



ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ ΞΑΝΘΗΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ Η/Υ

ΤΟΜΕΑΣ ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

**ΕΝΤΟΠΙΣΜΟΣ ΚΑΙ ΤΑΞΙΝΟΜΗΣΗ ΑΝΤΙΚΕΙΜΕΝΩΝ ΣΕ
ΔΟΡΥΦΟΡΙΚΕΣ ΕΙΚΟΝΕΣ**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΚΑΡΑΓΙΑΝΝΗ ΞΕΝΟΦΩΝ

A.E.M. 6001

Επιβλέπων: Αν. Καθηγητής Ιωάννης Πρατικάκης

Ξάνθη, Οκτώβριος 2018

Περιεχόμενα

Ευχαριστίες	4
Περίληψη	5
Εισαγωγή	7
1 Θεωρητικό υπόβαθρο - Τηλεπισκόπηση και Νευρωνικά δίκτυα	8
1.1 Βασικές αρχές τηλεπισκόπησης	8
1.1.1 Ηλεκτρομαγνητική ακτινοβολία	9
1.1.2 Το Ηλεκτρομαγνητικό Φάσμα	10
1.1.3 Ηλιακή ακτινοβολία και ατμοσφαιρικά παράθυρα	12
1.1.4 Άλληλεπίδραση της ακτινοβολίας με την ύλη	12
1.1.5 Χαρακτηριστικά των εικόνων τηλεπισκόπησης	15
1.1.6 Χωρική και φασματική ανάλυση (Spatial and spectral resolution)	17
1.1.7 Ταξινόμηση φασματικής πληροφορίας	20
1.2 Βασικές αρχές νευρωνικών δικτύων και συνελικτικών νευρωνικών δικτύων.	22
1.2.1 Νευρώνας	22
1.2.2 Perceptron	23
1.2.3 Νευρωνικά δίκτυα πολλών επιπέδων	25
1.2.4 Σιγμοειδείς νευρώνες	26
1.2.5 Επιβλεπόμενη και μη-επιβλεπόμενη μάθηση	28
1.2.6 Μέθοδος καθόδου με βάση την κλίση (Gradient Descent)	28
1.2.7 Μέθοδος οπισθοδιάδοσης (Backpropagation)	33
1.2.8 Υπερ-προσαρμογή (Overfitting)	38
1.2.9 Autoencoders και Stacked Autoencoders	38
1.2.10 Συνελικτικά νευρωνικά δίκτυα (Convolutional Neural Networks)	42
1.2.11 Συναρτήσεις ενεργοποίησης	46
1.2.12 Συναρτήσεις κόστους	47
1.2.13 Τι μαθαίνει ένα CNN	50
1.2.14 Μια ολοκληρωμένη αρχιτεκτονική CNN	51
2 Τα μοντέλα που μελετήθηκαν: Stacked Autoencoders-Logistic Regression και Diverse Region-Based CNN	52
2.1 Stacked Autoencoder - Logistic Regression	52
2.1.1 Ταξινόμηση με έμφαση στα φασματικά χαρακτηριστικά	52
2.1.2 Ταξινόμηση με έμφαση στα χωρικά χαρακτηριστικά	57

2.1.3 Ταξινόμηση με συνδυασμό των φασματικών και χωρικών χαρακτηριστικών	58
2.2 Diverse Region-Based CNN	59
3 Πειραματικά αποτελέσματα	66
3.1 Τα datasets που χρησιμοποιήθηκαν	66
3.1.1 Pavia University	66
3.1.2 Indian Pines	67
3.1.3 Salinas	69
3.1.4 Kaggle DSTL Competition dataset	70
3.2 Τα frameworks Theano, Tensorflow και Keras	72
3.3 Πειραματικά αποτελέσματα του Stacked Autoencoder	75
3.4 Πειραματικά αποτελέσματα του DR-CNN	79
3.5 Συμπεράσματα	86

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή της διπλωματικής εργασίας, Αν. Καθηγητή του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Η/Υ, Ιωάννη Πρατικάκη, για την εμπιστοσύνη που μου έδειξε και την καθοδήγησή του, καθ' όλη αυτή την προσπάθεια.

Θα ήθελα, επίσης να ευχαριστήσω τον υποψήφιο διδάκτορα Τσοχατζίδη Λάζαρο, για την πολύτιμη βοήθεια και συμβολή του στην ολοκλήρωση αυτής της εργασίας.

Ευχαριστώ τους φίλους μου, που βρίσκονταν πλάι μου.

Τέλος, ευχαριστώ τη μητέρα μου, που με στηρίζει όλα αυτά τα χρόνια. Τίποτα δε θα ήταν δυνατό, δίχως αυτή.

Περίληψη

Τα τελευταία χρόνια, ιδιαίτερα από το 2012 και μετά, η βαθιά μάθηση (deep learning) έχει διεισδύσει σε αρκετά διαφορετικά πεδία, όπως η τεχνητή όραση, η αναγνώριση ομιλίας και η φυσική επεξεργασία γλώσσας. Το αντικείμενο της τηλεπισκόπησης, παρ' όλες τις μοναδικές προκλήσεις του, εμφανίζει αρκετά κοινά στοιχεία με το πεδίο της τεχνητής όρασης. Στο πεδίο της τεχνητής όρασης, τα βαθιά νευρωνικά δίκτυα που έχουν εμφανιστεί από το 2012 μέχρι σήμερα έχουν οδηγήσει σε άλματα σε προβλήματα όπως η αναγνώριση και η ταξινόμηση αντικειμένων. Στο πεδίο της τηλεπισκόπησης, η αναγνώριση και η ταξινόμηση αντικειμένων σε υπερφασματικές εικόνες αποτελεί ένα από τα πιο δημοφιλή θέματα και ακόμα από το 2014 το ενδιαφέρον στράφηκε στην χρήση μοντέλων βαθιάς μάθησης για αυτό το σκοπό. Στην παρούσα διπλωματική εργασία, μελετήθηκαν δύο τέτοια μοντέλα, για την εξαγωγή χαρακτηριστικών και την ταξινόμηση σε εικόνες τηλεπισκόπησης: ένας stacked autoencoder - ίσως το πρώτο μοντέλο που χρησιμοποιήθηκε στο πεδίο της τηλεπισκόπησης και μία αρχιτεκτονική βασισμένη στα συνελικτικά νευρωνικά δίκτυα (Convolutional Neural Networks - CNN), που σήμερα θεωρείται το μοντέλο της τρέχουσας τεχνολογικής στάθμης.

Εισαγωγή

Τα τελευταία χρόνια το deep learning έχει οδηγήσει σε άλματα, σε πεδία όπως η τεχνητή οραση, η αναγνώριση ομιλίας και η φυσική επεξεργασία γλώσσας. Η ειρωνεία είναι ότι το deep learning, ως εξέλιξη των νευρωνικών δικτύων, παρακλάδι της τεχνητής νοημοσύνης, υπάρχει ως ιδέα εδώ και δεκαετίες. Βασικοί παράγοντες για την επαναφορά της ιδέας στο προσκήνιο είναι η ανάπτυξη νέων αλγορίθμων που καθιστούν δυνατή την εκπαίδευση νευρωνικών δικτύων με πολλά επίπεδα, η ανάπτυξη και χρήση των GPUs αξιοποιώντας την υψηλή υπολογιστική τους ισχύ και τα big data.

Η ιδέα του deep learning είναι απλή: η μηχανή μαθαίνει τα χαρακτηριστικά και συνήθως είναι πολύ καλή στη λήψη αποφάσεων (ταξινόμηση).

Το deep learning έκανε την επανεμφάνισή του μέσω της τεχνητής ορασης, το επιστημονικό πεδίο που στοχεύει στην οπτική κατανόηση μέσω υπολογιστικής ανάλυσης των εικόνων. Οι κλασσικές προσεγγίσεις, συχνά σήμερα αποκαλούνται "ρηχές", επειδή περιλαμβάνουν λίγα μόνο στάδια επεξεργασίας, όπως ένα στάδιο επεξεργασίας για την απομάκρυνση θορύβου ακολουθούμενο από έναν εξαγωγέα χαρακτηριστικών (feature extractor) και στη συνέχεια έναν ταξινομητή, για τη σύνδεση των ακατέργαστων δεδομένων με τις τελικές αποφάσεις. Παραδείγματα "ρηχών" προσεγγίσεων μάθησης περιλαμβάνουν τα SVMs (Support Vector Machines) και τα μοντέλα Markov. Αντιθέτως, τα μοντέλα deep learning έχουν πολλά επίπεδα ("βαθιά" μοντέλα, αντί για ρηχά), τα οποία επιτρέπουν στο δίκτυο να μάθει από τα δεδομένα σύνθετα, μη γραμμικά και ιεραρχικά χαρακτηριστικά.

Οι παραδοσιακές μέθοδοι για την εξαγωγή χαρακτηριστικών από υπερφασματικές εικόνες, απαιτούσαν ιδιαίτερες τεχνικές δεξιότητες και βαθιά γνώση του πεδίου εφαρμογής. Μόνον έτσι ήταν δυνατή η εξαγωγή χαρακτηριστικών βασισμένα στο χρώμα, την υφή, το σχήμα, το χωρικό και φασματικό περιεχόμενο, τα οποία έφεραν χρήσιμη για την ταξινόμηση πληροφορία.

Αναφορικά, κάποιοι παραδοσιακοί περιγραφείς χαρακτηριστικών (feature descriptors) είναι οι texture descriptors, scale invariant feature transform και το histogram of oriented gradients (HOG).

Με τα μοντέλα deep learning, το λεγόμενο feature engineering μπορεί να αντικατασταθεί από το ίδιο το μοντέλο, το οποίο μαθαίνει/εξάγει μόνο του τα χαρακτηριστικά από τα δεδομένα εισόδου. [BAC17]

Σήμερα, οι autoencoders, τα συνελικτικά νευρωνικά δίκτυα (Convolutional Neural Networks-CNNs), τα Deep Belief Networks (DBNs) και τα Recurrent Neural Networks (RNNs) αποτελούν τις τέσσερις κυριότερες αρχιτεκτονικές deep learning.

Από το 2012, μετά την επιτυχημένη χρήση ενός CNN στον διαγωνισμό ILSVRC από τους Alex Krizhevsky et al. [KSH12], όλο και περισσότεροι ερευνητές και επιστημονικές κοινότητες έστρεψαν το ενδιαφέρον τους στο deep learning. Σήμερα, αρχιτεκτονικές deep learning πετυχαίνουν state of the art αποτελέσματα σε πεδία απομάκρυσμένα από την οραση υπολογιστών με εφαρμογές στην βιοπληροφορική, τις μεταφορές, την γεωργία κ.ά. [MLY17].

Από το 2014, χρησιμοποιούνται προσεγγίσεις deep learning για την ταξινόμηση υπερφασματικών εικόνων, μία διαδικασία αρκετά δημοφιλή στις εφαρμογές τηλεπισκόπησης. Η ταξινόμηση στις υπερφασματικές εικόνες είναι η διαδικασία επισήμανσης των pixels σε μία ή περισσότερες κλάσεις εδαφοκάλυψης.

Η επεξεργασία των υπερφασματικών εικόνων περιλαμβάνει αρκετές προκλήσεις, όπως η διαχείριση των δεδομένων που συνήθως είναι πολλών διαστάσεων και ο μικρός όγκος διαθέσιμων δειγμάτων εκπαίδευσης.

Στην παρούσα διπλωματική εργασία, μελετήθηκαν δύο μοντέλα deep learning για την ταξινόμηση σε υπερφασματικές εικόνες.

Το πρώτο μοντέλο βασίζεται στην προσέγγιση των Chen et al. με την χρήση ενός stacked autoencoder για την εξαγωγή χαρακτηριστικών και ενός λογιστικού ταξινομητή (logistic regression classifier) για την απόδοση κλάσεων εδαφοκάλυψης. Αντιμετωπίζουν την ταξινόμηση με τρεις διαφορετικούς τρόπους, διατηρώντας κάθε φορά την ίδια αρχιτεκτονική με μικρές αλλαγές και δίνοντας ως είσοδο στο μοντέλο δεδομένα με έμφαση είτε στο φασματικό, είτε στο χωρικό περιεχόμενο είτε έναν συνδυασμό αυτών των δύο.

Το δεύτερο μοντέλο που μελετήθηκε και που σήμερα δίνει state of the art αποτελέσματα στη διαδικασία της ταξινόμησης σε υπερφασματικές εικόνες, είναι των Zhang et al. και είναι μία αρχιτεκτονική βασισμένη σε CNNs. Η καινοτομία του μοντέλου, είναι ότι για την εξαγωγή χαρακτηριστικών επιλέγει πολλαπλές περιοχές, γειτονικές του pixel ενδιαφέροντος, με διαφορετικά μεγέθη και προσανατολισμό ώστε να καταφέρει να συλλάβει τις διαφορετικές κατανομές αντικειμένων σε αυτές περιοχές και να μάθει διαφορετικές αναπαραστάσεις των δεδομένων εισόδου. Η συγχώνευση αυτών των αναπαραστάσεων τροφοδοτεί ένα νευρωνικό δίκτυο που λειτουργεί ως ταξινομητής, για την επισήμανση κάθε pixel με μία κλάση εδαφοκάλυψης.

Όσων αφορά τη δομή της εργασίας, το πρώτο κεφάλαιο αποτελείται από το απαραίτητο θεωρητικό υπόβαθρο σχετικά με το πεδίο της τηλεπισκόπησης και του deep learning, στο δεύτερο κεφάλαιο περιγράφονται αναλυτικά τα μοντέλα που μελετήθηκαν, στο τρίτο κεφάλαιο παρουσιάζονται τα datasets που χρησιμοποιήθηκαν, οι βιβλιοθήκες με τις οποίες υλοποιήθηκαν τα μοντέλα και τα πειραματικά αποτελέσματα. Η εργασία κλείνει με μία σειρά συμπερασμάτων όπως έχουν προκύψει μετά την ενασχόληση με το συγκεκριμένο αντικείμενο στα πλαίσια της διπλωματικής εργασίας.

Κεφάλαιο 1

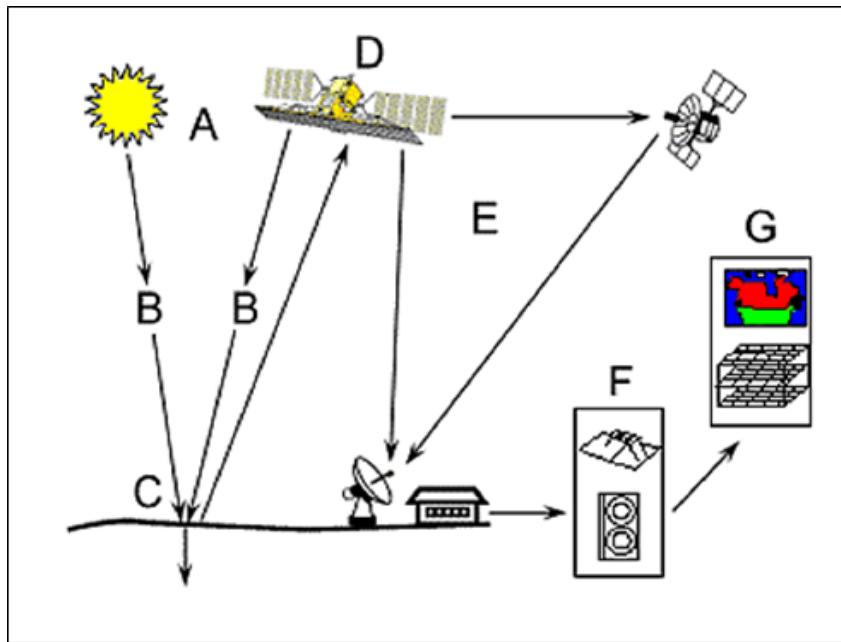
Θεωρητικό υπόβαθρο - Τηλεπισκόπηση και Νευρωνικά δίκτυα

1.1 Βασικές αρχές τηλεπισκόπησης

Τηλεπισκόπηση είναι η επιστήμη (και σε κάποιο βαθμό, τέχνη) απόκτησης πληροφορίας σχετικά με την επιφάνεια της Γης, χωρίς στην πραγματικότητα να είναι σε επαφή με αυτή (τηλεπισκόπηση = παρατήρηση από μακριά). Αυτό επιτυγχάνεται με την ανίχνευση και καταγραφή της ανακλώμενης ή εκπεμπόμενης ενέργειας και επεξεργασία, ανάλυση και εφαρμογή των εν λόγω πληροφοριών. [Nrc]

Σε μεγάλο μέρος της τηλεπισκόπησης, η διαδικασία περιλαμβάνει την αλληλεπίδραση μεταξύ της προσπίτουσας ακτινοβολίας και των στόχων ενδιαφέροντος. Αυτό δίδεται παραδειγματικά με τη χρήση των συστημάτων απεικόνισης, όπου συμμετέχουν τα ακόλουθα επτά στοιχεία. Σημειώστε, ωστόσο, ότι η τηλεπισκόπηση περιλαμβάνει επίσης την ανίχνευση της εκπεμπόμενης ενέργειας και τη χρήση μη οπτικών αισθητήρων. Παρόλο που ο όρος Τηλεπισκόπηση μπορεί να γενικευθεί σε οποιαδήποτε ενέργεια καταγραφής από απόσταση και ανάλυσης των εικόνων που προκύπτουν (ακόμη και αυτή των ανθρώπινων ματιών), λόγω της πολυφασματικότητάς της και της προγενέστερης ύπαρξης των αεροφωτογραφιών έχει κυριαρχήσει η χρήση του όρου για την καταγραφή και την ανάλυση δορυφορικών εικόνων.

- Πηγή Ενέργειας ή Φωτισμού (Α)** – η πρώτη προϋπόθεση για την τηλεπισκόπηση είναι να έχουμε μια πηγή ενέργειας που φωτίζει ή παρέχει ηλεκτρομαγνητική ενέργεια προς το στόχο του ενδιαφέροντος.
- Ακτινοβολία και Ατμόσφαιρα (Β)** – καθώς η ενέργεια ταξιδεύει από την πηγή έως το στόχο της, θα έρθει σε επαφή και θα αλληλεπιδράσει με την ατμόσφαιρα που περνά. Αυτή η αλληλεπίδραση μπορεί να λάβει χώρα και δεύτερη φορά ως η ενέργεια που ταξιδεύει από τον στόχο για τον αισθητήρα.
- Αλληλεπίδραση με τον Στόχο (C)** - όταν η ενέργεια πηγαίνει προς το στόχο μέσω της ατμόσφαιρας, αλληλεπιδρά με το στόχο, ανάλογα με τις ιδιότητες τόσο του στόχου όσο και της ακτινοβολίας.
- Καταγραφή της Ενέργειας από τον Αισθητήρα (D)** - αφού η ενέργεια έχει διασπαστεί ή εκπεμφθεί από το στόχο, χρειαζόμαστε έναν αισθητήρα (απομακρυσμένο – να μην



Σχήμα 1.1: Αυτά τα επτά στοιχεία περιλαμβάνουν τη μεθοδολογία της τηλεπισκόπησης από την αρχή μέχρι το τέλος.

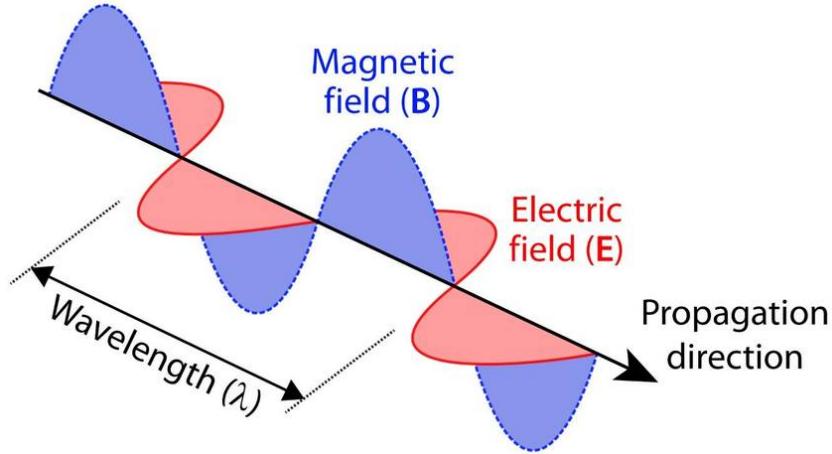
είναι σε επαφή με τον στόχο) για τη συλλογή και καταγραφή της ηλεκτρομαγνητικής ακτινοβολίας.

5. **Η Μετάδοση, Υποδοχή, και Επεξεργασία (Ε)** - η ενέργεια καταγεγραμμένη από τον αισθητήρα πρέπει να διαβιβάζεται, συχνά σε ηλεκτρονική μορφή, σε έναν σταθμό λήψης και επεξεργασίας, όπου τα δεδομένα επεξεργάζονται ώστε να αποδίδονται ως εικόνα (έντυπη ή / και ψηφιακή).
6. **Ερμηνεία και Ανάλυση (F)** - η επεξεργασμένη εικόνα ερμηνεύεται, οπτικά και / ή ψηφιακά ή ηλεκτρονικά, για την εξαγωγή πληροφοριών σχετικά με το στόχο που είχε “φωτιστεί”.
7. **Εφαρμογή (G)** - το τελικό στοιχείο της επεξεργασίας της τηλεπισκόπησης επιτυγχάνεται όταν εφαρμόζονται πληροφορίες τις οποίες έχουμε τη δυνατότητα να εξάγουμε από τις εικόνες σε σχέση με το στόχο για την καλύτερη κατανόηση του. Έτσι αποκαλύπτονται κάποιες νέες πληροφορίες ή βοήθειες για επίλυση ενός συγκεκριμένου προβλήματος.

1.1.1 Ηλεκτρομαγνητική ακτινοβολία

Η πρώτη απαίτηση για την τηλεπισκόπηση είναι να έχουμε μια ενεργειακή πηγή για να φωτίζει το στόχο (εκτός εάν η ανιχνευόμενη ενέργεια εκπέμπεται από το στόχο). Αυτή η ενέργεια έχει τη μορφή ηλεκτρομαγνητικής ακτινοβολίας.

Όλη η ηλεκτρομαγνητική ακτινοβολία έχει θεμελιώδεις ιδιότητες και συμπεριφέρεται με προβλέψιμο τρόπο σύμφωνα με τις βασικές αρχές της θεωρίας των κυμάτων. Η ηλεκτρομαγνητική ακτινοβολία αποτελείται από ένα ηλεκτρικό πεδίο (Ε) που ποικίλλει σε μέγεθος, σε



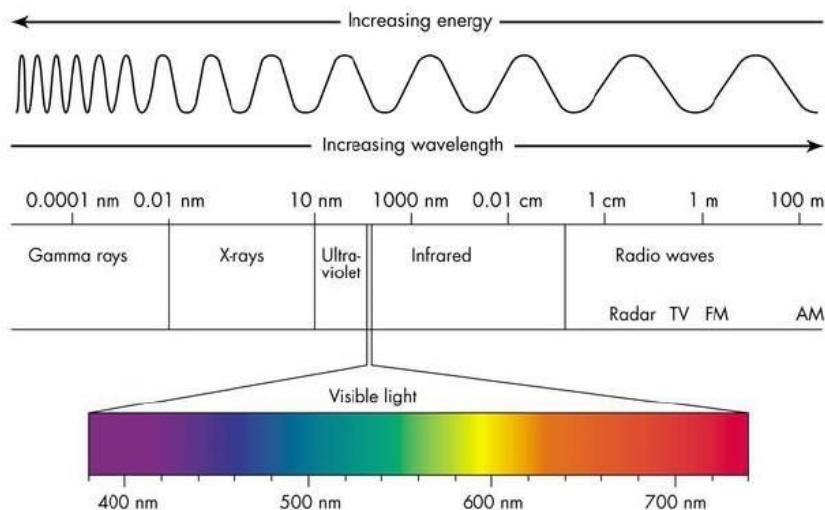
Σχήμα 1.2: Ηλεκτρομαγνητικό κύμα

μία διεύθυνση, κάθετη προς την κατεύθυνση στην οποία η ακτινοβολία ταξιδεύει, και ένα μαγνητικό πεδίο (B) προσανατολισμένο σε ορθές γωνίες προς το ηλεκτρικό πεδίο. Και τα δύο αυτά πεδία ταξιδεύουν με τη ταχύτητα του φωτός.

Δύο είναι τα χαρακτηριστικά της ηλεκτρομαγνητικής ακτινοβολίας που είναι ιδιαιτέρως σημαντικά για την κατανόηση της τηλεπισκόπησης, τα οποία είναι το μήκος κύματος και η συχνότητα.

1.1.2 Το Ηλεκτρομαγνητικό Φάσμα

Το ηλεκτρομαγνητικό φάσμα κυμαίνεται από τα μικρότερα μήκη κύματος (συμπεριλαμβανομένου γάμμα και Χ ακτίνες) στα μεγαλύτερα μήκη κύματος (συμπεριλαμβανομένων των μικροκυμάτων και των εκπεμπόμενων ραδιοκυμάτων). Είναι αρκετές οι περιοχές του ηλεκτρομαγνητικού φάσματος που είναι χρήσιμες στην τηλεπισκόπηση.



Σχήμα 1.3: Το ηλεκτρομαγνητικό φάσμα

Για τους περισσότερους σκοπούς, το υπεριώδες ή UV τμήμα του φάσματος έχει το μικρότερο μήκος κύματος που είναι πρακτικό για την τηλεπισκόπηση. Αυτή η ακτινοβολία είναι ακριβώς πέρα από το ιώδες τμήμα του ορατού μέρους του φάσματος, απ' όπου προήλθε και το όνομά του. Μερικά υλικά της επιφάνειας της Γης, κυρίως πετρώματα και ορυκτά, φθορίζουν ή εκπέμπουν ορατό φως όταν φωτίζονται με υπεριώδη ακτινοβολία.

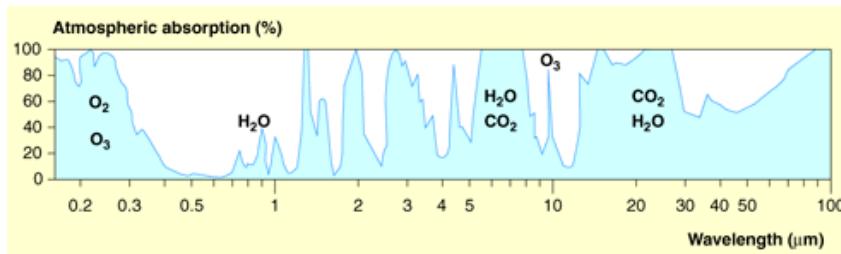
Το φάσμα που μπορούν τα μάτια μας, οι “τηλεανιχνευτές” μας, να ανιχνεύσουν είναι μέρος του ορατού φάσματος. Το ορατό τμήμα του φάσματος είναι πολύ μικρό σε σχέση με το υπόλοιπο φάσμα. Υπάρχει πολλή ακτινοβολία γύρω μας η οποία είναι ”αόρατη” στα μάτια μας, αλλά μπορεί να ανιχνευθεί από άλλα μέσα τηλεπισκόπησης και να χρησιμοποιηθούν προς όφελός μας. Το ορατό μήκος κύματος καλύπτει ένα εύρος περίπου από 0,4 έως 0,7 μμ. Το μεγαλύτερο ορατό μήκος κύματος είναι το ερυθρό και το μικρότερο το ιώδες. Τα κοινά μήκη κύματος τα οποία αντιλαμβανόμαστε ως ξεχωριστά χρώματα από το ορατό τμήμα του φάσματος, παρατίθενται παρακάτω.

- **Ιώδες:** 0.4 - 0.446 μμ
- **Μπλε:** 0.446 - 0.500 μμ
- **Πράσινο:** 0.500 - 0.578 μμ
- **Κίτρινο:** 0.578 - 0.592 μμ
- **Πορτοκαλί:** 0.592 - 0.620 μμ
- **Ερυθρό:** 0.620 - 0.7 μμ

Μπλε, πράσινο και κόκκινο είναι τα κύρια χρώματα ή μήκη κύματος του ορατού φάσματος. Αυτά ορίζονται έτσι, διότι κανένα πρωτεύον χρώμα δε μπορεί να δημιουργηθεί από άλλα δύο, αλλά όλα τα άλλα χρώματα μπορούν να σχηματιστούν συνδυάζοντας μπλε, πράσινο και κόκκινο σε διάφορες αναλογίες. Παρά το γεγονός ότι βλέπουμε το φως του ήλιου ως ομοιόμορφο ή ομοιογενές χρώμα, στη πραγματικότητα αποτελείται από διάφορα μήκη κύματος της ακτινοβολίας κυρίως στο υπεριώδες, ορατό και υπέρυθρο τμήμα του φάσματος.

Το επόμενο τμήμα του φάσματος που μας ενδιαφέρει είναι το υπέρυθρο (IR) τμήμα το οποίο καλύπτει το εύρος μήκους κύματος από περίπου 0,7 μμ έως 100 μμ, περισσότερο από 100 φορές το εύρος του ορατού τμήματος! Η υπέρυθρη περιοχή μπορεί να χωρισθεί σε δύο κατηγορίες με βάση τις ιδιότητες της ακτινοβολίας, η ανακλώμενη IR και η εκπεμπόμενη ή θερμική IR. Η ακτινοβολία στην ανακλώμενη περιοχή IR χρησιμοποιείται για σκοπούς της τηλεπισκόπησης κατά τρόπο παρόμοιο με της ακτινοβολίας στο ορατό τμήμα. Η ανακλώμενη IR καλύπτει μήκη κύματος περίπου από 0,7 μμ έως 3,0 μμ. Η θερμική περιοχή IR είναι αρκετά διαφορετική από το ορατό και ανακλώμενο IR τμήμα, διότι αυτή η ενέργεια είναι ουσιαστικά η ακτινοβολία που εκπέμπεται από την επιφάνεια της γης με τη μορφή θερμότητας. Η θερμική IR καλύπτει μήκη κύματος περίπου από 3,0 μμ έως 100 μμ.

Το τμήμα του φάσματος του πιο πρόσφατου ενδιαφέροντος για την τηλεπισκόπηση είναι το τμήμα αυτό των μικροκυμάτων, σε περιοχή περίπου από 1 μμ έως 1 m. Αυτό καλύπτει τα μακρύτερα μήκη κύματος που χρησιμοποιούνται για την τηλεπισκόπηση. Τα μικρότερα μήκη κύματος έχουν ιδιότητες παρόμοιες με την περιοχή του θερμικού υπερύθρου, ενώ τα μεγαλύτερα μήκη κύματος πλησιάζουν τα μήκη κύματος που χρησιμοποιούνται για την εκπομπή ραδιοκυμάτων.



Σχήμα 1.4: Ο όρος ατμοσφαιρικό παράθυρο έχει δοθεί στα μήκη κύματος στα οποία η ατμόσφαιρα είναι “διαφανής” και οι εκπομπές και ανακλάσεις “περνούν” σχεδόν ανεμπόδιστες. Σε άλλα μήκη κύματος η ακτινοβολία απορροφάται από τα διάφορα αέρια του θερμοκηπίου.

1.1.3 Ηλιακή ακτινοβολία και ατμοσφαιρικά παράθυρα

Η ηλιακή ακτινοβολία και η αντανάκλασή της από την επιφάνεια της γης διαπερνούν την ατμόσφαιρα πριν φτάσουν στους αισθητήρες των δορυφόρων. Τα “αέρια του θερμοκηπίου” στην ατμόσφαιρα απορροφούν ένα μέρος της ακτινοβολίας που ανακλάται από τη Γη. Το οζόν λειτουργεί σαν ένα απόλυτο φράγμα για την υπεριώδη ακτινοβολία απορροφώντας όλη την ακτινοβολία στην περιοχή του φάσματος μεταξύ 9.5 – 10 μμ. Τα ισχυρότερα “αέρια θερμοκηπίου” είναι οι υδρατμοί και το διοξείδιο του άνθρακα καθώς απορροφούν ακτινοβολία σε διάφορα μήκη κύματος.

Τα μήκη κύματος που επιτρέπουν σε μεγάλο τμήμα μιας ακτινοβολίας να περάσει μέσω της ατμόσφαιρας, ονομάζονται “ατμοσφαιρικά παράθυρα”. Ευτυχώς ανάμεσα σε αυτά είναι και το μεγαλύτερο μέρος των ορατών ακτίνων (διαφορετικά το φως δε θα έφτανε ποτέ στην επιφάνεια της γης). Η ατμόσφαιρα είναι σχεδόν 100% διαφανής σε ορισμένες περιοχές του εγγύς υπέρυθρου και αυτό επιτρέπει στους δορυφόρους να πραγματοποιούν μετρήσεις χωρίς μεγάλες ατμοσφαιρικές παρεμβολές. Το θερμικό υπέρυθρο φάσμα (10-12μμ) χρησιμοποιείται για τη μέτρηση της θερμοκρασίας του εδάφους, του νερού και των νεφών.

Παρά το γεγονός ότι η τηλεπισκόπηση γίνεται μέσα από τα ατμοσφαιρικά παράθυρα, εξακολουθούν σε κάποιο βαθμό να υπάρχουν παρεμβολές από τη διάχυση και την απορρόφηση της ακτινοβολίας στην ατμόσφαιρα.

Έτσι, πολλές φορές τα τηλεπισκοπικά δεδομένα είναι ελαφρώς αλλοιωμένα και πρέπει να διορθωθούν υποβάλλοντας τις εικόνες σε ψηφιακή επεξεργασία.

1.1.4 Αλληλεπίδραση της ακτινοβολίας με την ύλη

Ακτινοβολία που δεν απορροφάται ή σκεδάζεται στην ατμόσφαιρα μπορεί να φτάσει και να αλληλεπιδράσει με την επιφάνεια της Γης. Υπάρχουν τρεις (3) μορφές αλληλεπίδρασης που λαμβάνουν χώρα όταν η ενέργεια βρει το στόχο της, ή είναι προσπίπτουσα (I) πάνω στην επιφάνεια. Αυτές είναι:

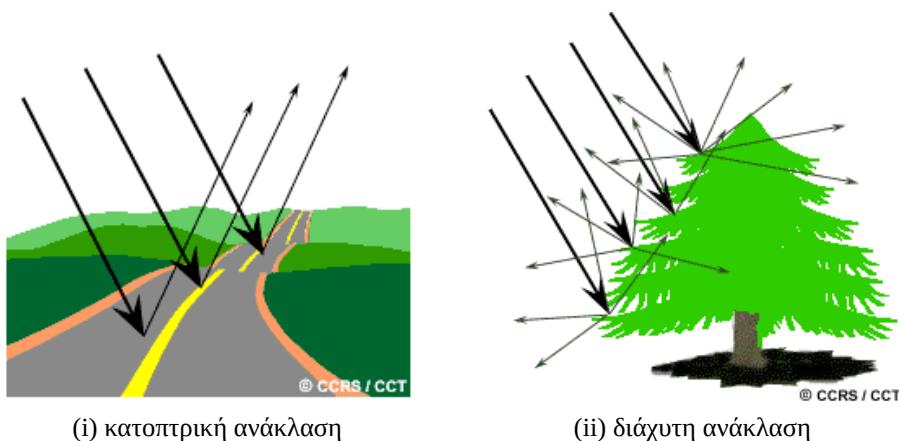
- η απορρόφηση (A),
- η μετάδοση (T),
- και η ανάκλαση (R).

Η συνολική προσπίπτουσα ενέργεια θα αλληλεπιδράσει με την επιφάνεια με έναν ή περισσότερους από αυτούς τους τρεις τρόπους. Οι αναλογίες του κάθε ένα θα εξαρτηθεί από το μήκος



Σχήμα 1.5: Απορρόφηση, μετάδοση και ανάκλαση σε ένα φύλλο.

κύματος της ενέργειας και της ύλης όπως επίσης και από την κατάσταση του υλικού. Η απορρόφηση (A) παρουσιάζεται όταν η ακτινοβολία (ενέργεια) απορροφάται από το στόχο, ενώ η μετάδοση (T) παρουσιάζεται όταν η ακτινοβολία διέρχεται σ' αυτόν. Η ανάκλαση (R) εμφανίζεται όταν η ακτινοβολία "αναπηδά" από το στόχο και γίνεται ανακατεύθυνση της πορείας της. Στην τηλεπισκόπηση, μας ενδιαφέρει περισσότερο η μέτρηση της ακτινοβολίας που ανακλάται από τους στόχους. Αναφερόμαστε σε δύο τύπους ανάκλασης, οι οποίοι αντιπροσωπεύουν τις δύο ακραίες περιπτώσεις του τρόπου με τον οποίο η ενέργεια ανακλάται από ένα στόχο: η **κατοπτρική ανάκλαση** και η **διάχυτη ανάκλαση**.



Σχήμα 1.6: Ανάκλαση σε λεία και τραχιά επιφάνεια

Όταν μία επιφάνεια είναι λεία παρατηρείται **κατοπτρική ανάκλαση** όπου όλη (ή σχεδόν όλη) η ενέργεια κατευθύνεται μακριά από την επιφάνεια σε μία κατεύθυνση. Η **διάχυτη ανάκλαση** λαμβάνει χώρα όταν η επιφάνεια είναι τραχιά και η ενέργεια ανακλάται σχεδόν ομοιόμορφα προς όλες τις κατευθύνσεις. Τα περισσότερα χαρακτηριστικά της επιφάνειας της γης απλώνονται σε τέτοια θέση έτσι ώστε να είναι είτε τέλειοι κατοπτρικοί ανακλαστήρες, είτε τέλειοι διάχυτοι ανακλαστήρες. Το πώς ένας συγκεκριμένος στόχος θα ανακλάσει την ακτινο-

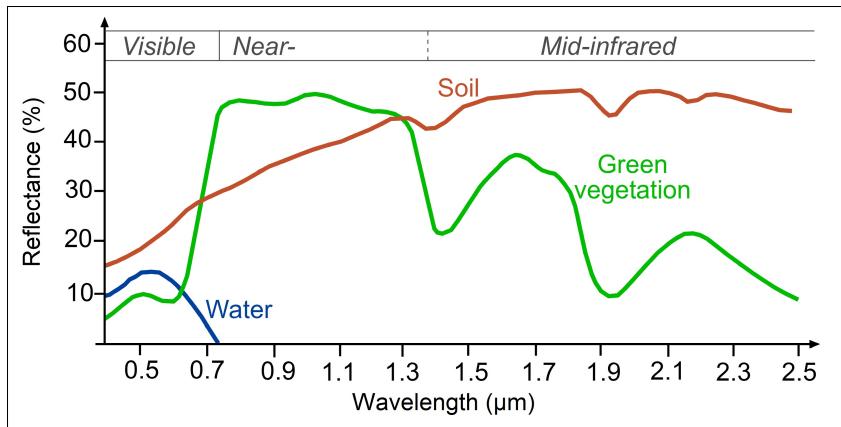
βολία, δηλαδή είτε τέλεια κατοπτρικά ή τέλεια διάχυτα, είτε κάπου ανάμεσα στα δύο, εξαρτάται από την επιφανειακή τραχύτητα του υλικού σε σύγκριση με το μήκος κύματος της εισερχόμενης ακτινοβολίας. Εάν τα μήκη κύματος είναι πολύ μικρότερα από τις διαφοροποιήσεις επιφανείας ή τα μεγέθη των σωματιδίων που αποτελούν την επιφάνεια, η διάχυτη ανάκλαση θα κυριαρχήσει. Για παράδειγμα, η λεπτόκοκκη άμμο θα εμφανιζόταν αρκετά λεία στα μεγάλα μήκη κύματος μικροκυμάτων αλλά σχετικά τραχιά στα ορατά μήκη κύματος.

Παρακάτω αναπτύσσονται δύο παραδείγματα στόχων στην επιφάνεια της Γης και πώς η ενέργεια στα ορατά και υπέρυθρα μήκη κύματος αλληλεπιδρά μαζί τους.

