



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΠΕΛΟΠΟΝΝΗΣΟΥ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ  
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

**Πτυχιακή Εργασία**

**Θέμα: Deep learning – Αλγόριθμοι και καινοτομικές  
ε-φαρμογές της τεχνολογίας των βαθέων νευρωνικών  
δικτύων**

των φοιτητών

Αργυρόπουλου Ηλία (ΑΜ: 2584)

Ελληνικού Σωκράτη (ΑΜ: 2550)

Επιβλέπων: Β. Ταμπακάς, Γ. Ε. Λιβιέρης

Ακαδημαϊκό έτος: 2018 - 2019

## Περιεχόμενα

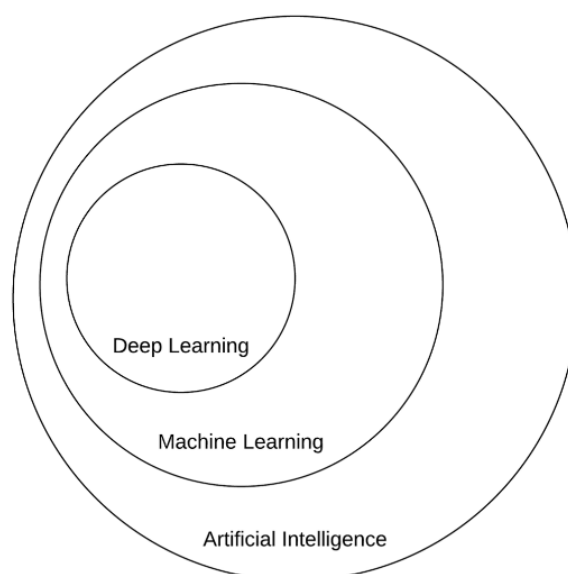
<b>1</b>	<b>Deep Learning και Υπολογιστικό νέφος</b>	3
1.1	Τι είναι το Deep Learning	3
1.2	Υπηρεσίες υπολογιστικού νέφους	5
1.3	Μοντέλα υπολογιστικού νέφους	6
<b>2</b>	<b>Νευρωνικά Δίκτυα (Neural Networks)</b>	9
2.1	Τεχνητά Μοντέλα (Artificial Models)	10
2.2	Μοντέλο ενεργοποίησης νευρώνα	12
2.2.1	Perceptron	12
2.3	Εκπαίδευση	18
2.3.1	Βασικές αρχές εκμάθησης	18
2.3.2	Αλγόριθμος οπισθοδρομικής διάδοσης του σφάλματος	18
2.3.3	Τεχνικές για βελτίωση της απόδοσης των νευρωνικών δικτύων	20
<b>3</b>	<b>Συνελκτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks)</b>	24
3.1	Επίπεδο συνέλιξης	26
3.2	Επίπεδο συγκέντρωσης	29
3.3	Μη γραμμικότητα	32
3.4	Dropout	33
3.5	Κανονικοποίηση	34
3.6	Προ-εκπαίδευση	35
3.7	Πλήρως διασυνδεδεμένο επίπεδο	36
3.8	Εκπαίδευση	36
<b>4</b>	<b>Μια εφαρμογή της μηχανικής μάθησης στη αιματολογική διάγνωση</b>	38
4.1	Εκπαίδευση	38
<b>5</b>	<b>Βιβλιογραφία</b>	39

# 1 Deep Learning και Υπολογιστικό νέφος

## 1.1 Τι είναι το Deep Learning

Οι μέθοδοι Deep Learning αποτελούν αντιπροσωπευτικές μεθόδους μάθησης με διαφορετικά επίπεδα αναπαράστασης τα οποία προκύπτουν συνθέτοντας απλά αλλά μη γραμμικά τμήματα (modules) καθένα από τα οποία μετασχηματίζει την αναπαράσταση σε ένα επίπεδο (ξεκινώντας από την είσοδο) σε μία αναπαράσταση ενός υψηλότερου και ελαφρώς πιο αφηρημένου επιπέδου. Η βασική πτυχή του Deep Learning είναι ότι αυτά τα επίπεδα σχεδιάζονται όχι από μηχανικούς: προκύπτουν από δεδομένα χρησιμοποιώντας μια γενική διαδικασία εκμάθησης (Y. LeCun, 2015).

Το Deep Learning αποτελεί υποπεδίο του machine learning το οποίο με τη σειρά του είναι υποπεδίο της τεχνητής νοημοσύνης. Μια γραφική αναπαράσταση που συνδέει τις προηγούμενες έννοιες φαίνεται στην επόμενη εικόνα:



*Εικόνα 1: Διάγραμμα Venn που περιγράφει το Deep Learning ως ένα υπο-πεδίο του machine learning το οποίο με τη σειρά του είναι ένα υποπεδίο της Τεχνητής Νοημοσύνης*

Βασικός στόχος της τεχνητής νοημοσύνης είναι να παρέχει ένα σύνολο αλγορίθμων και τεχνικών που μπορούν να χρησιμοποιηθούν για την επίλυση προβλημάτων τα οποία οι άνθρωποι μπορούν να λύσουν διαισθητικά και σχεδόν αυτοματοποιημένα αλλά από την άλλη πλευρά είναι δύσκολο για τους υπολογιστές. Ένα βασικό παράδειγμα τεχνητής νοημοσύνης είναι η ερμηνεία και η κατανόηση των περιεχομένων μιας

εικόνας. Αυτή η λειτουργία είναι κάτι που ο άνθρωπος μπορεί να κάνει πολύ εύκολα , ενώ αυτό έχει αποδειχθεί ότι είναι εξαιρετικά δύσκολο για τον υπολογιστή.

Ενώ η τεχνητή νοημοσύνη ενσωματώνει έναν μεγάλο όγκο δουλειάς για την περιγραφή των εικόνων, το υποπεδίο machine learning ενδιαφέρεται ιδιαίτερα για αναγνώριση προτύπων (pattern recognition) και για εκμάθηση από τα δεδομένα (learning from data).

Τα τεχνητά νευρωνικά δίκτυα είναι μια κατηγορία αλγορίθμων machine learning οι οποίοι «μαθαίνουν» από τα δεδομένα και ειδικεύονται στην αναγνώριση προτύπων και εμπνέονται από τη δομή και τη λειτουργία του εγκεφάλου. Το Deep Learning ανήκει στη κατηγορία των αλγορίθμων για τεχνητά νευρωνικά δίκτυα και στις περισσότερες περιπτώσεις οι δυο όροι χρησιμοποιούνται εναλλακτικά. Ίσως προκαλεί εντύπωση ότι το πεδίο του Deep Learning χρησιμοποιείται για περισσότερα από 60 χρόνια με διαφορετικά ονόματα ανάλογα με τις ερευνητικές μεθόδους που εφαρμόζονταν κάθε φορά και με βάση το διαθέσιμο υλικό και τα datasets που χρησιμοποιήθηκαν.

