εικόνα. Τα αποτελέσματα δείχνουν ότι τόσο η “Random” όσο και η “Grid-Random” μέθοδος λήψης των δειγμάτων παράγουν καλύτερα αποτελέσματα σε σχέση με την “Grid-Grid”. Επιπλέον, βελτιώνουν την απόδοση τους σύμφωνα με το μέγεθος του τμήματος εικόνας, κάτι το οποίο φαίνεται να μην συμβαίνει στις περισσότερες περιπτώσεις της “Grid-Grid” μεθόδου. Επιπρόσθετα, οι περιπτώσεις “Random” και “Grid-Random”, φαίνεται να παρέχουν υψηλή ακρίβεια εντοπισμού των κτιρίων ειδικά για τα μεγέθη

δειγμάτων εικόνας 128×128 και 244×224. Η αδυναμία της μεθόδου δειγματοληψίας “Grid-Grid” σε σχέση με τις άλλες δύο οφείλεται στο δεν έχει επιτραπεί στο δίκτυο να εκπαιδευτεί με δείγματα που να καλύπτουν όλο το χωρικό εύρος της αρχικής εικόνας και αυτής των επαληθευμένων δεδομένων παρά μόνο ένα τμήμα του με αποτέλεσμα το σύνολο των δεδομένων εκπαίδευσης να μην είναι τόσο αντιπροσωπευτικό όσο στις πρώτες δύο μεθόδους δειγματοληψίας.

Στα εννέα πειράματα που αφορούσαν την μέθοδο δειγματοληψίας και το μέγεθος των δειγμάτων εκπαίδευσης χρησιμοποιήθηκε σταθερή τιμή 8 για την παράμετρο mini-batch του συνελικτικού δικτύου. Ένα επιπλέον πείραμα πραγματοποιήθηκε με σκοπό την εξαγωγή συμπερασμάτων σχετικά με την επίδραση του μεγέθους του mini-batch, στην συνολική απόδοση. Η δοκιμή αφορούσε mini-batch με τιμή 64 και η αξιολόγηση των πειραματικών αποτελεσμάτων έδειξε ότι σχεδόν όλες οι μετρικές αποτίμησης είχαν καλύτερες τιμές στην περίπτωση όπου η εκπαίδευση έχει πραγματοποιηθεί με mini-batch μεγέθους 64.

Ελέγχοντας τον αντίκτυπο του βάθους κωδικοποίησης της αρχιτεκτονικής U-Net στην απόδοση των συνελικτικών δικτύων διαπιστώθηκε η αρχιτεκτονική U-Net, με βάθος κωδικοποίησης ίσο με 1, δίνει χαμηλά αποτέλεσμα σε αντίθεση με τις άλλες τρεις περιπτώσεις (τιμές 2 έως 4). Τα πειραματικά δεδομένα έδειξαν ότι υπάρχει μικρή υπεροχή της απόδοσης με βάθος 3 καθώς η υψηλή πολυπλοκότητα ενός δικτύου βάθους 4 συχνά οδηγεί σε προβλήματα υπέρ-προσαρμογής και είναι προφανώς πιο υπολογιστικά χρονοβόρα.

Βάσει των καταγραφών, όσον αναφορά τις τιμές της συνάρτησης κόστους και τα σφάλματα, ως προς τα δεδομένα εκπαίδευσης και επικύρωσης, ήταν αρκετές οι φορές που τα μεν κόστη και σφάλματα των δεδομένων εκπαίδευσης μειώνονταν σταθερά, δίχως όμως να ακολουθούνται από τις αντίστοιχες ενδείξεις για τα δεδομένα επικύρωσης. Για την αντιμετώπιση της υπέρ-προσαρμογής, προτάθηκε η τροποποίηση της ίδια της αρχιτεκτονικής του δικτύου χρησιμοποιώντας διαφορετικά επίπεδα Dropout. Με βάση τα πειραματικά αποτελέσματα η τροποποίηση αυτή δίνει θετικά αποτελέσματα και μάλιστα στο πλέον “αυστηρό” σενάριο, αυτό της δειγματοληψίας “Grid-Grid”.