Φύλλα: Μια χημική ένωση στα φύλλα που ονομάζεται χλωροφύλλη απορροφά έντονα την ακτινοβολία στα ερυθρά και μπλε μήκη κύματος, αλλά ανακλά τα πράσινα μήκη κύματος. Τα φύλλα εμφανίζονται «πράσινα» το καλοκαίρι, όταν το περιεχόμενο της χλωροφύλλης είναι στο μέγιστο. Το φθινόπωρο, υπάρχει λιγότερη χλωροφύλλη στα φύλλα, επομένως λιγότερη απορρόφηση και αναλογικά περισσότερη ανάκλαση στα ερυθρά μήκη κύματος, κάνοντας τα φύλλα να φαίνονται κόκκινα ή κίτρινα (κίτρινο είναι ένας συνδυασμός κόκκινου και πράσινου μήκους κύματος). Η εσωτερική δομή των υγιών φύλλων ενεργούν ως εξαιρετικοί διάχυτοι ανακλαστήρες των εγγύς υπέρυθρων μηκών κυμάτων. Αν τα μάτια μας ήταν ευαίσθητα στο εγγύς (κοντινό) υπέρυθρο, τα δέντρα θα φαίνονταν εξαιρετικά φωτεινά σε αυτά τα μήκη κύματος. Στην πραγματικότητα, η μέτρηση και η παρακολούθηση της εγγύς υπέρυθρης (IR) ανάκλασης είναι μιας κατεύθυνσης, με την οποία οι επιστήμονες μπορούν να καθορίσουν το πόσο υγιής (ή μη υγιής) είναι η βλάστηση.

Νερό: Μεγαλύτερα μήκη κύματος της ορατής και εγγύς υπέρυθρης ακτινοβολίας απορροφάται περισσότερο από το νερό από ότι τα μικρότερα ορατά μήκη κύματος. Έτσι το νερό συνήθως φαίνεται μπλε ή μπλε-πράσινο λόγω της ισχυρότερης ανάκλασης σε αυτά τα μικρότερα μήκη κύματος, και πιο σκούρο αν εκτεθεί σε ερυθρά ή εγγύς υπέρυθρα μήκη κύματος. Εάν υπάρχουν αιωρούμενα ίζηματα στα ανώτερα στρώματα του σώματος του νερού, τότε αυτό θα επιτρέψει την καλύτερη ανάκλαστικότητα και μία φωτεινότερη εμφάνιση του ύδατος. Το φαινομενικό χρώμα του νερού θα δείχνει μια ελαφρά μετατόπιση σε μεγαλύτερα μήκη κύματος. Το αιωρούμενο ίζημα (S) μπορεί εύκολα να συγχέεται με ρηχά (αλλά καθαρά) νερά, δεδομένου ότι τα δύο αυτά φαινόμενα εμφανίζονται πολύ παρόμοια. Η χλωροφύλλη στην άλγη απορροφά περισσότερο από τα μπλε μήκη κύματος και ανακλά τα πράσινα, καθιστώντας το νερό να φαίνεται πιο πράσινο με την παρουσία άλγης. Η τοπογραφία της επιφάνειας του νερού (τραχύ, λείο, επιπλέοντα υλικά, κλπ) μπορεί επίσης να οδηγήσει σε περίπλοκες ερμηνείες σχετικά με το νερό, που οφείλονται σε πιθανά προβλήματα της κατοπτρικής ανάκλασης, και άλλων επιδράσεων στο χρώμα και στη φωτεινότητα.

Μπορούμε να δούμε από αυτά τα παραδείγματα ότι, ανάλογα με το σύνθετο make-up του στόχου που εξετάζεται, και τα εμπλεκόμενα μήκη κύματος της ακτινοβολίας, είναι δυνατόν να παρατηρηθούν πολύ διαφορετικές αποκρίσεις στους μηχανισμούς απορρόφησης, μετάδοσης και αντανάκλασης. Με τη μέτρηση της ενέργειας που αντανακλάται (ή εκπέμπεται) από στόχους στην επιφάνεια της Γης μέσω μιας ποικιλίας διαφορετικών μήκων κύματος, μπορούμε να δομήσουμε μια **φασματική απόκριση** για αυτό το αντικείμενο - ένα διάγραμμα την ανακλώμενης ακτινοβολίας ως συνάρτηση του προσπίπποντος μήκους κύματος το οποίο αποτελεί την φασματική υπογραφή του υλικού. Συγκρίνοντας τα μοτίβα των αποκρίσεων των διαφορετικών χαρακτηριστικών είμαστε σε θέση να τα διακρίνουμε μεταξύ τους, όπου ίσως δεν ήταν δυνατό,



Σχήμα 1.7: Φασματική απόκριση εδάφους, βλάστησης και νερού

αν μόνο τα συγκρίναμε μόνο σε ένα μήκος κύματος. Για παράδειγμα, το νερό και η βλάστηση και μπορεί να αντανακλούν κάπως παρόμοια στα ορατά μήκη κύματος, αλλά είναι σχεδόν πάντα διαχωρίσιμα στο υπέρυθρο (σχήμα 1.7). Η φασματική απόκριση μπορεί να μεταβάλλεται σχετικά, ακόμη και για τον ίδιο τύπο στόχου και μπορεί επίσης να ποικίλει με το χρόνο (π.χ. πρασινάδα των φύλλων) και την τοποθεσία. Γνωρίζοντας που να εξετάσουμε και κατανοώντας τους παράγοντες που επηρεάζουν τη φασματική απόκριση των χαρακτηριστικών που μας ενδιαφέρουν είναι κρίσιμη η σωστή ερμηνεία των αλληλεπιδράσεων της ηλεκτρομαγνητικής ακτινοβολίας με την επιφάνεια.

1.1.5 Χαρακτηριστικά των εικόνων τηλεπισκόπησης

Η ηλεκτρομαγνητική ενέργεια μπορεί να ανιχνευθεί είτε με φωτογραφικά είτε με ηλεκτρονικά μέσα. Η φωτογραφική διαδικασία χρησιμοποιεί χημικές αντιδράσεις στην επιφάνεια του φωτο-ευαίσθητου φιλμ για την ανίχνευση και καταγραφή μεταβολών ενέργειας. Είναι σημαντική η διάκριση μεταξύ των όρων **εικόνες** και **φωτογραφίες** στην τηλεπισκόπηση. Μια **εικόνα** αναφέρεται σε οποιαδήποτε εικονογραφημένη αναπαράσταση, ανεξαρτήτως του μήκους κύματος ή συσκευής τηλεπισκόπησης που χρησιμοποιείται για την ανίχνευση και καταγραφή της ηλεκτρομαγνητικής ενέργειας. Μια **φωτογραφία** αναφέρεται ειδικότερα σε εικόνες που έχουν εντοπιστεί όπως επίσης και καταγράφει σε φωτογραφικό φιλμ. Η έγχρωμη φωτογραφία (i) του σχήματος 1.8, από τμήμα της Orange County της Καλιφόρνια, έχει ληφθεί στο ορατό μέρος του φάσματος. Οι φωτογραφίες συνήθως καταγράφονται πάνω από το εύρος μήκους κύματος από 0,3 μμ έως 0,9 μμ, που είναι το ορατό και ανακλώμενο υπέρυθρο. Με βάση τους ορισμούς αυτούς, μπορούμε να πούμε ότι όλες οι φωτογραφίες είναι εικόνες, αλλά όλες οι εικόνες δεν είναι φωτογραφίες. Ως εκ τούτου, εκτός και αν μιλάμε ειδικά για μια εικόνα που καταγράφεται φωτογραφικά, χρησιμοποιούμε τον όρο εικόνα.

Μια φωτογραφία θα μπορούσε επίσης να εκπροσωπείται και να εμφανίζεται σε ψηφιακή μορφή υποδιαιρώντας την εικόνα σε μικρά ίσου μεγέθους και σχηματοποιημένα τμήματα, που ονομάζονται εικονοστοιχεία ή pixels, και αντιπροσωπεύουν τη φωτεινότητα της κάθε περιοχής με μια αριθμητική τιμή ή ψηφιακό αριθμό. Πράγματι, αυτό ακριβώς έχει γίνει στη φωτογραφία (ii) (1.8 ii). Στην πραγματικότητα, χρησιμοποιώντας τους ορισμούς παραπάνω, αυτή είναι στην ουσία μια ψηφιακή εικόνα της αρχικής φωτογραφίας! Η φωτογραφία σαρώθηκε και υποδιαιρέθηκε σε pixels και σε κάθε ένα από αυτά να αποδίδεται ένας ψηφιακός αριθμός που

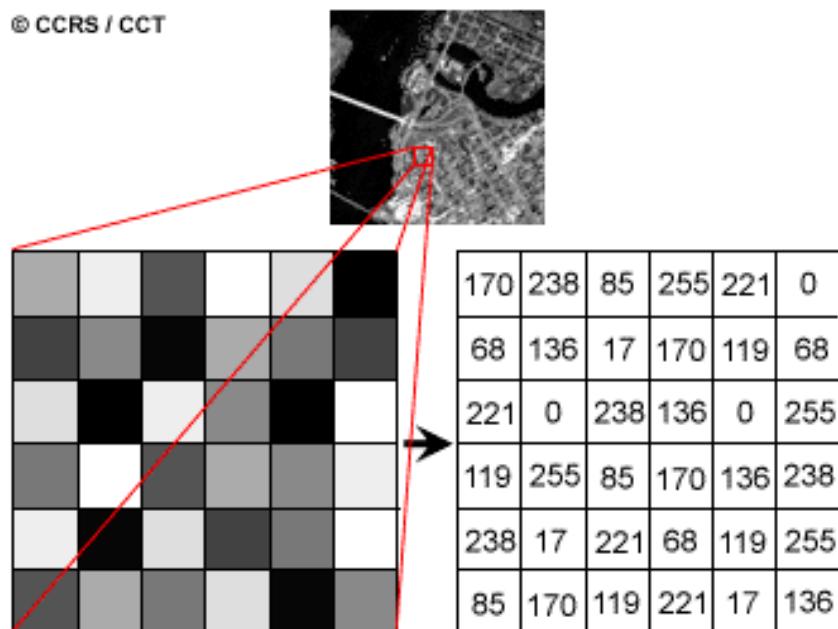


(i) 6inch αεροφωτογραφία των δρόμων της
Orange County
© Eagle Aerial Imaging 2009

(ii) Υπέρυθρη αεροφωτογραφία του
δέλτα του ποταμού Sacramento
Courtesy of HJW GeoSpatial

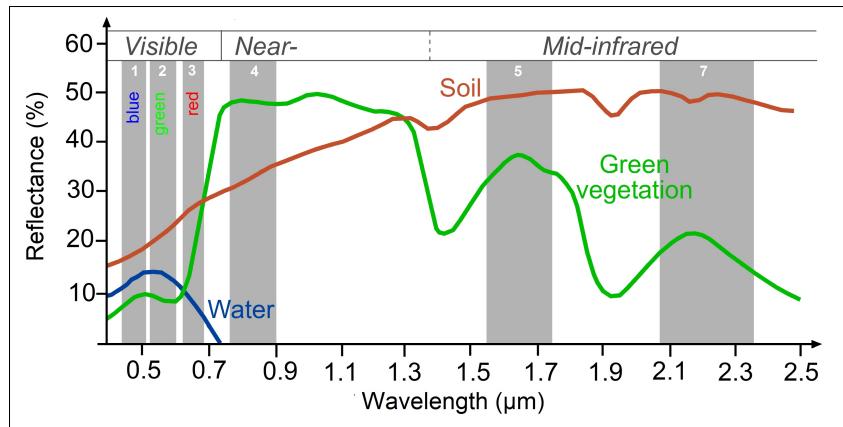
Σχήμα 1.8: Αεροφωτογραφίες στο ορατό και υπέρυθρο τμήμα του φάσματος

αντιπροσωπεύει τη σχετική φωτεινότητα της. Ο υπολογιστής εμφανίζει κάθε ψηφιακή τιμή σαν διαφορετικά επίπεδα φωτεινότητας. Οι αισθητήρες καταγραφής ηλεκτρομαγνητικής ενέργειας, καταγράφουν ηλεκτρονικά την ενέργεια ως μια σειρά από αριθμούς σε ψηφιακή μορφή από την αρχή. Αυτοί οι δύο διαφορετικοί τρόποι αντιπροσώπευσης και εμφάνισης των δεδομένων τηλεπισκόπησης, είτε εικονικά είτε ψηφιακά, είναι εναλλάξιμοι, καθώς μεταφέρουν τις ίδιες πληροφορίες (αν και κάποιες λεπτομέρειες μπορεί να χάνονται κατά τη μετατροπή από την μία στην άλλη μορφή).



Σχήμα 1.9: Η ψηφιακή αναπαράσταση της φωτογραφίας

Η οπτική τηλεπισκόπηση κάνει χρήση αισθητήρων που λειτουργούν στο ορατό, κοντινό υπέρθυρο και υπέρθυρο βραχέων κυμάτων (short-wave infrared - SWIR) του φάσματος, για τη σύνθεση εικόνων της επιφάνειας της γης ανιχνεύοντας, όπως έχει ήδη αναφερθεί, ηλιακή ακτινοβολία που αντανακλάται από τους στόχους που βρίσκονται στο έδαφος. Όπως φαίνεται στο σχήμα 1.7, διαφορετικά υλικά αντανακλούν και απορροφούν με διαφορετικό τρόπο σε διαφορετικά μήκη κύματος. Συνεπώς, μπορεί να γίνει διάκριση των στόχων με βάση τις φασμα-



Σχήμα 1.10: Ανάκλαση του νερού, του εδάφους και της βλάστησης σε διαφορετικά μήκη κύματος (από το 1.7 και κανάλια του Landsat TM 1 (0.45-0.52 μμ), 2 (0.52-0.60 μμ), 3 (0.63-0.69 μμ), 4 (0.76-0.90 μμ), 5 (1.55-1.75 μμ) και 7 (2.08-2.35 μμ)

τικές αποκρίσεις τους. Οι αισθητήρες λαμβάνουν σήματα σε διαφορετικά μήκη κύματος. Αυτές οι διαφορετικές φασματικές περιοχές ονομάζονται **κανάλια** (ή ζώνες συχνοτήτων). Το σχήμα 1.10 είναι το ίδιο με το 1.7 με την επιπλέον προσθήκη των καναλιών του αισθητήρα Landsat Thematic Mapper (TM) της NASA. Τα ορατό τμήμα του φάσματος ανιχνεύεται με τρία διαφορετικά κανάλια: το μπλε, το πράσινο και το κόκκινο και αντιστοιχίζονται στα κανάλια 1, 2 και 3.

1.1.6 Χωρική και φασματική ανάλυση (Spatial and spectral resolution)

Η λεπτομέρεια με την οποία μπορεί να γίνει διάκριση των στοιχείων μιας εικόνας, εξαρτάται από την **χωρική ανάλυση** του αισθητήρα και αναφέρεται στο μικρότερο δυνατό χαρακτηριστικό που μπορεί να ανιχνευθεί. Η χωρική ανάλυση των παθητικών αισθητήρων (αισθητήρες που βασίζονται στο προσπίπτον φως εξωτερικών πηγών - π.χ. τον ήλιο - για να κάνουν τις μετρήσεις), εξαρτάται κυρίως από το Στιγμιαίο Οπτικό Πεδίο (Instantaneous Field of View - IFOV). Το στιγμιαίο οπτικό πεδίο είναι η γωνία υπό την οποία γίνεται μία μέτρηση από τον αισθητήρα, μετράται συνήθως σε μrad και καθορίζει το εμβαδόν της επιφάνειας της Γης το οποίο ”βλέπει” ο αισθητήρας σε κάθε δεδομένη στιγμή (το οποίο ονομάζεται κελί ανάλυσης - resolution cell). Συνήθως το GIFOV είναι συνώνυμο με την χωρική ανάλυση της εικόνας (εκτός και αν η εικόνα προβάλλεται για κάποιο λόγο με μέγεθος pixel διαφορετικό της πλήρους χωρικής ανάλυσης).

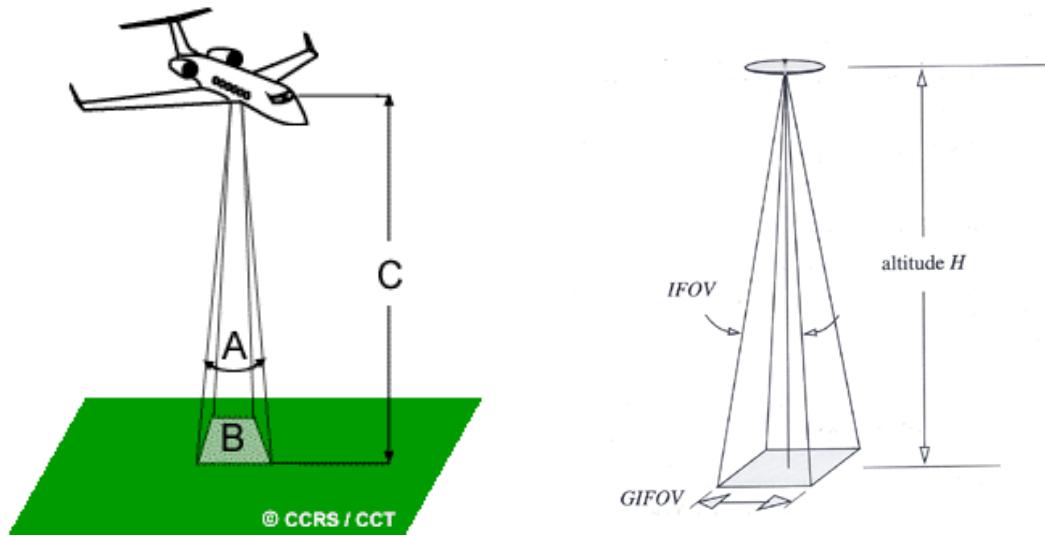
Το GIFOV υπολογίζεται ως εξής:

$$GIFOV(Nadir) = 2 \times Altitude \times 1000 \times \tan(IFOV \times (10^{-6})/2))$$

(πολλαπλασιάζουμε το υψόμετρο επί 1000 για να μετετρέψουμε το αποτέλεσμα σε μέτρα και το IFOV επί 10^{-6} για να μετατρέψουμε τα microradians σε radians.)

Για παράδειγμα, το IFOV του παγχρωματικού οργάνου του δορυφόρου QuickBird-2 είναι 1.37 μrad. Για υψόμετρο 450 χιλιομέτρων (στον ισημερινό) το GIFOV υπολογίζεται σε 0,6165 μέτρα.

Για να ανιχνευθεί ένα ομογενές στοιχείο, το μέγεθός του γενικά θα πρέπει να είναι ίσο ή μεγαλύτερο της ανάλυσης του κελιού. Αν περισσότερα από ένα στοιχεία βρίσκονται εντός



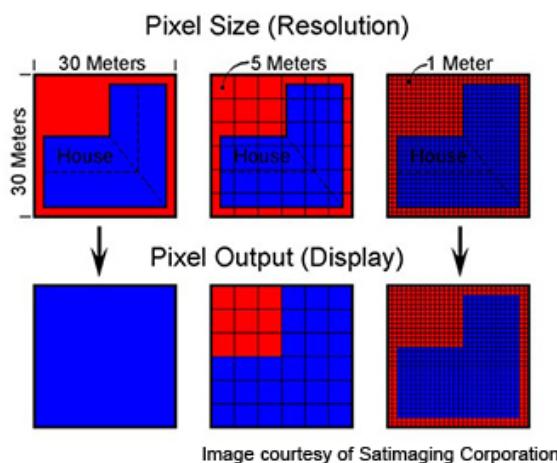
(i) IFOV (A), resolution cell (B) και υψόμετρο (C).

(ii) Το GIFOV είναι συνήθως συνώνυμο με την χωρική ανάλυση της εικόνας.

Σχήμα 1.11: IFOV, GIFOV και resolution cell

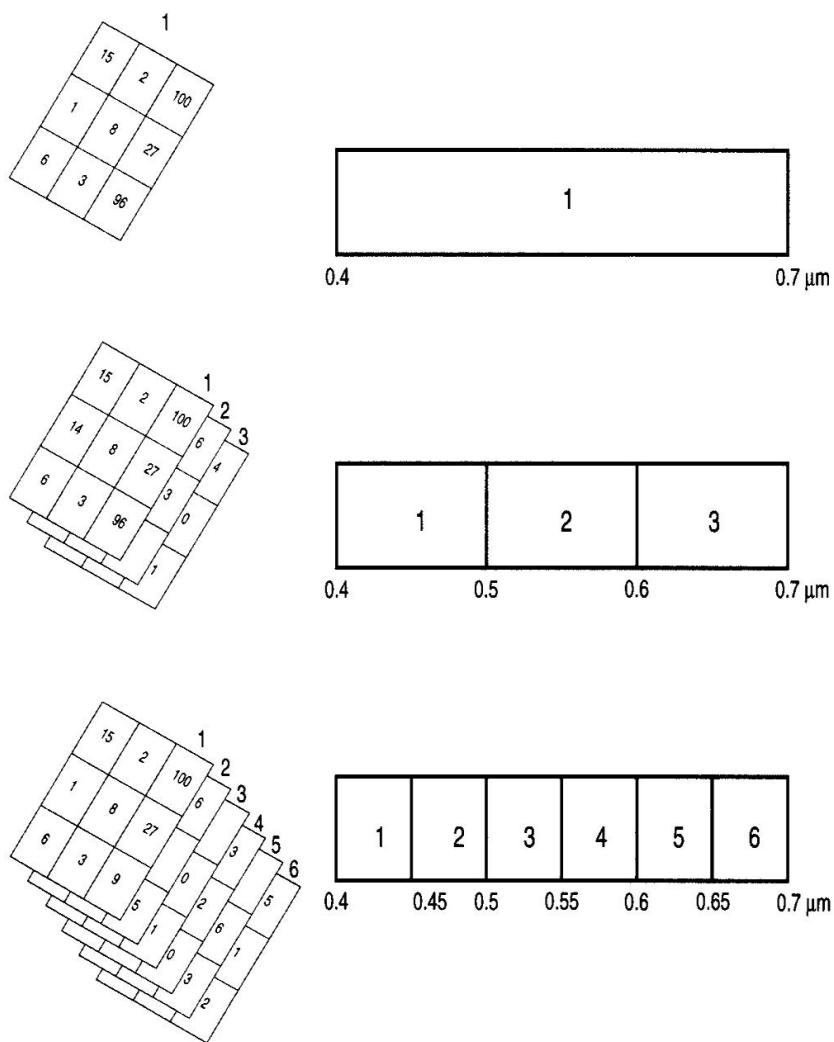
του κελιού η απόκριση που θα μετρηθεί θα περιλαμβάνει ένα μία μίξη σημάτων από όλα τα στοιχεία.

Στο σχήμα 1.12 φαίνεται πως η αναγνώριση ενός στόχου (ένα σπίτι στην προκειμένη περίπτωση) εμφανίζεται διαφορετικά ανάλογα με την χωρική ανάλυση. Στο παράδειγμα για την εικόνα ανάλυσης 30m η απόκριση του σπιτιού κυριαρχεί στο κελί και επομένως ολόκληρο το κελί ταξινομείται ως "σπίτι". Αντίθετα, στις εικόνες καλύτερης ανάλυσης, το σχήμα και η έκταση του στοιχείου αποτυπώνονται καλύτερα. Στην εικόνα με ανάλυση 5m, κάποια από τα κελιά που είναι μερικώς καλυμμένα από το στοιχείο ταξινομούνται ως κελιά βάσει της κυριαρχίας των σημάτων του στοιχείου. Στην καλύτερη ανάλυση (1m), το σχήμα και η έκταση του στοιχείου αναγνωρίζονται με μεγαλύτερη ακρίβεια.



Σχήμα 1.12: Σχηματική αναπαράσταση αναγνώρισης στόχου σε εικόνες διαφορετικής χωρικής ανάλυσης.

Η φασματική ανάλυση αναπαριστά το πλάτος της φασματικής ζώνης και την ευαισθησία του αισθητήρα. Μπορεί να οριστεί ως η ικανότητα του αισθητήρα να ορίσει διαστήματα μηκών κύματος για τον χαρακτηρισμό των διάφορων συστατικών της επιφάνειας της Γης. Όσο καλύτερη είναι η φασματική ανάλυση, τόσο πιο στενό είναι το διάστημα για κάθε κανάλι. Οι υπερφασματικοί αισθητήρες καταγράφουν την ανακλώμενη ενέργεια σε εκατοντάδες πολύ στενά φασματικά κανάλια. Το σχήμα 1.13 δείχνει την υποθετική αναπαράσταση συστημάτων τηλεπισκόπησης με διαφορετική φασματική ανάλυση. Η πρώτη αναπαράσταση δείχνει τις τιμές από 9 pixels χρησιμοποιώντας εικόνα που έχει ληφθεί σε μία ζώνη. Ομοίως, η δεύτερη και τρίτη αναπαράσταση δείχνουν τις τιμές που έχουν ληφθεί σε 3 και 6 ζώνες χρησιμοποιώντας τους κατάλληλους αισθητήρες. Αν η περιοχή των εικόνων είναι $A \text{ km}^2$, η ίδια περιοχή προβάλλεται χρησιμοποιώντας 1, 3 και 6 ζώνες.

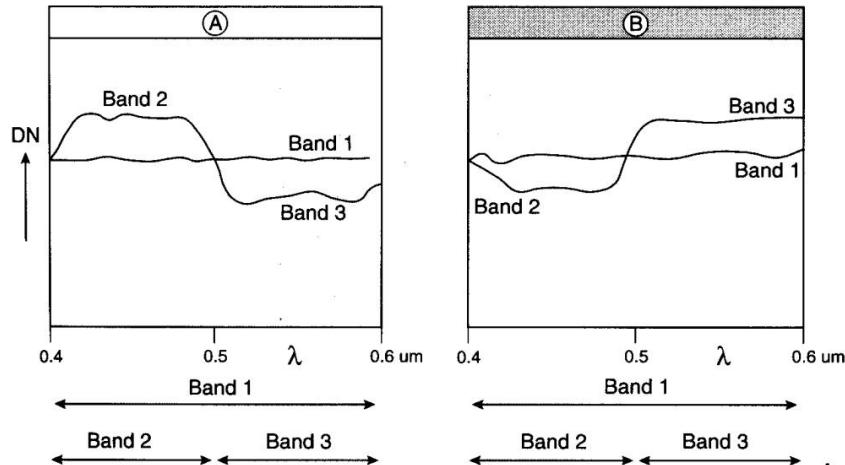


Σχήμα 1.13: Υποθετική αναπαράσταση συστημάτων τηλεπισκόπησης με διαφορετική φασματική ανάλυση (Source: Gibson, 2000)

Γενικά, μπορεί να γίνει καλύτερη διάκριση των χαρακτηριστικών της επιφάνειας από πολλαπλές στενές ζώνες, από ότι με μία μόνο ζώνη.

Για παράδειγμα, στο σχήμα 1.14, χρησιμοποιώντας την ζώνη 1 με την μεγάλη περιοχή μή-

κων κύματος, δε μπορεί να γίνει διάκριση των χαρακτηριστικών A και B. Όμως, οι φασματικές αποκρίσεις των δύο χαρακτηριστικών είναι διαφορετικές στις στενότερες ζώνες 2 και 3. Επομένως, μία πολυφασματική εικόνα που περιλαμβάνει στις ζώνες 2 και 3 μπορεί να χρησιμοποιηθεί για την διάκριση των χαρακτηριστικών A και B.



Σχήμα 1.14: Δύο διαφορετικές επιφάνειες (Α και Β) δεν είναι διαχωρίσιμες σε μία μονή ζώνη, αλλά μπορούν να διαχωριστούν σε δύο στενότερες ζώνες.

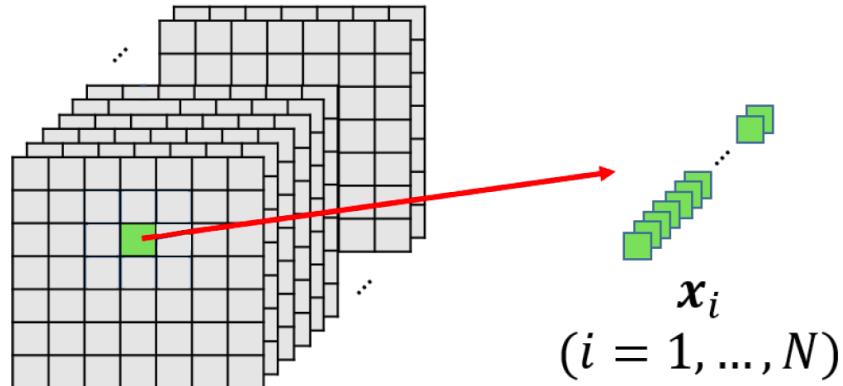
1.1.7 Ταξινόμηση φασματικής πληροφορίας

Με τον όρο ταξινόμηση εννοείται η διαδικασία επισήμανσης κάθε pixel σε κάποια κατηγορία εδαφοκάλυψης (επιφάνεια νερού, δρόμος, βλάστηση, κλπ). Προφανώς, ο άνθρωπος δε μπορεί να πάρει κάθε pixel σε μία εικόνα και να το αποδώσει σε μία κλάση εδαφοκάλυψης. Όμως το πρόβλημα δεν είναι μόνο αυτό. Ο όγκος των δεδομένων είναι πολύ μεγάλος και μεγαλώνει συνεχώς, καθώς όλο και περισσότεροι αισθητήρες σε δορυφόρους και αεροσκάφη συλλέγουν δεδομένα από την επιφάνεια της γης (περισσότεροι από 500 δορυφόροι με όργανα τηλεπισκόπησης βρίσκονται σε τροχιά). Επίσης, ο άνθρωπος περιορίζεται συνήθως στην ανάλυση δεδομένων ενός φασματικού καναλιού λόγω της δυσκολίας να ερμηνεύσει πολλές διαφορετικές εικόνες μαζί. Επιπλέον, για την ερμηνεία, ανάλυση και ταξινόμηση ενός στόχου, αποτελεί προϋπόθεση ο στόχος αυτός να είναι διακριτός - δηλαδή να μπορεί ο αναλυτής να τον διαχωρίσει από την υπόλοιπη εικόνα.

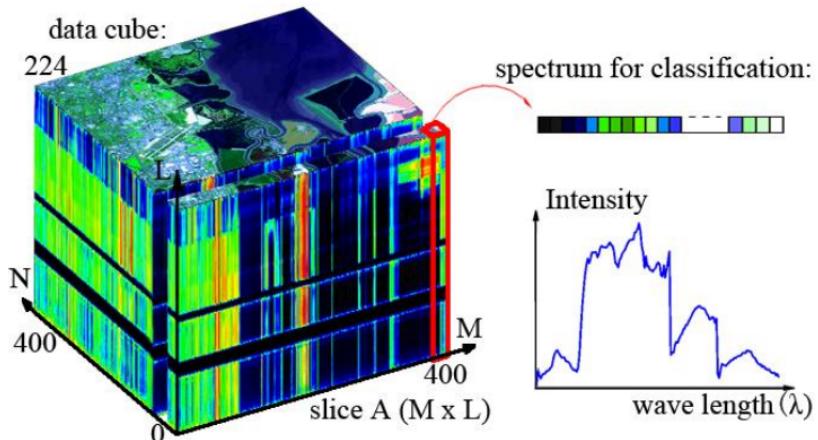
Οι παραπάνω λόγοι έχουν οδηγήσει στην ψηφιακή επεξεργασία και ανάλυση των δεδομένων τηλεπισκόπησης, η οποία μπορεί να χρησιμοποιηθεί για την αυτόματη αναγνώριση στόχων και εξαγωγή πληροφορίας χωρίς ανθρώπινη παρέμβαση.

Όταν ο υπολογιστής “βλέπει” μία εικόνα (λαμβάνει μία εικόνα ως είσοδο), ουσιαστικά βλέπει έναν πίνακα με τις τιμές των pixels. Ανάλογα με τι διαστάσεις και την ανάλυση της εικόνας, θα δει έναν π.χ. 500x500x100 πίνακα με αριθμούς (το 100 αναφέρεται στον αριθμό των φασματικών καναλιών - αν πρόκειται για μία RGB εικόνα, θα ήταν 500x500x3). Κάθε ένας από αυτούς τους αριθμούς έχει μία τιμή από 0 ως 255, που περιγράφει την ένταση του pixel σε εκείνο το σημείο (η εικόνα αποτελείται από 500x500 pixel vector, όπου κάθε διάνυσμα έχει 100 τιμές). Αυτοί οι αριθμοί αποτελούν, κατά τη διαδικασία της ταξινόμησης, τις μοναδικές εισόδους διαθέσιμες στον υπολογιστή. Η βασική ιδέα είναι ότι δίνουμε στον υπολογιστή αυτόν

τον πίνακα αριθμών και αυτός θα δώσει στην έξοδο την πιθανότητα το εν λόγω pixel vector να ανήκει σε μία κλάση (0.80 για επιφάνεια νερού, 0.15 για δρόμο 0.05 για κτήριο, κλπ).



Σχήμα 1.15: Μία πολυφασματική εικόνα και ένα pixel vector αυτής.



Σχήμα 1.16: Μία τυπική υπερφασματική εικόνα. Κάθε pixel φέρει πληροφορία για ολόκληρο το φάσμα.

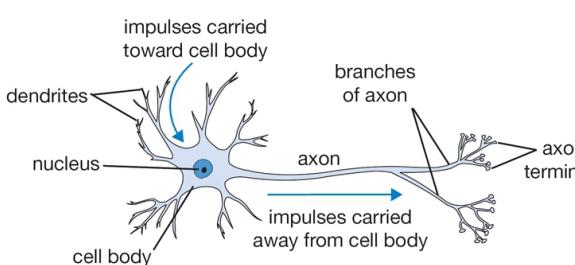
Όμως όπως έχει αναφερθεί, λόγω διαφορετικών ατμοσφαιρικών συνθηκών, φωτισμού, εποχής, υγρασίας και άλλων παραγόντων, τα τηλεπισκοπικά δεδομένα μπορεί να είναι αλλοιωμένα και αντικείμενα της ίδιας κλάσης να εμφανίζουν διαφορετικά φασματικά χαρακτηριστικά, κάνοντας δύσκολη την ταξινόμησή τους. Ακόμα, τα περισσότερα είδη βλάστησης δεν έχουν σταθερή φασματική απόκριση (π.χ. διαφορετικές συγκεντρώσεις χλωροφύλλης ανά εποχή), καθώς επίσης πολλά pixels δεν είναι καθαρά - δηλαδή μπορεί να έχουν περισσότερα από ένα στοιχεία εδαφοκάλυψης. Τέλος, σε κάποιες περιπτώσεις οι κλάσεις της ταξινόμησης μπορεί να έχουν παρόμοιες αποκρίσεις.

Από τα προηγούμενα φαίνεται ότι η ταξινόμηση φασματικών δεδομένων, αποτελεί ένα δύσκολο πρόβλημα και απαιτείται η εξαγωγή εύρωστων χαρακτηριστικών.

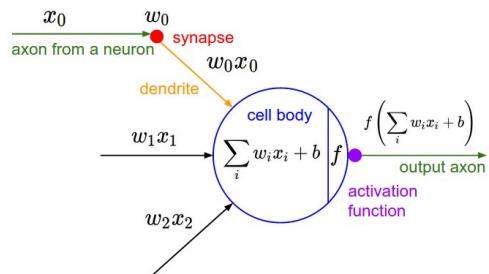
1.2 Βασικές αρχές νευρωνικών δικτύων και συνελικτικών νευρωνικών δικτύων.

1.2.1 Νευρώνας

Η κύρια υπολογιστική μονάδα του εγκεφάλου είναι ο νευρώνας. Περίπου 86 δισεκατομμύρια νευρώνες βρίσκονται στο ανθρώπινο νευρικό σύστημα και είναι συνδεδεμένοι μεταξύ τους με περίπου $10^{14} - 10^{15}$ συνάψεις. Το παρακάτω διάγραμμα δείχνει (i) έναν βιολογικό νευρώνα (i) και (ii) ένα μαθηματικό μοντέλο.



(i) Βιολογικός νευρώνας.



(ii) Μαθηματικό μοντέλο - τεχνητός νευρώνας

Σχήμα 1.17

Η αναλογία που περιγράφεται χρονολογείται στο 1943 και αναπτύχθηκε από του Warren McCulloch και Walter Pitts [MP43]. Κάθε νευρώνας λαμβάνει σήματα εισόδου από τους δενδρίτες και παράγει σήματα εξόδου κατά μήκος του άξονα. Ο άξονας τελικά διακλαδώνεται και συνδέεται με συνάψεις με τους δενδρίτες άλλων νευρώνων. Στο υπολογιστικό μοντέλο ενός νευρώνα, τα σήματα που ταξιδεύουν κατά μήκος του άξονα (π.χ x_0) αλληλεπιδρούν πολλαπλασιαστικά (w_0x_0) με τους δενδρίτες των άλλων νευρώνων με βάση το πόσο δυνατή είναι η σύναψη (w_0). Η βασική ιδέα είναι ότι η δύναμη της σύναψης (το βάρος w) μπορεί να αλλάξει και δηλώνει το πόσο επηρεάζει ο ένας νευρώνας τον άλλο. Ουσιαστικά, το βάρος τη σύναψης δηλώνει πόσο σημαντική είναι αυτή η σύναψη ως προς το σήμα εξόδου του νευρώνα. Στο βασικό μοντέλο οι δενδρίτες φέρνουν το σήμα εισόδου στο κύριο σώμα όπου όλα τα σήματα αθροίζονται. Αν το τελικό άθροισμα είναι μεγαλύτερο ενός κατωφλίου που έχει οριστεί, ο νευρώνας μπορεί να ενεργοποιηθεί, στέλνοντας έναν παλμό στον άξονά του. Στο υπολογιστικό μοντέλο, υποθέτουμε ότι οι ακριβείς χρόνοι αυτής της πυροδότησης του νευρώνα δεν έχουν σημασία και ότι μόνο η συχνότητα της πυροδότησης επικοινωνεί την πληροφορία. Βάσει αυτής της ερμηνείας μοντελοποιούμε τον ρυθμό πυροδότησης του νευρώνα με μία συνάρτηση ενεργοποίησης f , η οποία αναπαριστά τη συχνότητα των παλμών κατά μήκος του άξονα. Ιστορικά, μία σύνηθης επιλογή συνάρτησης ενεργοποίησης είναι η στιγμοειδής συνάρτηση, δεδομένου ότι λαμβάνει μία είσοδο πραγματικής τιμής (την ισχύ του σήματος μετά το άθροισμα) και έχει σύνολο τιμών εξόδου από 0 έως 1.

Με άλλα λόγια, κάθε νευρώνας εκτελεί ένα εσωτερικό γινόμενο της εισόδου με τα βάρη, προσθέτει έναν παράγοντα αμεροληψίας b (bias) και εφαρμόζει μία μη-γραμμικότητα (συνάρτηση ενεργοποίησης), σε αυτή την περίπτωση $f(x) = \frac{1}{1+\exp(-x)}$.

Πρέπει να τονιστεί ότι η παραπάνω προσέγγιση και αυτή η αναλογία του βιολογικού μοντέλου με το υπολογιστικό είναι αδρή. Για παράδειγμα, υπάρχουν διαφορετικοί τύποι νευρώνων, ο

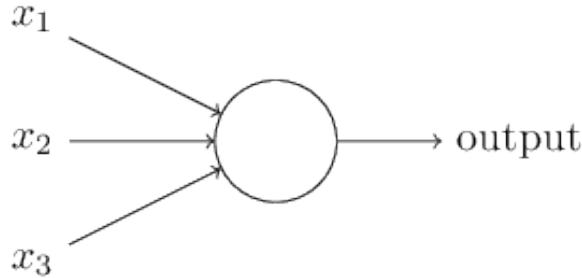
καθένας με διαφορετικές ιδιότητες, οι δενδρίτες των βιολογικών νευρώνων εκτελούν σύνθετους μη γραμμικούς υπολογισμούς, οι συνάψεις δεν είναι απλά βάρη αλλά σύνθετα, μη γραμμικά, δυναμικά συστήματα, κ.ο.κ.

1.2.2 Perceptron

Ο αλγόριθμος perceptron [Ros58] είναι ο πρώτος και πιο απλός αλγόριθμος που υλοποιεί έναν τεχνητό νευρώνα. Λαμβάνει εισόδους x_1, x_2, \dots με δυαδικές τιμές (0,1) και παράγει μία δυαδική έξοδο.