## 1.2 Υπηρεσίες υπολογιστικού νέφους

Το υπολογιστικό νέφος είναι μια τεχνολογία που χρησιμοποιεί το Διαδίκτυο και κεντρικούς απομακρυσμένους διακομιστές για τη διατήρηση των δεδομένων και εφαρμογών. Το υπολογιστικό νέφος επιτρέπει στους καταναλωτές και τις επιχειρήσεις να χρησιμοποιούν εφαρμογές χωρίς τη προγενέστερη εγκατάστασή τους και να έχουν πρόσβαση στα προσωπικά τους αρχεία σε οποιοδήποτε υπολογιστή που έχει πρόσβαση στο διαδίκτυο. Αυτή η τεχνολογία επιτρέπει μία πιο αποτελεσματική διαχείριση με κεντροποίηση της αποθήκευσης των δεδομένων της επεξεργασίας και του εύρους ζώνης. Με τον όρο υπολογιστικό νέφος εννοούμε την χρήση υπολογιστικών πόρων (τόσο υλικού όσο και λογισμικού), οι οποίοι παραδίδονται ως μία υπηρεσία μέσου δικτύου (συνήθως μέσου του Διαδικτύου). Το όρος «υπολογιστικό νέφος» προέρχεται από τη χρήση ενός συμβόλου που έχει τη μορφή νέφους (σύννεφο) ως μία αφαιρετική περιγραφή της πολύπλοκης υποδομής που περιέχεται στα διαγράμματα συστημάτων. Το υπολογιστικό νέφος αναθέτει απομακρυσμένες υπηρεσίες στα δεδομένα του χρήστη.

Σύμφωνα με το *Εθνικό Ινστιτούτο Προτύπων και Τεχνολογίας (NIST – National Institute of Standards and Technology)*, υπολογιστικό νέφος είναι ένα μοντέλο που καθιστά εφικτή την κατ' απαίτηση πρόσβαση σε ένα δίκτυο και στους υπολογιστικούς πόρους του. Ο συγκεκριμένος ορισμός απαριθμεί πέντε βασικά χαρακτηριστικά που πρέπει να διαθέτει το υπολογιστικό νέφος: ευρεία πρόσβαση στο δίκτυο, συγκέντρωση πόρων, κατ' απαίτηση εξυπηρέτηση, επέκταση και χρήση υπηρεσιών μέχρι ενός βαθμού. Το νέφος παρέχει υπηρεσίες, οι οποίες μπορούν να ομαδοποιηθούν στις ακόλουθες κατηγορίες:

- *IaaS (Infrastructure as a Service)*: Οι υπολογιστικοί πόροι του υλικού (όπως π.χ. η αποθήκευση) και η υπολογιστική ισχύς (CPU και μνήμη) προσφέρονται ως υπηρεσίες στους πελάτες. Αυτό επιτρέπει σε επιχειρήσεις να νοικιάσουν αυτούς τους πόρους, αντί να δαπανούν χρήματα για να αγοράσουν αποκλειστικούς διακομιστές και

δικτυακό εξοπλισμό. Ως παράδειγμα στην κατηγορία αυτή το Amazon προσφέρει χωρητικότητα S3<sup>1</sup>, υπολογιστική ισχύ EC2<sup>2</sup> και διαδικτυακή επικοινωνία SQS<sup>3</sup> για μικρές επιχειρήσεις και μεμονωμένους καταναλωτές.

- *SaaS (Software as a Service)*: Σε αυτό το μοντέλο οι εφαρμογές λογισμικού προσφέρονται ως υπηρεσίες στο διαδίκτυο και όχι ως πακέτα λογισμικού που θα αγοράζονται από μεμονωμένους πελάτες. Ένας από τους πρωτοπόρους παρόχους σ' αυτήν την υπηρεσία είναι Salesforce.com που προσφέρει εφαρμογές CRM (Customer Relationship Management) ως υπηρεσία. Άλλα παραδείγματα σ' αυτήν την κατηγορία περιλαμβάνουν web - based εφαρμογές γραφείου.

- *PaaS (Platform as a Service)*: Αυτό αναφέρεται στην παροχή διευκολύνσεων για την υποστήριξη της ανάπτυξης ολοκληρωμένων εφαρμογών. Πολύ συχνά χρησιμοποιούνται οι φυλλομέτρησες ιστού ως περιβάλλον ανάπτυξης. Παραδείγματα πλατφορμών αυτής της κατηγορίας είναι το Microsoft Azure Services platform, Google App Engine κ.λ.π.

### 1.3 Μοντέλα υπολογιστικού νέφους

Τα μοντέλα του υπολογιστικού νέφους είναι τα εξής:

- *Private Cloud*: Η υποδομή του υπολογιστικού νέφους χρησιμοποιείται για την αποκλειστική χρήση από έναν οργανισμό που περιέχει πολλαπλούς καταναλωτές. Μπορεί να είναι ιδιοκτησία του ίδιου του οργανισμού ή κάποιου τρίτου μέρους. Επίσης μπορεί να ισχύει και κάποιος συνδυασμός, με την έννοια ότι μπορεί κάποιο μέρος του εξοπλισμού να ανήκει στην ίδια την επιχείρηση και κάποιο άλλο μέρος σε άλλες επιχειρήσεις.

---

<sup>1</sup> Η υπηρεσία Amazon Simple Storage Service παρέχει στους προγραμματιστές ασφαλή, ανθεκτική και επεκτάσιμη αποθήκευση αντικειμένων και είναι πολύ εύκολη στη χρήση της διότι με ένα απλό περιβάλλον διαδικτυακών υπηρεσιών είναι εφικτή η αποθήκευση και η ανάκτηση οποιουδήποτε ποσού δεδομένων από οπουδήποτε στο διαδίκτυο.

<sup>2</sup> Η υπηρεσία Amazon Elastic Compute Cloud (EC2) είναι μία υπηρεσία βασισμένη στον παγκόσμιο ιστό που παρέχει τη δυνατότητα μεταβαλλόμενης υπολογιστικής χωρητικότητας στο νέφος και έχει σχεδιαστεί για να κάνει το επεκτάσιμο υπολογιστικό νέφος ευκολότερο για τους προγραμματιστές.