Ακόμα, πρέπει να τονιστεί ότι σημαντικό ρόλο, όσον αφορά τη συνολική διαδικασία, διαδραματίζει η κλίμακα του σαρωμένου χάρτη. Όπως γράφτηκε παραπάνω, η κλίμακα του σαρωμένου τοπογραφικού χάρτη, ο οποίος χρησιμοποιήθηκε στα παραπάνω πειράματα, είναι 1:5000. Αν τα διάφορα εκπαιδευμένα συνελικτικά νευρωνικά δίκτυα, των παραπάνω πειραμάτων, χρησιμοποιούνταν για προβλέψεις επί τμημάτων εικόνας εισόδου από σαρωμένο χάρτη διαφορετικής κλίμακας, η απόδοση δε θα ήταν εξίσου ικανοποιητική. Ο συγκεκριμένος περιορισμός, πρακτικά, δεν είναι πολύ σημαντικός, αφού οι συλλογές χαρτών συνήθως περιέχουν χάρτες ίδιας κλίμακας, όπως π.χ. οι χάρτες 1:25000 της Γ.Υ.Σ. Συνεπώς, αναλόγως την

κλίμακα του σαρωμένου χάρτη, στου οποίου τις εικόνες είναι επιθυμητό να εντοπιστούν γεωγραφικά χαρακτηριστικά, τα δεδομένα εκπαίδευσης του συνελικτικού δικτύου πρέπει προέρχονται αντίστοιχα από χάρτη κατάλληλης κλίμακας.

5.2 Προτάσεις βελτίωσης

Με βάση τα πειράματα της εργασίας και τα συμπεράσματα που προέκυψαν, είναι ενδιαφέρον να αναφερθούν κάποιες προτάσεις βελτίωσης για μελλοντικές εφαρμογές. Όσον αφορά την διαδικασία βελτιστοποίησης, θα ήταν ιδιαιτέρως ενδιαφέρουσα η εφαρμογή του πιο πρόσφατου αλγόριθμου AdamW αντί του Adam που χρησιμοποιήθηκε στην παρούσα εργασία. Επίσης, όπως ήδη επισημάνθηκε, ένα πρόβλημα το οποίο παρουσιάστηκε είναι αυτό της υπέρ-προσαρμογής του αλγόριθμου στα δεδομένα εκπαίδευσης. Ακόμα, αναφέρθηκε ότι στη συγκεκριμένη περίπτωση, αντιμετωπίστηκε σε κάποιο βαθμό αξιοποιώντας Dropout επίπεδα. Εκτός όμως των Dropout επιπέδων, τα οποία επιδρούν στην αρχιτεκτονική του δικτύου, υπάρχουν και μέθοδοι κανονικοποίησης οι οποίες επιδρούν στην συνάρτηση κόστους, επιβάλλοντας “ποινή” στα μεγάλα βάρη, με αποτέλεσμα να “συγκρατείται” η αύξηση τους και να αντιμετωπίζεται η υπέρ-προσαρμογή. Θα ήταν λοιπόν χρήσιμη η διερεύνηση διαφορετικών μεθόδων κανονικοποίησης και της επίδρασής τους στην τελική απόδοση του συστήματος. Η αποτελεσματικότητα του προτεινόμενου συστήματος θα μπορούσε επίσης να εφαρμοστεί σε χάρτες παλαιότερων περιόδων, διαφορετικής προέλευσης, τόσο όσον αφορά την γεωγραφική θέση τους όσο και όσον αφορά την χρονολογία τους. Ειδικά η εφαρμογή της μεθόδου στην ίδια γεωγραφική περιοχή αλλά από διαφορετικές χρονικές περιόδους θα παρείχε και επιπλέον χρήσιμα συμπεράσματα για την ικανότητα του συνελικτικού δικτύου να εκφράζει με πιο γενικό τρόπο τα κοινά χαρακτηριστικά που διέπουν τις εικόνες αυτές. Τέλος, η προτεινόμενη μεθοδολογία θα μπορούσε να ενταχθεί σε ένα συνολικό σύστημα το οποίο θα συνέθετε τα επιμέρους δείγματα εικόνων κτιρίων που παράγονται από την παρούσα μέθοδο σε μια ολιστική προσέγγιση. Μια τέτοια εφαρμογή θα είναι χρήσιμη για τον ενδιαφερόμενο μηχανικό/τοπογράφο/ερευνητή καθώς θα δέχεται σαν είσοδο μια γεωαναφερμένη εικόνα ενός ιστορικού χάρτη και θα δίνει στην έξοδο την ίδια περιοχή του χάρτη πάλι σε μορφή γεωαναφερμένης εικόνας περιέχοντας όμως μόνο το επιλεγμένο γεωγραφικό χαρακτηριστικό, στην συγκεκριμένη περίπτωση, τα κτίρια.