Στο σχήμα 1.18 φαίνεται ένα perceptron με τρεις εισόδους x_1, x_2, x_3 . Θα μπορούσε να έχει περισσότερες ή λιγότερες εισόδους. Ο Rosenblatt πρότεινε έναν απλό κανόνα για τον υπολογισμό της έξοδου. Εισήγαγε τα βάρη w_1, w_2, \dots , τα οποία είναι πραγματικοί αριθμοί που εκφράζουν πόσο σημαντική είναι η αντίστοιχη είσοδος ως προς την έξοδο. Η έξοδος του νευρώνα, 0 ή 1, καθορίζεται από το αν το σταθμισμένο άθροισμα $\sum w_j x_j$ είναι μικρότερο ή μεγαλύτερο μιας τιμής κατωφλίου (threshold value). Όπως τα βάρη, και τη τιμή κατωφλίου είναι ένας πραγματικός αριθμός και αποτελεί παράμετρο του νευρώνα.

$$\text{output} = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{threshold} \end{cases} \quad (1.1)$$

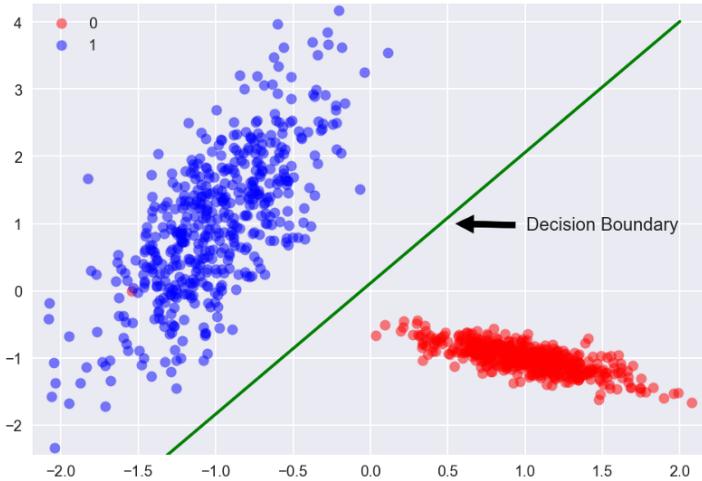


Σχήμα 1.18: Σχηματική απεικόνιση του perceptron.

Ο αλγόριθμος perceptron είναι ένας αλγόριθμος δυαδικής ταξινόμησης και μπορεί να εφαρμοσθεί μόνο σε γραμμικά διαχωρίσιμα δεδομένα. Αυτό που κάνει στην πραγματικότητα είναι να ορίζει **όρια απόφασης (decision boundaries)** για να διαχωρίζει τα δεδομένα (σχήμα 1.19).

Με άλλα λόγια, μπορούμε να φανταστούμε το perceptron ως μία μηχανή λήψης αποφάσεων (στην περίπτωσή μας οι απόφαση παίρνει τη μορφή 0 ή 1). Παίρνει αυτές τις αποφάσεις σταθιζόντας με βάρη τις εισόδους που λαμβάνει. Αλλάζοντας τις τιμές των βαρών και του κατωφλίου μπορούμε να ορίσουμε διαφορετικά μοντέλα για τη λήψη αποφάσεων, δηλαδή διαφορετικά decision boundaries.

Για απλοποίηση, θα κάνουμε δύο αλλαγές στην εξίσωση του perceptron. Αρχικά το $\sum_j w_j x_j > \text{threshold}$ θα το γράψουμε ως το εσωτερικό γινόμενο $w \cdot x \equiv \sum_j w_j x_j$, όπου τα w και x είναι διανύσματα που εκφράζουν τα βάρη και τις εισόδους αντίστοιχα. Η δεύτερη αλλαγή είναι η μετακίνηση του threshold στο αριστερό μέρος της εξίσωσης και η αντικατάστασή του με έναν συντελεστή γνωστό ως συντελεστή αμεροληψίας ($bias, b$). Ξαναγράφοντας την εξίσωση με το $bias$ αντί το $threshold$, προκύπτει:

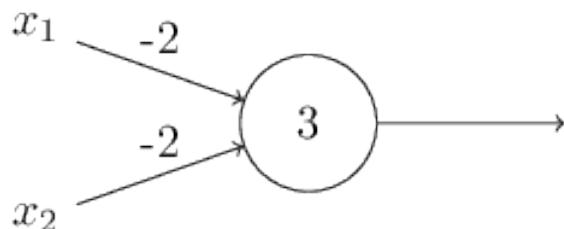


Σχήμα 1.19: Γραμμικός διαχωρισμός δεδομένων.

$$\text{output} = \begin{cases} 0 & \text{if } w \cdot x + b \leq 0 \\ 1 & \text{if } w \cdot x + b > 0 \end{cases} \quad (1.2)$$

Μπορούμε να φανταστούμε τον συντελεστή αμεροληψίας ως ένα μέτρο του πόσο εύκολο είναι να οδηγήσουμε την έξοδο του perceptron στο 1, ή βιολογικά μιλώντας, το πόσο εύκολο είναι να “πυροδοτήσουμε” τον νευρώνα. Μεγάλο *bias* οδηγεί πολύ εύκολα την έξοδο στο 1. Αντίθετα, αρνητικό *bias* κάνει δύσκολη αυτή την έξοδο [Nnd].

Παραπάνω, το perceptron έχει περιγραφεί ως μία μέθοδο στάθμισης των εισόδων για τη λήψη αποφάσεων. Ένας άλλος τρόπος χρήσης του είναι ο υπολογισμός λογικών πυλών όπως οι AND, OR και NAND. Για παράδειγμα ένα perceptron δύο εισόδους με βάρος -2 για κάθε είσοδο και *bias* 3 φαίνεται στο σχήμα 1.20.



Σχήμα 1.20: Ένα perceptron που υλοποιεί την λογική πύλη NAND.

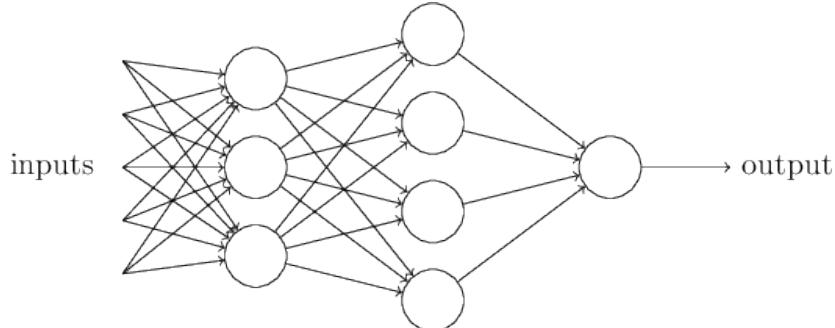
Η είσοδος 00 δίνει έξοδο 1, αφού το $(-2) * 0 + (-2) * 0 + 3 = 3$ είναι θετικό. Ομοίως οι είσοδοι 01 και 10 δίνουν έξοδο 1 ενώ η είσοδος 11 δίνει έξοδο 0 ($(-2) * 1 + (-2) * 1 + 3 = -1$). Άρα το perceptron υλοποιεί την πύλη NAND.

Έπειδή με πύλες NAND μπορεί να υλοποιηθεί οποιοδήποτε κύκλωμα, προκύπτει ότι τα perceptrons με τουλάχιστον ένα κρυμμένο επίπεδο μπορούν να μάθουν οποιαδήποτε συνάρτηση με οποιοδήποτε βαθμό ακρίβειας (Universal Approximation Theorem [Cyb89]).

1.2.3 Νευρωνικά δίκτυα πολλών επιπέδων

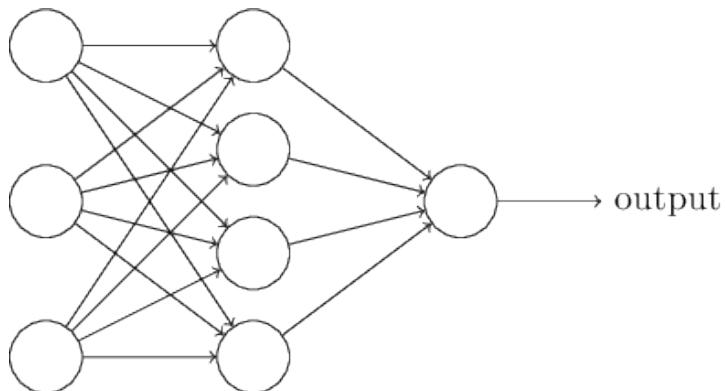
Το perceptron προφανώς δε προσεγγίζει και πολύ το μοντέλο του ανθρώπου για τη λήψη αποφάσεων. Ένα σύνθετο δίκτυο αποτελούμενο από πολλά perceptrons θα μπορούσε να πάρει αποφάσεις με διαφορετικό, πιο σύνθετο τρόπο.

Στο δίκτυο του σχήματος 1.21 η αριστερή “στήλη” με τα perceptrons, που ονομάζεται πρώτο επίπεδο, παίρνει τρεις πολύ απλές αποφάσεις, σταθμίζοντας τις εισόδους. Το δεύτερο επίπεδο παίρνει αποφάσεις σταθμίζοντας τα αποτελέσματα του πρώτου επιπέδου. Κατ’ αυτόν τον τρόπο ένα perceptron που βρίσκεται στο δεύτερο επίπεδο μπορεί πάρει αποφάσεις σε ένα πιο σύνθετο και αφαιρετικό επίπεδο από ένα perceptron του πρώτου επιπέδου.



Σχήμα 1.21: Δίκτυο perceptron πολλών επιπέδων (Multi Layer Perceptron - MLP).

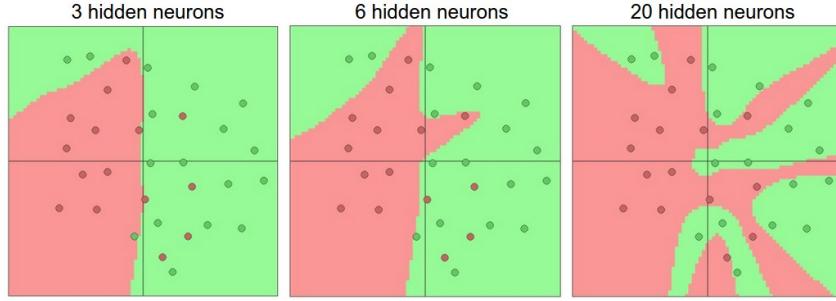
Το τρίτο επίπεδο του δικτύου στο σχήμα αποτελείται από ένα perceptron, συνεπώς παράγει μία τελική έξοδο (0 ή 1) και μπορεί να πάρει αποφάσεις, ή να διακρίνει δύο κλάσεις.



Σχήμα 1.22: Νευρωνικό δίκτυο με ένα κρυμμένο επίπεδο.

Τα δίκτυα πολλών επιπέδων μπορούν να μάθουν πιο σύνθετες συναρτήσεις, δηλαδή πιο σύνθετα decision boundaries για τον διαχωρισμό δεδομένων, από ένα απλό perceptron. Επιπλέον, το πόσο σύνθετα είναι τα decision boundaries που μαθαίνουν τα δίκτυα πολλών επιπέδων εξαρτάται και από τον αριθμό των νευρώνων στα κρυμμένα επίπεδα.

Συχνά, η τροφοδότηση των δεδομένων στο δίκτυο είναι σχετικά απλή. Για παράδειγμα, αν το πρόβλημα ήταν η ταξινόμηση grayscale εικόνων 64x64 pixels σε δύο κλάσεις (γάτα, σκύλος), ένας τρόπος θα ήταν να εισάγουμε τις τιμές των pixels στους νευρώνες εισόδου. Πιο συγκεκριμένα, θα χρειαζόμασταν $64 \times 64 = 4096$ νευρώνες εισόδου με τις τιμές των pixels κανονικοποιημένες ανάμεσα στο 0 και 1 και έναν νευρώνα εξόδου, ο οποίος θα έδινε έξοδο 0 για



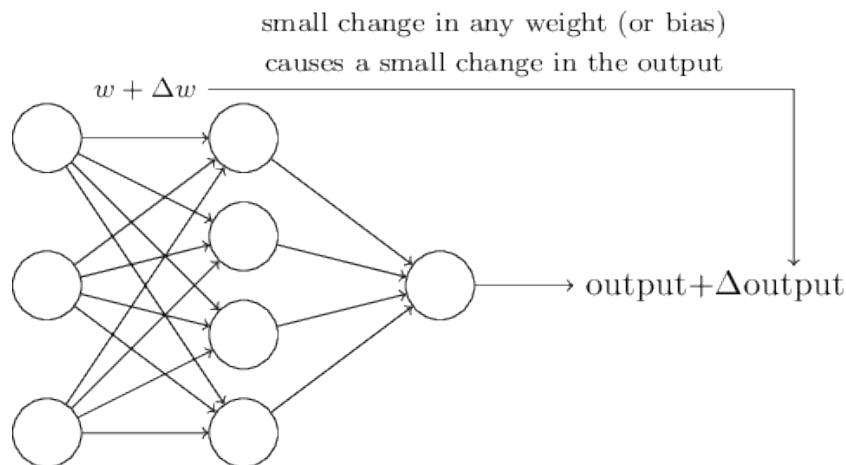
Σχήμα 1.23: Διαφορετικά decision boundaries για διαφορετικό αριθμό νευρώνων.

τιμές μικρότερες του 0.5 (αποδίδοντας την εικόνα στην κλάση γάτα) και 1 για τιμές μεγαλύτερες του 0.5 (αποδίδοντας την εικόνα στην κλάση σκύλος).

Να σημειωθεί ότι τα δίκτυα που έχουν παρουσιαστεί παραπάνω ονομάζονται **προσωτροφοδοτούμενα (feedforward)**, γιατί κάθε επίπεδο τροφοδοτείται από το προηγούμενο και μόνο από το προηγούμενο χωρίς να υπάρχει ανατροφοδότηση.

1.2.4 Σιγμοειδείς νευρώνες

Ας θεωρήσουμε ότι έχουμε ένα δίκτυο με perceptrons το οποίο θέλουμε να χρησιμοποιήσουμε ώστε να μάθει να επιλύει ένα πρόβλημα. Για παράδειγμα, η είσοδος του δικτύου μπορεί να είναι τα pixel vectors μιας πολυφασματικής εικόνας, και θέλουμε το δίκτυο να μάθει τα βάρη και τα *biases* ώστε η έξοδος του δικτύου να ταξινομεί σωστά τα pixel vectors στις κλάσεις (π.χ. νερό, δρόμος, χωράφι, κλπ). Όσων αφορά τη μάθηση αυτών των παραμέτρων, αυτό που θέλουμε είναι μικρές αλλαγές στα βάρη και στα *biases* να προκαλούν μικρές αλλαγές στην έξοδο του δικτύου. Αυτή είναι η ιδιότητα που κάνει δυνατή την μάθηση (σχήμα 1.24).



Σχήμα 1.24

Αν όντως μικρές αλλαγές στις παραμέτρους προκαλούσαν μικρές αλλαγές στην έξοδο, θα μπορούσαμε να τροποποιήσουμε τα βάρη και τα *biases*, ώστε να κάνουμε το δίκτυο να συμπεριφέρεται με τον τρόπο που θέλουμε. Για παράδειγμα, αν το δίκτυο ταξινομούσε λανθασμένα ένα pixel vector ως νερό ενώ ήταν δρόμος, θα μπορούσαμε να κάνουμε μία μικρή αλλαγή στις παραμέτρους ώστε το δίκτυο να οδηγούνταν προς τη σωστή ταξινόμηση. Επαναλαμβάνοντας

αυτή τη διαδικασία, αλλάζοντας δηλαδή τις παραμέτρους, θα οδηγούσαμε το δίκτυο να παράξει μία όλο και καλύτερη έξοδο, πιο κοντά στην πραγματική τιμή/κλάση (δρόμος). Κατ' αυτόν τον τρόπο το δίκτυο θα μάθαινε.

Αυτές οι μικρές αλλαγές στην έξοδο, όμως, δε συμβαίνουν στα perceptrons. Μικρές αλλαγές μπορεί να οδηγήσουν το σταθμισμένο άθροισμα σε τιμές λίγο μεγαλύτερες ή μικρότερες του κατωφλίου (0.1 ή -0.1) και επομένως πολύ μεγάλες αλλαγές στην έξοδο (το 0.1 δίνει έξοδο 1 ενώ το -0.1 δίνει 0). Μία αλλαγή από 0 σε 1 μπορεί να οδηγήσει τη συμπεριφορά του υπόλοιπου δικτύου σε μία απότομη αλλαγή με έναν σύνθετο τρόπο. Και ενώ με την αλλαγή αυτή το pixel vector να αναγνωρίζεται σωστά ως "δρόμος", η συμπεριφορά του δικτύου σε άλλα pixel vectors είναι πού πιθανό να έχει αλλάξει με έναν ανεξέλεγκτο τρόπο. Αυτό καθιστά δύσκολο το να οδηγήσουμε το δίκτυο να παράγει τις επιθυμητές εξόδους τροποποιώντας σταδιακά τα βάρη.

Το πρόβλημα αυτό αντιμετωπίζεται με τους σιγμοειδείς νευρώνες. Οι σιγμοειδείς νευρώνες είναι παρόμοιοι με τα perceptrons, με τη διαφορά ότι είναι τροποποιημένοι έτσι ώστε να έχουμε την επιθυμητή ιδιότητα που αναφέρθηκε παραπάνω: μικρές αλλαγές στις παραμέτρους να προκαλούν μικρές αλλαγές στην έξοδο.

Ένας σιγμοειδής νευρώνας μπορεί να σχεδιαστεί ακριβώς όπως ο perceptron στο σχήμα 1.18 με τις εξής διαφορές:

- Οι είσοδοι x_1, x_2, \dots αντί να παίρνουν τιμές 0 και 1, μπορούν να πάρουν οποιαδήποτε τιμή ανάμεσα στο 0 και 1 (π.χ. 0,638).
- Η έξοδος δεν είναι 0 ή 1, αλλά $\sigma(w \cdot x + b)$, όπου σ ονομάζεται η σιγμοειδής συνάρτηση (sigmoid function) και ορίζεται ως:

$$\sigma(z) \equiv \frac{1}{1 + e^{-z}}. \quad (1.3)$$

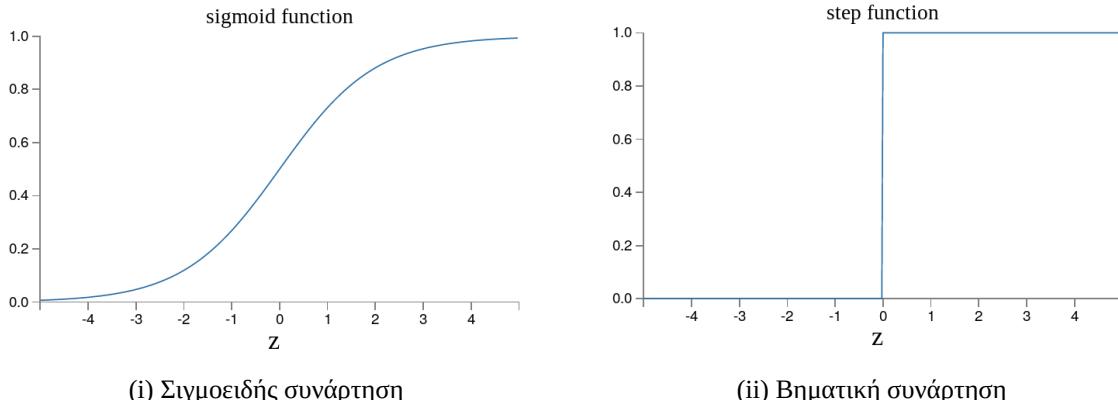
Η κάπως διαφορετικά, η έξοδος του σιγμοειδή νευρώνα με εισόδους x_1, x_2, \dots , βάρη w_1, w_2, \dots και bias b είναι:

$$\frac{1}{1 + \exp(-\sum_j w_j x_j - b)}. \quad (1.4)$$

Για πολύ μεγάλες θετικές τιμές του $z \equiv w \cdot x + b$ το $e^{-z} \approx 0$ και $\sigma(z) \approx 1$. Με άλλα λόγια, όταν το z είναι ένας μεγάλος και θετικός αριθμός η έξοδος της σιγμοειδούς τείνει στο 1. Όμοια μπορεί να δειχτεί ότι πολύ αρνητικές τιμές του z δίνουν έξοδο που τείνει στο 0. Φαίνεται από τα παραπάνω ότι για ακραίες τιμές του z η σιγμοειδής προσεγγίζει την βηματική συνάρτηση. Η διαφορά βρίσκεται στις ενδιάμεσες τιμές του z , οι οποίες δίνουν ενδιάμεσες και πιο ομαλές τιμές $\sigma(z)$. Αυτή η ομαλότητα της σ σημαίνει ότι μικρές αλλαγές Δw_j στα βάρη και Δb στο bias θα παράξουν μία μικρή αλλαγή Δoutput στην έξοδο του νευρώνα:

$$\Delta \text{output} \approx \sum_j \frac{\partial \text{output}}{\partial w_j} \Delta w_j + \frac{\partial \text{output}}{\partial b} \Delta b, \quad (1.5)$$

όπου το άθροισμα είναι καθόλα τα βάρη, w_j και οι $\partial \text{output}/\partial w_j$ και $\partial \text{output}/\partial b$ είναι οι μερικές παράγωγοι της εξόδου ως προς το w_j και το b , αντίστοιχα.



Σχήμα 1.25: Η σιγμοειδής συνάρτηση μοιάζει με μία πιο ομαλή έκδοση της βηματικής συνάρτησης.

Το Δουτρυτ είναι μία γραμμική συνάρτηση των Δw_j και Δb . Αυτή η γραμμικότητα καθιστά εύκολες τις μικρές αλλαγές της εξόδου και είναι δυνατή μόνο επειδή αντικαταστήσαμε την βηματική συνάρτηση με μία συνεχή και άρα παραγωγίσιμη συνάρτηση!

Αν και αποτελούνται από σιγμοειδείς νευρώνες, για ιστορικούς κυρίως λόγους τα δίκτυα πολλών επιπέδων συχνά ονομάζονται perceptrons πολλών επιπέδων (Multi Layer Perceptrons - MLP).

Καθ' όλο το υπόλοιπο της παρούσας εργασίας, καθώς και στη σύγχρονη βιβλιογραφία, τα MLP απαντώνται και με την ονομασία Fully-Connected Networks καθώς όλοι οι νευρώνες ενός επιπέδου συνδέονται με νευρώνες του προηγούμενου.

1.2.5 Επιβλεπόμενη και μη-επιβλεπόμενη μάθηση

Η επιβλεπόμενη και η μη-επιβλεπόμενη μάθηση είναι οι δύο τρόποι εκπαίδευσης των νευρωνικών δικτύων και έχουν άμεση σχέση με τα διαθέσιμα δεδομένα. Στην επιβλεπόμενη (supervised) μάθηση έχουμε διαθέσιμα τα δεδομένα (π.χ. τις εικόνες, ή τα pixel vectors) και τις επισημάνσεις (labels) τους. Η λογική είναι απλή: τροφοδοτούμε το δίκτυο με τα δεδομένα και συγκρίνουμε την έξodo του δικτύου με την πραγματική. Με αυτόν τον τρόπο το δίκτυο “μαθαίνει από τα λάθη του”. Στην μη-επιβλεπόμενη (unsupervised) μάθηση, οι επισημάνσεις δεν είναι διαθέσιμες. Παρακάτω, στην ενότητα όπου παρουσιάζονται οι Autoencoders, φαίνεται πως είναι δυνατή αυτή η εκπαίδευση.

1.2.6 Μέθοδος καθόδου με βάση την κλίση (Gradient Descent)

Αφού έχουμε ορίσει την αρχιτεκτονική του νευρωνικού δικτύου, θα πρέπει να δούμε πώς ακριβώς το δίκτυο μαθαίνει να υπολογίζει τα βάρη. Για αυτό το στάδιο, την εκπαίδευση του δικτύου, είναι απαραίτητο να υπάρχουν διαθέσιμα δεδομένα τα οποία το δίκτυο θα χρησιμοποιήσει, και ονομάζονται δεδομένα εκπαίδευσης. Τα δεδομένα αυτά, στην περίπτωση της ταξινόμησης εικόνας, περιέχουν τις εικόνες μαζί με την πληροφορία της κλάσης στην οποία ανήκουν.

Συνήθως, το σύνολο των δεδομένων που έχουμε στη διάθεσή μας, χωρίζεται σε δεδομένα εκπαίδευσης (training data) και δεδομένα ελέγχου (test data), τα οποία χρησιμοποιούμε για να αξιολογήσουμε πόσο καλά έχει μάθει το δίκτυο (να ταξινομεί τις εικόνες). Τα δύο αυτά σύνολα

πρέπει να είναι ανεξάρτητα και να μην περιέχουν κοινά δείγματα. Αφού το δίκτυο εκπαιδευτεί στα δεδομένα εκπαίδευσης, ελέγχεται η ικανότητα γενίκευσής του, δηλαδή η ικανότητά του να ταξινομήσει άγνωστες εικόνες, τις οποίες δεν έχει “ξαναδεί”. Να σημειωθεί ότι αποτελεί κοινή πρακτική, από το σύνολο των δεδομένων εκπαίδευσης να σχηματίζεται το σύνολο δεδομένων επαλήθευσης (validation set). Αυτό είναι συνήθως το 10-20% του συνόλου των δεδομένων και χρησιμοποιείται κατά τη διάρκεια της εκπαίδευσης, αντί του συνόλου ελέγχου, ώστε να υπάρχει μία ένδειξη αν το δίκτυο “μαθαίνει” ή όχι.

Για να συνδέσουμε όσα αναφέρονται στην ενότητα αυτή με την τηλεπισκόπηση, θεωρούμε ένα νευρωνικό δίκτυο το οποίο χρησιμοποιείται για την ταξινόμηση των pixel vectors μιας υπερφασματικής εικόνας εκατό καναλιών, στις κλάσεις “δρόμος”, “νερό”, “δέντρα”, “κτήρια”. Σύμφωνα με όσα έχουν προηγηθεί, φαίνεται πως το επίπεδο εσόδου θα πρέπει να έχει εκατό νευρώνες και το επίπεδο εξόδου τέσσερις νευρώνες. Για παράδειγμα, αν ένα συγκεκριμένο δείγμα x των δεδομένων εκπαίδευσης είναι rixel νερού, τότε η επιθυμητή έξοδος y δηλώνεται ως $y(x) = (0, 1, 0, 0)^T$. (Το T χρησιμοποιείται για τη μετατροπή του διανύσματος γραμμής σε διάνυσμα στήλης.).

Χρειαζόμαστε έναν αλγόριθμο ο οποίος θα μας επιτρέπει να βρίσκουμε βάρη και συντελεστές αμεροληψίας, ώστε η έξοδος του δικτύου να προσεγγίζει τις επιθυμητές εξόδους $y(x)$ για όλα τα δεδομένα εκπαίδευσης x . Για να ποσοτικοποιήσουμε πόσο καλή είναι αυτή η προσέγγιση, ορίζουμε μία συνάρτηση κόστους (cost function):

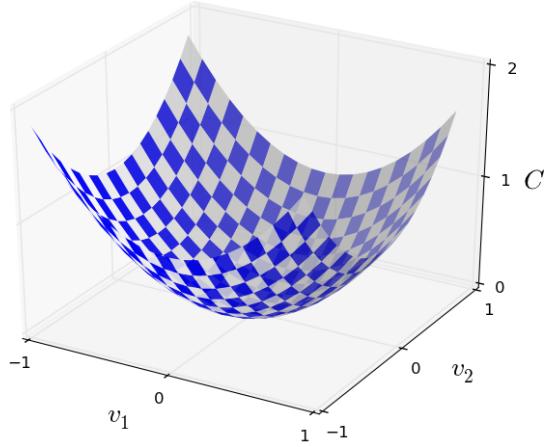
$$C(w, b) \equiv \frac{1}{2n} \sum_x \|y(x) - a\|^2 \quad (1.6)$$

Το w είναι όλα τα βάρη στο δίκτυο, το b όλα τα biases, το n ο συνολικός αριθμός των δεδομένων εκπαίδευσης, το a είναι το διάνυσμα εξόδου του δικτύου με το x ως είσοδο και το άθροισμα γίνεται στο σύνολο των δεδομένων εκπαίδευσης x .

Φυσικά, το a εξαρτάται από την είσοδο x , το w και το b . Η συνάρτηση C ονομάζεται τετραγωνική συνάρτηση κόστους ή μέσο τετραγωνικό σφάλμα (Mean Square Error - MSE). Η συνάρτηση $C(w, b)$ είναι θετική και ελαχιστοποιείται όταν η έξοδος του δικτύου a προσεγγίζει την επιθυμητή/πραγματική τιμή $y(x)$ για όλες τις εισόδους των δεδομένων εκπαίδευσης. Συνεπώς, ο αλγόριθμος εκπαίδευσης είναι επιτυχημένος αν μάθει βάρη και biases τέτοια ώστε $C(w, b) \approx 0$. Σκοπός, λοιπόν, του αλγόριθμου εκπαίδευσης είναι η εύρεση βαρών και biases που ελαχιστοποιούν όσο το δυνατόν περισσότερο την συνάρτηση κόστους. Αυτή η ελαχιστοποίηση που περιγράφεται παραπάνω, γίνεται με χρήση του αλγόριθμου καθόδου με βάση την κλίση (gradient descent).

Επειδή δε μπορούμε να οπτικοποιήσουμε την συνάρτηση κόστους στις πραγματικές διαστάσεις των ανεξάρτητων μεταβλητών w και b , καθώς αυτές μπορεί να είναι εκατοντάδες χιλιάδες, θα θεωρήσουμε ότι θέλουμε να ελαχιστοποιήσουμε την συνάρτηση κόστους $C(v)$, η οποία είναι μία συνάρτηση δύο μεταβλητών (v_1, v_2) , όπως φαίνεται στο σχήμα 1.26.

Επειδή στην πραγματικότητα η συνάρτηση κόστους μπορεί να εξαρτάται από εκατομμύρια ανεξάρτητες μεταβλητές, η εύρεση του ελαχίστου της συνάρτησης κόστους με την αναλυτική προσέγγιση καθίσταται πρακτικά αδύνατη. Το πρόβλημα εν τέλει αντιμετωπίζεται με διαφορετική προσέγγιση. Σε πολλές εργασίες και αναφορές, έχει γίνει η αναλογία της καμπύλης του σχήματος 1.26 με μία κοιλάδα (ή έναν λόφο) και μία μπάλα που κατρακυλάει μέχρι να φτάσει σε ένα ελάχιστο σημείο. Μετακινώντας την μπάλα κατά μία πολύ μικρή ποσότητα Δv_1 στην κατεύθυνση του v_1 και Δv_2 στην κατεύθυνση του v_2 , αλγεβρικά προκύπτει:



Σχήμα 1.26: Τετραγωνική συνάρτηση κόστους δύο μεταβλητών.

$$\Delta C \approx \frac{\partial C}{\partial v_1} \Delta v_1 + \frac{\partial C}{\partial v_2} \Delta v_2 \quad (1.7)$$

Ορίζοντας την κλίση της συνάρτηση ∇C , έχουμε:

$$\Delta C \approx \nabla C \cdot \Delta v \quad (1.8)$$

Δεδομένου ότι θέλουμε αρνητικές τιμές ΔC , πρέπει να επιλέξουμε κατάλληλο Δv , όπως:

$$\Delta v = -\eta \nabla C, \quad (1.9)$$

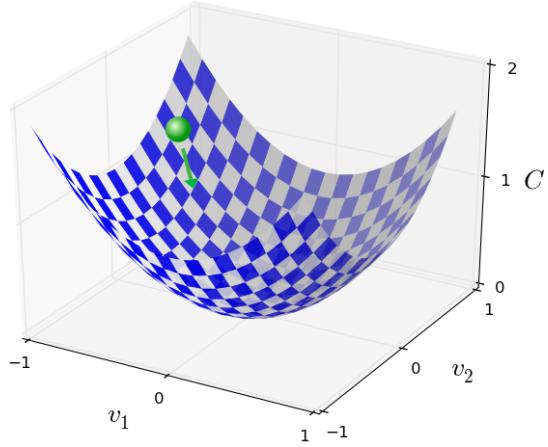
, όπου το η είναι μία μικρή θετική ποσότητα, γνωστή ως ρυθμός μάθησης (learning rate). Η εξίσωση 1.8 γίνεται: $\Delta C \approx -\eta \nabla C \cdot \nabla C = -\eta \|\nabla C\|^2$. Επειδή $\|\nabla C\|^2 \geq 0$, εξασφαλίζεται ότι $\Delta C \leq 0$, δηλαδή ότι η συνάρτηση C πάντα θα μειώνεται, αν μεταβάλλουμε το Δv σύμφωνα με την εξίσωση 1.9.

Συνεπώς, χρησιμοποιούμε την εξίσωση 1.9 για να υπολογίσουμε μία τιμή για το Δv και μετακινούμε την μπάλα στη θέση v σύμφωνα με:

$$v \rightarrow v' = v - \eta \nabla C. \quad (1.10)$$

Επαναλαμβάνοντας αυτόν τον κανόνα πολλές φορές, γίνονται βήματα που μειώνουν την συνάρτηση και, ιδανικά, οδηγούν στο ολικό ελάχιστο της συνάρτησης.

Συνοπτικά, ο τρόπος που δουλεύει ο αλγόριθμος gradient descent είναι με τον επαναληπτικό υπολογισμό της κλίσης ∇C και της κίνησης προς την αντίθετη κατεύθυνση, κατεβαίνοντας τον "λόφο" (σχήμα 1.27).



Σχήμα 1.27: Κίνηση της μπάλας για την ελαχιστοποίηση της συνάρτησης.

Φαίνεται πως πρέπει να γίνει προσεκτική επιλογή του ρυθμού μάθησης η , ώστε να μη προκύψει $\Delta C > 0$ και ταυτόχρονα να μην έχει πολύ μικρή τιμή, γιατί ο αλγόριθμος θα είναι αργός, καθώς θα χρειασθεί πολλές επαναλήψεις μέχρι να φτάσει στο ελάχιστο της συνάρτησης.

Η παρουσίαση του αλγορίθμου συναρτήσει δύο μεταβλητών έγινε καθαρά για λόγους οπτικοποίησης. Προφανώς, η συνάρτηση C μπορεί να είναι μία συνάρτηση περισσότερων μεταβλητών (v_1, \dots, v_m) και η μεταβολή ΔC που προκύπτει από μία μικρή αλλαγή $\Delta v = (\Delta v_1, \dots, \Delta v_m)^T$ να είναι:

$$\Delta C \approx \nabla C \cdot \Delta v, \quad (1.11)$$

όπου η κλίση ∇C είναι ένα διάνυσμα:

$$\nabla C \equiv \left(\frac{\partial C}{\partial v_1}, \dots, \frac{\partial C}{\partial v_m} \right)^T \quad (1.12)$$

Οι εξισώσεις 1.9 και 1.10 εξακολουθούν να ισχύουν, εξασφαλίζοντας τον υπολογισμό των θέσεων v που οδηγούν στο ελάχιστο της συνάρτησης, ανεξάρτητα από τον αριθμό των μεταβλητών.

Ο αλγόριθμος gradient descent εφαρμόζεται για τη μάθηση στα νευρωνικά δίκτυα. Η ιδέα είναι να χρησιμοποιήσουμε τον αλγόριθμο για να βρούμε τα βάρη w_k και τα biases b_l τα οποία ελαχιστοποιούν την συνάρτηση κόστους (εξίσωση 1.6). Αντικαθιστώντας τα βάρη και το bias στον κανόνα ανανέωσης των τιμών του αλγόριθμου προκύπτει:

$$w_k \rightarrow w'_k = w_k - \eta \frac{\partial C}{\partial w_k} \quad (1.13)$$

$$b_l \rightarrow b'_l = b_l - \eta \frac{\partial C}{\partial b_l} \quad (1.14)$$

Εφαρμόζοντας επαναληπτικά τον παραπάνω κανόνα ανανέωσης των βαρών, οδηγούμαστε στο ελάχιστο της συνάρτησης κόστους. Με άλλα λόγια, αυτός είναι ο κανόνας με τον οποίο μαθαίνει ένα νευρωνικό δίκτυο.

Stochastic Gradient Descent

Ένα πρόβλημα που εμφανίζεται στον αλγόριθμο gradient descent είναι ότι για τον υπολογισμό της κλίσης ∇C πρέπει να υπολογίσουμε πρώτα τις κλίσεις ∇C_x , ξεχωριστά για κάθε είσοδο x και έπειτα να πάρουμε τη μέση τιμή τους $\nabla C = \frac{1}{n} \sum_x \nabla C_x$. Όταν ο αριθμός των εισόδων είναι μεγάλος αυτή η διαδικασία γίνεται χρονοβόρα. Για αυτό, χρησιμοποιείται μια παραλλαγή του αλγορίθμου που ονομάζεται stochastic gradient descent [KW52], ώστε να επιταχυνθεί η διαδικασία. Η ιδέα είναι να υπολογίσουμε την κλίση ∇C υπολογίζοντας τις κλίσεις ∇C_x για ένα μικρό δείγμα εισόδων που έχουν επιλεγεί τυχαία και παίρνοντας τον μέσο όρο αυτού του μικρού δείγματος. Με αυτόν τον τρόπο, υπολογίζεται μία καλή προσέγγιση της πραγματικής κλίσης ∇C και επιταχύνει τον αλγόριθμο gradient descent, άρα και την μάθηση.

Πιο συγκεκριμένα, ο αλγόριθμος stochastic gradient descent δουλεύει επιλέγοντας τυχαία έναν μικρό αριθμό m εισόδων από τα δεδομένα εκπαίδευσης. Συμβολίζουμε αυτά τα δείγματα ως X_1, X_2, \dots, X_m και αναφερόμαστε σε αυτά ως *mini-batch*. Δεδομένου ότι το μέγεθος m του mini-batch είναι κατάλληλα μεγάλο, μπορούμε να υποθέσουμε ότι η μέση τιμή των ∇C_{x_j} θα είναι προσεγγιστικά ίση με την μέση τιμή όλων των ∇C_x , δηλαδή:

$$\frac{\sum_{j=1}^m \nabla C_{X_j}}{m} \approx \frac{\sum_x \nabla C_x}{n} = \nabla C, \quad (1.15)$$

$$\nabla C \approx \frac{1}{m} \sum_{j=1}^m \nabla C_{X_j}, \quad (1.16)$$

όπου το δεύτερο άθροισμα είναι καθ' όλο το σύνολο των δεδομένων εκπαίδευσης.

Οι εξισώσεις 1.13 και 1.14 γίνονται:

$$w_k \rightarrow w'_k = w_k - \frac{\eta}{m} \sum_j \frac{\partial C_{X_j}}{\partial w_k} \quad (1.17)$$

$$b_l \rightarrow b'_l = b_l - \frac{\eta}{m} \sum_j \frac{\partial C_{X_j}}{\partial b_l}, \quad (1.18)$$

όπου τα αθροίσματα γίνονται για όλα τα δείγματα εκπαίδευσης X_j που βρίσκονται στο παρόν mini-batch. Κάθε φορά επιλέγεται ένα άλλο mini-batch μέχρις ότου εξαντλήσουμε τα δεδομένα εκπαίδευσης, δηλαδή μία εποχή εκπαίδευσης. Επειτα ξεκινάει η επόμενη εποχή.

Συνοπτικά είναι ευκολότερο να εφαρμόσουμε τον αλγόριθμο με mini-batches σε σχέση με την εφαρμογή σε ολόκληρο το σύνολο δεδομένων εκπαίδευσης. Αν είχαμε εισόδους $n = 60000$ και επιλέγαμε μέγεθος mini-batch $m = 10$ θα είχαμε επιτάχυνση (x6000) στον υπολογισμό της κλίσης! Φυσικά, οι τιμές θα ήταν προσεγγίσεις της "πραγματικής" κλίσης (εννοώντας την κλίση στο σύνολο των δεδομένων), και θα υπήρχαν στατιστικές ανωμαλίες, αλλά και πάλι θα είχαμε μία γενική κατεύθυνση προς την ελαχιστοποίηση της συνάρτησης κόστους.

Ως τελική βελτιστοποίηση για ταχύτερη σύγκλιση της στοχαστικής μεθόδου, έγινε εισαγωγή μιας παραμέτρου της ροπής (momentum) [Qia99]. Με αυτό τον τρόπο προστίθεται η

παράμετρος γ στο προηγούμενο "βήμα" v_{t-1} για τον υπολογισμό του τωρινού βήματος v_t στον κανόνα ανανέωσης. Η παράμετρος αυτή συνήθως ορίζεται στο 0.9.