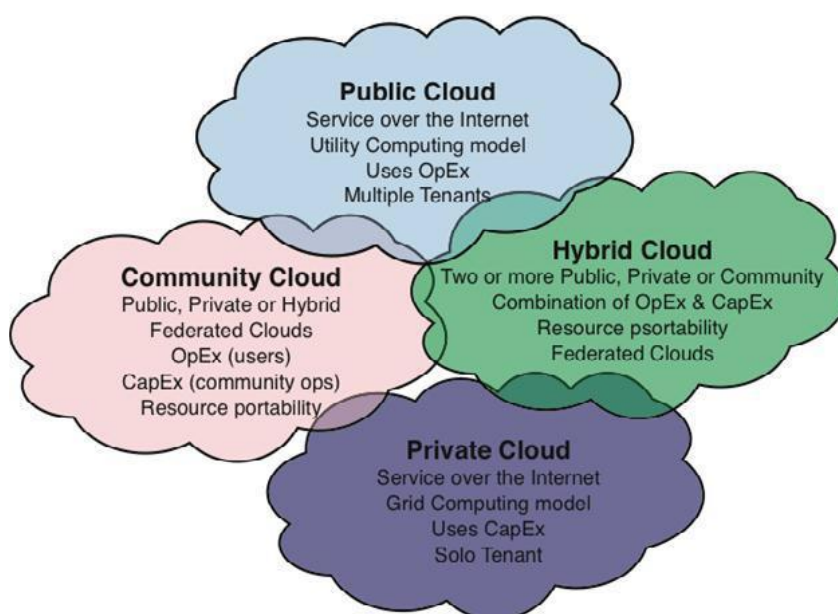
<sup>3</sup> Η υπηρεσία Amazon Simple Queue Service (SQS) είναι μία γρήγορη, αξιόπιστη, επεκτάσιμη υπηρεσία τοποθέτησης μηνυμάτων σε ουρά αναμονής και χρησιμοποιείται για τη μετάδοση οποιουδήποτε όγκου δεδομένων χωρίς την απώλεια μηνυμάτων ή την απαίτηση διαθεσιμότητας άλλων υπηρεσιών.

- *Public Cloud*: Στην περίπτωση αυτή οι εφαρμογές του δημόσιου νέφους, η αποθήκευση και άλλοι πόροι διατίθενται στο ευρύ κοινό μέσω ενός Παρόχου υπηρεσιών. Αυτές οι υπηρεσίες προσφέρονται είτε δωρεάν είτε σε ένα μοντέλο πληρωμή ανά χρήση (Pay per Use). Οι εφαρμογές στα δημόσια νέφη παρέχονται από διαφορετικούς και ανεξάρτητους διακομιστές, συστήματα αποθήκευσης και δίκτυα. Σε γενικές γραμμές οι πάροχοι δημόσιων υπηρεσιών Cloud, όπως Amazon AWS διατηρούν και εκμεταλλεύονται την υποδομή και παρέχουν προσπέλαση μόνο μέσω του διαδικτύου (δεν παρέχεται άμεση συνδεσιμότητα).

- *Community Cloud*: Οι κοινοτικές υποδομές υπολογιστικού νέφους διανέμουν την υποδομή μεταξύ διαφορετικών οργανισμών από μία συγκεκριμένη κοινότητα με κοινές ανησυχίες (όπως π.χ. ασφάλεια, συμμόρφωση κλπ.). Στην περίπτωση αυτή οι δαπάνες κατανέμονται σε λιγότερους χρήστες απ' ό,τι σε ένα δημόσιο νέφος, έτσι ώστε να πραγματοποιούνται μόνο μερικές από τις δυνατότητες μείωσης του υπολογιστικού νέφους.

- *Hybrid Cloud*: Το υβριδικό νέφος είναι μία σύνθεση από δύο ή περισσότερα σύννεφα (ιδιωτικά, κοινοτικά ή δημόσια) που παραμένουν ως μοναδικές οντότητες, αλλά ταυτόχρονα συνδέονται μεταξύ τους, προσφέροντας έτσι τα πλεονεκτήματα των πολλαπλών μοντέλων.

Η εικόνα που ακολουθεί, περιγράφει τα μοντέλα του υπολογιστικού νέφους που περιγράφηκαν παραπάνω (US-CERT, 2012):



*Εικόνα 2: Μοντέλα Cloud Computing*

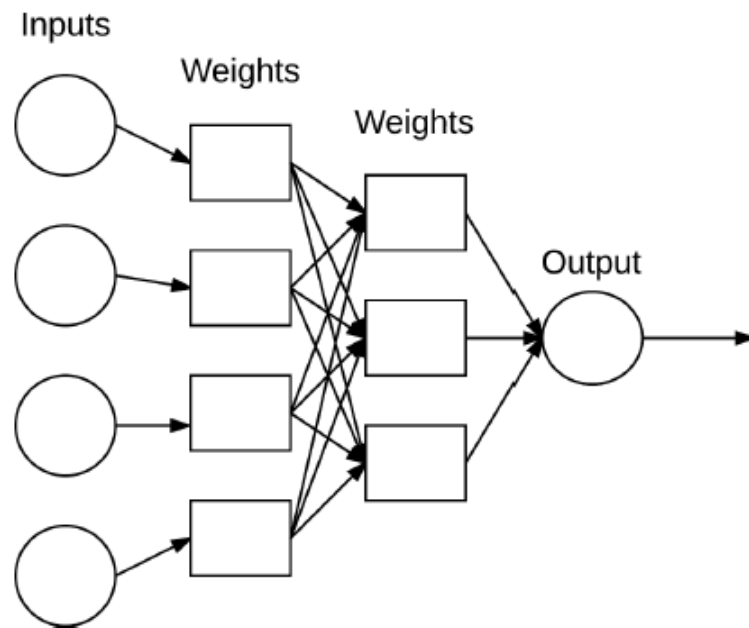


## 2 Νευρωνικά Δίκτυα (Neural Networks)

Πριν ασχοληθούμε με τα συνελκτικά νευρωνικά δίκτυα πρέπει πρώτα να κατανοήσουμε τις βασικές αρχές των νευρωνικών δικτύων. Τα νευρωνικά δίκτυα αποτελούν τα δομικά στοιχεία των συστημάτων deep learning. Προκειμένου αυτά να έχουν σωστή εφαρμογή στο χώρο του deep learning θα πρέπει πρώτα να περιγράψουμε τα βασικά στοιχεία των νευρωνικών δικτύων συμπεριλαμβανομένων της αρχιτεκτονικής τους, του τύπου των κόμβων τους και των αλγορίθμων για τη ‘μάθηση’ των δικτύων.