Βιβλιογραφία

Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. «ImageNet Classification with Deep Convolutional Neural Networks.» Στο *Advances in neural information processing systems*, 1097-1105. 2012.

Anders Krogh, John Hertz. «A Simple Weight Decay Can Improve Generalization.» Στο *Advances in Neural Information Processing Systems 4*, 950-957. San Mateo, Calif: Kaufmann, 1992.

Andrew L. Maas, Awni Y. Hannun, Andrew Y. Ng. «Rectifier Nonlinearities Improve Neural Network Acoustic Models.» *ICML Workshop on Deep Learning for Audio, Speech and Language Processing.* Atlanta,Georgia: Computer Science Department, Stanford University, 2013.

Andrew Ng, Jiquan Ngiam, Chuan Yu Foo. *Deep Learning Tutorial.* Stanford University. χ.χ. http://ufldl.stanford.edu/tutorial/.

Aston Zhang, Zachary C. Lipton, Mu Li, Alexander J. Smola. «Dive into Deep Learning.» 25 July 2021. https://d2l.ai/.

B.T.Polyak. «Some methods of speeding up the convergence of iteration methods.» *USSR Computational Mathematics and Mathematical Physics* 4, αρ. 5 (1964): 1-17.

Chigozie Enyinna Nwankpa, Winifred Ijomah, Anthony Gachagan, Stephen Marshall. «Activation Functions: Comparison of Trends in Practice and Research for Deep Learning.» 2018.

Datta, Leonid. «A Survey on Activation Functions and their relation with Xavier and He Normal Initialization.» Delft University of Technology, 2020.

David E. Rumelhart, Geoffrey E. Hinton, Ronald J. Williams. «Learning representations by back-propagating errors.» *Nature* 323 (1986): 533-536.

Diederik P. Kingma, Jimmy Lei Ba. «ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION.» *International Conference on Learning Representations.* San Diego, CA, 2015.

Dominik Scherer, Andreas Muller, Sven Behnke. «Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition.» *20th International conference on artificial neural networks (ICANN), pp. 92-101.* Berlin, 2010.

Fei-Fei Li, Ranjay Krishna, Danfei Xu. *CS231n: Convolutional Neural Networks for Visual Recognition.* Stanford University. 2020. https://cs231n.github.io/.

Fukushima, Kunihiko. «Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position.» *Biological Cybernetics* 36 (1980): 193-202.

Genevieve Orr, Nici Schraudolph, Fred Cummins. *CS-449: Neural Networks.* Willamette University. χ.χ. https://www.willamette.edu/~gorr/classes/cs449/momrate.html.

Hebb, Donald O. *The Organization of Behavior: A Neuropsychological Theory.* New York: Wiley, 1949.

Herbert Robbins, Sutton Monro. «A Stochastic Approximation Method.» *Ann. Math. Statist.* 22, αρ. 3 (1951): 400-407.

Ian Goodfellow, Yoshua Bengio, Aaron Courville. *Deep Learning book.* MIT Press. 2016. https://www.deeplearningbook.org/.

Ian J. Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, Yoshua Bengio. «Maxout Networks.» Montreal: Département d'informatique et de recherche opérationnelle, Université de Montréal, 2013.

Ilya Loshchilov, Frank Hutter. «Decoupled Weight Decay Regularization.» *International Conference on Learning Representations (ICLR) 2019.* New Orleans, 2019.