$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta)$$

$$\theta = \theta - v_t$$

Οπτικά, βλέποντας στην εικόνα της εικόνα του σχήματος 1.27 από πάνω με τη μέθοδο να συγκλίνει στο κέντρο της έλλειψης, στο σχήμα 1.28 φαίνεται η μέθοδος SGD και SGD με την εισαγωγή της ροπής (momentum). Η ταχύτερη σύγκλιση είναι προφανής.



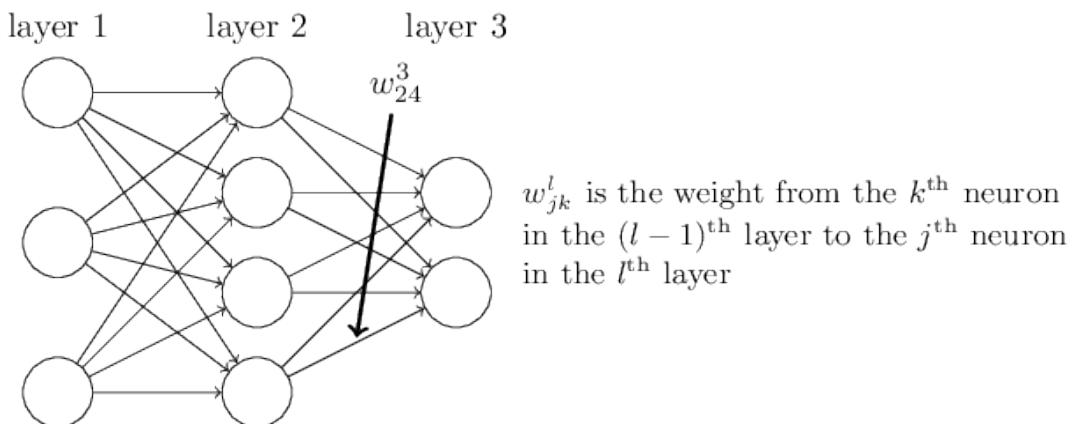
Σχήμα 1.28: SGD (αριστερά) και SGD με momentum (δεξιά)

1.2.7 Μέθοδος οπισθοδιάδοσης (Backpropagation)

Ο αλγόριθμος backpropagation αναπτύχθηκε αρχικά τη δεκαετία του 1970', όμως η σημασία του εκτιμήθηκε μετά το διάστημα paper των David Rumelhart, Geoffrey Hinton και Ronald Williams, το 1986 [RHW88].

Στο κέντρο του αλγορίθμου βρίσκεται μία έκφραση για την μερική παράγωγο $\partial C / \partial w$ της συνάρτησης κόστους C ως προς όλα τα βάρη w (ή bias b) στο δίκτυο. Η έκφραση μας λέει πόσο γρήγορα μεταβάλλεται η συνάρτηση κόστους, όταν μεταβάλλουμε τα βάρη και τα biases.

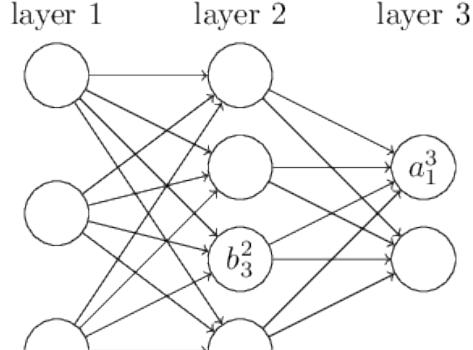
Συμβολισμοί



Σχήμα 1.29

Στο σχήμα 1.29 φαίνονται τα σύμβολα που χρησιμοποιούνται. Συγκεκριμένα, το w_{jk}^l δηλώνει το βάρος της σύνδεσης από τον k^{th} νευρώνα του $(l-1)^{\text{th}}$ επιπέδου στον j^{th} νευρώνα του l^{th} επιπέδου.

Παρόμοια, συμβολίζονται και τα *biases* και οι ενεργοποιήσεις των νευρώνων. Συγκεκριμένα, χρησιμοποιείται το b_j^l για τον j^{th} νευρώνα του l^{th} επιπέδου και α_j^l για την ενεργοποίηση του j^{th} νευρώνα του l^{th} επιπέδου (σχήμα 1.30).



Σχήμα 1.30

Με αυτούς τους συμβολισμούς, η ενεργοποίηση α_j^l του νευρώνα στο l^{th} επίπεδο σχετίζεται με την ενεργοποίηση στο $(l - 1)^{th}$ επίπεδο με την εξίσωση:

$$a_j^l = \sigma \left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l \right), \quad (1.19)$$

όπου το άθροισμα γίνεται για όλους τους νευρώνες k στο $(l - 1)^{th}$ επίπεδο.

Μπορούμε να ξαναγράψουμε αυτή την έκφραση με τη μορφή πινάκων, αν ορίσουμε έναν πίνακα βαρών w^l για κάθε επίπεδο l , όπου τα στοιχεία του πίνακα w^l είναι τα βάρη προς το l^{th} επίπεδο, δηλαδή το στοιχείο στην j^{th} γραμμή και k^{th} στήλη είναι το w_{jk}^l . Ομοίως, για κάθε επίπεδο l ορίζουμε ένα διάνυσμα bias, b^l , με τις τιμές b_j^l , ένα στοιχείο για κάθε νευρώνα στο l^{th} επίπεδο. Τέλος, με τον ίδιο τρόπο ορίζουμε διάνυσμα ενεργοποίησης α^l του οποίου τα στοιχεία είναι οι ενεργοποιήσεις α_j^l .

Εφαρμόζοντας την σιγμοειδή συνάρτηση σ διανυσματικά (με $f(x) = x^2$:

$$f \left(\begin{bmatrix} 2 \\ 3 \end{bmatrix} \right) = \begin{bmatrix} f(2) \\ f(3) \end{bmatrix} = \begin{bmatrix} 4 \\ 9 \end{bmatrix}, \quad (1.20)$$

και με τους παραπάνω συμβολισμούς, η εξίσωση 1.19 ξαναγράφεται:

$$a^l = \sigma(w^l a^{l-1} + b^l). \quad (1.21)$$

Με αυτή την έκφραση φαίνεται αφαιρετικά το πως σχετίζονται οι ενεργοποιήσεις ενός επιπέδου με αυτές του προηγούμενου: αρχικά εφαρμόζεται ένας πίνακας βαρών στις ενεργοποιήσεις, προστίθεται το διάνυσμα με τα biases και τέλος εφαρμόζεται η σιγμοειδής συνάρτηση σ .

Στον υπολογισμό της ενεργοποίησης α^l μέσω τις 1.21, έμμεσα υπολογίζεται η ποσότητα $z^l \equiv w^l a^{l-1} + b^l$, η οποία ονομάζεται σταθμισμένη είσοδος στους νευρώνες του l^{th} επιπέδου. Η εξίσωση 1.21 μπορεί να γραφτεί και ως $a^l = \sigma(z^l)$.

Έστω ότι έχουμε και πάλι ως συνάρτηση κόστους την τετραγωνική συνάρτηση σφάλματος

$$C = \frac{1}{2n} \sum_x \|y(x) - a^L(x)\|^2, \quad (1.22)$$

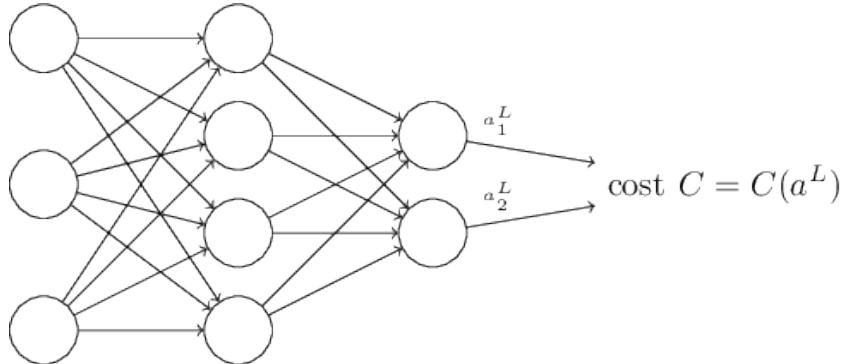
όπου το n είναι ο συνολικός αριθμός των δειγμάτων εκπαίδευσης, το άθροισμα γίνεται στο σύνολο των δεδομένων εκπαίδευσης, το $y = y(x)$ είναι η επιθυμητή έξοδος, το L δηλώνει τον αριθμό των επιπέδων του δικτύου και $a^L = a^L(x)$ είναι το διάνυσμα των ενεργοποιήσεων του επιπέδου εξόδου του δικτύου για είσοδο x .

Για να χρησιμοποιήσουμε τον αλγόριθμο backpropagation υπάρχουν δύο προϋποθέσεις: αρχικά, ότι η συνάρτηση κόστους μπορεί να γραφεί ως ο μέσος όρος $C = \frac{1}{n} \sum_x C_x$ των συναρτήσεων κόστους C_x κάθε δείγματος εκπαίδευσης, x , ξεχωριστά. (Στην τετραγωνική συνάρτηση σφάλματος, $C_x = \frac{1}{2} \|y - a^L\|^2$)

Αυτό απαιτείται, γιατί στην πραγματικότητα ο αλγόριθμος μας επιτρέπει να υπολογίσουμε τις μερικές παραγώγους $\partial C_x / \partial w$ και $\partial C_x / \partial b$ για κάθε δείγμα εκπαίδευσης.

Επιπλέον, προϋπόθεση αποτελεί ότι η συνάρτηση κόστους μπορεί να γραφεί συναρτήσει των εξόδων του δικτύου (σχήμα 1.31). Για την τετραγωνική συνάρτηση σφάλματος:

$$C = \frac{1}{2} \|y - a^L\|^2 = \frac{1}{2} \sum_j (y_j - a_j^L)^2, \quad (1.23)$$



Σχήμα 1.31

Το γινόμενο Hadamard, $s \odot t$

Ο αλγόριθμος backpropagation βασίζεται σε κοινές αλγεβρικές πράξεις - πρόσθεση διανυσμάτων, πολλαπλασιασμός διανύσματος με πίνακα, κλπ. Μία λιγότερο κοινή πράξη είναι αυτή του "κατά στοιχείο" πολλαπλασιαμού, η οποία συμβολίζεται $s \odot t$, όπου τα διανύσματα s και t είναι ίδιων διαστάσεων και είναι γνωστή ως γινόμενο Hadamard.

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix} \odot \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 1 * 3 \\ 2 * 4 \end{bmatrix} = \begin{bmatrix} 3 \\ 8 \end{bmatrix}. \quad (1.24)$$

Για να υπολογίσουν τις μερικές παραγώγους $\partial C / \partial w_{jk}^l$ και $\partial C / \partial b_j^l$, οι Rumelhart et al. χρησιμοποίησαν μία ενδιάμεση ποσότητα δ_j^l , την οποία ονομάζουμε σφάλμα (error) του j^{th}

νευρώνα στο l^{th} επίπεδο. Ο αλγόριθμος backpropagation θα μας δώσει μία διαδικασία για τον υπολογισμό του σφάλματος δ_j^l και τη συσχέτιση του με τις μερικές παραγώγους.

Έστω ότι υπάρχει ο j^{th} νευρώνας στο επίπεδο l και γίνεται μία μεταβολή Δz_j^l στην σταθμισμένη είσοδο του νευρώνα, ώστε αντί να δίνει στην έξοδο $\sigma(z_j^l)$, δίνει $\sigma(z_j^l + \Delta z_j^l)$. Η μεταβολή αυτή διαδίδεται στα επίπεδα του δικτύου, προκαλώντας την μεταβολή του συνολικού κόστους κατά μία ποσότητα $\frac{\partial C}{\partial z_j^l} \Delta z_j^l$.

Το σφάλμα δ_j^l του νευρώνα j στο επίπεδο l ορίζεται ως:

$$\delta_j^l \equiv \frac{\partial C}{\partial z_j^l}. \quad (1.25)$$

Ο συμβολισμός δ^l δηλώνει το διάνυσμα σφαλμάτων για το επίπεδο l και ο αλγόριθμος backpropagation θα μας δώσει έναν τρόπο υπολογισμού του δ^l για κάθε επίπεδο και τη συσχέτιση αυτών των σφαλμάτων με τις πραγματικές ποσότητες που μας ενδιαφέρουν, $\partial C / \partial w_{jk}^l$ και $\partial C / \partial b_j^l$.

Τέσσερις θεμελιώδεις εξισώσεις για τον αλγόριθμο backpropagation

Εξίσωση σφάλματος του επιπέδου εξόδου, δ^L

$$\delta_j^L = \frac{\partial C}{\partial a_j^L} \sigma'(z_j^L). \quad (1.26)$$

Ο πρώτος όρος, $\partial C / \partial a_j^L$ εκφράζει πόσο γρήγορα μεταβάλλεται το κόστος συναρτήσει της j^{th} ενεργοποίησης εξόδου. Άν, για παράδειγμα, το C δεν εξαρτάται πολύ από τον συγκεκριμένο νευρώνα εξόδου, j , τότε το δ_j^L θα είναι μικρό. Ο δεύτερος όρος, $\sigma'(z_j^L)$ εκφράζει πόσο γρήγορα μεταβάλλεται η συνάρτηση ενεργοποίησης σ για z_j^L . Για την τετραγωνική συνάρτηση κόστους: $C = \frac{1}{2} \sum_j (y_j - a_j^L)^2$ και $\partial C / \partial a_j^L = (a_j^L - y_j)$.

Η εξίσωση 1.26 μπορεί να γραφεί με τη μορφή πινάκων:

$$\delta^L = \nabla_a C \odot \sigma'(z^L), \quad (1.27)$$

και για την τετραγωνική συνάρτηση κόστους:

$$\delta^L = (a^L - y) \odot \sigma'(z^L). \quad (1.28)$$

Εξίσωση για το σφάλμα δ^l , συναρτήσει του σφάλματος στο επόμενο επίπεδο δ^{l+1}

$$\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l), \quad (1.29)$$

όπου ο $(w^{l+1})^T$ είναι ο αντίστροφος του πίνακα βαρών w^{l+1} για το $(l+1)^{th}$ επίπεδο. Αν γνωρίζουμε το σφάλμα δ^{l+1} στο $(l+1)^{th}$ επίπεδο, μπορούμε να εφαρμόσουμε τον αντίστροφο πίνακα βαρών $(w^{l+1})^T$ και να ”κινηθούμε προς τα πίσω” στο δίκτυο, λαμβάνοντας μία μέτρηση του σφάλματος στην έξοδο του l^{th} επιπέδου. Τι γινόμενο Hadamard $\odot \sigma'(z^l)$ ”μετακινεί” το

σφάλμα προς τα πίσω, πριν την συνάρτηση ενεργοποίησης του επιπέδου l , επιστρέφοντας το σφάλμα δ^l στην σταθμισμένη είσοδο του επιπέδου l .

Συνδυάζοντας τις εξισώσεις 1.26 και 1.29, μπορούμε να υπολογίσουμε το σφάλμα δ^l για κάθε επίπεδο του δικτύου. Αρχικά χρησιμοποιούμε την 1.26 για τον υπολογισμό του δ^L και εφαρμόζουμε την εξίσωση 1.29 για τον υπολογισμό του δ^{L-1} και ξανά την 1.29 για τον υπολογισμό του δ^{L-2} , μέχρι το πρώτο επίπεδο του δικτύου.

Εξίσωση για το ρυθμό μεταβολής του κόστους ως προς κάθε bias του δικτύου

$$\frac{\partial C}{\partial b_j^l} = \delta_j^l. \quad (1.30)$$

δηλαδή, το σφάλμα δ_j^l είναι ακριβώς ίσο με τον ρυθμό μεταβολής $\partial C / \partial b_j^l$.

Εξίσωση για το ρυθμό μεταβολής του κόστους ως προς κάθε βάρος του δικτύου

$$\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l. \quad (1.31)$$

Ο αλγόριθμος Backpropagation

1. Είσοδος x : Θέσε την αντίστοιχη ενεργοποίηση α^1 για το επίπεδο εισόδου.
2. Πρωστροφοδότηση: Για κάθε $l = 2, 3, \dots, L$, υπολόγισε $z^l = w^l a^{l-1} + b^l$ και $a^l = \sigma(z^l)$.
3. Σφάλμα εξόδου δ^L : Υπολόγισε το διάνυσμα $\delta^L = \nabla_a C \odot \sigma'(z^L)$.
4. Οπισθοδιάδοση του σφάλματος: Για κάθε $l = L-1, L-2, \dots, 2$, υπολόγισε το $\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l)$.
5. Έξοδος: η παράγωγος της συνάρτησης κόστους δίνεται από τις $\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l$ και $\frac{\partial C}{\partial b_j^l} = \delta_j^l$.

Όπως έχει αναφερθεί, ο αλγόριθμος backpropagation υπολογίζει την παράγωγο της συνάρτησης κόστους για κάθε δείγμα εκπαίδευσης, $C = C_x$. Στην πράξη, ο αλγόριθμος backpropagation συνδυάζεται με έναν αλγόριθμο μάθησης, όπως ο gradient descent, με τον οποίο υπολογίζεται η παράγωγος για πολλά δείγματα εισόδου. Για παράδειγμα, δεδομένου ενός mini-batch μεγέθους m , ο παρακάτω αλγόριθμος εφαρμόζει τον gradient descent με mini-batches.

1. Είσοδος ενός συνόλου δειγμάτων εκπαίδευσης.
2. Για κάθε δείγμα εκπαίδευσης x : Θέσε την αντίστοιχη ενεργοποίηση $\alpha^{x,1}$, και εκτέλεσε τα παρακάτω βήματα:
 - Πρωστροφοδότηση: Για κάθε $l = 2, 3, \dots, L$, υπολόγισε τα $z^{x,l} = w^l a^{x,l-1} + b^l$ και $a^{x,l} = \sigma(z^{x,l})$.
 - Σφάλμα εξόδου: $\delta^{x,L}$: Υπολόγισε το διάνυσμα $\delta^{x,L} = \nabla_a C_x \odot \sigma'(z^{x,L})$.

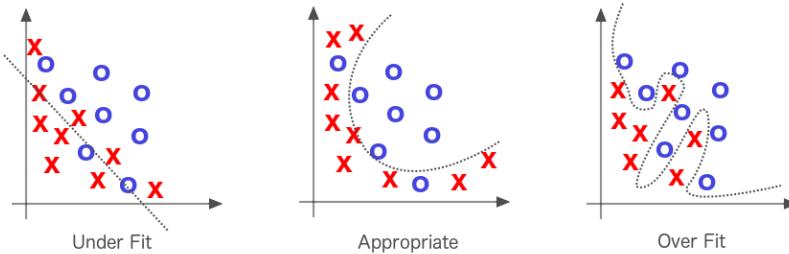
- Οπισθοδιάδοση του σφάλματος: Για κάθε $l = L - 1, L - 2, \dots, 2$, υπολόγισε το $\delta^{x,l} = ((w^{l+1})^T \delta^{x,l+1}) \odot \sigma'(z^{x,l})$.
3. Gradient Descent: Για κάθε $l = L, L - 1, \dots, 2$, ανανέωσε τα βάρη σύμφωνα με τον κανόνα $w^l \rightarrow w^l - \frac{\eta}{m} \sum_x \delta^{x,l} (a^{x,l-1})^T$ και τα biases σύμφωνα με τον κανόνα $b^l \rightarrow b^l - \frac{\eta}{m} \sum_x \delta^{x,l}$.

1.2.8 Υπερ-προσαρμογή (Overfitting)

Η λογική στη χρήση νευρωνικών δικτύων είναι ότι τροφοδοτούμε ένα δίκτυο με δεδομένα εκπαίδευσης, τα οποία για πρόβλημα της ταξινόμησης φέρουν και τις αντίστοιχες κλάσεις στις οποίες ανήκουν και αφού το δίκτυο εκπαιδευτεί και μάθει τα βάρη, το δίκτυο θα πρέπει να είναι ικανό να δεχθεί άγνωστα δεδομένα (τα οποία δεν συμπεριλαμβάνονταν στα δεδομένα εκπαίδευσης) και να εκτελέσει επιτυχώς την ταξινόμηση. Με άλλα λόγια, επιθυμούμε να γενικεύσει με βάσει τα δεδομένα εκπαίδευσης (π.χ. αν το δίκτυο έχει μάθει να ταξινομεί αυτοκίνητα, θα πρέπει να μπορεί να ταξινομήσει την εικόνα ενός αυτοκινήτου την οποία δεν έχει “ξαναδεί”).

Το overfitting συμβαίνει όταν ένα δίκτυο εκπαιδεύεται στα δεδομένα εκπαίδευσης, όμως δε μπορεί να γενικεύσει τη γνώση σε άγνωστα δεδομένα (δηλαδή, μπορεί να μην είναι ικανό να διακρίνει την εικόνα ενός αυτοκινήτου αν έχει διαφορετικό προσανατολισμό).

Στο σχήμα (i) του 1.32 φαίνεται ένα παράδειγμα overfitting. Τα δίκτυο έχει μάθει ένα decision boundary που δεν αφήνει περιθώρια γενίκευσης. Αντίθετα, υπάρχει και ο όρος υπο-προσαρμογή (underfitting) που δηλώνει ότι το δίκτυο δεν έχει μάθει τα κατάλληλα βάρη για να διακρίνει επιτυχώς τα δεδομένα εισόδου.



Σχήμα 1.32: Υπο-προσαρμογή και υπερ-προσαρμογή

1.2.9 Autoencoders και Stacked Autoencoders

Ακόμα από το 1989 έχει αποδειχθεί η εκφραστική δύναμη των δικτύων τριών επιπέδων (είσοδος, κρυμμένο επίπεδο, έξοδος). Η απόδειξη έδειξε ότι κάθε συνεχής συνάρτηση μπορεί να υλοποιηθεί με ένα δίκτυο τριών επιπέδων, δεδομένων επαρκούς αριθμού κρυμμένων νευρώνων και κατάλληλης μη-γραμμικότητας στις συναρτήσεις ενεργοποίησης.

Παρόλα αυτά, εξαιτίας της έλλειψης κατάλληλων αλγορίθμων εκπαίδευσης στα πρώτα χρόνια, δεν ήταν δυνατή η εκμετάλλευση αυτού του μοντέλου, μέχρις ότου οι Hinton και Salakhutdinov [HS06] πρότειναν την ιδέα τους για τη βαθιά μάθηση (deep learning) το 2006. Μέχρι τότε, δεν ήταν ξεκάθαρο πως ήταν δυνατό να εκπαιδευτούν βαθιά νευρωνικά δίκτυα, καθώς οι αλγόριθμοι εκπαίδευσης που ήταν βασισμένοι σε κάποια μορφή του gradient-descent

με τυχαίες αρχικοποιήσεις των βαρών, ”κολλούσαν” σε κακές λύσεις. Οι Hinton et al. πρότειναν έναν μη επιβλεπόμενο αλγόριθμο μάθησης με εκπαίδευση κατά επίπεδο για Deep Belief δίκτυα.

Η βαθιά μάθηση περιλαμβάνει μία κλάση μοντέλων τα οποία προσπαθούν ιεραρχικά να μάθουν χαρακτηριστικά των δεδομένων εισόδου με βαθιά νευρωνικά δίκτυα, τυπικά μεγαλύτερα των τριών επιπέδων. Το δίκτυο, αρχικοποιείται, ή προ-εκπαίδευται κατά επίπεδο (layer-wise) με μη επιβλεπόμενη εκπαίδευση (unsupervised training) και στη συνέχεια συντονίζεται με έναν επιβλεπόμενο (supervised) τρόπο.

Με τον τρόπο αυτό, το δίκτυο μπορεί να μάθει χαρακτηριστικά υψηλού επιπέδου από αυτά των χαμηλών επιπέδων, καθώς επίσης, τελικά μπορούν να εξαχθούν χαρακτηριστικά με στόχο την ταξινόμηση προτύπων. Τα βαθιά μοντέλα μπορούν να οδηγήσουν προοδευτικά σε ολοένα και πιο αφαιρετικά και σύνθετα χαρακτηριστικά στα υψηλά επίπεδα, και αυτά τα χαρακτηριστικά είναι γενικά αμετάβλητα σε τοπικές διακυμάνσεις της εισόδου. Σύμφωνα με παλιότερες εργασίες, τα βαθιά μοντέλα μπορούν να δώσουν καλύτερες προσεγγίσεις μη γραμμικών συναρτήσεων από τα ”ρηχά” μοντέλα.

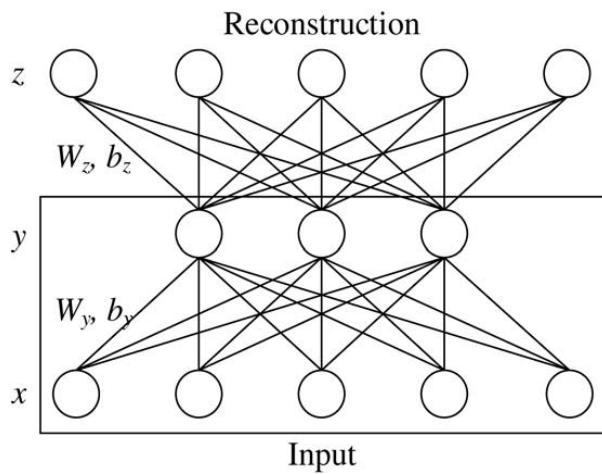
Βασισμένοι στην εργασία των Hinton et al., οι Vincent et al. [Vin+08] εισήγαγαν μία μέθοδο για μη επιβλεπόμενη μάθηση μιας αναπαράστασης. Με την προσέγγιση αυτή, εκπαίδευσαν autoencoders και την αντίστοιχη βαθιά αρχιτεκτονική, stacked autoencoders.

Η περιγραφή του autoencoder του stacked autoencoder γίνεται παρακάτω.

Autoencoders

Ένας autoencoder έχει ένα ορατό επίπεδο d εισόδων, ένα κρυμμένο επίπεδο h νευρώνων, ένα επίπεδο ανακατασκευής d νευρώνων, και μία συνάρτηση ενεργοποίησης f .

Κατά την εκπαίδευσή του, πρώτα αντιστοιχίζει τις εισόδους $x \in \mathbb{R}^d$ στο κρυμμένο επίπεδο και παράγει τις ενεργοποίησεις $y \in \mathbb{R}^h$. Το τμήμα που περιγράφηκε σε αυτό το βήμα, φαίνεται μέσα στο περίγραμμα του σχήματος 1.33 και ονομάζεται κωδικοποιητής (encoder).



Σχήμα 1.33: Autoencoder ενός επιπέδου. Το μοντέλο μαθαίνει μία αναπαράσταση ” y ” της εισόδου ” x ”, ανακατασκευάζοντας την στο ” z ”.

Έπειτα, το y αντιστοιχίζεται με έναν αποκωδικοποιητή (decoder) σε ένα επίπεδο εξόδου, το οποίο έχει το ίδιο μέγεθος με το επίπεδο εισόδου. Αυτή η αντιστοίχιση είναι η ”ανακατασκευή”.

Οι ανακατασκευασμένες τιμές συμβολίζονται ως $z \in \Re^d$. Αυτά τα δύο βήματα, περιγράφονται μαθηματικά:

$$\begin{aligned} y &= f(W_y x + b_y) \\ z &= f(W_z x + b_z) \end{aligned}$$

όπου τα W_y και W_z συμβολίζουν τα βάρη εισόδου-κρυμμένου επιπέδου και κρυμμένου επιπέδου-εξόδου, αντίστοιχα και τα b_y και b_z τα biases του κρυμμένου επιπέδου και του επιπέδου εξόδου, αντίστοιχα. Η $f()$ δηλώνει την συνάρτηση ενεργοποίησης, στην οποία παρέχεται η μη-γραμμικότητα. Στο μοντέλο που θα παρουσιαστεί στο επόμενο κεφάλαιο υπάρχει ο περιορισμός:

$$W_y = W_z = W$$

Με αυτόν τον περιορισμό λέμε πως ο autoencoder έχει "δεμένα βάρη" (tied weights), μειώνοντας τις παραμέτρους του μοντέλου στο μισό. Συνεπώς, υπάρχουν τρεις παράμετροι προς μάθηση: (W, b_y, B_z) .

Σκοπός της εκπαίδευσης είναι η ελαχιστοποίηση του κόστους μεταξύ της εισόδου και της ανακατασκευής, δηλαδή:

$$\operatorname{argmin}_{W, b_y, b_z} [c(x, z)],$$

όπου το z εξαρτάται από τις παραμέτρους W, b_y, b_z , ενώ το x είναι δεδομένο. Το $[c(x, z)]$ δηλώνει το σφάλμα, το οποίο μπορεί να οριστεί με ποικίλους τρόπους. Συνεπώς, ο κανόνας ανανέωσης των βαρών μπορεί να οριστεί ως: (όπου η δηλώνει τον ρυθμό μάθησης):

$$W = W - \eta \frac{\partial \operatorname{cost}(x, z)}{\partial W} \quad (1.32)$$

$$b_y = b_y - \eta \frac{\partial \operatorname{cost}(x, z)}{\partial b_y} \quad (1.33)$$

$$b_z = b_z - \eta \frac{\partial \operatorname{cost}(x, z)}{\partial b_z} \quad (1.34)$$

Μετά το τέλος της εκπαίδευσης του δικτύου, το επίπεδο ανακατασκευής μαζί με τις παραμέτρους του αφαιρούνται και η αναπαράσταση που έχει μάθει το δίκτυο βρίσκεται στο κρυμμένο επίπεδο, το οποίο μπορεί στη συνέχεια να χρησιμοποιηθεί για ταξινόμηση ή ως είσοδος ενός υψηλότερου επιπέδου με σκοπό την εξαγωγή βαθύτερων (πιο αφαιρετικών) χαρακτηριστικών.

Η δύναμη του autoencoder βρίσκεται σε αυτή τη μορφή της εκπαίδευσης μέσω ανακατασκευής. Είναι σημαντικό να σημειωθεί ότι κατά την ανακατασκευή, γίνεται χρήση μόνο της πληροφορίας που βρίσκεται στις ενεργοποιήσεις του κρυμμένου επιπέδου, y , το οποίο αποτελεί την κωδικοποίηση της εισόδου με τη μορφή χαρακτηριστικών. Αν το μοντέλο μπορεί να ανακτήσει πλήρως την αρχική είσοδο από το y , αυτό σημαίνει ότι το y διατηρεί αρκετή από την πληροφορία εισόδου. Και ο μή-γραμμικός μετασχηματισμός που έχει μάθει το δίκτυο, ο οποίος ορίζεται από τα βάρη και τα biases, μπορεί να θεωρηθεί ως ένα καλό βήμα εξαγωγής χαρακτηριστικών. Συνεπώς, "στοιβάζοντας" τους encoders κατ' αυτόν τον τρόπο, και εκπαιδεύοντάς τους όπως έχει περιγραφεί παραπάνω, επιτυγχάνεται η ελαχιστοποίηση του κόστους.

Ταυτόχρονα, διατηρούν μία αφαιρετική και ανεκτική σε μεταβολές (ή και θόρυβο) της εισόδου πληροφορία.

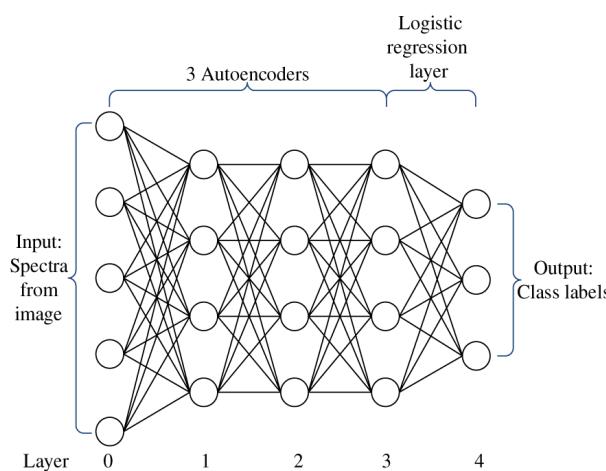
Σαν υποσημείωση, και με σκοπό μία διαισθητική κατανόηση της αρχιτεκτονικής, αξίζει να αναφερθεί ότι οι autoencoders επιτυγχάνουν τη μείωση των διαστάσεων της εισόδου (όταν φυσικά τα κρυμμένα επίπεδα έχουν λιγότερους κόμβους από αυτό της εισόδου), μαθαίνοντας ένα manifold, μία αναπαράσταση δηλαδή σε έναν χώρο λιγότερων διαστάσεων από αυτών της εισόδου. Μπορούμε να το παρομοιάσουμε με τη μαθηματική διαδικασία Principal Component Analysis. Στην πραγματικότητα, αν ο autoencoder είχε μόνο γραμμικές συναρτήσεις ενεργοποίησης θα εκτελούσε PCA.

Stacked Autoencoders

Τοποθετώντας διαδοχικά τα επίπεδα εισόδου και τα κρυμμένα επίπεδα των autoencoders κατασκευάζουμε έναν Stacked Autoencoder. Το μοντέλο χρησιμοποιείται για την εξαγωγή βαθιών χαρακτηριστικών. Το σχήμα 1.34 δείχνει έναν τυπικό stacked autoencoder ακολουθούμενο από έναν logistic regression ταξινομητή.

Ο πρώτος autoencoder αντιστοιχίζει τις εισόδους του επιπέδου "0" στα χαρακτηριστικά του πρώτου επιπέδου. Η εκπαίδευση γίνεται όπως περιγράφηκε προηγουμένως. Μετά την εκπαίδευση του πρώτου autoencoder, τα επόμενα επίπεδα των autoencoders εκπαιδεύονται χρησιμοποιώντας ως είσοδο την έξοδο του προηγούμενου autoencoder. Για παράδειγμα, ενώ εκπαιδεύουμε τον autoencoder μεταξύ του δεύτερου και τρίτου επιπέδου, προσπαθούμε να ανακατασκευάσουμε την έξοδο του δεύτερου επιπέδου, σύμφωνα με τις ενεργοποιήσεις του τρίτου επιπέδου. Μετά την εκπαίδευση αυτή, ο decoder του τρίτου επιπέδου αφαιρείται και διατηρούνται μόνο οι παράμετροι εισόδου-κρυμμένου επιπέδου (W_y, b_y) του autoencoder), ως βάρη μεταξύ του δεύτερου και του τρίτου επιπέδου.

Αν ο ταξινομητής που ακολουθείται είναι και αυτός ένα νευρωνικό δίκτυο, κατά την εκπαίδευσή του, οι παράμετροι καθ' όλο το δίκτυο προσαρμόζονται ελαφρώς, με πολύ μικρότερο ρυθμό μάθησης. Αυτό το βήμα ονομάζεται fine-tuning. Για το επίπεδο logistic regression, η εκπαίδευση είναι απλά η εφαρμογή της μεθόδου οπισθοδιάσης, με μεταβολές στα βάρη καθ' όλο το δίκτυο.



Σχήμα 1.34: Stacked Autoencoder συνδεδεμένος με ένα επίπεδο logistic regression. Έχει πέντε επίπεδα: επίπεδο εισόδου, τρία κρυμμένα επίπεδα και ένα επίπεδο εξόδου.

1.2.10 Συνελικτικά νευρωνικά δίκτυα (Convolutional Neural Networks)

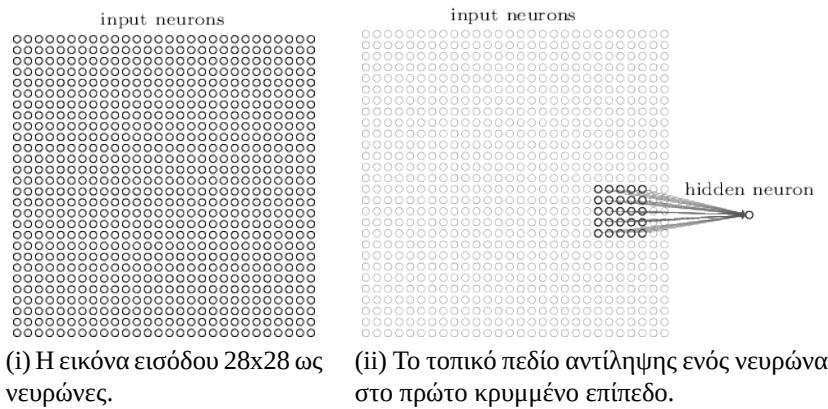
Τα συνελικτικά νευρωνικά δίκτυα [LeC+99] (από δω και πέρα CNNs), μοιάζουν με τα “κλασσικά” νευρωνικά δίκτυα που έχουν περιγραφεί παραπάνω: αποτελούνται από νευρώνες που έχουν βάρη και biases. Κάθε νευρώνας λαμβάνει εισόδους, εκτελεί το εσωτερικό γινόμενο με τα βάρη και εφαρμόζει μία (μη γραμμική) συνάρτηση ενεργοποίησης. Τα CNNs έχουν και αυτά συνάρτηση κόστους και όλα όσα έχουν αναφερθεί παραπάνω.

Τα συνελικτικά νευρωνικά δίκτυα (από δω και πέρα CNNs) χρησιμοποιούν τρεις βασικές αρχές: **τοπικά πεδία αντίληψης (local receptive fields)**, **κοινά βάρη (shared weights)** και **υποδειγματοληψία (pooling)**.

Για την περιγραφή τους, παρακάτω, θεωρούμε ως παράδειγμα μία εικόνα διαστάσεων 28x28 pixels, η οποία δίνεται ως είσοδος στο CNN.

Τοπικά πεδία αντίληψης (local receptive fields)

Σε ένα fully-connected network, όπως παρουσιάστηκε προηγουμένως, η είσοδος φαινόταν ως μία κάθετη στήλη με νευρώνες. Στα CNNs, μπορούμε να φανταστούμε την είσοδο ως ένα 28x28 τετράγωνο με νευρώνες, των οποίων οι τιμές αντιστοιχούν στην ένταση των pixel της εικόνας 28x28 που χρησιμοποιούμε ως είσοδο (i στο σχήμα 1.35).



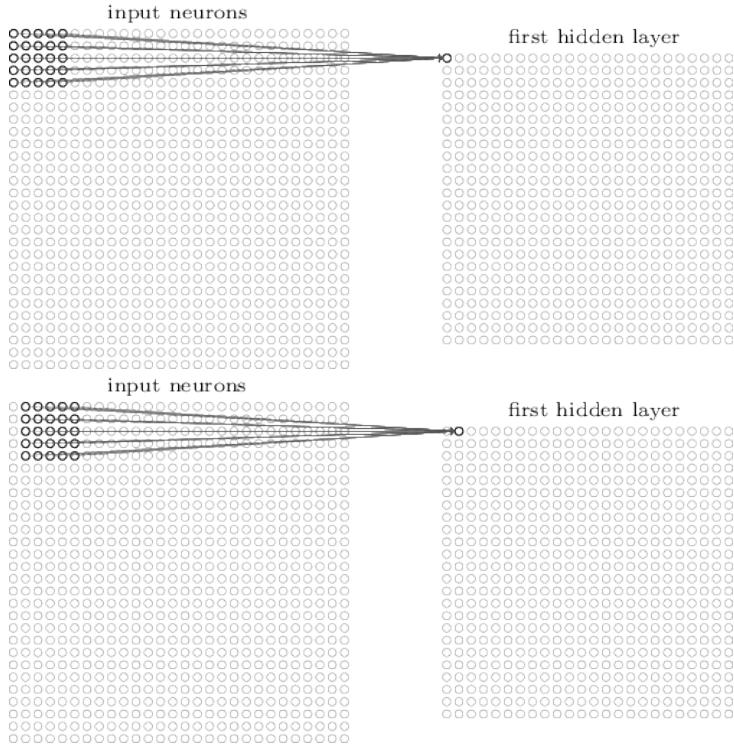
Σχήμα 1.35: Τοπικό πεδίο αντίληψης.

Όπως και στα ”κλασσικά” νευρωνικά δίκτυα, η είσοδος των CNNs συνδέεται με ένα επίπεδο κρυμμένων νευρώνων, με τη διαφορά ότι κάθε pixel εισόδου δε συνδέεται με κάθε νευρώνα στο κρυμμένο επίπεδο. Αντίθετα, οι συνδέσεις γίνονται σε μικρές, τοπικές περιοχές της εικόνας εισόδου.