Πολλά έργα που αφορούν νοημοσύνη, αναγνώριση προτύπων και ανίχνευση αντικειμένων είναι εξαιρετικά δύσκολο να αυτοματοποιηθούν, αλλά εκτελούνται ιδιαίτερα εύκολα και φυσικά από ζώα και μικρά παιδιά. Για παράδειγμα ένα μικρό παιδί μπορεί εύκολα να ξεχωρίζει τη διαφορά ανάμεσα σε ένα σχολικό λεωφορείο και σε ένα λεωφορείο της γραμμής, ενώ αντίστοιχα ένας φίλος μπορεί εύκολα να αναγνωρίσει τον ιδιοκτήτη του από έναν ξένο. Για το πώς γίνεται αυτό η απάντηση βρίσκεται μέσα στο σώμα μας. Κάθε άνθρωπος διαθέτει ένα πραγματικό δίκτυο νευρώνων το οποίο συνδέεται με το νευρικό του σύστημα. Το δίκτυο αυτό αποτελείται από έναν μεγάλο αριθμό διασυνδεδεμένων νευρώνων (νευρικών κυττάρων). Ως εκ τούτου ένα τεχνητό νευρωνικό δίκτυο είναι ένα υπολογιστικό σύστημα που προσπαθεί να μιμηθεί (ή τουλάχιστον εμπνέεται από) τις νευρικές συνδέσεις στο νευρικό μας σύστημα. Τα τεχνητά νευρωνικά δίκτυα αναφέρονται σε συντομογραφία ως ANN (Artificial Neural Networks).

Για να θεωρηθεί ένα σύστημα ως νευρωνικό δίκτυο πρέπει να έχει μια δομή κατευθυνόμενου γραφήματος όπου κάθε κόμβος στο γράφημα εκτελεί μερικούς απλούς υπολογισμούς. Στην επόμενη εικόνα παρουσιάζουμε την αρχιτεκτονική ενός απλού νευρωνικού δικτύου (A. Dar, 2018).

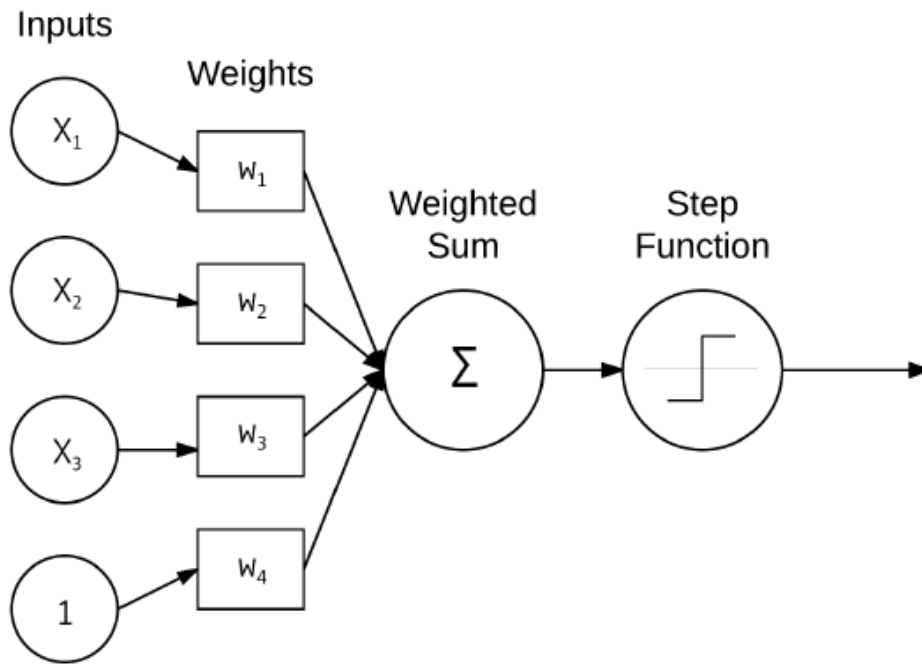


Εικόνα 3: Αρχιτεκτονική ενός απλού νευρωνικού δικτύου. Στο δίκτυο υπάρχουν εισοδοί (inputs). Κάθε σύνδεση μεταφέρει ένα σήμα μέσω των δύο κρυμμένων επιπέδων στο δίκτυο. Μια τελική συνάρτηση υπολογίζει την ετικέτα κλάσης εξόδου

Κάθε κόμβος εκτελεί έναν απλό υπολογισμό. Κάθε σύνδεσμος μεταφέρει ένα σήμα (δηλαδή το αποτέλεσμα ενός υπολογισμού) από έναν κόμβο στον επόμενο και διαθέτει ένα βάρος που υποδεικνύει την έκταση στην οποία το σχήμα ή επεκτείνεται ή μειώνεται. Ορισμένες συνδέσεις διαθέτουν μεγάλα θετικά βάρη τα οποία ενισχύουν το σήμα υποδεικνύοντας ότι το σήμα είναι πολύ σημαντικό όταν κάνουμε μία ταξινόμηση. Άλλες συνδέσεις έχουν αρνητικά βάρη τα οποία μειώνουν την ισχύ του σήματος διευκρινίζοντας έτσι ότι η έξοδος του κόμβου είναι λιγότερο σημαντική στην τελική ταξινόμηση. Ονομάζουμε ένα τέτοιο σύστημα ως τεχνητό νευρωνικό δίκτυο αν αποτελείται από μια δομή γραφήματος (όπως η προηγούμενη εικόνα) με βάρη σύνδεσης τα οποία είναι τροποποιήσιμα χρησιμοποιώντας έναν αλγόριθμο μάθησης.

## 2.1 Τεχνητά Μοντέλα (Artificial Models)

Ας δούμε ένα βασικό NN που εκτελεί ένα απλό σταθμισμένο άθροισμα των εισόδων του όπως φαίνεται στην επόμενη εικόνα. Οι τιμές  $x_1$ ,  $x_2$  και  $x_2$  είναι οι εισοδοί στο NN και τυπικά αντιστοιχούν σε μια απλή ροή (δηλ. σημείο δεδομένων) από τον πίνακα σχεδίασης. Μπορούμε να θεωρήσουμε αυτές τις εισόδους ως διανύσματα χαρακτηριστικών εισόδου στο NN



Εικόνα 4: Ένα απλό νευρωνικό δίκτυο που παίρνει το σταθμισμένο άθροισμα της εισόδου  $x$  και των βαρών  $w$ . Αυτό το σταθμισμένο άθροισμα μεταβιβάζεται στη συνέχεια μέσω της συνάρτησης ενεργοποίησης για να αποφασιστεί αν το νεύρο ενεργοποιείται

Στην πράξη αυτές οι εισοδοι θα μπορούσαν να είναι διανύσματα τα οποία ποσοτικοποιούν τα περιεχόμενα μιας εικόνας με ένα συστηματικό προκαθορισμένο τρόπο (π.χ., ιστογράμματα χρώματος, ιστογράμματα προκαθορισμένων βαθμίδων κλπ.).

## 2.2 Μοντέλο ενεργοποίησης νευρώνα

Όπως αναφέρθηκε παραπάνω η ενεργοποίηση του κάθε νευρώνα γίνεται με μία μη γραμμική συνάρτηση  $h$ , η οποία δέχεται το άθροισμα του ηλεκτρικού φορτίου που δέχεται ο νευρώνας, εφαρμόζει τη μη γραμμικότητα και δίνει αποτέλεσμα την απόκριση του νευρώνα.