Ilya Sutskever, James Martens, George Dahl, Geoffrey Hinton. «On the importance of initialization and momentum in deep learning.» *Proceedings of the 30th International Conference on Machine Learning* 28, αρ. 3 (2013): 1139-1147.

John Duchi, Elad Hazan, Yoram Singer. «Adaptive Subgradient Methods for Online Learning and Stochastic Optimization.» *Journal of Machine Learning Research*, αρ. 12 (2011): 2121-2159.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. «Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification.» Microsoft Research, 2015.

Keiron O'Shea, Ryan Nash. «An Introduction to Convolutional Neural Networks.» arXiv, 2015.

Matthew D. Zeiler, Dilip Krishnan, Graham W. Taylor, Rob Fergus. «Deconvolutional networks.» *2010 IEEE Computer Society Conference on computer vision and pattern recognition (pp. 2528-2535).* San Francisco, CA, 2010.

Nesterov, Yurii. «A Method for Solving a Convex Programming Problem with Convergence Rate 0(1/K^2).» *Soviet Mathematics Doklady* 27 (1983): 372-367.

Nielsen, Michael. «Neural Networks and Deep Learning.» December 2019. http://neuralnetworksanddeeplearning.com/.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov. «Dropout: A Simple Way to Prevent Neural Networks from Overfitting.» *Journal of Machine Learning Research 15*, June 2014: 1929-1958.

Olaf Ronneberger, Philipp Fischer, Thomas Brox. «U-net: Convolutional networks for biomedical image segmentation.» *International Conference on Medical image computing and computer-assisted intervention (pp. 234-241).* Springer, Cham, 2015.

Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, Kaori Togashi. «Convolutional neural networks: an overview and application in radiology.» *Insights into Imaging*, αρ. 9 (2018): 611-629.

Rosenblatt, Frank. «The perceptron: a probabilistic model for information storage and organization in the brain.» *Psychological Reviews* 65, αρ. 6 (1958): 386-408.

Ruder, Sebastian. «An overview of gradient descent optimization algorithms.» Dublin: Insight Centre for Data Analytics, NUI Galway, 2017.

Sarkar, Sudeshna. *CS60010: Deep Learning.* Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur. 22 January 2018. https://cse.iitkgp.ac.in/~sudeshna/courses/DL18/lec9-CNN-25Jan18.pdf.

Thangarajah Akilan, Q. M. Jonathan Wu, Wandong Zhang. «Video foreground extraction using multi-view receptive field and encoder–decoder DCNN for traffic and surveillance applications.» *IEEE Transactions on Vehicular Technology* 68, αρ. 10 (2019): 9478-9493.

Turing, Alan M. «Computing Machinery and Intelligence.» *Mind* 59, αρ. 236 (1950): 433-460.

Vincent Dumoulin, Francesco Visin. «A guide to convolution arithmetic for deep learning.» ArXiv, 2018.

Vinod Nair, Geoffrey E. Hinton. *Rectified Linear Units Improve Restricted Boltzmann Machines.* Toronto: Department of Computer Science, University of Toronto, 2010.

Warren S. McCulloch, Walter Pitts. «A logical calculus of the ideas immanent in nervous activity.» *The Bulletin of Mathematical Biophysics* 5 (1943): 115-133.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel. «Backpropagation Applied to Handwritten Zip Code Recognition.» *Neural Computation* 1, αρ. 4 (1989): 541-551.

Y. Lecun, L. Bottou,Y. Bengio,P. Haffner. «Gradient-based learning applied to document recognition.» *Proceedings of the IEEE* 86, αρ. 11 (1998): 2278-2324.

Yehuda Koren, Robert Bell, Chris Volinsky. «Matrix Factorization Techniques for Recommender Systems.» IEEE Computer Society, 2009.

Y-Lan Boureau, Jean Ponce, Yann LeCun. «A theoretical analysis of feature pooling in visual recognition.» *Proceedings of the 27th international conference on machine learning (ICML-10), pp. 111-118.* Haifa, 2010.

Zhou, Chellappa. «Computation of optical flow using a neural network.» *IEEE 1988 International Conference on Neural Networks, 1988, pp. 71-78 vol.2.* χ.χ.