Συγκεκριμένα, κάθε νευρώνας στο πρώτο κρυμμένο επίπεδο συνδέεται με μία μικρή περιοχή νευρώνων εισόδου, για παράδειγμα με μία περιοχή 5x5, που αντιστοιχίζεται σε 25 pixels εισόδου. Συνεπώς, οι συνδέσεις για έναν νευρώνα στο πρώτο κρυμμένο επίπεδο μπορεί να μοιάζουν με αυτές στη δεξιά εικόνα του 1.35).

Η περιοχή αυτή στην εικόνα εισόδου ονομάζεται **τοπικό πεδίο αντίληψης (local receptive field)** για τον κρυμμένο νευρώνα. Είναι ένα μικρό παράθυρο των pixel εισόδου, το οποίο ”βλέπει” ο νευρώνας και όπου για κάθε σύνδεση προς το παράθυρο ο νευρώνας μαθαίνει ένα βάρος και ένα συνολικό bias.

Το παράθυρο μετακινείται καθ’ολη την εικόνα και για κάθε θέση αυτού του παραθύρου αντιστοιχεί ένας διαφορετικός νευρώνας στο πρώτο κρυμμένο επίπεδο (εικόνα 1.36).



Σχήμα 1.36

Καθώς το τοπικό πεδίο αντίληψης μετακινείται κατά ένα pixel δεξιά (δηλαδή κατά έναν νευρώνα) ”κατασκευάζεται” το πρώτο επίπεδο. Για μία εικόνα 28x28 με τοπικό πεδίο αντίληψης 5x5 θα υπάρχουν 24x24 νευρώνες στο κρυμμένο επίπεδο. Αυτό συμβαίνει γιατί το παράθυρο μπορεί να μετακινηθεί μόνο κατά 24 νευρώνες χωρίς να ”βγει” από την εικόνα. (Στην πραγματικότητα το πεδίο μπορεί να μετακινείται σε διαστήματα μεγαλύτερα του ενός. Αυτό το διάστημα ονομάζεται **stride length** και αποτελεί υπερ-παράμετρο του δικτύου. Επιπλέον, στην περίπτωση που θέλουμε να διατηρήσουμε τις διαστάσεις της εικόνας (x, y), μπορούμε να προσθέσουμε μηδενικά στα άκρα της εισόδου. Η διαδικασία αυτή ονομάζεται **zero-padding** και αποτελεί και αυτή υπερ-παράμετρο του δικτύου. Δηλαδή στην 28x28 εικόνα με την πρόσθεση δύο μηδενικών σε κάθε πλευρά, στο κρυμμένο επίπεδο θα βρίσκονται και πάλι 28x28 νευρώνες.)

Κοινά βάρη (shared weights) και biases

Κάθε κρυμμένος νευρώνας έχει bias και 5x5 βάρη συνδεδεμένα στο τοπικό πεδίο αντίληψή του. Αυτά τα βάρη και το bias είναι κοινά για κάθε έναν από τους 24x24 κρυμμένους νευρώνες. Με άλλα λόγια, για κάθε j, k^{th} κρυμμένο νευρώνα, η έξοδος είναι:

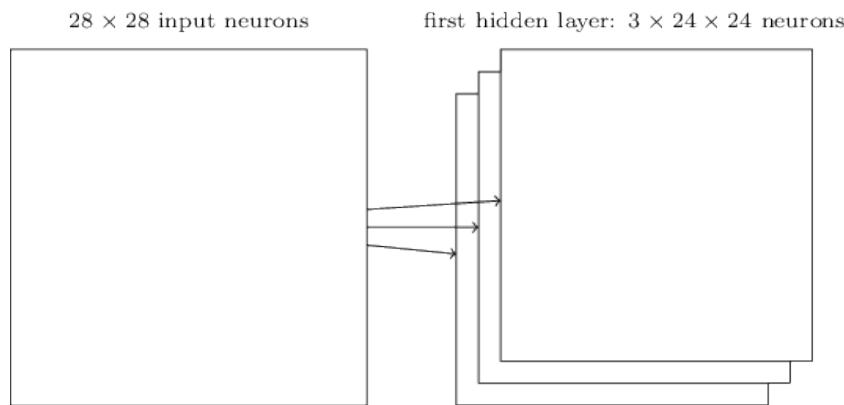
$$\tilde{y} = \sigma \left(b + \sum_{l=0}^4 \sum_{m=0}^4 w_{l,m} a_{j+l, k+m} \right). \quad (1.35)$$

Εδώ, το σ είναι η συνάρτηση ενεργοποίησης, η οποία μπορεί να είναι η σιγμοειδής συνάρτηση, το b είναι μία κοινή τιμή για το bias. Ο $w_{l,m}$ είναι ένας 5x5 πίνακας κοινών βαρών. Και με α συμβολίζεται η ενεργοποίηση της εισόδου στο σημείο (x, y) .

Αυτό σημαίνει ότι όλοι οι νευρώνες στο πρώτο κρυμμένο επίπεδο ανιχνεύουν ακριβώς το ίδιο χαρακτηριστικό (π.χ. κάθετες ακμές, αλλά σε διαφορετικές θέσεις της εικόνας εισόδου).

Για αυτό το λόγο, συνηθίζεται αυτή η αντιστοίχιση του επιπέδου εισόδου στο κρυμμένο επίπεδο να ονομάζεται **χάρτης χαρακτηριστικών (feature map)**. Όλα τα βάρη μέσω των οποίων κατασκευάζεται το feature map ονομάζονται **κοινά βάρη (shared weights)**, αφού τα μοιράζονται όλοι οι νευρώνες του κρυμμένου επιπέδου. Τα κοινά βάρη και το bias ορίζουν έναν **πυρήνα (kernel) ή φίλτρο**.

Στην εικόνα 1.37 βρίσκονται τρία feature maps. Κάθε feature map ορίζεται από ένα σύνολο φίλτρων (εδώ τρία φίλτρα διαστάσεων 5x5). Το αποτέλεσμα είναι το δίκτυο να ανιχνεύει τρία διαφορετικά χαρακτηριστικά, με κάθε ένα χαρακτηριστικό να είναι ανιχνεύσιμο σε ολόκληρη την εικόνα.



Σχήμα 1.37: 3 feature maps

Στην πράξη, τα CNNs χρησιμοποιούν δεκάδες ή και εκατοντάδες φίλτρα σε κάθε επίπεδο όπως το παραπάνω. Στην πραγματικότητα, το επίπεδο του δικτύου που μόλις περιγράφηκε ονομάζεται επίπεδο συνέλιξης (convolutional layer ή conv layer) και αποτελεί τη βάση των CNN όπως υποδηλώνει και το όνομα. Η ονομασία του επιπέδου προκύπτει από την εξίσωση 1.35, στην οποία εμπεριέχεται η πράξη της συνέλιξης.

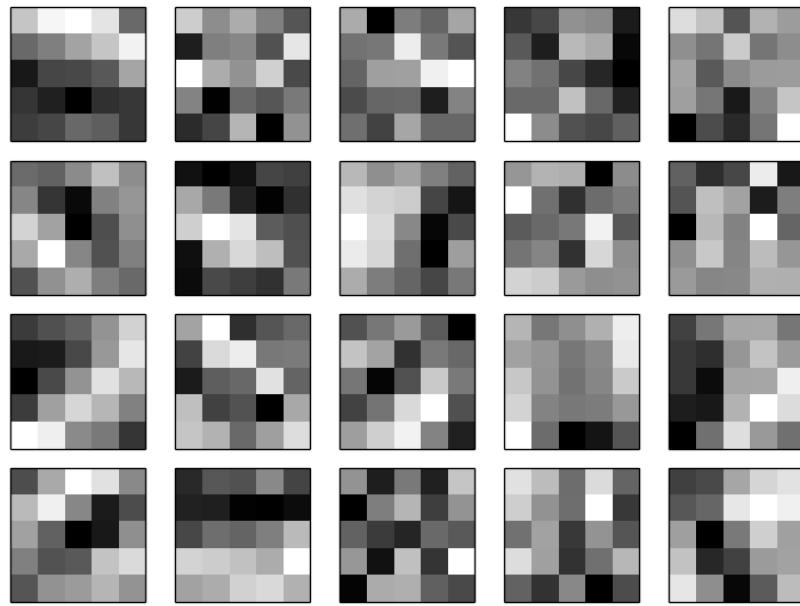
Οι 20 εικόνες στο σχήμα 1.38 αντιστοιχούν σε 20 φίλτρα 5x5. Λευκότερα σημεία δηλώνουν μικρότερα (πιο αρνητικά) βάρη, οπότε τα φίλτρα ανταποκρίνονται λιγότερο στα αντίστοιχα pixels εισόδου. Σκουρότερα σημεία υποδηλώνουν μεγαλύτερα βάρη, οπότε τα φίλτρα ανταποκρίνονται περισσότερο σε αυτά τα pixels εισόδου. Θα μπορούσαμε να πούμε ότι αυτές οι εικόνες δείχνουν το είδος των χαρακτηριστικών στα οποία το επίπεδο συνέλιξης ανταποκρίνεται.

Επίπεδο υποδειγματοληψίας (Pooling layer)

Επιπρόσθετα των επιπέδων συνέλιξης που περιγράφηκαν παραπάνω, τα CNNs περιλαμβάνουν και επίπεδα υποδειγματοληψίας (pooling layers). Τα pooling layers χρησιμοποιούνται συνήθως αμέσως μετά τα επίπεδα συνέλιξης. Αυτό που κάνουν είναι να απλοποιούν την πληροφορία την έξοδο ενός επιπέδου συνέλιξης.

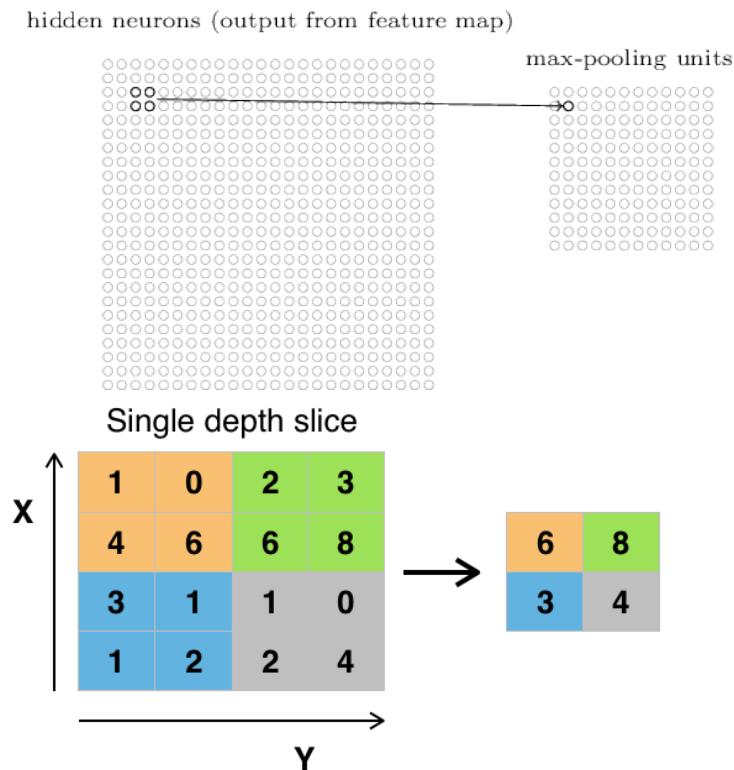
Συγκεκριμένα, ένα pooling layer λαμβάνει ως είσοδο κάθε feature map, ή καλύτερα τις ενεργοποιήσεις των εξόδων των νευρώνων από το conv layer.

Για παράδειγμα, κάθε μονάδα στο pooling layer μπορεί να συνοψίσει μιά περιοχή (ας πούμε) 2x2 νευρώνων του προηγούμενου επιπέδου. Μία σύνθησης τακτική είναι η υποδειγματοληψία



Σχήμα 1.38: 20 5x5 φίλτρα

που είναι γνωστή ως max-pooling. Σε αυτή, μια μονάδα υποδειγματοληψίας απλά δίνει στην έξοδο τη μέγιστη ενεργοποίηση μιας περιοχής 2x2, όπως φαίνεται στο σχήμα 1.39.

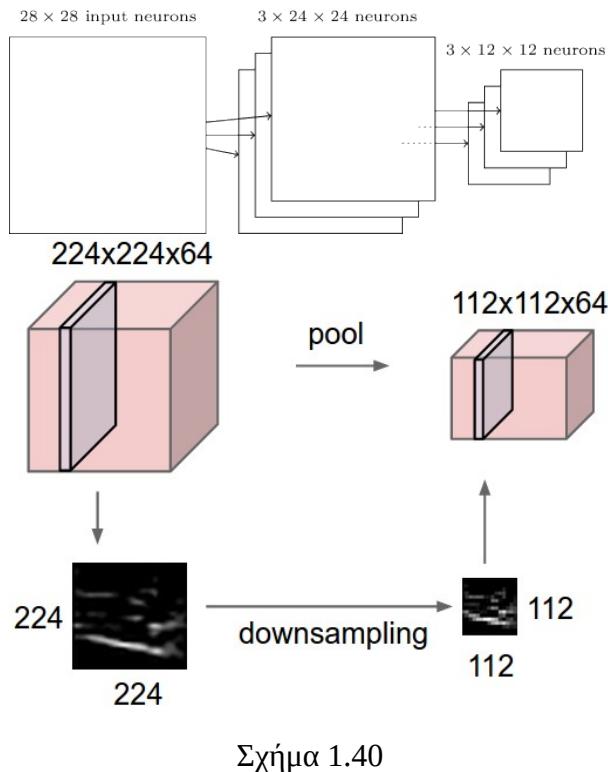


Example of Maxpool with a 2x2 filter and a stride of 2

Σχήμα 1.39

Στο σχήμα 1.39 φαίνονται 24×24 νευρώνες ως έξοδος του conv layer. Μετά το pooling έχουμε 12×12 νευρώνες.

Όπως έχει αναφερθεί, το conv layer περιλαμβάνει περισσότερα από ένα feature maps (τόσα όσα και τα φίλτρα του επιπέδου). Η εφαμοργή του max-pooling γίνεται σε κάθε feature map ξεχωριστά (1.40).



Σχήμα 1.40

1.2.11 Συναρτήσεις ενεργοποίησης

Μέχρι τώρα έχουμε δει την βηματική και την σιγμοειδή συνάρτηση ως συναρτήσεις ενεργοποίησης. Στην εξίσωση 1.35 εμπεριέχεται και η συνάρτηση ενεργοποίησης ως σ . Θα μπορούσαμε να πούμε ότι η συνάρτηση ενεργοποίησης ανήκει σε ένα δικό της επίπεδο στο δίκτυο, ένα επίπεδο ενεργοποίησης αν θέλετε, το οποίο βρίσκεται αμέσως μετά το επίπεδο συνέλιξης.

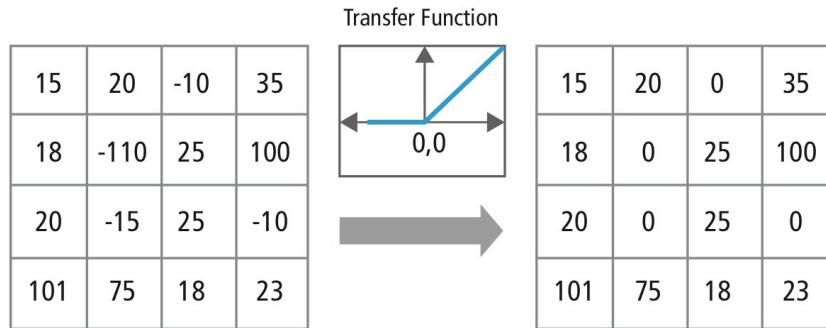
Σήμερα η πιο σύνηθης συναρτησης ενεργοποίησης για τα κρυμμένα επίπεδα είναι η **ReLU (Rectified Linear Units)**, καθώς εμφανίζει πλεονεκτήματα έναντι άλλων μη γραμμικών συναρτήσεων που εφαρμόζονταν παλαιότερα, όπως η σιγμοειδής και η tanh. Αρχικά, φαίνεται πως με τη χρήση της ReLU το δίκτυο εκπαιδεύεται ταχύτερα [GBB11]. Επιπλέον, με χρήση της ReLU αντιμετωπίζεται το **πρόβλημα της εκλειπόμενης παραγώγου (vanishing gradient)**.

Σύμφωνα με το πρόβλημα αυτό, καθώς μέσω του backpropagation υπολογίζονται οι παράγωγοι της συνάρτησης κόστους ως προς τα βάρη του επιπέδου, αυτές γίνονται όλο και μικρότερες καθώς κινούμαστε προς τα πίσω στο δίκτυο. Αυτό έχει ως συνέπεια τα πρώτα επίπεδα να εκπαιδεύονται με πολύ αργούς ρυθμούς σε σύγκριση με τα επίπεδα που βρίσκονται πιο κοντά στην έξοδο του δικτύου. Τα πρώτα επίπεδα είναι εξαιρετικά σημαντικά, διότι είναι αυτά που μαθαίνουν να ανιχνεύουν τα απλά χαρακτηριστικά (κάθετες γραμμές, κλπ) και στα οποία βασίζονται τα επόμενα επίπεδα

Η ReLU δίνει στην έξοδο 0 αν η είσοδος είναι αρνητική. Σε κάθε άλλη περίπτωση δίνει έξοδο ίση με την είσοδο.

$$f(x) = \max(x, 0)$$

Με άλλα λόγια, η συνάρτηση αυτή μετατρέπει τις αρνητικές τιμές που λαμβάνει σε μηδενικά (σχήμα 1.41).



Σχήμα 1.41: Εφαρμογή της ReLU.

Στο επίπεδο εξόδου, όπως έχει περιγραφεί στις "κλασσικές" αρχιτεκτονικές πολλών επιπέδων, φαίνοταν πως γινόταν χρήση της σιγμοειδούς συνάρτησης. Αυτό σημαίνει πως στο παράδειγμα της ταξινόμησης ενός pixel vector στις κλάσεις "δρόμος", "νερό", "δέντρα", "κτήρια" που είχε την πραγματική τιμή $y(x) = (0, 1, 0, 0)^T$, η σιγμοειδής συνάρτηση θα μπορούσε να δώσει στην έξοδο $\alpha(x) = (0.349239449, 0.97321848, 0.129448329, 0.29348294)^T$. Δεν έχει αναφερθεί έως τώρα, αλλά θα ήταν επιθυμητό η έξοδος του δικτύου να είναι μία κατανομή πιθανότητας με άθροισμα τη μονάδα. Με χρήση της σιγμοειδούς συνάρτησης αυτό μπορεί να συμβεί μόνο στην περίπτωση της δυαδικής ταξινόμησης, όταν δηλαδή η διάκριση γίνεται ανάμεσα σε δύο μόνο κλάσεις. Στο παράδειγμα με τις τέσσερις κλάσεις, μπορούμε να φανταστούμε κάθε νευρώνα εξόδου ως έναν δυαδικό ταξινομητή (π.χ. "δρόμος" και "όχι δρόμος"), για αυτό και το άθροισμα των τιμών της εξόδου είναι μεγαλύτερο της μονάδας.

Το πρόβλημα αυτό αντιμετωπίζεται με τη χρήση μιας διαφορετικής συνάρτησης ενεργοποίησης στο επίπεδο εξόδου (ή αν θέλετε, ένα επίπεδο ενεργοποίησης αμέσως μετά το επίπεδο εξόδου). Η συνάρτηση που χρησιμοποιείται για αυτό τον σκοπό ονομάζεται **softmax**:

$$\text{softmax}_i = \frac{\exp(a_i)}{\sum_j \exp(a_j)}$$

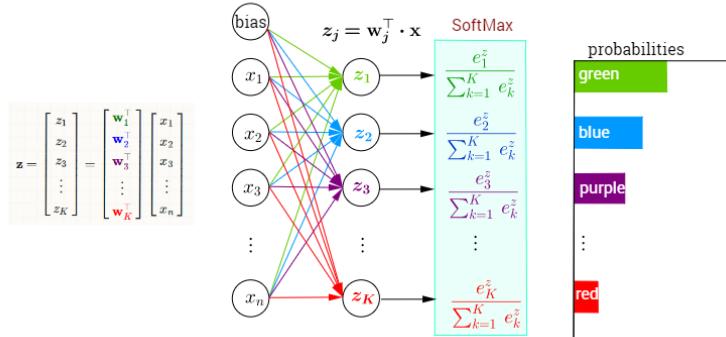
1.2.12 Συνάρτησεις κόστους

Η συνάρτηση κόστους που έχει εμφανισθεί μέχρι τώρα είναι η συνάρτηση τετραγωνικού σφάλματος (MSE).

Στην πραγματικότητα, όπως και στο μοντέλο που μελετήθηκε στα πλαίσια αυτής της εργασίας, η συνήθης συνάρτηση κόστους που χρησιμοποιείται είναι η cross-entropy loss function. Ο κύριος λόγος που η συνάρτηση MSE εγκαταλήφθηκε είναι ότι εξ' ορισμού, η MSE μπορεί να προκαλέσει αργούς ρυθμούς μάθησης του μοντέλου. Αυτό ο πρόβλημα αντιμετωπίζεται με την εισαγωγή της συνάρτησης **cross-entropy**.

$$\begin{bmatrix} 1.2 \\ 0.9 \\ 0.4 \end{bmatrix} \xrightarrow{\text{Softmax}} \begin{bmatrix} 0.46 \\ 0.34 \\ 0.20 \end{bmatrix}$$

(i) Εφαρμογή της softmax



(ii) Εφαρμογή της softmax στο επίπεδο εξόδου.

Μπορούμε να θεωρήσουμε ότι ένας νευρώνας μαθαίνει αλλάζοντας τα βάρη και το bias με έναν ρυθμό που καθορίζεται από τις μερικές παραγώγους της συνάρτησης κόστους, $\partial C / \partial w$ και $\partial C / \partial b$. Αργός ρυθμός μάθησης ουσιαστικά σημαίνει ότι αυτές οι μερικές παράγωγοι είναι μικρές.

Η συνάρτηση κόστους:

$$C = \frac{(y - a)^2}{2}, \quad (1.36)$$

όπου a είναι η έξοδος ενός νευρώνα για είσοδο $x = 1$ και $y = 0$ η επιθυμητή τιμή.

Γνωρίζοντας ότι $a = \sigma(z)$ και $z = wx + b$ και εφαρμόζοντας τον κανόνα της αλυσίδας, μπορούμε να παραγωγίσουμε τη συνάρτηση κόστους ως προς τα βάρη και το bias:

$$\frac{\partial C}{\partial w} = (a - y)\sigma'(z)x = a\sigma'(z)$$

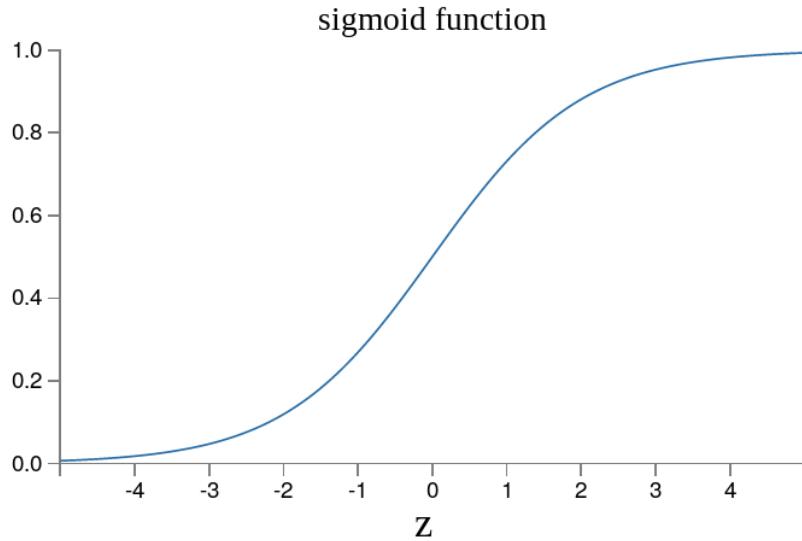
$$\frac{\partial C}{\partial b} = (a - y)\sigma'(z) = a\sigma'(z),$$

όπου έχει αντικατασταθεί $x = 1$ και $y = 0$. Από την καμπύλη της σιγμοειδούς συνάρτησης (σχήμα 1.42), φαίνεται πως όταν η έξοδος του νευρώνα είναι κοντά στο 1, η καμπύλη τείνει να γίνει επίπεδη με αποτέλεσμα η $\sigma'(z)$ να παίρνει πολύ μικρές τιμές. Σε αυτή την περίπτωση, δηλαδή όταν ένας νευρώνας δίνει μία εξαιρετικά εσφαλμένη τιμή (στην περίπτωσή μας δίνει $a = 1$ ενώ η επιθυμητή τιμή είναι $y = 0$), από τις παραπάνω εξισώσεις προκύπτει ότι οι $\partial C / \partial w$ και $\partial C / \partial b$ παίρνουν πολύ μικρές τιμές. Από εδώ προέρχεται το πρόβλημα της αργής μάθησης.

Το πρόβλημα αυτό αντιμετωπίζεται με την αντικατάσταση της συνάρτησης MSE με την cross-entropy, η οποία ορίζεται:

$$C = -\frac{1}{n} \sum_x [y \ln a + (1 - y) \ln(1 - a)], \quad (1.37)$$

όπου n είναι ο συνολικός αριθμός των δειγμάτων των δεδομένων εκπαίδευσης, και το άθροισμα γίνεται καθ' όλα τα δείγματα, x . y είναι η αντίστοιχη επιθυμητή τιμή.



Σχήμα 1.42: Καμπύλη σιγμοειδούς συνάρτησης.

Δύο ιδιότητες της cross-entropy την καθιστούν κατάλληλη για συνάρτηση κόστους: Πρώτον, $C > 0$ και δεύτερον, αν η έξοδος ενός νευρώνα προσεγγίζει την επιθυμητή έξοδο για όλα τα δεδομένα εισόδου, x , η cross-entropy θα τείνει στο μηδέν. Αν υποθέσουμε ότι $y = 0$ και $a \approx 0$ για μία είσοδο x , ο νευρώνας δίνει μία καλή έξοδο και ο πρώτος όρος στην έκφραση 1.37 απαλούφεται αφού $y = 0$, ενώ ο δεύτερος όρος είναι $-\ln(1 - a) \approx 0$. Ομοίως, μπορεί να δειχθεί και στην περίπτωση που $y = 1$ και $a \approx 0$.

Η μερική παράγωγος της cross-entropy ως προς τα βάρη, αντικαθιστώντας όπου $a = \sigma(z)$ και εφαρμόζοντας τον κανόνα της αλυσίδας είναι:

$$\frac{\partial C}{\partial w_j} = -\frac{1}{n} \sum_x \left(\frac{y}{\sigma(z)} - \frac{(1-y)}{1-\sigma(z)} \right) \frac{\partial \sigma}{\partial w_j} \quad (1.38)$$

$$= -\frac{1}{n} \sum_x \left(\frac{y}{\sigma(z)} - \frac{(1-y)}{1-\sigma(z)} \right) \sigma'(z) x_j. \quad (1.39)$$

η οποία μπορεί να γραφεί ως:

$$\frac{\partial C}{\partial w_j} = \frac{1}{n} \sum_x \frac{\sigma'(z) x_j}{\sigma(z)(1-\sigma(z))} (\sigma(z) - y). \quad (1.40)$$

Κάνοντας χρήση της σιγμοειδούς συνάρτησης $\sigma(z) = 1/(1 + e^{-z})$ μπορεί να δειχθεί ότι $\sigma'(z) = \sigma(z)(1 - \sigma(z))$, απλοποιώντας τη μερική παράγωγο:

$$\frac{\partial C}{\partial w_j} = \frac{1}{n} \sum_x x_j (\sigma(z) - y). \quad (1.41)$$

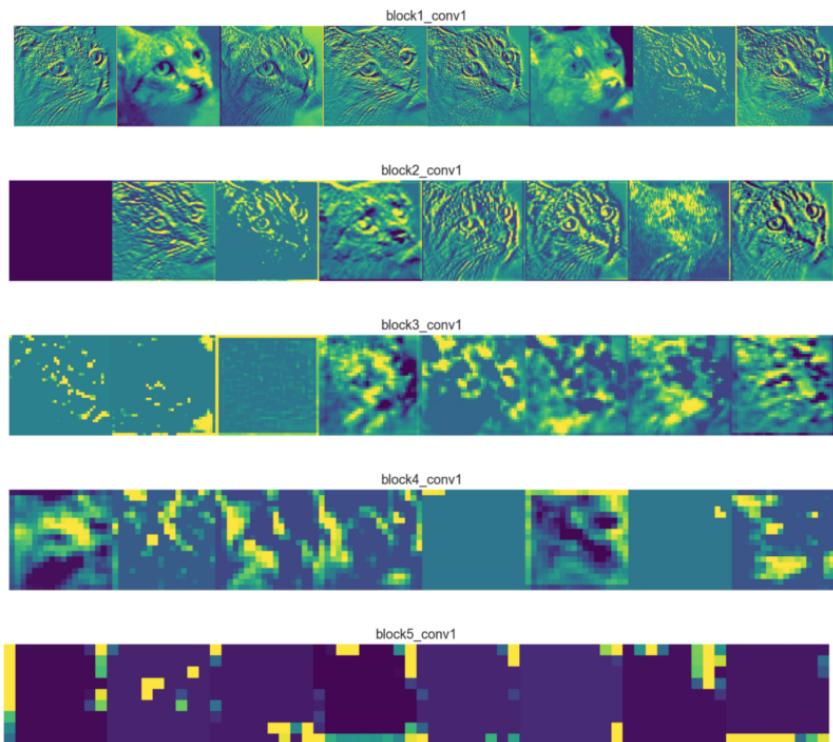
Η έκφραση αυτή δηλώνει ότι ο ρυθμός με τον οποίο τα γίνεται η μάθηση, ελέγχετε από το $\sigma(z) - y$, δηλαδή από το σφάλμα της εξόδου. Όσο μεγαλύτερο το σφάλμα, τόσο ταχύτερα μαθαίνει ο νευρώνας.

Η χρήση της cross-entropy ως συνάρτηση κόστους, οδηγεί στην απαλοιφή του όρου $\sigma'(z)$, οπότε μπορούμε να αγνοήσουμε το γεγονός ότι μπορεί να πάρει μικρές τιμές.

Ομοίως, υπολογίζεται η μερική παράγωγος του κόστους ως προς το bias.

$$\frac{\partial C}{\partial b} = \frac{1}{n} \sum_x (\sigma(z) - y). \quad (1.42)$$

1.2.13 Τι μαθαίνει ένα CNN



Σχήμα 1.43

Στην εικόνα 1.43 έχουν επιλεγεί 8 feature maps από κάθε ένα από τα 5 επίπεδα συνέλιξης ενός CNN. Μπορούν να γίνουν κάποιες αξιοσημείωτες παρατηρήσεις σχετικά με τα feature maps, προχωρώντας βαθύτερα στα επίπεδα του δικτύου.

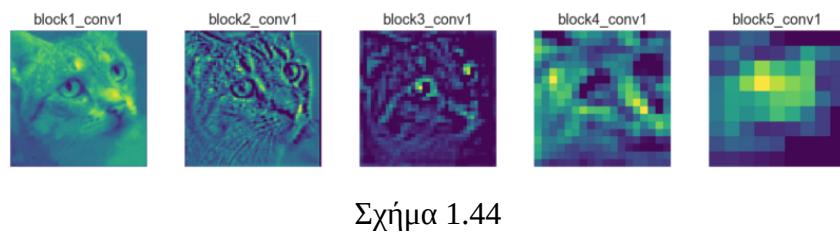
Οι φωτεινές περιοχές είναι οι "ενεργοποιημένες" περιοχές, δηλαδή οι περιοχές στις οποίες το φίλτρο έχει ανιχνεύσει το πρότυπο/χαρακτηριστικό που έψαχνε.

Στο σχήμα 1.44 έχει επιλεγεί ένα feature map για κάθε επίπεδο. Το πρώτο feature map (block1_conv1) διατηρεί το μεγαλύτερο κομμάτι της πληροφορίας που βρίσκεται στην εικόνα. Στις αρχιτεκτονικές των CNN τα πρώτα επίπεδα συνήθως λειτουργούν ως ανιχνευτές ακμών (edge detectors).

Προχωρώντας βαθύτερα στο δίκτυο, τα feature maps μοιάζουν όλο και λιγότερο με την αρχική εικόνα και περισσότερο με μία αφαιρετική αναπαράσταση αυτής. Συγκεκριμένα, στο τρίτο επίπεδο συνέλιξης, φαίνεται πως το χαρακτηριστικό που ανιχνεύεται στην εικόνα είναι τα μάτια της γάτας. Στο επίπεδο αυτό μπορούμε να αναγνωρίσουμε ότι πρόκειται για μία γάτα, όμως από το επόμενο επίπεδο και έπειτα δε μπορούμε να αναγνωρίσουμε την αρχική εικόνα

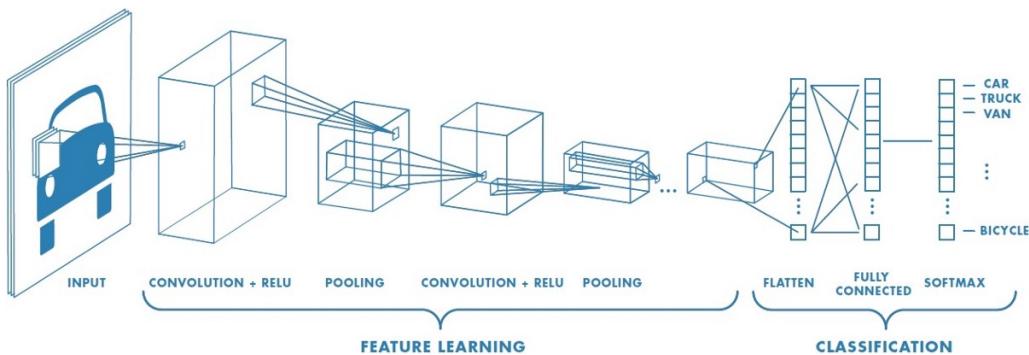
στο feature map. Ο λόγος είναι ότι βαθύτερα feature maps κωδικοποιούν έννοιες υψηλού επιπέδου όπως μύτη γάτας” ή ”αυτί σκύλου”, ενώ feature maps χαμηλότερου επιπέδου ανιχνεύουν απλές ακμές και σχήματα. Για αυτό βαθύτερα feature maps περιλαμβάνουν όλο και λιγότερη πληροφορία σχετικά με την εικόνα και περισσότερη σχετικά με την κλάση της εικόνας. Αν και εξακολουθούν να κωδικοποιούν χρήσιμα χαρακτηριστικά, αυτά δεν είναι οπτικά κατανοητά από εμάς.

Τα feature maps γίνονται όλο και πιο αραιά όσο πιο βαθιά βρίσκονται στο δίκτυο, δηλαδή τα φίλτρα αναγνωρίζουν λιγότερα χαρακτηριστικά. Αυτό συμβαίνει επειδή τα φίλτρα στα πρώτα επίπεδα ανιχνεύουν γραμμές και σχήματα, τα οποία περιλαμβάνονται σε κάθε εικόνα, αλλά όσο πιο βαθιά πηγαίνουμε στο δίκτυο τα χαρακτηριστικά είναι πιο σύνθετα όπως ”ουρά σκύλου”, τα οποία δε βρίσκονται σε κάθε εικόνα. Για αυτό στην εικόνα 1.43 με τα 8 feature maps για κάθε επίπεδο, βλέπουμε πως υπάρχουν κενά feature maps βαθύτερα στο δίκτυο (block4_conv1 και block5_conv1).



1.2.14 Μια ολοκληρωμένη αρχιτεκτονική CNN

Συνοψίζοντας, τα CNNs αποτελούνται από επίπεδα συνέλιξης, υποδειγματοληψίας και fully-connected επίπεδα. Συνηθίζεται η συνάρτηση ενεργοποίησης των επιπέδων συνέλιξης να είναι η ReLU και η συνάρτηση του επιπέδου εξόδου να είναι η softmax.



Σχήμα 1.45: Αρχιτεκτονική CNN για την ταξινόμηση εικόνων σε 1000 κλάσεις.

Ένα δίκτυο CNN φαίνεται στην εικόνα 1.45. Η είσοδος είναι μία RGB εικόνα (3 κανάλια), και το δίκτυο αποτελείται από 2 επίπεδα συνέλιξης ακολουθούμενα από επίπεδα δειγματοληψίας. Τα επίπεδα συνέλιξης έχουν τον ρόλο να μάθουν χαρακτηριστικά των εικόνων και το τελευταίο επίπεδο οδηγείται ως είσοδος σε έναν ταξινομητή που έχει ένα softmax επίπεδο εξόδου.

Κεφάλαιο 2

Τα μοντέλα που μελετήθηκαν: Stacked Autoencoders-Logistic Regression και Diverse Region-Based CNN

Στα πλαίσια της διπλωματικής εργασίας, μελετήθηκαν τα μοντέλα δύο ομάδων ερευνητών: των Yushi Chen et al. [Che+14], οι οποίοι πρώτοι πρότειναν την εφαρμογή βαθιών νευρωνικών δικτύων για την ταξινόμηση υπερφασματικών εικόνων, με τη χρήση stacked autoencoders και των Mengmeng Zhang et al. [ZLD18], οι οποίοι πρότειναν μία καινοτόμα αρχιτεκτονική την οποία ονόμασαν Diverse Region CNN.

Στο κεφάλαιο αυτό γίνεται εκτενής ανάλυση των παραπάνω μοντέλων.

2.1 Stacked Autoencoder - Logistic Regression

Οι Chen et al. στην εργασία τους, αρχικά επιβεβαίωσαν την εγκυρότητα χρήσης του stacked autoencoder, χρησιμοποιώντας το μοντέλο για την ταξινόμηση υπερφασματικών εικόνων με χρήση μόνο της φασματικής πληροφορίας.

Στη συνέχεια, πρότειναν έναν τρόπο ταξινόμησης με χρήση κυρίως της χωρικής πληροφορίας των εικόνων.

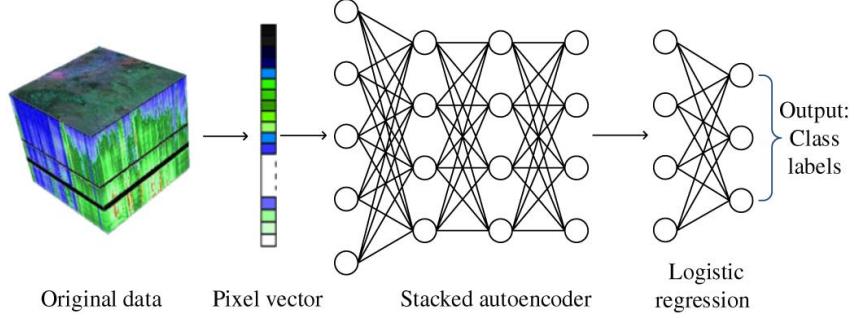
Στο τέλος, πρότειναν μία καινοτόμο μέθοδο, στην οποία συνδύασαν την χωρική και την φασματική πληροφορία της εικόνας με σκοπό την αύξηση της ακρίβειας στην ταξινόμηση.

2.1.1 Ταξινόμηση με έμφαση στα φασματικά χαρακτηριστικά

Στην πρώτη απόπειρα, στόχος ήταν η μάθηση βαθιών φασματικών χαρακτηριστικών, προεκπαιδεύοντας έναν stacked autoencoder με έναν ιεραρχικό τρόπο, όπως αυτός περιγράφηκε στο προηγούμενο κεφάλαιο. Στο παρόν κεφάλαιο, γίνεται μία εκτενέστερη ανάλυση της αρχιτεκτονικής.

Με σκοπό την απόκτηση μιας μη γραμμικής αντιστοίχισης της εισόδου στα ενδιάμεσα κρυμμένα επίπεδα του μοντέλου, ορίζεται η σιγμοειδής συνάρτησης ενεργοποίησης, τόσο στον encoder όσο και στον decoder.

Σε όλα τα στάδια της εργασίας τους, χρησιμοποίησαν tied weights. Αν και υπάρχουν αρκετές επιλογές όσων αφορά τη συνάρτηση κόστους, για τον υπολογισμό του σφάλματος μεταξύ της



Σχήμα 2.1

εισόδου και της ανακατασκευής στην έξοδο, στο συγκεκριμένο μοντέλο η επιλογή έγινε βάσει της συνάρτησης ενεργοποίησης.