### 2.2.1 Perceptron

Η πρώτη και πιο απλή συνάρτηση που χρησιμοποιήθηκε ονομάζεται μοντέλο perceptron που στην ουσία είναι ένας δυαδικός ταξινομητής. Χρησιμοποιείται κυρίως στον διαχωρισμό των δεδομένων εισόδου σε 2 επιμέρους κατηγορίες. Στην περίπτωση που οι 2 αυτές κατηγορίες έχουν τιμές 0 και 1, τότε η συνάρτηση κατωφλίου ονομάζεται και βηματική συνάρτηση (step function) ή αντίστοιχα ως συνάρτηση πρόσημού (signum function) όταν οι τιμές είναι -1 και 1.

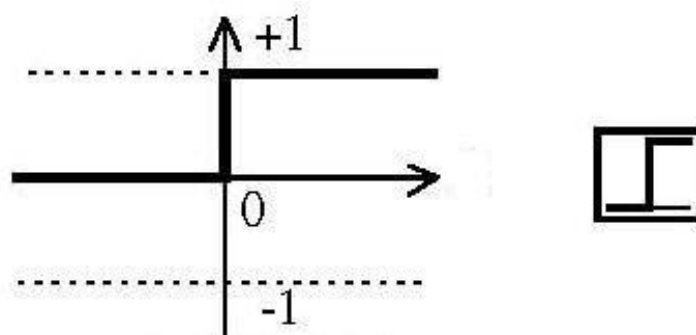
$$f(x) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

#### 2.2.1.1 Βηματική συνάρτηση ενεργοποίησης

Η βηματική συνάρτηση (Hard-Limit Transfer Function) ενεργοποίησης μπορεί να περιγραφεί μαθηματικά ως εξής:

$$\varphi(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

Η έξοδος του νευρώνα ισούται με 0 αν το *Output* είναι μικρότερο από 0 και με 1 αν το *Output* είναι μεγαλύτερο ή ίσο του 0.

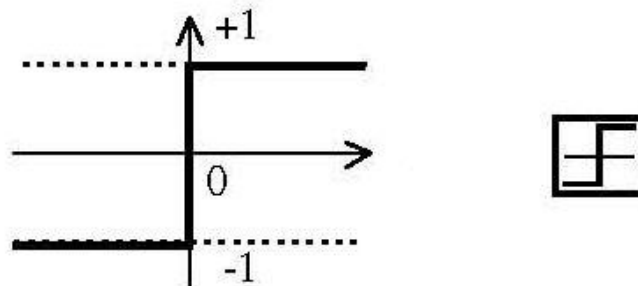


Εικόνα 5: Βηματική Συνάρτηση Ενεργοποίησης

### 2.2.1.2 Συμμετρική βηματική συνάρτηση ενεργοποίησης

Η συμμετρική βηματική συνάρτηση (Symmetric Hard Transfer Function) έχει παρόμοια μορφή με την απλή βηματική συνάρτηση με τη διαφορά ότι όταν το *Output* είναι μικρότερο του 0, η έξοδος του νευρώνα είναι -1.

$$\varphi(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases}$$

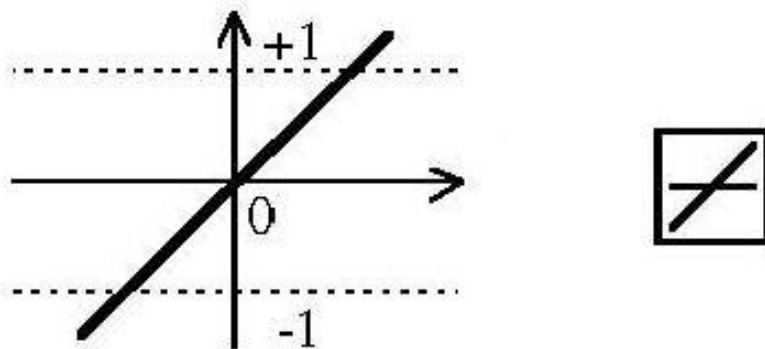


Εικόνα 6: Συμμετρική Βηματική Συνάρτηση Ενεργοποίησης

Οι βηματικές συναρτήσεις παρόλο που προσομοιάζουν πολύ καλά τη λειτουργία που θέλουμε να εκτελεί μια συνάρτηση ενεργοποίησης δε θεωρούνται χρήσιμες για τέτοιο ρόλο στα NN καθώς σύμφωνα με τον απειροστικό λογισμό έχουν το βασικό μειονέκτημα ότι η παράγωγός τους απειρίζεται. Έτσι προέκυψε η ανάγκη συναρτήσεων ενεργοποίησης που η γραφική τους παράσταση να μοιάζει με τη βηματική, αλλά ταυτόχρονα να είναι συνεχείς και παραγωγίσιμες σε όλο το πεδίο ορισμού τους.

### 2.2.1.3 Γραμμική συνάρτηση ενεργοποίησης

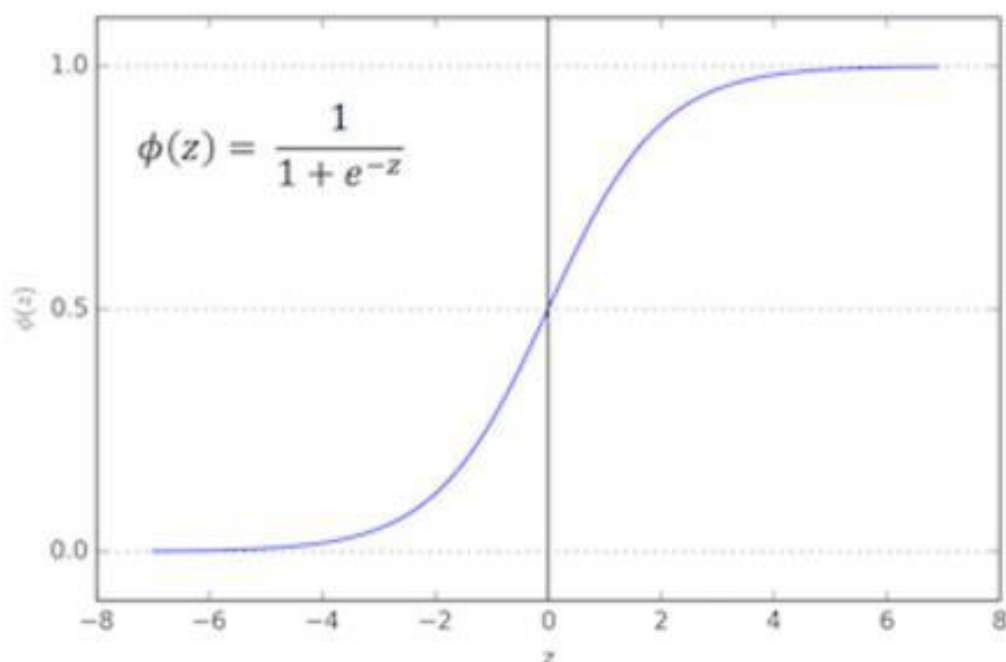
Μια γραμμική συνάρτηση ενεργοποίησης μπορεί να έχει τη μορφή  $\varphi(x)=cx$