Πιο συγκεκριμένα, δεδομένου ότι χρησιμοποιείται η σιγμοειδής συνάρτηση f , καθώς η έξοδος του νευρώνα τείνει στο 0 ή στο 1, η παράγωγός της σιγμοειδούς, f' , τείνει ασυμπτωτικά στο μηδέν. Με τη χρήση μιας συνηθισμένης συνάρτησης κόστους, όπως η συνάρτηση μέσου τετραγωνικού σφάλματος, η κλίση του κόστους θα εμφανίσει το ίδιο πρόβλημα, κάτι το οποίο οδηγεί σε ανεπίτρεπτα αργούς ρυθμούς εκπαίδευσης. Παρόλα αυτά, έχει αποδειχτεί από τους παρακάτω υπολογισμούς ότι η συνάρτηση κόστους cross entropy επιτρέπει τη μεταβολή των βαρών ακόμα και όταν οι κόμβοι τείνουν στον κορεσμός (δηλαδή οι έξοδοι τους είναι κοντά στο 0 ή 1). Καθ' όλη την εργασία τους, γίνεται χρήση της cross entropy σε συνδυασμό με την σιγμοειδή συνάρτηση ενεργοποίησης.

Στην παρούσα υλοποίηση, ο υπολογισμός του κόστους γίνεται με χρήση mini-batch των εισόδων:

$$c = -\frac{1}{m} \sum_{i=1}^m \sum_{k=1}^d [x_{ik} \log z_{ik} + (1 - x_{ik}) \log (1 - z_{ik})] \quad (2.1)$$

όπου το d δηλώνει το μέγεθος του διανύσματος εισόδου και το m δηλώνει το μέγεθος του mini-batch. Το $x_{ik}(z_{ik})$ δηλώνει το k^{th} στοιχείο της i^{th} εισόδου(ανακατασκευής) στο mini-batch. Το εσωτερικό άθροισμα γίνεται καθ' όλες τις διαστάσεις της εισόδου, ενώ το εξωτερικό άθροισμα γίνεται για όλο το mini-batch.

Ο υπολογισμός του κόστους 2.1 γίνεται τη χρήση της στοχαστικής μεθόδου με βάση την κλίση με mini-batch. Χάριν εμβάθυνσης, παρακάτω αναγράφονται οι διαφορικές εξισώσεις που ορίζουν τον κανόνα ανανέωσης των βαρών.

Αρχικά, η ανακατασκευή σε βαθμωτή μορφή γράφεται:

$$net_{ip}^y = \sum_{q=1}^d x_{iq} W_{qp} + b_{yp} \quad (2.2)$$

$$net_{ik}^z = \sum_{p=1}^h W_{kp} f(net_{ip}^y) + b_{zk} \quad (2.3)$$

$$z_{ik} = f(net_{ik}^z) = f(\sum_{p=1}^h x_{ip} W_{zp} + b_{zp}) + b_{zk} \quad (2.4)$$

όπου το $net_{ip}^y(net_{ik}^z)$ υποδηλώνει την είσοδο στον p^{th} κρυμμένο νευρώνα, δεδομένου του i^{th} δείγματος του mini-batch.

Οι πρώτη και δεύτερη παράγωγοι της σιγμοειδούς συνάρτησης $\frac{1}{1+\exp-x}$ είναι:

$$f'(x) = f(x)[1 - f(x)] \quad (2.5)$$

$$f''(x) = f(x)[1 - f(x)][1 - 2f(x)] \quad (2.6)$$

Γνωρίζονται τις 2.2 - 2.4 μπορούν να υπολογιστούν οι μερικές παράγωγοι της ανακατασκευής z ως προς τις παραμέτρους W , b_y και b_z .

$$\frac{\partial z_{ik}}{\partial W_{rs}} = \begin{cases} f'(net_{ik}^z)[W_{ks}f'(net_{is}^s)x_{ir} + f(net_{is}^y)] & k = r \\ f'(net_{ik}^z)[W_{ks}f'(net_{is}^s)x_{ir}] & k \neq r \end{cases} \quad (2.7)$$

$$\frac{\partial z_{ik}}{\partial b_{yr}} = f'(net_{ik}^z)W_{kr}f'(net_{ir}^y) \quad (2.8)$$

$$\frac{\partial z_{ik}}{\partial b_{zr}} = f'(net_{ik}^z) \quad (2.9)$$

όπου W_{rs} δηλώνει το βάρος που συνδέει την r^{th} είσοδο και τον s^{th} κρυμμένο νευρών. Το $b_{yr}(b_{zr})$ δηλώνει το bias για τον r^{th} νευρώνα στο κρυμμένο επίπεδο (ανακατασκευής).

Βασιζόμενοι στις 2.2 - 2.9 οδηγούμαστε στις μερικές διαφορικές εξισώσεις του κόστους 2.1, ως προς τις παραμέτρους W , b_y και b_z .

$$\frac{\partial c}{\partial W_{rs}} = -\frac{1}{m} \sum_{k=1}^m \left\{ \sum_{i=1}^d \left[\frac{x_{ik} - z_{ik}}{z_{ik}(1 - z_{ik})} f'(net_{ik}^z) W_{ks} f'(net_{is}^y) x_{ir} \right] + f'(net_{ik}^z) f(net_{is}^y) \right\} \quad (2.10)$$

$$\frac{\partial c}{\partial b_{yr}} = -\frac{1}{m} \sum_{k=1}^m \sum_{i=1}^d \frac{x_{ik} - z_{ik}}{z_{ik}(1 - z_{ik})} f'(net_{ik}^z) W_{kr} f'(net_{ir}^y) \quad (2.11)$$

$$\frac{\partial c}{\partial b_{zr}} = \frac{1}{m} \sum_{k=1}^m \sum_{i=1}^d \frac{x_{ik} - z_{ik}}{z_{ik}(1 - z_{ik})} f'(net_{ik}^z) \quad (2.12)$$

Αντικαθιστώντας τις παραπάνω εξισώσεις στις 1.32 - 1.34, που εμφανίστηκαν στην εισαγωγή της αρχιτεκτονικής στο προηγούμενο κεφάλαιο, ορίζουμε τους κανόνες ανανέωσης των βαρών.

Μετά την εκπαίδευση του δικτύου, το επίπεδο ανακατασκευής απομακρύνεται και οι ενεργοποιήσεις του κρυμμένου θεωρούνται ορίζονται πλέον τα χαρακτηριστικά τα οποία έμαθε το μοντέλο. Επόμενα επίπεδα εκπαιδεύονται με τον ίδιο τρόπο, αλλά οι είσοδοι τους είναι οι έξοδοι

των προηγούμενων επιπέδων. Επομένως, οι stacked autoencoders κατασκευάζονται με διαδοχικά επίπεδα encoders.

Για την χρήσιμη εφαρμογή του δικτύου με τα χαρακτηριστικά που έχουν αποκτηθεί από την παραπάνω εκπαίδευση στη διαδικασία της ταξινόμησης, απαιτείται το λεγόμενο fine-tuning, δηλαδή μία επιπρόσθετη εκπαίδευση για τη μεταβολή των βαρών, καθ' όλο το προ-εκπαιδευμένο δίκτυο, μαζί με έναν ταξινομητή λογιστικής παλινδρόμησης (logistic regression classifier). Ουσιαστικά, το επίπεδο εισόδου του ταξινομητή, δέχεται ως είσοδο την έξοδο του προεκπαιδευμένου stacked autoencoder, δηλαδή τα χαρακτηριστικά που έχει εξάγει το δίκτυο. Η επιπρόσθετη εκπαίδευση που πραγματοποιείται, υλοποιεί τον αλγόριθμο οπισθοδιάδοσης με υπολογισμό του σφάλματος, όχι στο επίπεδο εξόδου του stacked autoencoder αλλά σε αυτό του ταξινομητή και το σφάλμα διαδίδεται μέχρι την είσοδο του δικτύου. Ο ταξινομητής χρησιμοποιεί την συνάρτηση softmax ως συνάρτηση ενεργοποίησης στο επίπεδο εξόδου, για να εξασφαλισθεί ότι το άθροισμα όλων των νευρώνων εξόδου είναι μονάδα, ώστε η έξοδος να μπορεί να θεωρηθεί μία κατανομή πιθανότητας.

Για παράδειγμα, δεδομένου ενός διανύσματος εισόδου R , το οποίο αποτελεί την έξοδο ενός προηγούμενου επιπέδου autoencoder, η πιθανότητα η είσοδος να ανήκει σε μία κλάση i είναι:

$$P(Y = i|R, W, b) = s(WR + b) = \frac{\exp^{W_i R + b_i}}{\sum_j \exp^{W_j R + b_j}} \quad (2.13)$$

όπου W και b είναι τα βάρη και τα biases του επιπέδου logistic regression και το άθροισμα γίνεται καθ' όλους τους νευρώνες εξόδου.

Το μέγεθος του επιπέδου εξόδου είναι ίσο με τον συνολικό αριθμό των κλάσεων.

Εδώ, πρέπει να τονιστούν τρία σημεία: Αρχικά, αφού ο stacked autoencoder έχει εξάγει τα φασματικά χαρακτηριστικά της εισόδου, σε έναν χώρο διαστάσεων ίσο με τον αριθμό των νευρώνων στην έξοδό του, τα χαρακτηριστικά αυτά είναι δυνατό να οδηγηθούν ως είσοδος σε έναν ταξινομητή που δεν υλοποιείται ως ένα νευρωνικό δίκτυο. Επιπλέον, ο λόγος για τον οποίο είναι δυνατή η συνέχιση της εκπαίδευσης του δικτύου μαζί με την εκπαίδευση του ταξινομητή, είναι ότι ο ταξινομητής υλοποιείται ακριβώς ως ένα νευρωνικό δίκτυο και μόνον έτσι είναι δυνατή η οπισθοδιάδοση του σφάλματος που εμφανίζεται στην έξοδο του ταξινομητή. Τέλος, τόσο στη συγκεκριμένη εργασία, όσο ως καθολική πρακτική, συνηθίζεται ο ρυθμός με τον οποίο μεταβάλλονται τα βάρη του προ-εκπαιδευμένου δικτύου να είναι μικρότερος από αυτόν της εκπαίδευσης του ταξινομητή (έτσι άλλωστε προέκυψε και ο όρος fine-tuning).

Συνοπτικά, ο προτεινόμενος αλγόριθμος των Chen et al. δίνεται παρακάτω:

Algorithm 1 SAE With Logistic Regression (SAE-LR)

initialize mini-batch size \mathbf{b} , pretraining epochs \mathbf{pt} , pre-training learning rate \mathbf{pl} , fine-tuning epochs \mathbf{ft} , fine-tuning learning rate \mathbf{fl} , number of layers \mathbf{d} and number of neurons in each hidden layer $\mathbf{n}[\mathbf{d}]$. Input dimension \mathbf{D} , total number of classes \mathbf{C} .

for every layer $L(1 \leq L \leq d)$ **do**
 Construct an autoencoder with \mathbf{d}_{vis} input neurons, \mathbf{d}_{hid} hidden neurons.
 if L is the first layer (i.e., $L = 1$) **then**
 $\mathbf{d}_{\text{vis}} = \mathbf{D}$
 $\mathbf{d}_{\text{hid}} = \mathbf{n}[1]$
 Set input of the autoencoder x to be initial data.
 else
 $\mathbf{d}_{\text{vis}} = \mathbf{n}[L-1]$
 $\mathbf{d}_{\text{hid}} = \mathbf{n}[L]$
 Set input of the autoencoder x to be the output of its former layer.
 end if
 initialize AE weight matrix \mathbf{W} with random variables, and biases b_y and b_z as zeros.

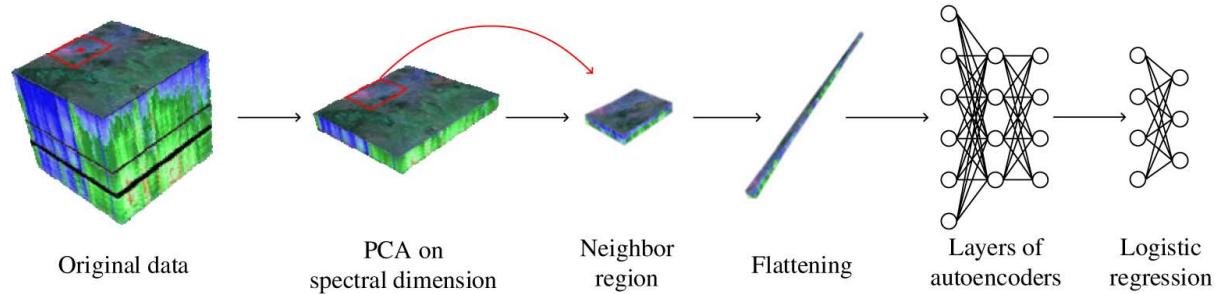
for every pretraining epoch **do**
 for every mini-batch **do**
 Compute reconstruction: $z = f(Wf(Wx + b_y) + b_z)$
 Compute cost: $c = -\frac{1}{b}[x \log z - (1 - x) \log 1 - z]$
 Update weights using 2.10, with learning rate \mathbf{pl} .
 end for
 end for
 Cast away the reconstruction layer.
end for

initialize logistic regression layer input neurons as $\mathbf{n}[\mathbf{d}]$, output neurons as \mathbf{C} .
for every fine-tuning epoch **do**
 for every mini-batch **do**
 Compute probability of each class according to 2.13.
 Update weights from top layer to the bottom using ordinary back propagation, with learning rate \mathbf{fl} .
 end for
 end for

2.1.2 Ταξινόμηση με έμφαση στα χωρικά χαρακτηριστικά

Το δεύτερο μοντέλο που μελετήθηκε, εκτελούσε την ταξινόμηση σε μία υπερφασματική εικόνα, λαμβάνοντας υπόψιν τη χωρική πληροφορία της εικόνας. Η σκέψη που οδηγεί αρκετά μοντέλα στην εκμετάλλευση της χωρικής πληροφορίας της εικόνας, η οποία θα αναφερθεί αρκετές φορές στην παρούσα εργασία, είναι αρκετά απλή και κατανοητή ακόμα και διαισθητικά:

Αν ένα pixel vector ανήκει σε μία κλάση, είναι αρκετά πιθανό και ένα γειτονικό του pixel vector να ανήκει στην ίδια κλάση.



Σχήμα 2.2: Ταξινόμηση με χωρικά-κυριαρχούμενα χαρακτηριστικά. Το πρώτο βήμα κατά την επεξεργασία είναι η εφαρμογή της διαδικασίας PCA για τη συμπίεση των διαστάσεων του φασματικού περιεχομένου. Ακολουθεί ο μετασχηματισμός της εικόνα σε διάνυσμα και η εισαγωγή του σε έναν stacked autoencoder για την εξαγωγή χαρακτηριστικών.

Στο σχήμα 2.2 φαίνεται το εν λόγω μοντέλο. Συγκεκριμένα, στο πρώτο επίπεδο χρησιμοποιείται η διαδικασία PCA (Principal Component Analysis) σε ολόκληρη την εικόνα, για την μείωση των διαστάσεων των δεδομένων εισόδου. Δεδομένου ότι σε αυτή τη μέθοδο δίνεται έμφαση στη χωρική πληροφορία της εικόνας, η διαδικασία PCA εφαρμόζεται στο φασματικό περιεχόμενο, μειώνοντας τα φασματικά κανάλια της εικόνας και διατηρώντας το χωρικό της περιεχόμενο. Στο δεύτερο βήμα, γίνεται εξαγωγή της γειτονικής περιοχής κάθε επισημασμένου pixel που θα χρησιμοποιηθεί για την εκπαίδευση (δηλαδή κάθε pixel για το οποίο γνωρίζουμε εκ των προτέρων την κλάση στην οποία ανήκει).

Θα πρέπει να αναφερθεί, ότι κατά την υλοποίηση του μοντέλου, ο αριθμός των principal components κατά την εφαρμογή της PCA ορίστηκε σε $n = 4$, δηλαδή διατηρήθηκαν 4 από τα συνολικά >100 φασματικά κανάλια της εικόνας, και το παράθυρο που εξάχθηκε έχοντας στο κέντρο το pixel που αναφέρεται είχε διαστάσεις 7×7 . Πρόκειται δηλαδή για ένα "κομμάτι" της εικόνας $7 \times 7 \times 4$. Στο τρίτο βήμα, αυτή η περιοχή μετασχηματίζεται σε ένα διάνυσμα 196×1 και τελικά δίνεται στο δίκτυο ως είσοδος.

Στην περίπτωση που δεν εφαρμόζονταν PCA στα φασματικά κανάλια της εικόνας, για μία εικόνας με 100 κανάλια θα προέκυπτε μία περιοχή $7 \times 7 \times 100$ που θα μετασχηματίζονταν σε ένα διάνυσμα 4900×1 , ανεπίτρεπτα πολλών διαστάσεων για το δίκτυο στο οποίο τροφοδοτείται.

Algorithm 2 Classification With Spatial-Dominated Feature

initialize neighborhood region size **a**, number of principle components **n**, image height **h**, width **w**.

PCA transform the image.

Retain the first **n** principle components. Thus we have an image of **h x w x n** size.

for each pixel **do**

Crop a neighboring region for each point.

For those points near the edge that don't have enough surrounding pixels, fill in with its mirror.

Flatten the **a x a x n** array into a vector of **a²n x 1** size.

end for

Concatenate all vectors to form a matrix **M**.

Train a SAE-LR with **M** as the input. Training procedures are the same to Algorithm 1.

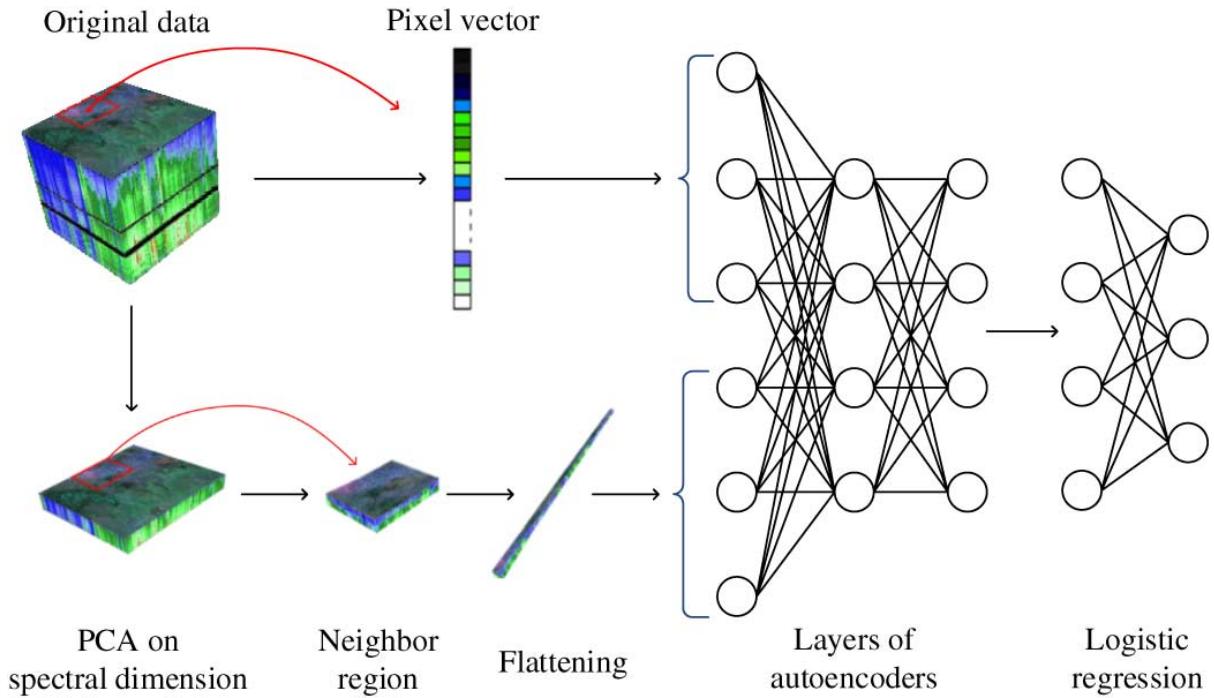
2.1.3 Ταξινόμηση με συνδυασμό των φασματικών και χωρικών χαρακτηριστικών

Η καινοτομία που εισήγαγαν οι Chen et al. βρίσκεται στο μοντέλο που παρουσιάζεται σε αυτή την υποενότητα, και το οποίο όπως θα φανεί στο επόμενο κεφάλαιο, είχε μεγαλύτερη ακρίβεια στην ταξινόμηση από τις δύο προσεγγίσεις που περιγράφηκαν μέχρι τώρα.

Έδω, συνδυάζονται η φασματική και η χωρική πληροφορία της εικόνας για την κατασκευή ενός συνδυαστικού μοντέλου, το οποίο φαίνεται στο σχήμα 2.3.

Πρώτα, λαμβάνεται υπόψιν το φασματικό περιεχόμενο του pixel, αφού περιλαμβάνει την σημαντικότερη πληροφορία όσων αφορά τη διάκριση διαφορετικών τύπων εδαφοκάλυψης. Για να συμπεριληφθεί η χωρική πληροφορία της εικόνας, γίνεται όπως προηγουμένως, εξαγωγή των πρώτων principal components της γειτονικής περιοχής. Για κάθε pixel, το διάνυσμα (196x1) που προκύπτει μετασχηματίζοντας την γειτονική περιοχή, προστίθεται στο φασματικό pixel vector, σχηματίζοντας μία υβριδική είσοδο που αποτελείται τόσο από φασματική όσο και από χωρική πληροφορία.

Η προ-εκπαίδευση και το fine-tuning είναι ίδιο με αυτό που έχει περιγραφεί προηγουμένως.



Σχήμα 2.3: Συνδυαστικό μοντέλο φασματική-χωρική ταξινόμησης. Η φασματική και η χωρική πληροφορία εξάγονται ξεχωριστά με χρήση των μεθόδων που έχουν ήδη περιγραφεί, και η εξαγωγή των χαρακτηριστικών γίνεται με βαθιές αρχιτεκτονικές stacked autoencoders. Τελικά, η ταξινόμηση επιτυγχάνεται με χρήση ενός λογιστικού ταξινομητή.

Algorithm 3 Joint Spectral-Spatial Classification

Extract Spatial-dominated feature for each pixel according to Algorithm 2 to form a matrix \mathbf{M} .

Scale \mathbf{M} into unit interval.

Normalize the whole initial image onto unit interval.

for each pixel **do**

Add spectrum of each pixel on tail of each pixel's feature vector, (i.e., rows in \mathbf{M}).

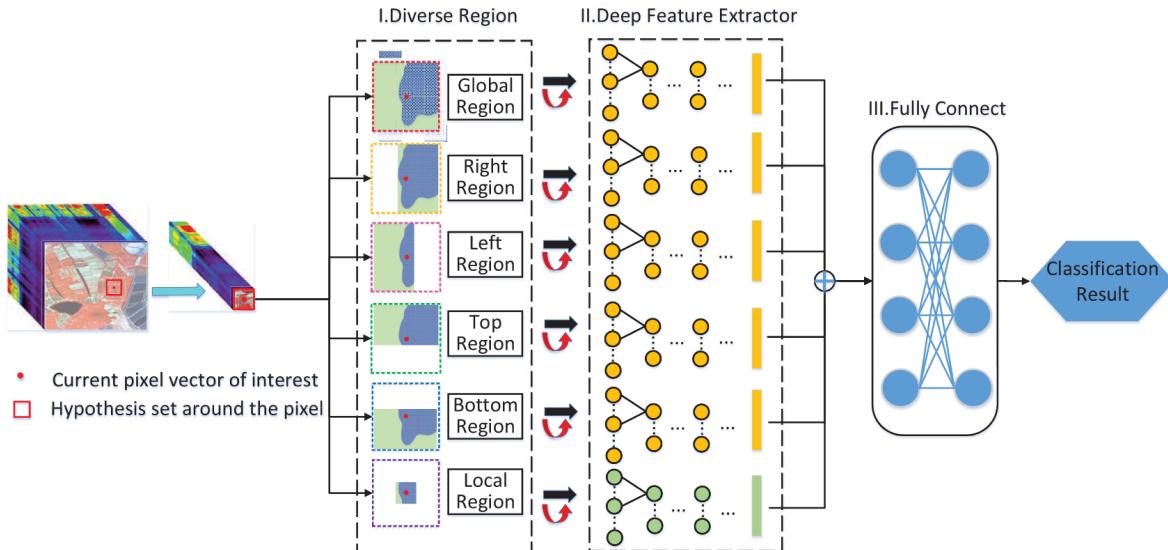
end for

Train a SAE-LR with \mathbf{M} as the input. Training procedures are the same to Algorithm 1.

2.2 Diverse Region-Based CNN

Στα πλαίσια της παρούσας διπλωματικής εργασίας, μελετήθηκε το μοντέλο που θεωρείται σήμερα state of the art, και το οποίο οι Zhang et al. ονομάζουν Diverse Region-Based CNN (DR-CNN).

Τα κλασσικά μοντέλα CNN χρησιμοποιούν ένα φίλτρο/παράθυρο συγκεκριμένου μεγέθους το οποίο ολισθαίνει στην εικόνα για την εξαγωγή χαρακτηριστικών (ουσιαστικά, τη μάθηση των βαρών του φίλτρου, όπως αυτή περιγράφηκε στο προηγούμενο κεφάλαιο).



Σχήμα 2.4: Το συνολικό διάγραμμα της αρχιτεκτονικής DR-CNN.

Αντίθετα, το DR-CNN τροφοδοτείται στην είσοδο όχι με ολόκληρη την εικόνα, αλλά με μικρά κομμάτια αυτής - τοπικές περιοχές (local regions) διαφορετικών μεταξύ τους διαστάσεων (κεντρική περιοχή, αρχική περιοχή και τέσσερις περιοχές με διαφορετικό προσανατολισμό). Η συνέλιξη, όπως γίνεται στα κλασσικά CNN, εκτελείται πάνω σε αυτές τις μικρές περιοχές και όχι με ολίσθηση σε ολόκληρη την εικόνα.

Το μοντέλο χρησιμοποιεί τις έξη εισόδους που αναφέρθηκαν παραπάνω για την εξαγωγή των φασματικών-χωρικών χαρακτηριστικών της υπερφασματικής εικόνας και έναν softmax ταξινομητή για την ταξινόμηση του κάθε pixel.

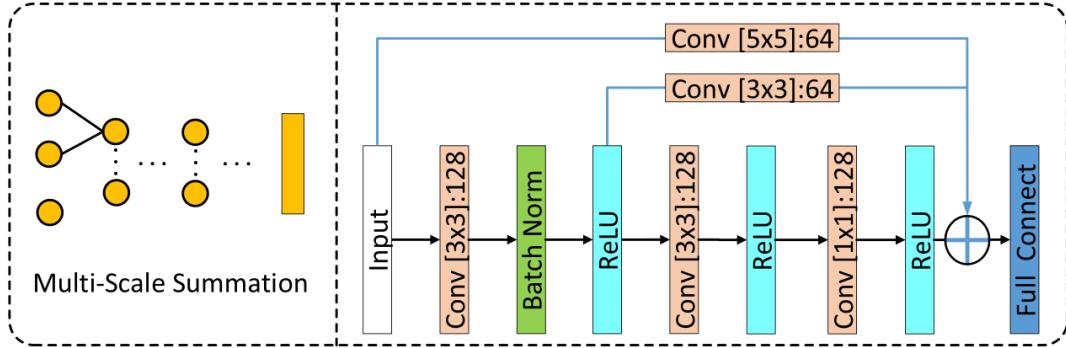
Πιο αναλυτικά, το βαθύ νευρωνικό δίκτυο που μελετήθηκε, αποτελείται από ένα συνδυασμό έξη τοπολογιών/διακλαδώσεων CNN, όπου κάθε διακλάδωση αντιστοιχεί σε μία από τις έξη περιοχές. Η αρχιτεκτονική που περιγράφηκε συντόμως παραπάνω, απεικονίζεται στο σχήμα 2.4.

Η έμπνευση για αυτή τη δομή της αρχιτεκτονικής βασίζεται στην υπόθεση ότι τα γειτονικά pixels συνήθως αποτελούνται από παρόμοια υλικά και με μεγάλη πιθανότητα ανήκουν στην ίδια κλάση με το κεντρικό pixel. Αυτό που είναι εξαιρετικά σημαντικό είναι ο τρόπος με τον οποίο επιλέγονται οι γειτονικές περιοχές - οι έξη περιοχές αντιστοιχίζονται σε ορθογώνια σχήματα με ποικίλα μεγέθη και κάθε περιοχή οδηγείται σε μία από τις έξη διακλαδώσεις CNN.

Σε κάθε μία από τις έξη διακλαδώσεις CNN που εμπεριέχονται στο μοντέλο, γίνεται χρήση ενός στοιχείου που οι ερευνητές ονόμασαν *multi-scale summation module*. Η χρήση του έγινε με κύριο στόχο την αντιμετώπιση του overfitting λόγω των περιορισμένων δειγμάτων εκπαίδευσης. Το στοιχείο αυτό φαίνεται στο σχήμα 2.5.

Σε ένα τυπικό μοντέλο CNN, τα πρώτα επίπεδα συνέλιξης, έχοντας στη διάθεσή τους εισόδους υψηλής χωρικής ανάλυσης, μαθαίνουν περισσότερες τοπικές λεπτομέρειες, ενώ επόμενα επίπεδα που δέχονται εισόδους μικρής χωρικής ανάλυσης διατηρούν πιο αφαιρετικές πληροφορίες, υψηλού σημασιολογικού επιπέδου.

Το multi-scale summation είναι εμπνευσμένο από μοντέλα όπως το densenet και το ResNet και είναι σχεδιασμένο ώστε να συνδυάζει τις τοπικές λεπτομέρειές οι οποίες διατηρούνται στα πρώτα επίπεδα με την πληροφορία υψηλού επιπέδου. Στη συγκεκριμένη τοπολογία, αυτό το πε-

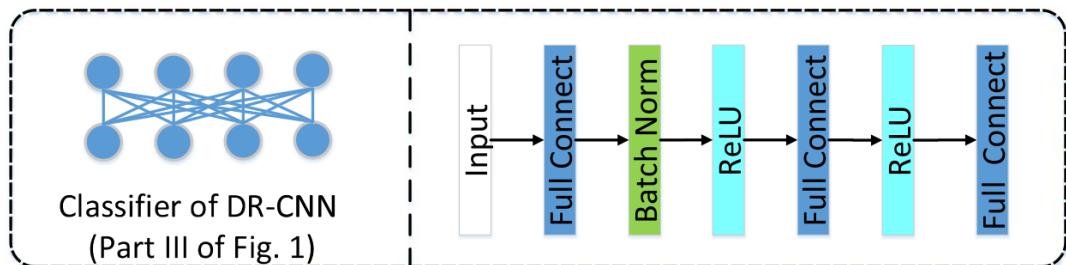


Σχήμα 2.5: Αναλυτικό διάγραμμα της διακλάσωσης CNN με το "multi-scale summation module".

τυχαίνει συγχωνεύοντας την είσοδο, τις ενεργοποιήσεις του πρώτου επιπέδου συνέλιξης και του τελευταίου επιπέδου. Στο σχήμα 2.5 οι δύο είσοδοι στο multi-scale summation που προέρχονται από προηγούμενα επίπεδα, περιλαμβάνουν επίπεδα συνέλιξης με διαφορετικά μεγέθη φίλτρων. Αυτό συμβαίνει ώστε οι διαστάσεις τους να προσαρμοστούν κατάλληλα στις διαστάσεις του feature map (ή με άλλα λόγια, των χαρακτηριστικών) του υψηλότερου επιπέδου συνέλιξης, και στο οποίο συγχωνεύονται.

Ουσιαστικά, στο σημείο αυτό γίνεται συνένωση των χαρακτηριστικών του υψηλότερου επιπέδου με τα χαρακτηριστικά που έχουν εξαχθεί από χαμηλότερα επίπεδα.

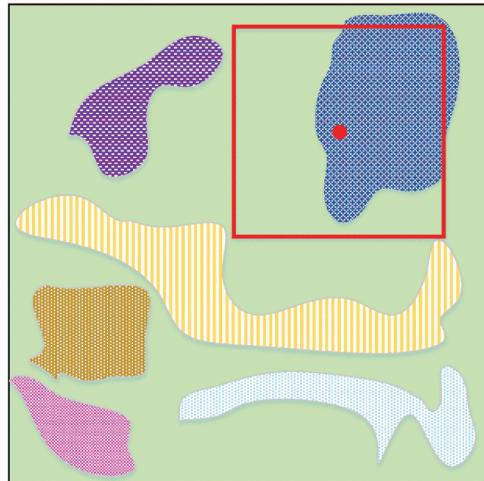
Στη συνέχεια, το σύνολο αυτό των επαυξημένων χαρακτηριστικών οδηγείται σε ένα fully-connected δίκτυο, όπως φαίνεται στο τμήμα III του 2.4. Περισσότερες λεπτομέρειες για το fully connected δίκτυο, φαίνονται στο σχήμα 2.6.



Σχήμα 2.6: Λεπτομέρειες του fully-connected δικτύου.

Οι τεχνικές φασματικής-χωρικής ταξινόμησης με CNN εκμεταλλεύονται τη χωρική συσχέτιση των γειτονικών pixel. Δηλαδή, τα υπερφασματικά pixels σε μία μικρή περιοχή γύρω από ένα κεντρικό pixel συμμετέχουν συνολικά στην εξαγωγή χαρακτηριστικών από το CNN. Παρόλα αυτά, όπως έχει ήδη αναφερθεί, τα κλασσικά μοντέλα CNN κάνουν χρήση εισόδων με σταθερές διαστάσεις καθώς και επιλογή φίλτρων με σταθερό και πάλι μέγεθος (π.χ. 3x3, 5x5, κλπ). Αυτή η επιλογή περιοχών μπορεί να περιλαμβάνει γειτονικά pixels διαφορετικών κλάσεων, ειδικά για εικόνες αστικού περιβάλλοντος οι οποίες έχουν σύνθετες κατανομές κλάσεων. Σε τέτοιες ιδιαίτερα ετερογενείς περιοχές, τα υλικά/αντικείμενα ακόμα και σε ένα πολύ μικρό τμήμα μπορεί να ανήκουν σε διαφορετικές κλάσεις. Αυτό είναι και το μειονέκτημα της χρήσης σταθερών τετράγωνων παραθύρων.

Όσα περιγράφηκαν παραπάνω, φαίνονται γραφικά στο σχήμα 2.7. Θεωρούμε ότι υπάρχει μόνο μία τοπική περιοχή (κόκκινο πλαίσιο), γειτονική του κεντρικού pixel, το οποίο ανήκει στο ”μπλε” υλικό. Επειδή τα περισσότερα pixels της τοπικής περιοχής ανήκουν στο ”πράσινο” υλικό, το αποτέλεσμα της ταξινόμησης του pixel ενδιαφέροντος, με χρήση της τετράγωνης περιοχής που έχει επιλεχθεί, θα είναι απογοητευτικό.



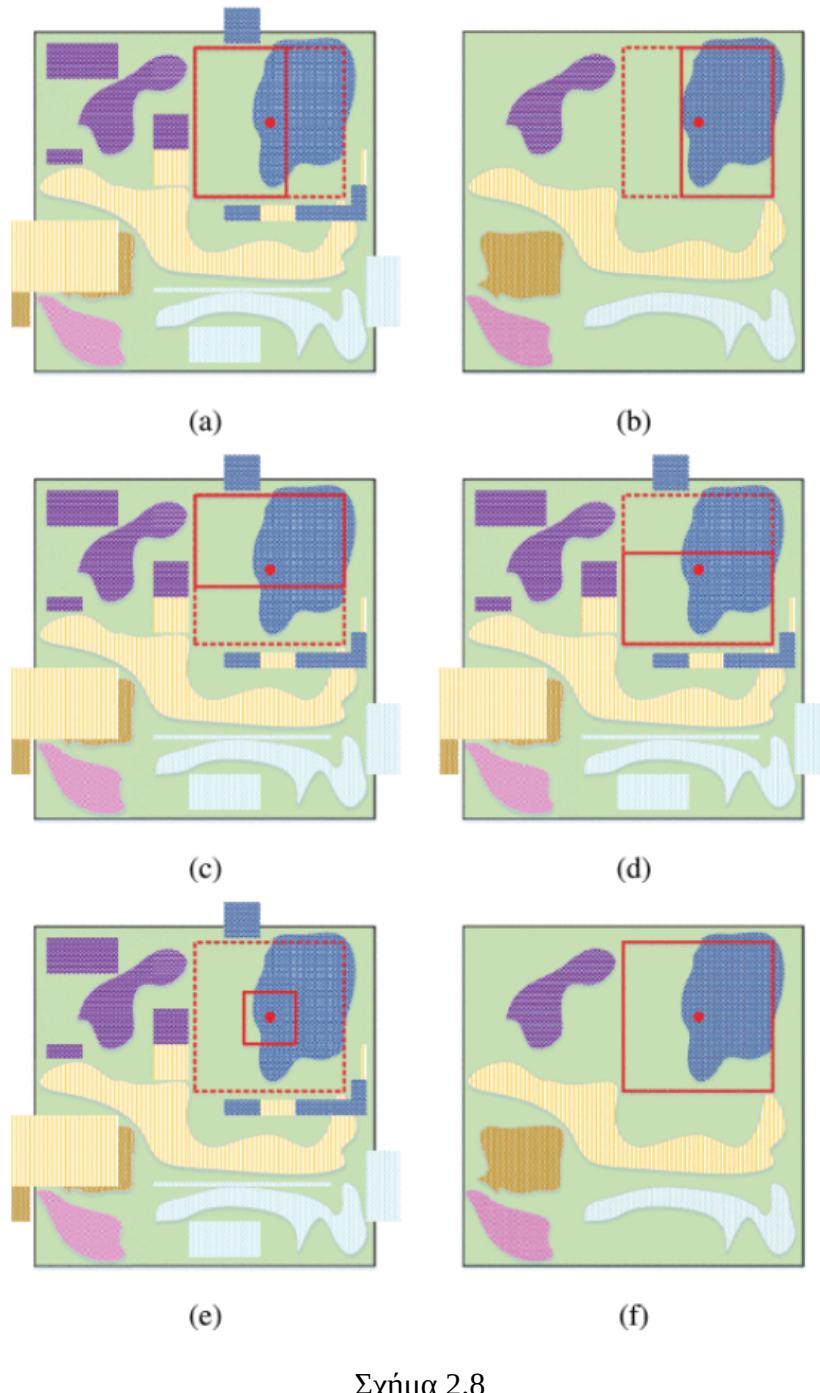
Σχήμα 2.7: Γραφική αναπαράσταση μιας τετράγωνης τοπικής περιοχής, επιλεγμένη από την αρχική εικόνα.

Στην ίδια εικόνα, παρατηρούμε ότι η περιοχή με pixels που ανήκουν στην ίδια κλάση με το κεντρικό, είναι κατανεμημένα κυρίως στη δεξιά πλευρά της τετράγωνης περιοχής. Σύμφωνα με αυτή τη παρατήρηση, αν αντί για το τετράγωνο κόκκινο πλαίσιο, εξάγουμε μόνο τη δεξιά γειτονική περιοχή του pixel ενδιαφέροντος, τότε γίνεται καλύτερη αξιοποίηση της χωρικής πληροφορίας της εικόνας.