Εικόνα 7: Γραμμική Συνάρτηση Ενεργοποίησης

#### 2.2.1.4 Σιγμοειδής συνάρτηση ενεργοποίησης

Πρόκειται για την πιο ευρέως χρησιμοποιούμενη ιστορικά συνάρτηση, καθώς ερμηνεύεται άμεσα ως ρυθμός πυροδότησης ενός νευρώνα. Δέχεται κάθε πραγματικό αριθμό και τον κανονικοποιεί στο διάστημα  $[0,1]$ . Με άλλα λόγια, μεγάλοι αρνητικοί αριθμοί γίνονται 0, ενώ μεγάλοι θετικοί γίνονται 1. Τα τελευταία χρόνια, όμως, η χρήση της έχει περιοριστεί αφού διαθέτει δυο βασικά μειονεκτήματα: Για πολύ μεγάλες ή μικρές τιμές η κλίση της γίνεται πρακτικά μηδενική, κάτι που οδηγεί στην παύση της εκπαίδευσης αφού η κλίση μεταφέρεται σε όλους τους νευρώνες σύμφωνα με τον αλγόριθμο οπισθοδρομικής διάδοσης που θα μελετηθεί παρακάτω. Επιπλέον, η αρχικοποίηση των βαρών πρέπει να είναι ιδιαίτερα προσεκτική, καθώς βάρη με μεγάλες τιμές μπορούν να οδηγήσουν τον νευρώνα σε κορεσμό.



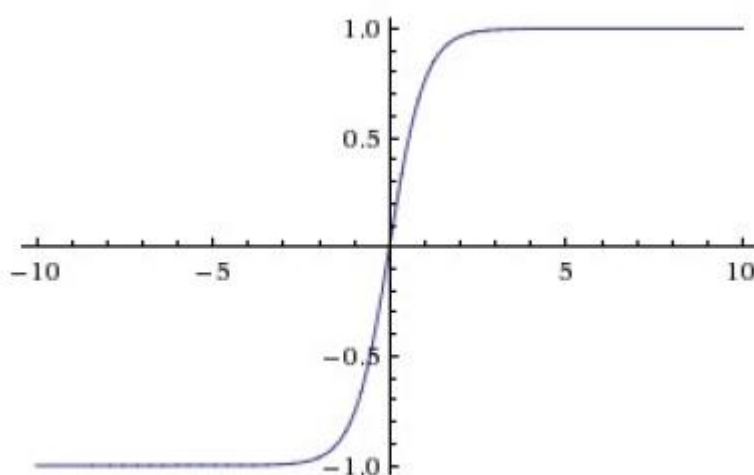
Εικόνα 8: Σιγμοειδής Συνάρτηση

### 2.2.1.5 Σιγμοειδής Υπερβολική εφαπτομένη

Η σιγμοειδής συνάρτηση υπερβολικής εφαπτομένης (Tanh Sigmoid Transfer Function) είναι πολύ κοντά στη λογική της απλής σιγμοειδούς που παρουσιάστηκε παραπάνω. Η μαθηματική της έκφραση είναι  $\varphi(x)=\tanh x$  και στην ουσία εκφράζει έναν κλιμακούμενο σιγμοειδή νευρώνα σύμφωνα με τη σχέση:

$$\tanh x = 2\sigma(2x) - 1$$

Η καμπύλη της συνάρτησης αυτής έχει την εξής μορφή:



Εικόνα 9: Tanh Sigmoid Function

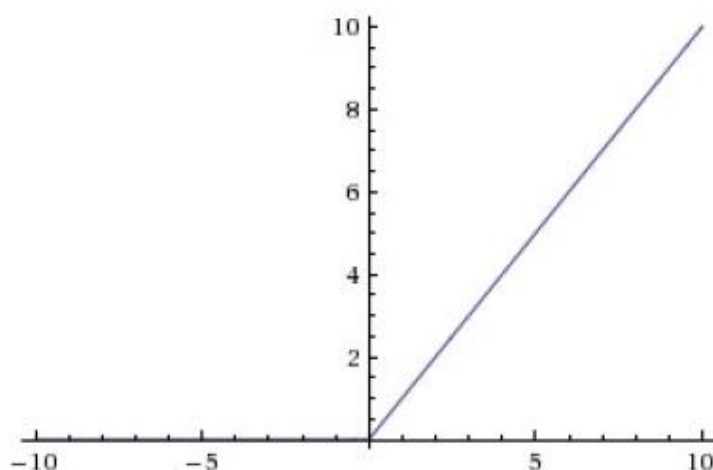
Εύκολα μπορεί κανείς να διακρίνει ότι λειτουργεί πανομοιότυπα με τη λογαριθμική σιγμοειδή υποχρεώνοντας το *Output* να βρίσκεται ανάμεσα στις τιμές 1 και -1.

Παρόλο που στην πράξη φάνηκε να λειτουργεί πιο αποδοτικά από την απλή σιγμοειδή και χρησιμοποιήθηκε περισσότερο από αυτή, υποκύπτει στον ίδιο κίνδυνο κορεσμού όταν αναπροσαρμοστεί αρκετά και έτσι περιορίστηκε η χρήση της.

### 2.2.1.6 Γραμμική Μονάδα Επιδιόρθωσης ReLU

Η Γραμμική Μονάδα Επιδιόρθωσης ή ReLU όπως αποκαλείται εν συντομία, είναι τα τελευταία χρόνια η πιο διαδεδομένη συνάρτηση ενεργοποίησης για εφαρμογές NN. Η μαθηματική της έκφραση είναι:  $f(x)=\max(0,x)$ .