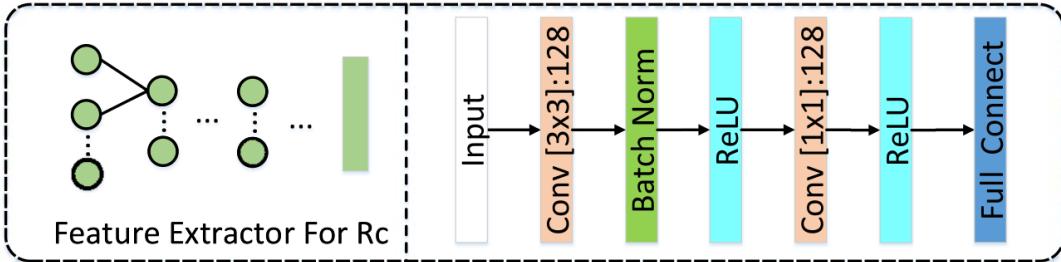
Βασιζόμενοι στην παραπάνω ανάλυση, οι Zhang et al., πρότειναν την εξαγωγή ποικίλων περιοχών, σύμφωνα με τις δυνατές κατανομές των υλικών στην εικόνα. Οι παρακάτω αποτελούν μερικές αντιπροσωπευτικές καταστάσεις:

1. ”Μισές” περιοχές διαφορετικού προσανατολισμού: υπάρχουν αρκετά ”μισά” τμήματα της τοπικής περιοχής (π.χ. αριστερά, δεξιά, πάνω, κάτω), όπως φαίνεται στο σχήμα 2.8 (a-d). Η διακλάδωση CNN του σχήματος 2.5 εκπαιδεύεται σε κάθε μία από αυτές τις ”μισές” περιοχές και εξάγει χαρακτηριστικά.
2. Κεντρική περιοχή: όπως φαίνεται στο 2.8 (e), η περιοχή αυτή δίνει έμφαση σε μία πολύ μικρή περιοχή γύρω από το pixel ενδιαφέροντος (3x3). Ειδικότερα, η διακλάδωση CNN που εκπαιδεύεται σε αυτή την πολύ μικρή περιοχή μπορεί να οδηγηθεί στην εξαγωγή ”καθαρών” φασματικών χαρακτηριστικών. Ειδικότερα, μπορεί η φασματική πληροφορία, ουσιαστικά η φασματική απόκριση, του pixel να είναι αλλοιωμένη εξαιτίας κάποιων παραγόντων. Όμως μία περιοχή 3x3 εξασφαλίζει τις ομοιότητες των γειτονικών pixels, μετριάζοντας την επίδραση αυτών των παραγόντων στη φασματική πληροφορία. Τα χαρακτηριστικά που εξάγονται από τις κεντρικές περιοχές συχνά δεν επηρεάζονται από σύνθετες χωρικές κατανομές σε ετερογενείς περιοχές. Σε αντίθεση με τις τοπικές περιοχές που έχουν αναφερθεί προηγουμένως, τα χαρακτηριστικά της κεντρικής περιοχής εξάγονται από το κομμάτι του δικτύου (μία άλλη διακλάδωση) του σχήματος 2.9.

3. Παγκόσμια περιοχή (global region): όπως φαίνεται στο 2.8 (f), είναι η τετράγωνη περιοχή όπου η διακλάδωση CNN εξάγει χαρακτηριστικά μιας μεγαλύτερης περιοχής, που πιθανότατα περιλαμβάνει pixels διαφορετικών κλάσεων.



Όλες οι περιοχές που αναφέρθηκαν παραπάνω, δίνονται ως δεδομένα εισόδου στο προτεινόμενο μοντέλο DR-CNN, όπως φαίνεται στο τμήμα I της 2.4. Οι διακλαδώσεις που φαίνονται στα σχήματα 2.5 και 2.9 αποτελούν τους feature extractors όπως υποδηλώνεται και στο τμήμα II του 2.4. Στο μοντέλο DR-CNN, κάθε διαδικασία συνέλιξης περιλαμβάνει κάποια επίπεδα: το



Σχήμα 2.9: Η διακλάδωση CNN για την εξαγωγή χαρακτηριστικών από την κεντρική περιοχή.

επίπεδο συνέλιξης, το επίπεδο μη-γραμμικού μετασχηματισμού (με την εφαρμογή της συνάρτησης ενεργοποίησης Relu) και ένα επίπεδο batch normalization. Συγκεκριμένα, όσων αφορά την υλοποίηση, η συνέλιξη γίνεται χωρίς zero padding, και με βήμα (stride) 1.

Ο αριθμός των φίλτρων στα επίπεδα συνέλιξης υποδεικνύεται στα σχήματα 2.5 και 2.9. Εστω R_C, R_G, R_L, R_R, R_T και R_B οι περιοχές εισόδου του CNN (center, global, left, right,top, bottom). Το επίπεδο συνέλιξης δημιουργεί ένα φίλτρο W το οποίο συνελλίσεται με τα δεδομένα εισόδου και προσθέτει ένα bias b για να παράξει ένα tensor εξόδων Z :

$$Z = W \otimes R_q + b, \quad q \in \{C, G, L, R, T, B\}$$

όπου το \otimes συμβολίζει την συνέλιξη. Υπάρχουν πολλές εναλλακτικές για τον μη-γραμμικό μετασχηματισμό, όπως η σιγμοειδής συνάρτηση και η υπερβολική εφαπτομένη. Εδώ, έχει επιλεγεί η συνάρτηση rectified linear unit (ReLU) [Hin] για το επίπεδο του μη-γραμμικού μετασχηματισμού, και υπολογίζει την ενεργοποίηση εξόδου

$$\tilde{Z} = \max\{0, Z\}$$

Για την κανονικοποίηση του batch (batch normalization - BN), γίνεται κανονικοποίηση των ενεργοποιήσεων του προηγούμενου επιπέδου σε κάθε batch. Με άλλα λόγια, εφαρμόζεται ένας μετασχηματισμός που διατηρεί τη μέση ενεργοποίηση κοντά στο 0 και την τυπική απόκλιση της ενεργοποίησης κοντά στο 1. Ας θεωρήσουμε ότι το batch έχει μέγεθος m και το \tilde{Z} προκύπτει από τα δείγματα ολόκληρου του batch. Σύμφωνα με αυτά, υπάρχουν m τιμές αυτής της ενεργοποίησης στο batch $\tilde{Z} = \{\tilde{Z}_1, \tilde{Z}_2, \dots, \tilde{Z}_m\}$ που υπολογίζονται ως εξής:

$$\hat{Z} = \gamma \frac{\tilde{Z} - \varepsilon(\tilde{Z})}{\sqrt{\text{Var}(\tilde{Z}) + \xi}} + \beta$$

όπου το

$$\hat{Z} = \{\hat{Z}_1, \hat{Z}_2, \dots, \hat{Z}_m\}$$

συμβολίζει την έξοδο των δειγμάτων στο batch αφού δεχθεί κανονικοποίηση, το $\varepsilon(\tilde{Z})$ και το $\text{Var}(\tilde{Z})$ συμβολίζουν την προσδοκώμενη τιμή (expectation) και την απόκλιση του \tilde{Z} , αντίστοιχα. Τα γ και β είναι υπερ-παράμετροι προς μάθηση.

Γενικά, η αλυσίδα του feature extractor τελειώνει στο fully-connected επίπεδο, και ολόκληρη η διαδικασία της εξαγωγής χαρακτηριστικών για μία συγκεκριμένη περιοχή ορίζεται ως:

$$f_{R_q} = F(R_q, \theta), \quad q \in \{C, G, L, R, T, B\}$$

όπου η συνάρτηση F αποτελείται από τη διαδικασία της συνέλιξης και της fully-connected διαδικασίας, το R_q αναπαριστά την συγκεκριμένη περιοχή, το $f_{R_q} \in \Re^{1 \times 1}$ είναι τα χαρακτηριστικά που έχουν εξαχθεί από την R_q και το θ αποτελείται από τα W, b, γ και β .

Αφού γίνει η εξαγωγή των χαρακτηριστικών από όλες τις περιοχές, αυτά συγχωνεύονται μαζί. Αρχικά, τα χαρακτηριστικά που προέρχονται από διαφορετικές διακλαδώσεις ενώνονται για να σχηματίσουν ένα διάνυσμα χαρακτηριστικών (feature vector)

$$f = \{f_{R_C}, f_{R_G}, f_{R_L}, f_{R_R}, f_{R_T}, f_{R_B}\}$$

Έπειτα όπως φαίνεται στο τμήμα III του 2.4, τα fully-connected επίπεδα συνδυάζουν αυτά τα χαρακτηριστικά, θεωρώντας το f ως είσοδο. Τέλος, το επίπεδο softmax αποδίδει την κλάση στο pixel ελέγχου.

Ας σημειωθεί ότι η συνολική εκπαίδευση του μοντέλου ξεκινάει με την εκπαίδευση κάθε διακλάδωσης ξεχωριστά, όπου κάθε διακλάδωση έχει στο τέλος ένα softmax επίπεδο. Αυτό το fully-connected επίπεδο υπάρχει ώστε να γίνεται υπολογισμός του σφάλματος και ανανέωση των βαρών και αφού ολοκληρωθεί η εκπαίδευση των διακλαδώσεων, τα fully-connected επίπεδα που φέρουν στην έξοδό τους απομακρύνονται. Εν τέλει, αυτό που απομένει σε κάθε δίκτυο-διακλάδωση είναι τα χαρακτηριστικά που έχουν μάθει και αυτά συγχωνεύονται δίνονται συνολικά σε έναν ταξινομητή.

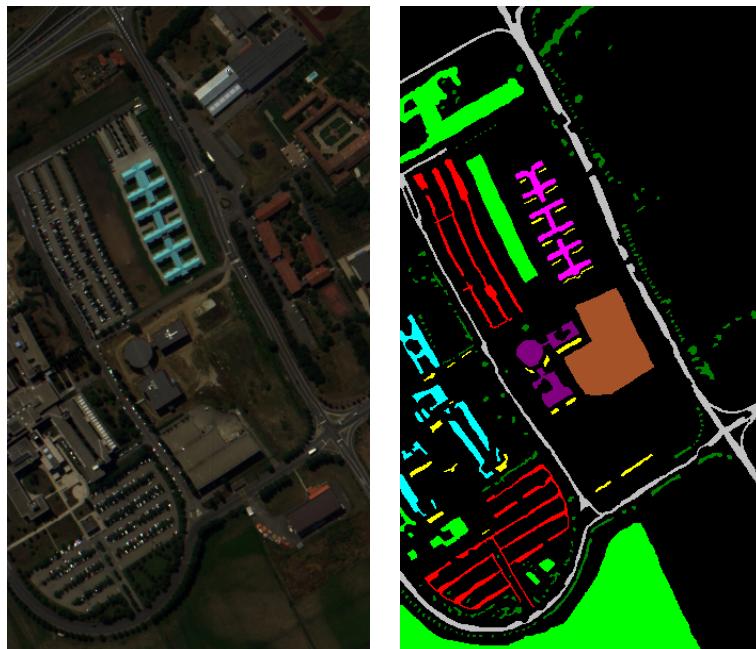
Κεφάλαιο 3

Πειραματικά αποτελέσματα

3.1 Τα datasets που χρησιμοποιήθηκαν

3.1.1 Pavia University

Η υπερφασματική εικόνα του παντεπιστημίου της Pavia στην Ιταλία, έχει αποκτηθεί με τον εναέριο αισθητήρα ROSIS-3 (Reflective Optics System Imaging Spectrometer), το 2003. Αποτελείται από 610x340 pixels με χωρική ανάλυση 1.3m. Από τα 115 αρχικά φασματικά κανάλια που υποστηρίζει το όργανο, έχουν αφαιρεθεί 12 κανάλια με τον περισσότερο θόρυβο και διατηρήθηκαν 103 φασματικά κανάλια, τα οποία καλύπτουν την περιοχή του φάσματος 0.43 με 0.86 μμ. Το ground truth περιλαμβάνει 9 κλάσεις, η οποίες βρίσκονται στον παρακάτω πίνακα.



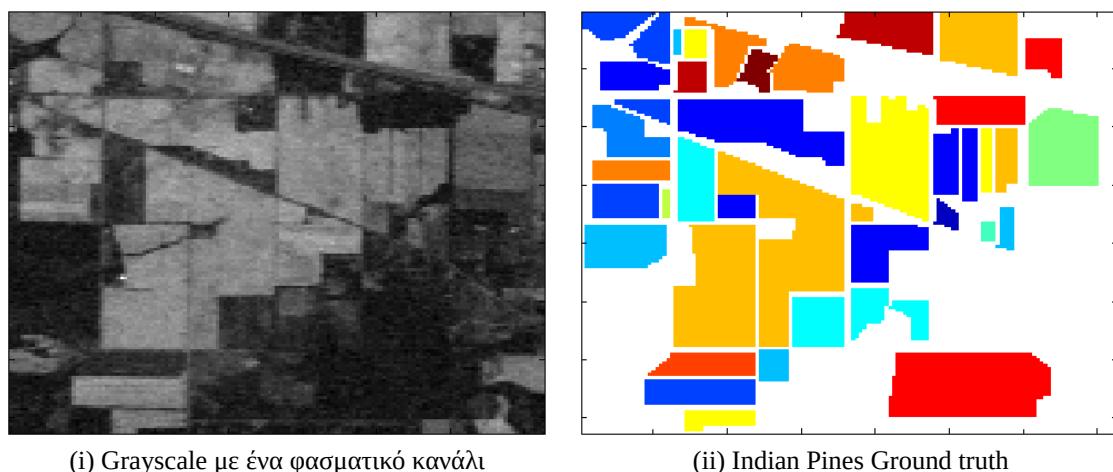
(i) False-color σύνθεση με χρήση των φασματικών καναλιών 53,31 και 8.

Σχήμα 3.1: False-color composite και ground truth

#	Class	Samples
1	Water	6631
2	Trees	18649
3	Asphalt	2099
4	Self-Blocking Bricks	3064
5	Bitumen	1345
6	Tiles	5029
7	Shadows	1330
8	Meadows	3682
9	Bare Soil	947
	Sum of labeled pixels	42776

3.1.2 Indian Pines

Η εικόνα αυτή αποκτήθηκε από το όργανο AVIRIS, στην περιοχή Indian Pines της Indiana και αποτελείται από 145x145 pixels με χωρική ανάλυση 20m. και 224 φασματικές μπάντες, οι οποίες καλύπτουν το μέρος του φάσματος από 0.4 μέχρι 2.5μμ. Η εικόνα Indian Pines περιλαμβάνει κατά 2/3 γεωργικές εκτάσεις και 1/3 δασικές περιοχές και βλάστηση. Υπάρχουν δύο δρόμοι με διπλές λωρίδες, γραμμές τραίνου, καθώς και αραιές κατοικίες, άλλα κτήρια και μικρότεροι δρόμοι. Δεδομένου ότι η εικόνα αποκτήθηκε τον Ιούνιο κάποιες από τις καλλιέργειες που βρίσκονται στην περιοχή, όπως οι καλλιέργειες καλαμποκιού, και σόγια είναι στα πρώτα στάδια με κάλυψη μικρότερη του 5% της συνολικής εδαφοκάλυψης. Το διαθέσιμο ground truth αποτελείται από 16 κλάσεις. Επιπλέον, ο αριθμός των φασματικών καναλιών έχει μειωθεί, με την απομάκρυνση των καναλιών [104-108], [150-163] και 220 λόγω υψηλής απορρόφησης από την υγρασία της ατμόσφαιρας και χαμηλού SNR. Τελικά, έχουν διατηρηθεί 200 μπάντες.



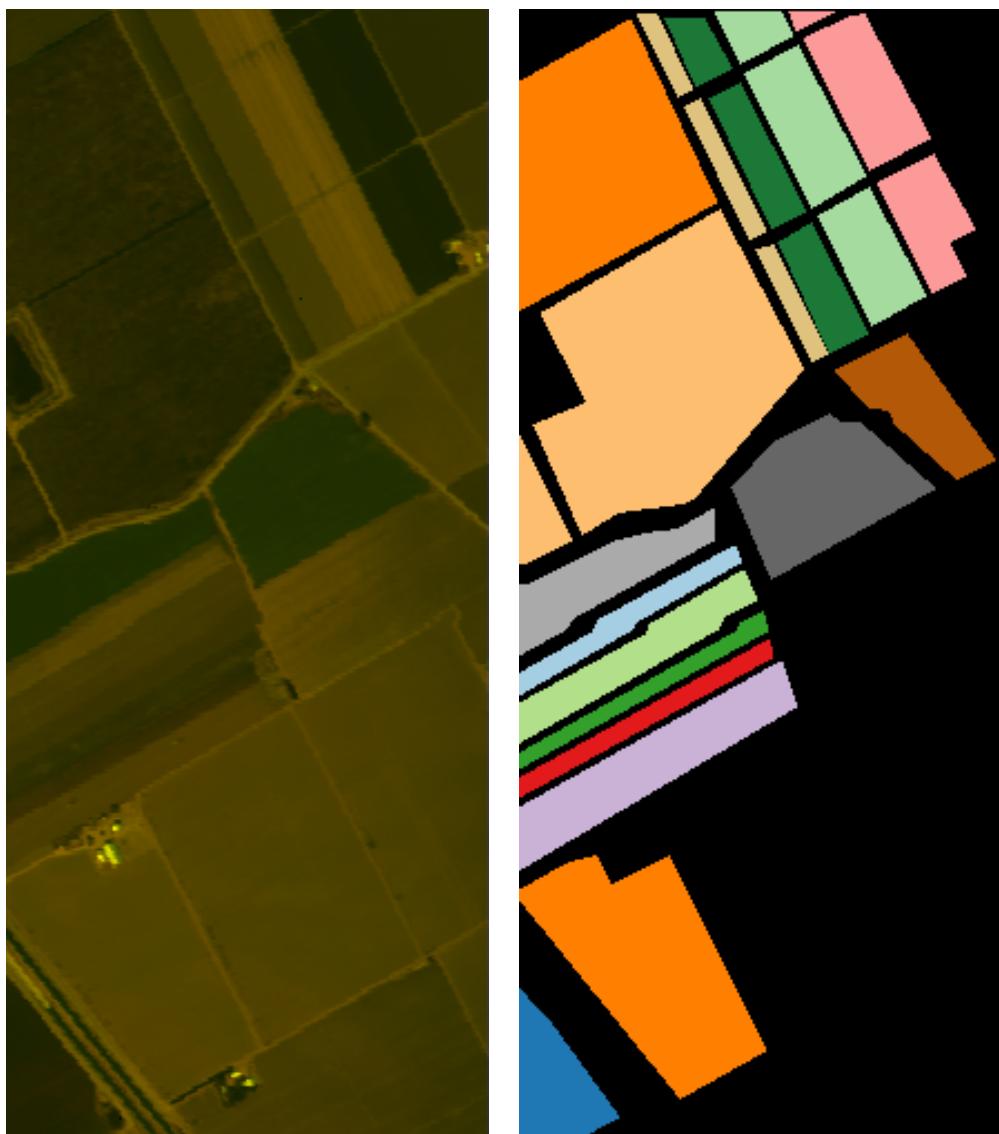
Σχήμα 3.2: Grayscale εικόνα και ground truth

#	Class	Samples
1	Alfalfa	46
2	Corn-notill	1428
3	Corn-mintill	830
4	Corn	237
5	Grass-pasture	483
6	Grass-trees	730
7	Grass-pasture-mowed	28
8	Hay-windrowed	478
9	Oats	20
10	Soybean-notill	972
11	Soybean-mintill	2455
12	Soybean-clean	593
13	Wheat	205
14	Woods	1265
15	Buildings-Grass-Trees-Drives	386
16	Stone-Steel-Towers	93
	Sum of labeled pixels	10249

Πίνακας 3.1: Επισημασμένα pixels ανά κλάση για το Indian Pines dataset.

3.1.3 Salinas

Αυτή η υπερφασματική εικόνα συλλέχθηκε από το AVIRIS, στην περιοχή Salinas Valley, στην California. Αποτελείται από 512x217 pixels και έχει χωρική ανάλυση 3.7m. Όπως και στο Indian Pines dataset, και εδώ αφαιρέθηκαν οι 20 φασματικές μπάντες απορρόφησης νερού: στη συγκεκριμένη περίπτωση οι [108-112], [154-167] και 224. Περιλαμβάνει καλλιέργειες, γυμνό έδαφος και αμπέλια. Το ground truth περιλαμβάνει 16 κλάσεις.



(i) False-color σύνθεση με χρήση των φασματικών καναλιών 25, 10 και 5.

(ii) Salinas Ground truth

Σχήμα 3.3: False-color composite και ground truth

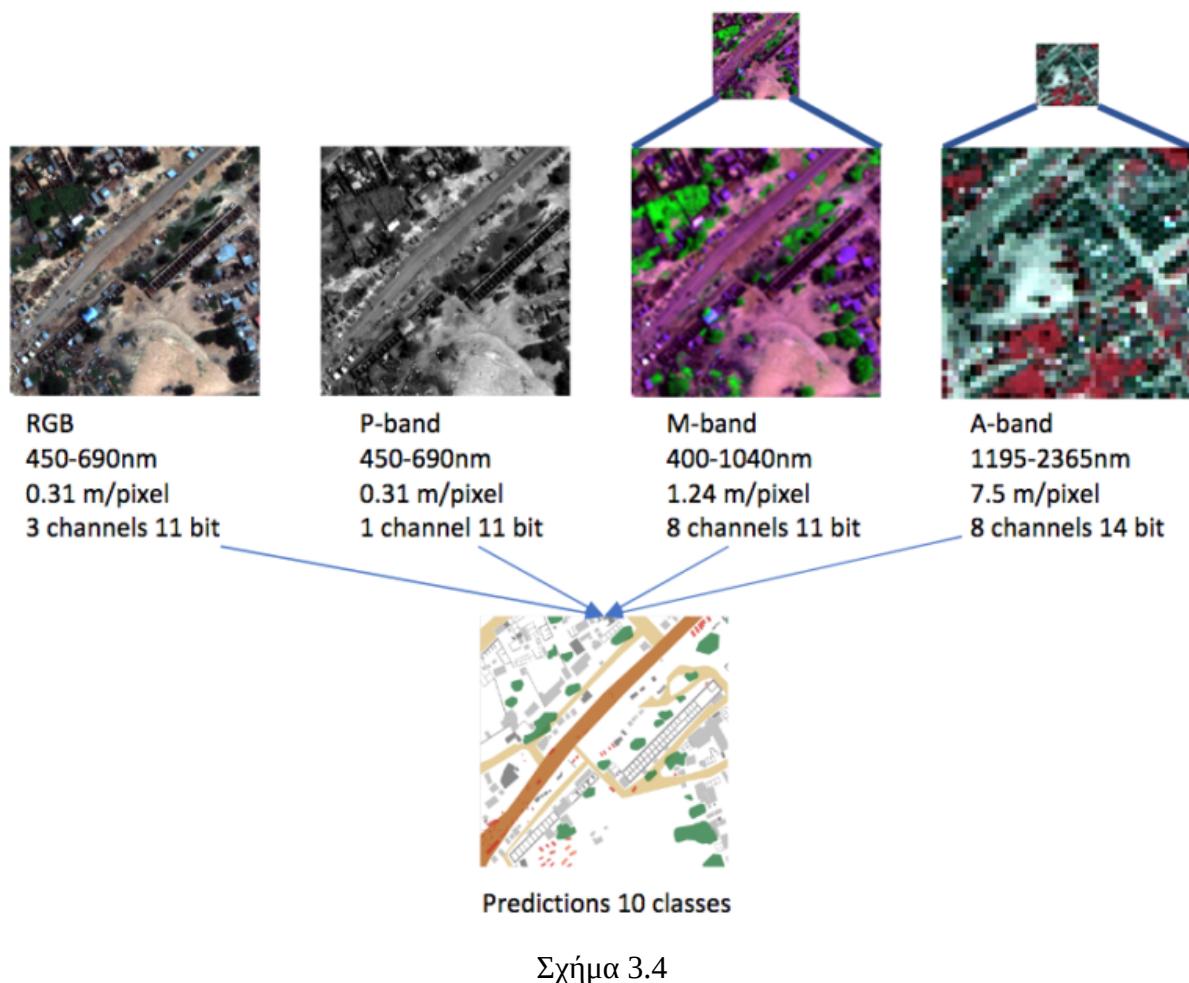
#	Class	Samples
1	Brocoli green weeds 1	2009
2	Brocoli green weeds 2	3726
3	Fallow	1976
4	Fallow rough plow	1394
5	Fallow smooth	2678
6	Stubble	3959
7	Celery	3579
8	Grapes untrained	11271
9	Soil vineyard develop	6203
10	Corn senesced green weeds	3278
11	Lettuce romaine 4wk	1068
12	Lettuce romaine 5wk	1927
13	Lettuce romaine 6wk	916
14	Lettuce romaine 7wk	1070
15	Vinyard untrained	7268
16	Vinyard vertical trellis	1807
	Sum of labeled pixels	54129

3.1.4 Kaggle DSTL Competition dataset

Στα πλαίσια του διαγωνισμού Dstl Satellite Imagery Feature Detection στην πλατφόρμα Kaggle, δημοσιεύτηκε ένα dataset με εικόνες του δορυφόρου WorldView-3 της DigitalGlobe.

Αυτό περιλαμβάνει 57 εικόνες. Κάθε εικόνα καλύπτει επιφάνειες ενός τετραγωνικού χιλιομέτρου και αποτελείται από:

- μία παγχρωματική εικόνα, 450-800nm, με χωρική ανάλυση 0.31m
- μία πολυφασματική εικόνα 8 καναλιών (red, red edge, coastal, blue, green, yellow, near-IR1 and near-IR2), 400nm - 1040nm, με χωρική ανάλυση 1.24m και
- μία ακόμα πολυφασματική εικόνα 8 καναλιών στο short wave infrared (SWIR) τμήμα του φάσματος, 1195nm - 2365nm με χωρική ανάλυση 7.5m.



Class	Additional Description
Buildings	large buildings, residential, non-residential, fuel storage facilities, fortified building
Structures	man-made structures
Road	-
Track	poor/dirt/cart tracks, footpaths/trails
Trees	woodland, hedgerows, groups of trees, stand-alone trees
Crops	contour ploughing/cropland, grain crops, row (potatoes, turnips) crops
Waterway	-
Standing water	-
Vehicle Large	large vehicle (e.g. lorry, truck, bus), logistics vehicle
Vehicle Small	small vehicle (car, van), motorbike

Πίνακας 3.2: Οι κλάσεις του dataset του διαγωνισμού.

3.2 Τα frameworks Theano, Tensorflow και Keras

Theano

Το Theano αναπτύχθηκε το 2007 από ερευνητές (Bengio, Pascal, κ.ά.) στο Montreal Institute for Learning Algorithms (MILA) του Πανεπιστημίου του Μοντρέαλ. Το Theano είναι ταυτόχρονα μία βιβλιοθήκη και ένας μαθηματικός μεταγλωττιστής (compiler) σε CPU και GPU, το οποίο επιτρέπει στον χρήστη να ορίσει, να βελτιστοποιήσει και να υπολογίσει μαθηματικές εκφράσεις που περιλαμβάνουν πολυδιάστατους πίνακες, με έναν αποδοτικό τρόπο. Από την εμφάνισή του χρησιμοποιήθηκε από την κοινότητα της μηχανικής μάθησης και υπήρχε ενεργή και συνεχής ανάπτυξη από το 2008 μέχρι το 2017.

Το Theano επιτρέπει στον χρήστη να ορίσει συμβολικά μαθηματικές εκφράσεις (τις οποίες αναπαριστά ως καθοδηγούμενους μη-κυκλικούς γράφους) και να τις μεταγλωτίσει με έναν βελτιστοποιημένο τρόπο είτε στη CPU είτε στη GPU (με τη χρήση CUDA). Επιπλέον, το Theano μπορεί να υπολογίσει τη συμβολική παραγώγιση σύνθετων εκφράσεων, να αγνοήσει μεταβλητές που δεν απαιτούνται στον υπολογισμό της τελικής εξόδου, να επαναχρησιμοποιήσει προηγούμενα μερικά αποτελέσματα, ώστε να αποφύγει περιττούς υπολογισμούς, να εφαρμόσει μαθηματικές απλοποιήσεις, καθώς και βελτιστοποιήσεις που ελαχιστοποιούν τα σφάλματα λόγω προσεγγίσεων του hardware.

Η διεπαφή (API) του Theano είναι η γλώσσα Python. Το API του Theano είναι παρόμοιο με το NumPy, μία ευρέως διαδεδομένη βιβλιοθήκη στην Python, η οποία παρέχει έναν n-διάστατο πίνακα ως δομή δεδομένων καθώς και πολλές συναρτήσεις με indexing, μετασχηματισμούς και βασικούς υπολογισμούς (exp, log, sin, κλπ) σε έναν ολόκληρο πίνακα μονομιάς. [TDT+16]

Η ανάπτυξη του Theano από το MILA σταμάτησε το 2017 μετά την έκδοση 1.0.

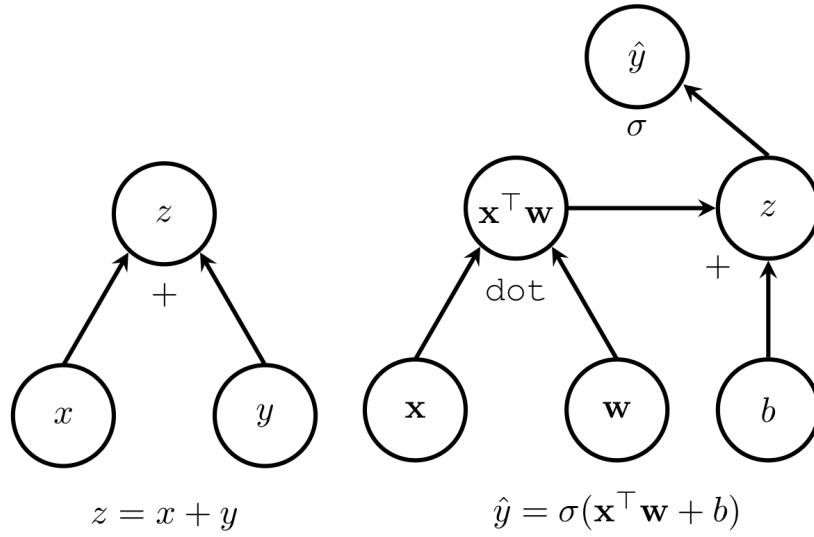
Tensorflow

Το Νοέμβριο του 2015, η Google δημοσίευσε το Tensorflow, μία deep learning βιβλιοθήκη ανοιχτού κώδικα, για μοντέλα μηχανικής μάθησης. Χαρακτηριστικά, σύμφωνα με την αρχική δημοσίευση: "an interface for expressing machine learning algorithms".

Στο Tensorflow, οι αλγόριθμοι μηχανικής μάθησης αναπαρίστανται ως υπολογιστικοί γράφοι. Ένα υπολογιστικός γράφος ή γράφος ροής δεδομένων (dataflow graph) είναι ένα είδος καθοδηγούμενου γράφου, στον οποίο οι κόμβοι περιγράφουν διαδικασίες ενώ οι ακμές αναπαριστούν τη ροή των δεδομένων μεταξύ αυτών των διαδικασιών. Αν μία μεταβλητή εξόδου z είναι το αποτέλεσμα μιας δυαδικής πράξης των εισόδων x και y , τότε προκύπτει ένας γράφος όπως αυτός του σχήματος 3.6.

Εν συντομία, παρουσιάζονται τα βασικά στοιχεία ενός τέτοιου γράφου, και συγκεκριμένα οι διαδικασίες (operations), tensors και οι μεταβλητές (variables).

1. Operations: Το μεγαλύτερο πλεονέκτημα της αναπαράστασης ενός αλγορίθμου με τη μορφή γράφου είναι όχι μόνο οπτική αντίληψη των εξαρτήσεων μεταξύ των στοιχείων ενός υπολογιστικού μοντέλου, αλλά και το γεγονός ότι ο κόμβος του γράφου έχει έναν γενικό ορισμό. Στο Tensorflow, οι κόμβοι αναπαριστούν διαδικασίες, οι οποίες με τη σειρά τους εκφράζουν το συνδυασμό ή τον μετασχηματισμό των δεδομένων που "réouν" μέσα στον γράφο. Μία διαδικασία μπορεί να έχει καμία ή πολλές εισόδους και να παράγει μηδενικές ή πολλές εξόδους. Ετσι, μπορεί να αναπαραστήσει μία μαθηματική εξίσωση, μία μεταβλητή ή σταθερά, μία συνθήκη για τον έλεγχο ροής, μία I/O διαδικασία ή ακόμα και μία θύρα δικτύου. Στον πίνακα 3.3 αναγράφονται διαφορετικοί τύποι διαδικασιών που μπορούν να εμφανιστούν σε έναν γράφο.



Σχήμα 3.5: Παραδείγματα υπολογιστικών γράφων. Ο αριστερός γράφος δείχνει έναν απλό υπολογισμό, ο οποίος αποτελείται μόνο από την πρόσθεση δύο μεταβλητών εισόδου, x , και y . Σε αυτή την περίπτωση, το z είναι το αποτέλεσμα της πράξης $+$, όπως υποδηλώνει ο συμβολισμός. Ο γράφος στα δεξιά δίνει ένα πιο σύνθετο παράδειγμα, υπολογίζοντας μία logistic regression μεταβλητή, \hat{y} για ένα διάνυσμα x , διάνυσμα βαρών w και ένα βαθμωτό bias b . Όπως φαίνεται στον γράφο, το \hat{y} είναι το αποτέλεσμα μιας σιγμοειδούς ή λογιστικής συνάρτησης σ .

Σχήμα 3.6

2. **Tensors:** Στο Tensorflow, οι ακμές αναπαριστούν τα δεδομένα που ρέουν από τη μία διαδικασία στην άλλη και αναφέρονται ως tensors (λόγω έλλειψης καλύτερης μετάφρασης, πέρα από "n-διάστατους πίνακες", θα εμφανίζονται ως tensors, στην παρούσα εργασία). Ένας tensor είναι μία πολυδιάστατη συλλογή ομογενών τιμών με έναν σταθερό τύπο. Ο αριθμός των διαστάσεων του tensor ονομάζεται τάξη (rank). Το σχήμα (shape) του είναι ένα ζευγάρι τιμών (tuple) που περιγράφει το μέγεθός του, δηλαδή των αριθμών στοιχείων σε κάθε διάσταση. Από μαθηματική σκοπιά, ο tensor αποτελεί γενίκευση του πίνακα δύο-διαστάσεων, του μονοδιάστατου διανύσματος και των βαθμωτών τιμών (που θεωρούνται tensors μηδενικής τάξης). Ο ίδιος ο tensor δεν κρατάει ούτε αποθηκεύει τιμές στη μνήμη, αλλά παρέχει μια διεπαφή για την ανάκτηση των τιμών που αντιστοιχίζονται σε αυτόν. Όταν δημιουργούμε μία διαδικασία στο Tensorflow, όπως η έκφραση $x + y$, επιστρέφεται έναν αντικείμενο τύπου tensor. Αυτό μπορεί να αποτελέσει την είσοδο άλλων υπολογισμών, συνδέοντας μία διαδικασία με μία άλλη μέσω μιας ακμής. Με αυτή την έννοια, τα δεδομένα "ρέουν" μέσα στον γράφο.
3. **Μεταβλητές:** Σε μία τυπική περίπτωση, όπως όταν εκτελείται η στοχαστική μέθοδος με βάση την κλίση (Stochastic Gradient Descent), ο γράφος ενός μοντέλου μηχανικής μάθησης εκτελείται από την αρχή ως το τέλος πολλές φορές για ένα μόνο πείραμα. Μεταξύ δύο τέτοιων εκτελέσεων, οι tensors στην πλειοψηφία τους καταστρέφονται. Παρόλα αυτά, είναι συχνά αναγκαίο να διατηρηθεί η κατάσταση (π.χ. των βαρών και άλλων παραμέτρων του δικτύου) του γράφου σε διαφορετικές εκτελέσεις. Για το λόγο αυτό, υπάρχουν οι μεταβλητές, οι οποίες είναι ειδικές διαδικασίες. [Gol16]

Category	Examples
Element-wise operations	Add, Mul, Exp
Matrix operations	MatMul, MatrixInverse
Value-producing operations	Constant, Variable
Neural network units	SoftMax, ReLU, Conv2D
Checkpoint operations	Save, Restore

Πίνακας 3.3: Παραδείγματα διαδικασιών του Tensorflow.

Keras

Το Keras είναι ένα API υψηλότερου επιπέδου, γραμμένο σε Python και το οποίο λειτουργεί, μεταξύ άλλων, "πάνω" από το Tensorflow ή το Theano. Αναπτύχθηκε τον Μάρτιο του 2015 και το 2017 η ομάδα του Tensorflow της Google το έχει εντάξει στο Tensorflow Framework.

Το Keras κάνει την ανάπτυξη μοντέλων πιο εύκολη και γρήγορη. Χαρακτηριστικά, παρατίθεται ένα παράδειγμα που επιβεβαιώνει τα παραπάνω.

TF example:

```
kernel = tf.Variable(tf.truncated_normal([3, 3, 64, 64], type=tf.float32, stddev=1e-1), name='weights')
conv = tf.nn.conv2d(self.conv1_1, kernel, [1, 1, 1, 1], padding='SAME')
biases = tf.Variable(tf.constant(0.0, shape=[64], dtype=tf.float32), trainable=True, name='biases')
out = tf.nn.bias_add(conv, biases)
self.conv1_2 = tf.nn.relu(out, name='block1_conv2')
```

Keras:

```
x = Convolution2D(64, 3, 3, activation='relu', border_mode='same', name='block1_conv2')(x)
```

Σχήμα 3.7: Υλοποίηση ενός επιπέδου συνέλιξης, με zero padding και ReLU, σε (low-level) Tensorflow και (high-level) Keras.

3.3 Πειραματικά αποτελέσματα του Stacked Autoencoder

Για τον stacked autoencoder των Chen et al., χρησιμοποιήθηκε το dataset Pavia University. Τα labeled pixels χωρίστηκαν σε τρία σύνολα: εκπαίδευσης (training), επαλήθευσης (validation) και ελέγχου (testing), με αναλογία 6:2:2. Δηλαδή, το 60% των συνολικών επισημασμένων pixel επιλέχθηκαν τυχαία ως δεδομένα εκπαίδευσης και 20% και 20% για δεδομένα επαλήθευσης και ελέγχου, αντίστοιχα.

#	Class	No. of training samples	No. of validation samples	No. of testing samples
1	Asphalt	4110	1370	1371
2	Bare soil	11212	3735	3739
3	Bitumen	1324	441	442
4	Gravel	2062	689	685
5	Meadows	854	268	256
6	Metal sheets	3066	1016	1022
7	Bricks	824	274	258
8	Shadow	2345	781	752
9	Trees	626	192	208
	Total	25550	8550	8450

Πίνακας 3.4: Κλάσεις εδαφοκάλυψης και αριθμός των pixel στο Pavia University dataset

Κατά την εκπαίδευση, τα δεδομένα εκπαίδευσης χρησιμοποιούνται ώστε το μοντέλο να μάθει τα βάρη και τα biases για κάθε νευρώνα. Τα δεδομένα ελέγχου χρησιμοποιούνται για τον υπολογισμό των αποτελεσμάτων της ταξινόμησης.

Ως υπενθύμιση, οι Chen et al. πρότειναν τρεις προσεγγίσεις για την ταξινόμηση υπερφασματικών εικόνων με τη χρήση ενός stacked autoencoder για την εξαγωγή χαρακτηριστικών και ενός λογιστικού ταξινομητή:

1. ταξινόμηση με έμφαση στα φασματικά χαρακτηριστικά, όπου το μοντέλο δέχεται μόνο το φασματικό pixel vector της εικόνας,
2. ταξινόμηση με έμφαση στα χωρικά χαρακτηριστικά, όπου η είσοδος του μοντέλου προκύπτει από την γειτονική περιοχή του pixel ενδιαφέροντος και
3. ταξινόμηση με συνδυασμό των παραπάνω δύο προσεγγίσεων, τροφοδοτώντας το μοντέλο με μία υβριδική είσοδο.

Για το υπόλοιπο του κεφαλαίου, οι τρεις αυτές προσεγγίζεις θα εμφανίζονται ως spectral, spatial και joint, αντίστοιχα.

Οι δείκτες που χρησιμοποιήθηκαν για τη μέτρηση της απόδοση των μοντέλων είναι οι: overall accuracy (OA) και Kappa coefficients. Στον πίνακα 3.5 φαίνονται τα καλύτερα αποτελέσματα από το σύνολο των δοκιμών εκπαίδευσης των μοντέλων.