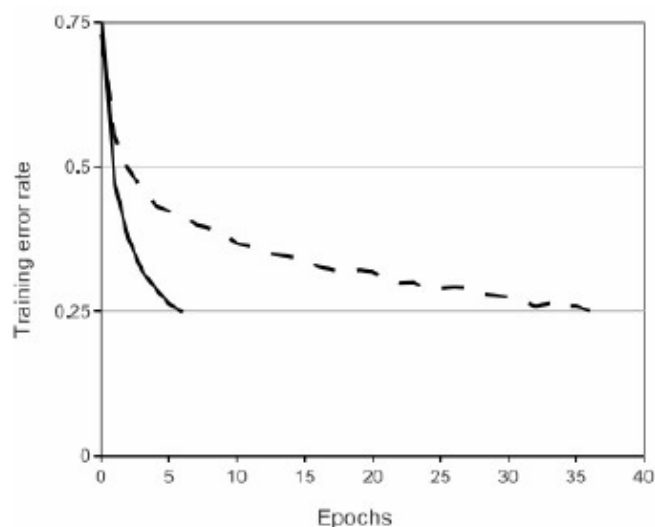
Όταν η τιμή *Output* είναι αρνητική τότε τη μηδενίζει ενώ αν είναι θετική την αφήνει όπως είναι (γίνεται γραμμική με κλίση 1) και ουσιαστικά αυτό που κάνει είναι να βάζει ένα κάτω κατώφλι στο 0. Η γραφική της παράσταση έχει την παρακάτω μορφή:



Εικόνα 10: Rectified Linear Unit

Τα πλεονεκτήματα που παρουσιάζει η ReLU σε σχέση με τις υπόλοιπες συναρτήσεις ενεργοποίησης που έβρισκαν εφαρμογή πριν από αυτή είναι δύο. Αρχικά τα πειραματικά δεδομένα από τις έρευνες του A. Krizhevsky και της ομάδας του από το πανεπιστήμιο του Τορόντο έδειξαν ότι η συνάρτηση αυτή φαίνεται να επιταχύνει σε μεγάλο βαθμό (σχεδόν 6 φορές) τον αλγόριθμο σύγκλισης του κόστους εκπαίδευσης του δικτύου σε σχέση με τις σιγμοειδείς συναρτήσεις (εικόνα 3.10) λόγω της γραμμικότητάς της και της τάσης της να μη φτάνει στον κορεσμό. Ένα επιπλέον πλεονέκτημά της είναι ότι η υπολογιστική της πολυπλοκότητα είναι πολύ μικρότερη σε σχέση με τις σιγμοειδείς αφού η μεν απλά κατωφλιών ει έναν πίνακα από τιμές στο 0 ενώ οι δε περιλαμβάνουν εκθετικούς υπολογισμούς και έτσι είναι πολύ πιο εύκολο να υλοποιηθεί προγραμματιστικά.





Εικόνα 11: Σύγκριση ταχύτητας σύγκλισης ReLU και Tanh

Επιπλέον οι μονάδες ReLU παρουσιάζουν το μειονέκτημα ότι είναι ευάλωτες στον κίνδυνο να «πεθάνουν» κατά τη φάση της εκπαίδευσης ή με άλλα λόγια να παραμένουν απενεργοποιημένες σε οποιαδήποτε είσοδο όπως συμβαίνει με άλλες συναρτήσεις ενεργοποίησης. Αυτό το πρόβλημα αναφέρεται ως Vanishing Gradient Problem. Όταν για τη μοντελοποίηση του δικτύου, για την αναπροσαρμογή των βαρών χρησιμοποιούνται μέθοδοι βασισμένες στην παράγωγο της συνάρτησης κόστους ως προς το εκάστοτε βάρος είναι πιθανό κάποιες φορές οι παράγωγοι να είναι τόσο μικρές που πρακτικά δεν επιτρέπουν στο βάρος να αναπροσαρμοστεί. Όταν αυτό συμβαίνει σε μεγάλη κλίμακα μέσα στο νευρωνικό δίκτυο τότε η εκπαίδευση γίνεται όλο και πιο αργή με εκθετικό ρυθμό και από ένα σημείο και μετά το δίκτυο ουσιαστικά δεν εκπαιδεύεται πια. Αυτό το πρόβλημα απαντάται όταν η παράμετρος του ρυθμού εκπαίδευσης (learning rate) η οποία επεξηγείται παρακάτω είναι πολύ υψηλή. Το θετικό είναι ότι με κατάλληλη ρύθμιση της παραμέτρου αυτής ο αλγόριθμος στην πλειοψηφία των περιπτώσεων δε θα υποπέσει σε τέτοιο πρόβλημα.

## 2.3 Εκπαίδευση

### 2.3.1 Βασικές αρχές εκμάθησης

Αφού οριστεί η τοπολογία του νευρωνικού δικτύου, δηλαδή ο αριθμός των νευρώνων και των επιπέδων που θα το απαρτίζουν, αυτό πρέπει να εκπαιδευτεί για να μπορέσει να ταξινομήσει τα πρότυπα. Η διαδικασία εκπαίδευσης ανήκει στην κατηγορία της επιβλεπόμενης μάθησης, αφού έχουμε τα δεδομένα εκπαίδευσης  $\mathbf{x}$  και την ετικέτα της κλάσης που ανήκουν  $t$ . Η εκπαίδευση είναι στην πραγματικότητα μια ενέργεια βελτιστοποίησης της συνάρτησης σφάλματος/κόστους, που εκφράζει τη απόκλιση της επιθυμητής εξόδου, και είναι ένα μέτρο απόδοσης του αλγορίθμου εκπαίδευσης από την έξοδο του νευρωνικού δικτύου.

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \|\mathbf{y}(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n\|^2$$

Οι τιμές των  $w$  για τις οποίες η συνάρτηση σφάλματος βρίσκεται σε ένα τοπικό ή στο ολικό ελάχιστο της αποτελούν και τη γνώση που έχει αποκτήσει το νευρωνικό δίκτυο. Η διαδικασία εκπαίδευσης είναι συνήθως μια επαναληπτική διαδικασία που ανανεώνει τα βάρη  $w$  στην κατεύθυνση αρνητικής κλίσης (gradient descent).

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_n(\mathbf{w}^{(\tau)})$$

Ο πιο διαδεδομένος αλγόριθμος είναι ο αλγόριθμος οπισθοδρομικής διάδοσης του σφάλματος που τυποποιεί την παραπάνω διαδικασία και είναι αυτός που εφαρμόζεται στη πλειοψηφία των σημερινών εφαρμογών.

### 2.3.2 Αλγόριθμος οπισθοδρομικής διάδοσης του σφάλματος

Ο αλγόριθμος οφείλει την ονομασία του στον τρόπο με τον οποίο εκτελείται. Αποτελείται από 2 φάσεις: τη διάδοση και την ανανέωση των βαρών. Κατά τη διάδοση εκτελείται πρώτα η εμπρόσθια διάδοση: στην αρχή ενεργοποιείται το δίκτυο και οι υπολογισμοί αρχίζουν από το πρώτο επίπεδο, συνεχίζουν στα κρυφά επίπεδα και τέλος στο επίπεδο εξόδου όπου και παράγουν την έξοδο του. Εκ των υστέρων, εκτελείται η οπισθοδρομική διάδοση στην οποία υπολογίζεται η απόκλιση της εξόδου από την επιθυμητή, δηλαδή η παράγωγος της συνάρτησης σφάλματος, και η πληροφορία διαδίδεται κατά την αντίστροφη φορά (προς την αρχή του δικτύου). Τελικά, με βάση αυτή την απόκλιση οι τιμές των βαρών ανανεώνονται και η διαδικασία ξεκινάει από την αρχή.