Dataset	Μετρικές	Spectral-dominated	Spatial-dominated	Joint
Pavia	OA	0.956145	0.986024	0.989091
	Kappa coef.	0.942647	0.981718	0.985714

Πίνακας 3.5: Τα καλύτερα από τα αποτελέσματα των μοντέλων για τα δύο datasets

Στον πίνακα 3.5 είναι φανερό ότι η με την από κοινού αξιοποίηση της φασματικής και της χωρικής πληροφορίας της εικόνας, κατασκευάζοντας υβριδική είσοδο που έχει περιγραφεί στο προηγούμενο κεφάλαιο, το εκπαιδευμένο μοντέλο δίνει την μεγαλύτερη ακρίβεια στην ταξινόμηση συγκριτικά με τις άλλες δύο προσεγγίσεις.

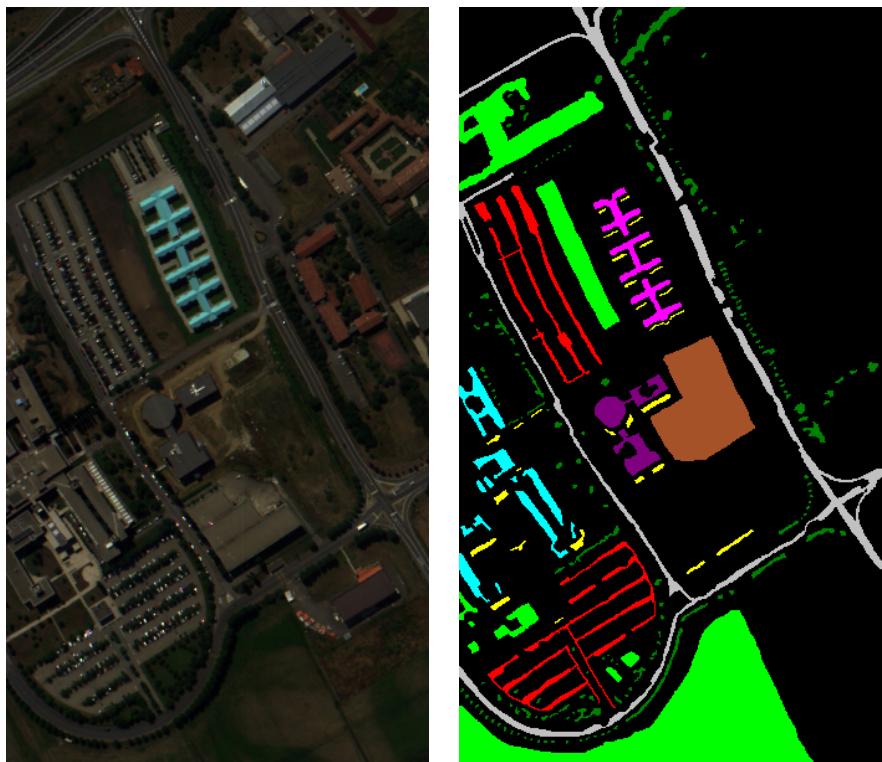
Για την εκπαίδευση των τριών μοντέλων στο Pavia University dataset, οι υπερ-παράμετροι για τις οποίες τα μοντέλα δίνουν τα καλύτερα αποτελέσματα, βρίσκονται στον παρακάτω πίνακα.

Hyperparameters	spectral	spatial	joint
pretraining learning rate	0.5	0.5	0.5
pretraining epochs	800	500	500
finetuning learning rate	0.05	0.03	0.03
finetuning epochs	250000	58000	80000
batch size	100	50	50
hidden layers [sizes]	[60, 60, 60, 60]	[310, 100, 100, 100, 100]	[280, 100, 100, 100]

Πίνακας 3.6: Υπερ-παράμετροι που δίνουν τα καλύτερα αποτελέσματα για το Pavia dataset.

Πέρα από τα ποσοστά ακρίβειας που έχουν παρουσιαστεί, μπορούν να γίνουν κάποιες παρατηρήσεις ακόμα και οπτικά. Στις εικόνες 3.9 (i-iii), μπορούμε να παρατηρήσουμε ότι το μοντέλο που χρησιμοποιεί την φασματική πληροφορία εμφανίζει αρκετό θόρυβο. Χαρακτηριστικά, υπάρχουν κουκίδες σε επιφάνειες που στην πραγματικότητα παρουσιάζουν μία συνέχεια και στο πάνω δεξιά μέρος της εικόνας, το μοντέλο έχει ταξινομήσει λανθασμένα μία αρκετά μεγάλη επιφάνεια ως "δρόμο" ενώ πρόκειται για τη μεταλλική σκεπή ενός κτηρίου (metal sheets). Παρόλα αυτά, ακόμα και το μοντέλο που κάνει χρήση της χωρική πληροφορίας παρουσιάζει και αυτό προβλήματα. Είναι εμφανές ότι αποτυγχάνει να αναγνωρίζει την περιοχή γυμνού εδάφους στο κάτω δεξιά τμήμα της εικόνας. Τελικά, το μοντέλο που συνδυάζει τις παραπάνω προσεγγίσεις, δίνει ένα πιο ικανοποιητικό αποτέλεσμα.

Για λόγους ευχρηστίας στις συγκρίσεις, παραθέτω και εδώ την εικόνα και το ground truth.



(i) False-color σύνθεση με χρήση των φασματικών καναλιών 53,31 και 8.

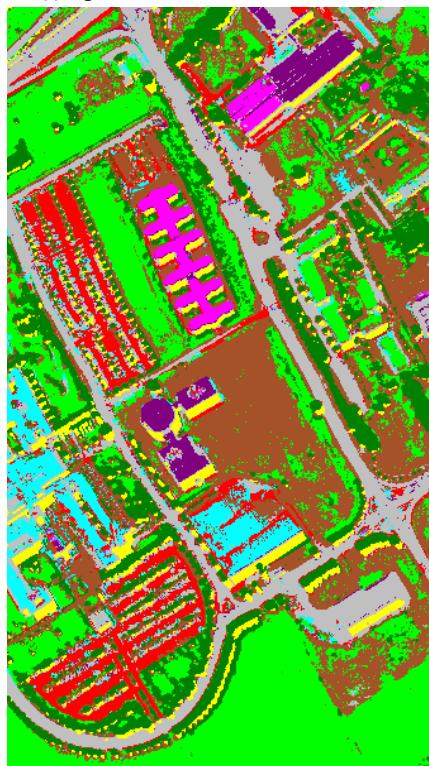
(ii) Pavia University Ground truth

Σχήμα 3.8: False-color composite και ground truth



(i) Spectral-dominated

(ii) Spatial-dominated



(iii) Joint spectral-spatial

Asphalt
Meadows
Gravel
Trees
Sheets
Baresoil
Biturmen
Bricks
Shadows

(iv) Υπόμνημα

Σχήμα 3.9: Οι τρεις προσεγγίσεις

3.4 Πειραματικά αποτελέσματα του DR-CNN

Η απόδοση του μοντέλου αξιολογείται με τη χρήση τριών datasets: (i) Indian Pines, (ii) Salinas και (iii) Pavia University. Για κάθε dataset επιλέχθηκαν τυχαία 200 επισημασμένα pixels κάθε κλάσης ως δεδομένα εκπαίδευσης και όλα τα υπόλοιπα ως δεδομένα ελέγχου.

Στο Indian Pines dataset, αν και αρχικά υπάρχουν 16 κλάσεις, απορρίφθηκαν οι 8 κλάσεις με τα λιγότερα δείγματα. Στον πίνακα 3.7 φαίνεται ο αριθμός δειγμάτων που χρησιμοποιήθηκαν για το Indian Pines dataset.

#	Class	Training	Test
1	Corn-notill	200	1228
2	Corn-mintill	200	630
3	Grass-pasture	200	283
4	Hay-windrowed	200	278
5	Soybean-notill	200	772
6	Soybean-mintill	200	2255
7	Soybean-clean	200	393
8	Woods	200	1065
-	Total	1600	6904

Πίνακας 3.7: Δείγματα εκπαίδευσης και ελέγχου για το Indian Pines dataset.

Τα datasets, Salinas και Pavia University, χρησιμοποιήθηκε στο σύνολό τους με όλες τις κλάσεις τους.

#	Class	Training	Test
1	Brocoli green weeds 1	200	1809
2	Brocoli green weeds 2	200	3526
3	Fallow	200	1776
4	Fallow rough plow	200	1194
5	Fallow smooth	200	2478
6	Stubble	200	3759
7	Celery	200	3379
8	Grapes untrained	200	11071
9	Soil vinyard develop	200	6003
10	Corn senesced green weeds	200	3078
11	Lettuce romaine 4wk	200	868
12	Lettuce romaine 5wk	200	1727
13	Lettuce romaine 6wk	200	716
14	Lettuce romaine 7wk	200	870
15	Vinyard untrained	200	7068
16	Vinyard vertical trellis	200	1607
-	Total	3200	50929

Πίνακας 3.8: Δείγματα εκπαίδευσης και ελέγχου για το Salinas dataset.

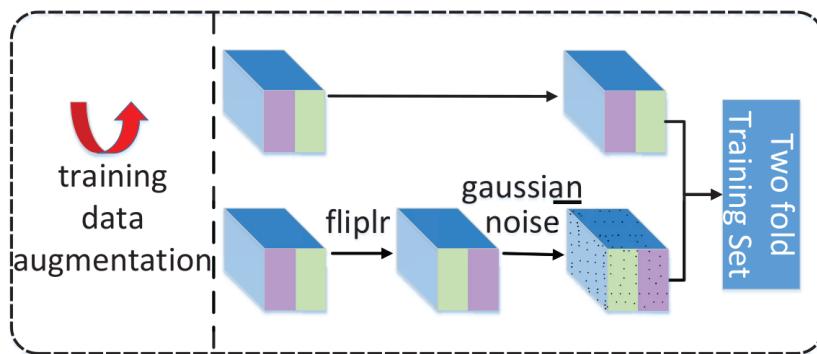
#	Class	Training	Test
1	Asphalt	200	6431
2	Meadows	200	18449
3	Gravel	200	1899
4	Trees	200	2864
5	Sheets	200	1145
6	Baresoil	200	4829
7	Bitumen	200	1130
8	Bricks	200	3482
9	Shadows	200	747
-	Total	1800	40976

Πίνακας 3.9: Δείγματα εκπαίδευσης και ελέγχου για το Pavia University dataset.

Μία βαθιά αρχιτεκτονική συνήθως απαιτεί πολλά δείγματα εκπαίδευσης ώστε να μπορέσει να μάθει μία καλή αναπαράσταση των δεδομένων. Όμως στην πράξη, για την ταξινόμηση υπερφασματικών εικόνων, τα datasets είναι περιορισμένα με λίγα επισημασμένα δείγματα.

Ένας από τους κύριους λόγους για την έλλειψη αρκετών επισημασμένων δειγμάτων είναι το κόστος δημιουργίας τους. Η αντιστοίχιση των pixel μιας εικόνας σε κλάσεις εδαφοκάλυψης περιλαμβάνει μεταξύ άλλων, εργασία πεδίου σε μεγάλες σε έκταση περιοχές.

Για αυτό, χρησιμοποιήθηκαν μέθοδοι (τεχνητής) επαύξησης των διαθέσιμων δεδομένων (data augmentation). Αυτές φαίνονται στην εικόνα 3.10.



Σχήμα 3.10: Η διαδικασία επαύξησης των δεδομένων.

Για κάθε δείγμα εκπαίδευσης εκτελούνται δύο βήματα επαύξησης των δεδομένων για την παραγωγή δεδομένων. Το πρώτο είναι η αναστροφή (flip) και το δεύτερο είναι η εισαγωγή γκαουσιανού θορύβου στα αρχικά δείγματα. Κατ’ αυτόν τον τρόπο, ο αριθμός των δειγμάτων εκπαίδευσης αυξάνεται κατά δύο φορές και εξασφαλίζεται καλύτερη προσέγγιση των παραμέτρων του δικτύου.

Για κάθε pixel εκπαίδευσης, χρησιμοποιείται μία γειτονική περιοχή διαστάσεων 11x11 pixels, και με βάση αυτή την τετράγωνη περιοχή εξάγονται μικρότερες περιοχές διαφορετικών διαστάσεων, όπως έχει περιγραφεί παραπάνω. Η επιλογή της 11x11 περιοχής έγινε από τους ερευνητές εμπειρικά και μετά από δοκιμές.

Επιπλέον, χρησιμοποιείται η στοχαστική μέθοδος με βάση την κλίση (stochastic gradient descent) με μέγεθος batch 450 δείγμάτων, ροπή (momentum) 0.99 και weight decay D = 0.0001.

Αρχικά ο ρυθμός μάθησης (learning rate - L) ορίζεται 0.001 και στη συνέχεια μειώνεται $\hat{L} = L * (\frac{1}{(1+Dx\hat{L})})$, όπου \hat{L} είναι η νέα τιμή του ρυθμού μάθησης και ο αριθμός των επαναλήψεων. Όλα τα επύπεδα συνέλιξης αρχικοποιούνται βάσει μιας γκαουσιανής κατανομής με μηδενική μέση τιμή και τυπική απόκλιση $\frac{2}{f_{an_i}n + f_{an_o}out}$ όπου $f_{an_i}n$ είναι οι διαστάσεις εισόδου και $f_{an_i}n$ οι διαστάσεις εξόδου. Τα biases όλων των επιπέδων συνέλιξης αρχικοποιούνται με μηδενικά.

Στους παρακάτω πίνακες εμφανίζονται τα αποτελέσματα κάθε διακλάδωσης (με τις αντίστοιχες τοπικές περιοχές $R_L, R_R, R_U, R_B, R_C, R_G$, ως αριστερή, δεξιά, άνω, κάτω κεντρική και global) και του συνολικού μοντέλου στα datasets που χρησιμοποιήθηκαν.

	OA	Kappa coef.
$R_L(7 \times 11)$	95.01	93.9
$R_R(7 \times 11)$	95.94	95.0
$R_U(11 \times 7)$	95.74	94.8
$R_B(11 \times 7)$	94.24	93.0
$R_C(3 \times 3)$	83.09	79.5
$R_G(11 \times 11)$	96.88	96.2
DR-CNN	99.2	99.0

Πίνακας 3.10: Αποτελέσματα στο Indian Pines dataset

	OA	Kappa coef.
$R_L(7 \times 11)$	96.24	95.8
$R_R(7 \times 11)$	95.84	95.3
$R_U(11 \times 7)$	95.72	95.2
$R_B(11 \times 7)$	96.40	96.0
$R_C(3 \times 3)$	92.07	91.2
$R_G(11 \times 11)$	96.49	96.1
DR-CNN	98.22	98.0

Πίνακας 3.11: Αποτελέσματα στο Salinas dataset

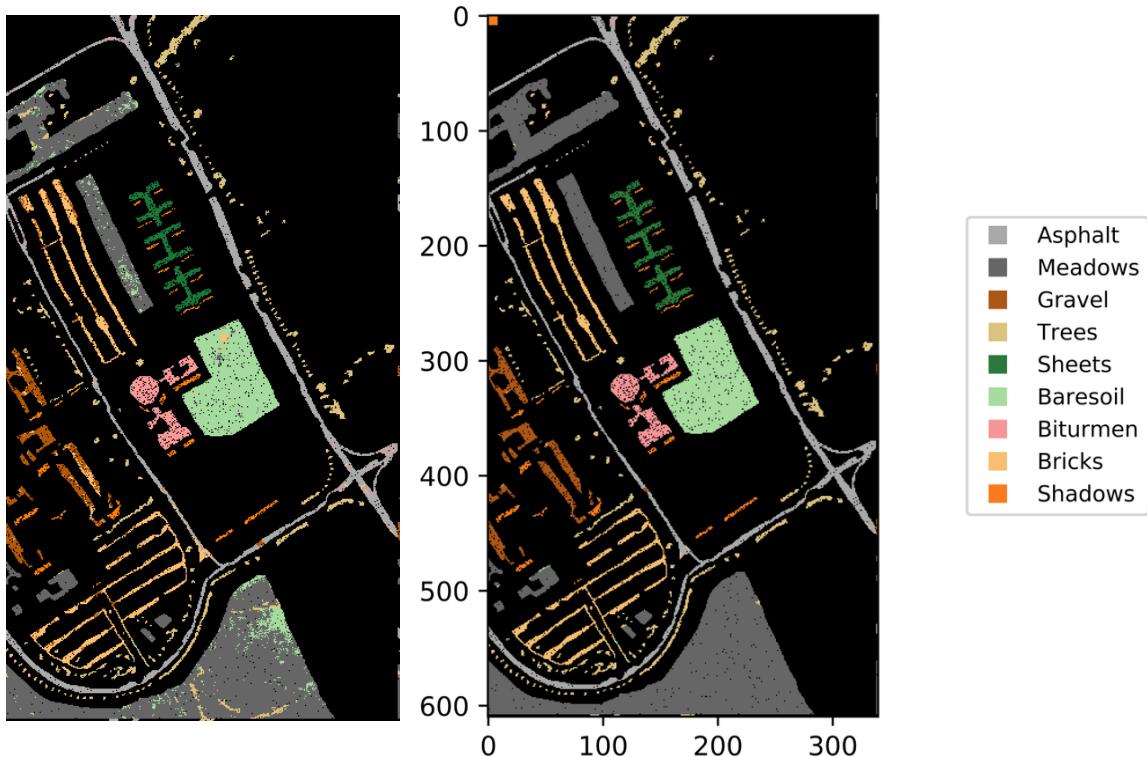
	OA	Kappa coef.
R_L	98.18	97.6
R_R	98.28	97.7
R_U	98.58	98.1
R_B	96.82	95.8
R_C	92.39	89.9
R_G	97.85	97.1
DR-CNN	99.60	99.4

Πίνακας 3.12: Αποτελέσματα στο Pavia University dataset

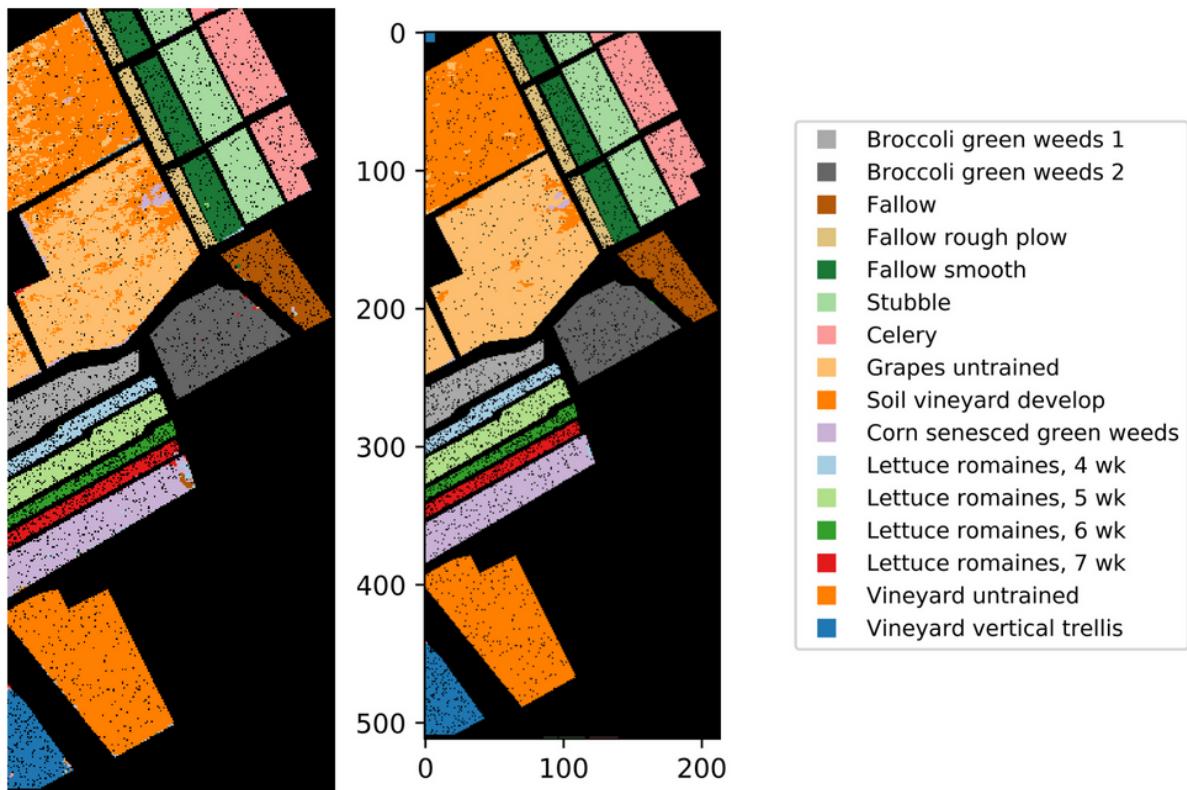
Στα παραπάνω αποτελέσματα φαίνεται ότι η κεντρική περιοχή R_C αποδίδει πολύ χαμηλή ακρίβεια σε σύγκριση με τις υπόλοιπες περιοχές. Ο κύριος λόγος που συμβαίνει αυτό είναι ότι η είσοδος απολείται από μία μικρή περιοχή που περιέχει καθαρή φασματική πληροφορία χωρίς

ουσιαστικά αξιοποίηση του χωρικού περιεχομένου των γειτονικών pixel. Στις υπόλοιπες περιπτώσεις, η ταξινόμηση δίνει καλά αποτελέσματα. Χαρακτηριστικό είναι ότι στα αποτελέσματα του Pavia University οι άνω, αριστερά και δεξιά περιοχές όταν δίνονται ως είσοδοι στο αντίστοιχο δίκτυο-διακλάδωση, δίνουν καλύτερα αποτελέσματα από την global περιοχή (11x11). Αυτό υποδηλώνει ότι οι διαφορετικές κατανομές των αντικειμένων στις επιλεγμένες περιοχές επηρεάζουν την εκπαίδευση και την ικανότητα γενίκευσης του δικτύου στην διαδικασία της ταξινόμησης. Συνολικά, το τελικό μοντέλο με τη συγχώνευση των χαρακτηριστικών, αξιοποιεί τις διαφορετικές κατανομές στις τοπικές περιοχές και τις διαφορετικές αναπαραστάσεις των δεδομένων εισόδου και δίνει μεγαλύτερη ακρίβεια στα αποτελέσματα σε σχέση με την κάθε διακλάδωση μεμονωμένα.

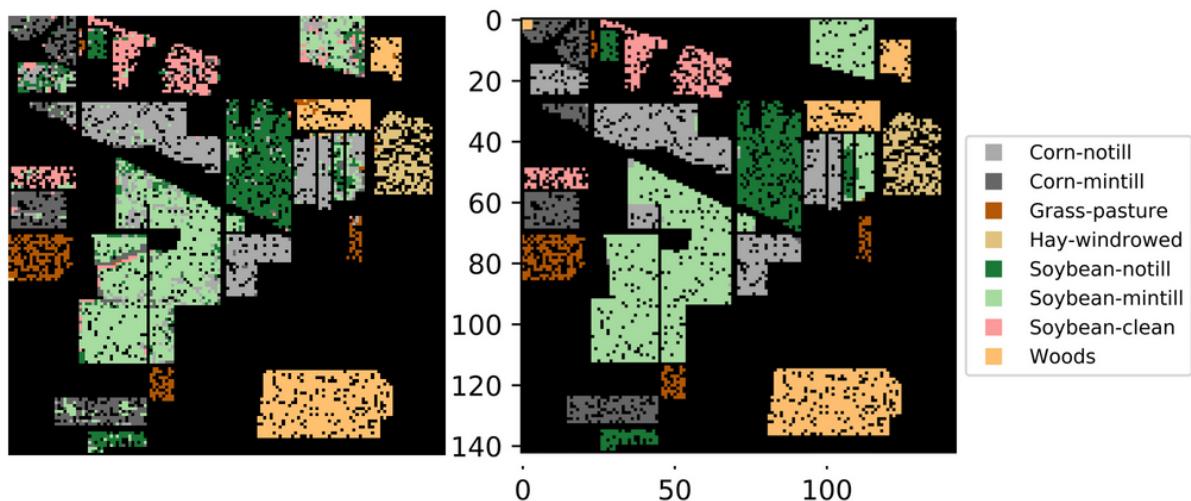
Όπως και στα αποτελέσματα του προηγούμενου μοντέλου, κάποιες παρατηρήσεις μπορούν να γίνουν και οπτικά.



Σχήμα 3.11: Το αποτέλεσμα για το Pavia University μετά την εκπαίδευση μόνο της διακλάδωσης CNN με είσοδο την κεντρική περιοχή (3x3) R_C (αριστερά) και το αποτέλεσμα του συνολικού μοντέλου DR-CNN (δεξιά).



Σχήμα 3.12: Το αποτέλεσμα για το Salinas μετά την εκπαίδευση μόνο της διακλάδωσης CNN με είσοδο την κεντρική περιοχή (3×3) R_C (αριστερά) και το αποτέλεσμα του συνολικού μοντέλου DR-CNN (δεξιά).



Σχήμα 3.13: Το αποτέλεσμα για το Indian Pines μετά την εκπαίδευση μόνο της διακλάδωσης CNN με είσοδο την κεντρική περιοχή (3×3) R_C (αριστερά) και το αποτέλεσμα του συνολικού μοντέλου DR-CNN (δεξιά).

Κατά τη μελέτη των δύο μοντέλων, θεωρήθηκε αρκετές φορές προβληματικός ο τρόπος με τον οποίο γίνεται ο έλεγχος του εκπαιδευμένου μοντέλου και ο περιορισμός που προκύπτει ως αποτέλεσμα. Συγκεκριμένα, για να είναι δυνατή η εκπαίδευση της αρχιτεκτονικής με τους stacked autoencoders στο Pavia University dataset χρησιμοποιήθηκε το 60% των συνολικών 42776 επισημασμένων pixel. Το πρώτο πρόβλημα που προκύπτει είναι ότι δεν υπάρχουν διαθέσιμες υπερφασματικές εικόνες με αυτόν τον αριθμό επισημασμένων pixel και συνεπώς ένα μοντέλο που κάνει χρήση περισσότερων των 22.000 pixel για την εκπαίδευσή του είναι αδύνατο πρακτικά να εκπαιδευτεί σε διαφορετικές σκηνές (αστικές περιοχές, δασικές εκτάσεις, καλλιέργειες, κλπ). Επιπλέον, ο έλεγχος του εκπαιδευμένου μοντέλου για τον υπολογισμό των αποτελεσμάτων ακρίβειας, γίνεται στην ίδια υπερφασματική εικόνα. Ο διαχωρισμός των δεδομένων σε σύνολα εκπαίδευσης, επαλήθευσης και ελέγχου γίνεται τυχαία. Αυτό συνεπάγεται ότι είναι πολύ πιθανό δύο διαδοχικά pixels να χρησιμοποιηθούν κατά την εκπαίδευση και τον έλεγχο αντίστοιχα. Όμως, ενώ ο σκοπός του ελέγχου είναι να γίνει εξακρίβωση της ικανότητας γενίκευσης του μοντέλου, όταν του δοθούν άγνωστα δεδομένα. Παρόλα αυτά, στην πραγματικότητα, με την επιλογή αυτής της 7x7 γειτονικής περιοχής υπάρχει επικάλυψη των τοπικών περιοχών που έχουν σχηματιστεί αρχικά για το pixel εκπαίδευσης και στη συνέχεια για το διαδοχικό pixel ελέγχου. Με άλλα λόγια, κατά τον έλεγχο, το μοντέλο τροφοδοτείται με δεδομένα που μοιάζουν ως ένα βαθμό με τα δεδομένα εκπαίδευσης. Από την άλλη, το μοντέλο DR-CNN, αν και χρησιμοποιεί πολύ λιγότερα pixel ως δεδομένα εκπαίδευσης (συγκεκριμένα για το Pavia University o stacked autoencoder χρησιμοποιεί 25550 επισημασμένα pixels, ενώ το DR-CNN 1800), χρησιμοποιεί και αυτό τοπικές περιοχές που μπορεί να εμφανίζουν επικάλυψη. Για τους παραπάνω λόγους, χρησιμοποιήθηκε το dataset με τις πολυφασματικές (όχι υπερφασματικές) εικόνες του WorldView-3. Συγκεκριμένα, χρησιμοποιήθηκε μία εικόνα του dataset για την εκπαίδευση του μοντέλου DR-CNN και μία διαφορετική εικόνα (που όμως είχε παρόμοιες κατανομές των κλάσεων) για τον έλεγχο. Ιδιαίτερα, χρησιμοποιήθηκαν οι πολυφασματικές εικόνες με τα 8 κανάλια του φάσματος από 400nm μέχρι 1040nm, καθώς επίσης απορρίφθηκαν οι κλάσεις "Waterway", "Vehicle Large" και "Vehicle Small".

Στον πίνακα 3.13 φαίνονται οι κλάσεις και ο αριθμός των επισημασμένων pixel που χρησιμοποιήθηκαν για την εκπαίδευση του μοντέλου.

Class	Labeled Pixels
Buildings	22857
Misc_Structures	10702
Road	14114
Tracks	23308
Trees	54215
Crops	578517
Waterways	-
Standing Water	10764
Vehicle large	-
Vehicle small	-

Πίνακας 3.13: Οι κλάσεις και τα επισημασμένα pixels που χρησιμοποιήθηκαν για την εκπαίδευση του μοντέλου.

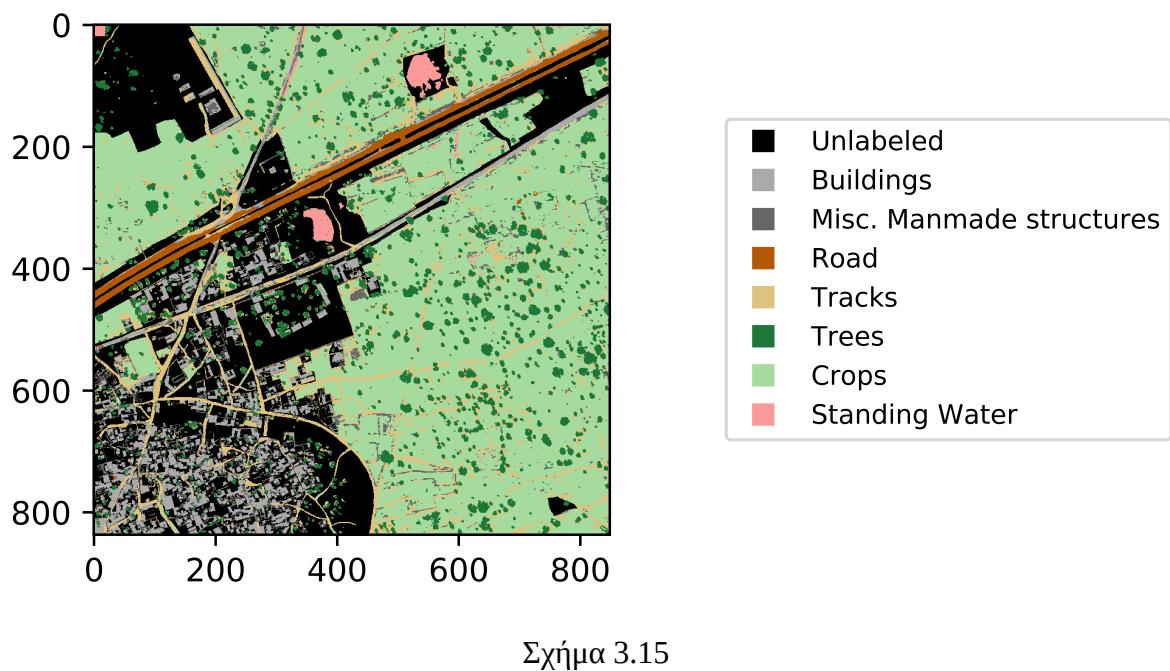
Στο σχήμα 3.14 φαίνεται (i) η εικόνα από την οποία εξάχθηκαν τα δεδομένα εκπαίδευσης του DR-CNN και (ii) η εικόνα στην οποία έγινε ο έλεγχος. Στο σχήμα 3.15 φαίνεται οπτικά το

αποτέλεσμα της ταξινόμησης στην εικόνα ελέγχου και παρακάτω στον πίνακα ?? τα αποτελέσματα για κάθε τοπική περιοχή και για το συνολικό μοντέλο.



(i) Η τοποθεσία στην οποία έγινε η εκπαίδευση του μοντέλου.
 (ii) Η τοποθεσία στην οποία έγινε ο έλεγχος του μοντέλου.

Σχήμα 3.14: Η εκπαίδευση και ο έλεγχος του μοντέλου έγινε σε διαφορετικές εικόνες, με παρόμοιες κατανομές των κλάσεων.



Σχήμα 3.15

	OA	Kappa coef.
R_L	76.70	60.4
R_R	73.16	55.8
R_U	75.55	58.5
R_B	74.73	57.7
R_C	67.25	49.6
R_G	77.77	61.7
DR-CNN	79.93	64.3

Πίνακας 3.14: Αποτελέσματα στην εικόνα ελέγχου του WorldView dataset.

3.5 Συμπεράσματα

Στην εργασία αυτή μελετήθηκαν δύο μοντέλα deep learning για την εξαγωγή "βαθιών" χαρακτηριστικών (deep features) από υπερφασματικές εικόνες και την ταξινόμησή τους σε κλάσεις εδαφοκάλυψης. Φάνηκε ότι η αξιοποίηση από κοινού της χωρικής και φασματικής πληροφορίας από το δίκτυο, δίνει αποτελέσματα μεγαλύτερης ακρίβειας στην ταξινόμηση. Είναι χαρακτηριστικό ότι όταν έγινε αξιοποίηση της χωρικής πληροφορίας, με την εξαγωγή μιας γειτονικής περιοχής, ακόμα και μετά την εφαρμογή του PCA για τη μείωση των διαστάσεων του φασματικού περιοχομένου, το μοντέλο πέτυχε υψηλή ακρίβεια. Αυτό υποδεικνύει ότι υπάρχει πλεονάζουσα φασματική πληροφορία και είναι πιθανόν να μπορούσαν είτε να απορριφθούν συγκεκριμένες ζώνες συχνοτήτων εξ' ολοκλήρου, ή όπως συνέβει στην παραπάνω περίπτωση να υπόκεινται σε μία διαδικασία μείωσης διαστάσεων. Σημαντικός, επίσης, φαίνεται ότι είναι ο τρόπος με τον οποίο επιλέγεται η γειτονική περιοχή γύρω από το pixel ενδιαφέροντος. Όπως παρουσιάστηκε και επιβεβαιώνουν τα πειραματικά αποτελέσματα, γειτονικές περιοχές με διαφορετικές διαστάσεις ή προσανατολισμό μπορεί να έχουν διαφορετικές κατανομές των αντικειμένων, ειδικά σε εξαιρετικά ετερογενείς (από σκοπιά εδαφοκάλυψης) εικόνες. Η συγχώνευση των αναπαραστάσεων που μαθαίνει το μοντέλο από κάθε τέτοια γειτονική περιοχή οδηγεί σε αποτελέσματα μεγαλύτερης ακρίβειας.

Συνεπώς, είναι ασφαλές να ειπωθεί ότι επόμενες εργασίες και βαθιές αρχιτεκτονικές που θα αναπτυχθούν θα πρέπει να αναζητήσουν ειδικότερα τον τρόπο με τον οποίο επιλέγονται κατάλληλες γειτονικές περιοχές.

Τα datasets που χρησιμοποιήθηκαν στα πλαίσια της εργασίας είναι τα πιο δημοφιλή καθώς χρησιμοποιούνται σε κάθε εργασία ως μέτρο σύγκρισης με προηγούμενες εργασίες. Τέλος, είναι πολύ σημαντικό να αναπτυχθούν νέα datasets, που να καλύπτουν ποικιλία διαφορετικών τοποθεσιών (αστικό περιβάλλον, καλλιέργειες, δασικές εκτάσεις, κ.ά) και να διαθέτουν περισσότερες κατηγορίες εδαφοκάλυψης.

Bibliography

- [BAC17] John Ball, Derek Anderson, and Chee Seng Chan. “A Comprehensive Survey of Deep Learning in Remote Sensing: Theories, Tools and Challenges for the Community”. In: 11 (Aug. 2017).
- [Che+14] Yushi Chen et al. “Deep learning-based classification of hyperspectral data”. In: *IEEE Journal of Selected topics in applied earth observations and remote sensing* 7.6 (2014), pp. 2094–2107.
- [Cnn] *Applied Deep Learning: CNN*. 2017. URL: <https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2>.
- [Cyb89] G. Cybenko. “Approximation by superpositions of a sigmoidal function”. In: *Mathematics of Control, Signals and Systems* 2.4 (1989), pp. 303–314. ISSN: 1435-568X. DOI: 10.1007/BF02551274. URL: <https://doi.org/10.1007/BF02551274>.
- [GBB11] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. “Deep Sparse Rectifier Neural Networks”. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Geoffrey Gordon, David Dunson, and Miroslav Dudík. Vol. 15. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR, 2011, pp. 315–323. URL: <http://proceedings.mlr.press/v15/glorot11a.html>.
- [Gol16] Peter Goldsborough. “A Tour of TensorFlow”. In: (Oct. 2016).
- [Hin] Geoffrey E. Hinton. *Rectified Linear Units Improve Restricted Boltzmann Machines* Vinod Nair.
- [HS06] Geoffrey E Hinton and Ruslan R Salakhutdinov. “Reducing the dimensionality of data with neural networks”. In: *science* 313.5786 (2006), pp. 504–507.
- [KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. NIPS’12. Lake Tahoe, Nevada: Curran Associates Inc., 2012, pp. 1097–1105. URL: <http://dl.acm.org/citation.cfm?id=2999134.2999257>.
- [KW52] Joe Kiefer and J Wolfowitz. “Stochastic Estimation of the Maximum of A Regression Function”. In: 23 (Sept. 1952).
- [LeC+99] Yann LeCun et al. “Object Recognition with Gradient-Based Learning”. In: *Shape, Contour and Grouping in Computer Vision*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 319–345. ISBN: 978-3-540-46805-9. DOI: 10.1007/3-540-46805-6_19. URL: https://doi.org/10.1007/3-540-46805-6_19.

- [MLY17] Seonwoo Min, Byunghan Lee, and Sungroh Yoon. “Deep learning in bioinformatics”. In: *Briefings in Bioinformatics* 18.5 (2017), pp. 851–869. DOI: 10.1093/bib/bbw068. eprint: /oup/backfile/content_public/journal/bib/18/5/10.1093_bib_bbw068/2/bbw068.pdf. URL: <http://dx.doi.org/10.1093/bib/bbw068>.
- [MP43] Warren S. McCulloch and Walter Pitts. “A logical calculus of the ideas immanent in nervous activity”. In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133. ISSN: 1522-9602. DOI: 10.1007/BF02478259. URL: <https://doi.org/10.1007/BF02478259>.
- [Nnd] *Neural Networks and Deep Learning*. URL: <https://www.nrcan.gc.ca/node/9309>.
- [Nrc] *Fundamentals of Remote Sensing*. URL: <https://www.nrcan.gc.ca/node/9309>.
- [Qia99] Ning Qian. “On the Momentum Term in Gradient Descent Learning Algorithms”. In: *Neural Netw.* 12.1 (Jan. 1999), pp. 145–151. ISSN: 0893-6080. DOI: 10.1016/S0893-6080(98)00116-6. URL: [http://dx.doi.org/10.1016/S0893-6080\(98\)00116-6](http://dx.doi.org/10.1016/S0893-6080(98)00116-6).
- [RHW88] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. “Neurocomputing: Foundations of Research”. In: ed. by James A. Anderson and Edward Rosenfeld. Cambridge, MA, USA: MIT Press, 1988. Chap. Learning Representations by Back-propagating Errors, pp. 696–699. ISBN: 0-262-01097-6. URL: <http://dl.acm.org/citation.cfm?id=65669.104451>.
- [Ros58] F. Rosenblatt. “The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain”. In: *Psychological Review* (1958), pp. 65–386.
- [TDT+16] The Theano Development Team et al. “Theano: A Python framework for fast computation of mathematical expressions”. In: (May 2016).
- [Vin+08] Pascal Vincent et al. “Extracting and composing robust features with denoising autoencoders”. In: *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1096–1103.
- [ZF14] Matthew D Zeiler and Rob Fergus. “Visualizing and understanding convolutional networks”. In: *European conference on computer vision*. Springer, 2014, pp. 818–833.
- [ZLD18] Mengmeng Zhang, Wei Li, and Qian Du. “Diverse Region-Based CNN for Hyperspectral Image Classification”. In: *IEEE Transactions on Image Processing* 27.6 (2018), pp. 2623–2634.