Πιο αναλυτικά, κατά την εμπρόσθια φάση του αλγορίθμου κάθε νευρώνας υπολογίζει την έξοδο του ως ένα σταθμισμένο άθροισμα των εισόδων του σε μορφή

$$a_j = \sum_i w_{ji} z_i$$

Όπου  $z_i$  η έξοδος των νευρώνων του προηγούμενου επιπέδου και  $w_{ji}$  το βάρος της σύνδεσης του νευρώνα  $i$  με τον νευρώνα  $j$ . Να σημειωθεί και ότι είναι πιθανό να συμπεριλαμβάνεται μια επιπλέον είσοδος με τιμή 1 (bias). Έπειτα, εφαρμόζεται η μη γραμμική συνάρτηση  $h$  που δίνει την τελική έξοδο του νευρώνα σύμφωνα με τη σχέση:

$$z_j = h(a_j)$$

Η εμπρόσθια διάδοση συνεχίζεται σειριακά σε όλα τα κρυφά επίπεδα μέχρι να φτάσει στο επίπεδο εξόδου. Στο σημείο αυτό είναι αναγκαίο να υποθέσουμε ότι η συνάρτηση σφάλματος έχει τη γενική μορφή:

$$E(\mathbf{w}) = \sum_{n=1}^N E_n(\mathbf{w})$$

Για να αξιολογήσουμε τα τωρινά βάρη και την επίδρασή τους στην έξοδο αλλά και να ανανεώσουμε τα βάρη πρέπει να υπολογίζουμε τη πρώτη παράγωγο της συνάρτησης σφάλματος σύμφωνα με το κανόνα της αλυσίδας:

$$\frac{\partial E_n}{\partial w_{ji}} = \frac{\partial E_n}{\partial a_j} \frac{\partial a_j}{\partial w_{ji}}$$

Επιπλέον ορίζοντας ότι

$$\delta_j \equiv \frac{\partial E_n}{\partial a_j}$$

και παρατηρώντας ότι

$$\frac{\partial a_j}{\partial w_{ji}} = z_i$$

Καταλήγουμε στην εξής σχέση:

$$\frac{\partial E_n}{\partial w_{ji}} = \delta_j z_i$$

η οποία δηλώνει ότι η παράγωγος είναι το γινόμενο της τιμής  $\delta_j$  επί την έξοδο από τον αντίστοιχο νευρώνα του προηγούμενου επιπέδου. Επομένως, μένει μόνο να υπολογίσουμε τις τιμές των  $\delta$  για τον κάθε νευρώνα. Για τους νευρώνες του επιπέδου εξόδου έχουμε:

$$\delta_k = y_k - t_k$$

ενώ για τα κρυφά επίπεδα αποδεικνύεται πως ο τύπος για την οπισθοδρομική διάδοση είναι:

$$\delta_j = h'(a_j) \sum_k w_{kj} \delta_k$$

Ο αλγόριθμος μπορεί τελικά να συνοψιστεί στα παρακάτω βήματα:

1. Εμπρόσθια διάδοση για την εύρεση των εξόδων κάθε νευρώνα
2. Αξιολόγηση των  $\delta_k$  για το επίπεδο εξόδου
3. Οπισθοδρομική διάδοση με την απόκτηση των  $\delta_j$  για τα κρυφά επίπεδα
4. Ανανέωση των βαρών σύμφωνα με τη παράγωγο της συνάρτησης σφάλματος πολλαπλασιασμένη με μια σταθερά  $a$  που ονομάζεται ρυθμός εκμάθησης και ορίζει το πόσο «γρήγορα» θα αναπροσδιορίζονται τα βάρη.

### 2.3.3 Τεχνικές για βελτίωση της απόδοσης των νευρωνικών δικτύων

#### 2.3.3.1 Αρχικοποίηση βαρών

Ένα σημαντικό ζήτημα για τη βελτίωση της εκπαίδευσης των νευρωνικών δικτύων είναι οι τιμές με τις οποίες αρχικοποιούνται τα βάρη  $W$ . Ένα πρώτο βήμα είναι να αναθέσουμε τελείως τυχαίες τιμές, που ενώ δε δημιουργεί πρόβλημα στον αλγόριθμο, σίγουρα δε βοηθά στην σύγκλιση του. Μια συνήθης πρακτική είναι αυτό να γίνει τυχαία με βάση την κατανομή Gauss με μέση τιμή 0 και διασπορά  $\sigma$ . Το βασικό πρό-

βλημα αυτής της προσέγγισης είναι η πιθανότητα ο νευρώνας να κορεσθεί. Επειδή αρκετά συχνά οι ενεργοποιήσεις θα είναι κοντά στο 0 ή στο 1, η παράγωγος της συνάρτησης σφάλματος θα έχει πολύ μικρή τιμή και συνεπώς το δίκτυο θα μαθαίνει πολύ αργά. Μια πολύ καλύτερη τακτική είναι η κατανομή Gauss να έχει μέση τιμή 0 αλλά διασπορά  $1/\sqrt{n_{in}}$ , όπου  $n_{in}$  το πλήθος των παραδειγμάτων εκπαίδευσης.

### 2.3.3.2 Προεπεξεργασία δεδομένων

Η αναπαράσταση και αποθήκευση εικόνων και γενικά κάθε μορφής δεδομένων μπορεί να διαφέρει από σύστημα σε σύστημα. Για να μπορέσουν τα δεδομένα να αποτελέσουν πρότυπα εκπαίδευσης, πολλές φορές είναι αναγκαίο να υποστούν προεπεξεργασία. Υπάρχουν διάφοροι τρόποι και τεχνικές για να γίνει αυτό, που εξαρτώνται κυρίως από το είδος της εφαρμογής και τα ίδια τα δεδομένα. Εδώ θα αναφερθούν κάποιοι βασικοί και συνηθισμένοι τρόποι που αφορούν την περίπτωση που η είσοδος έχει τη μορφή εικόνας:

- **Κανονικοποίηση:** Τόσο τα δεδομένα εκμάθησης όσο και τα δεδομένα δοκιμής σχεδόν πάντα κανονικοποιούνται είτε στο διάστημα  $[0,1]$  είτε ως προς τη μέση τιμή τους. Επιπλέον, μπορεί να γίνει και κανονικοποίηση ως προς τη διασπορά τους, για να γίνει μοναδιαία. Μια άλλη μέθοδος είναι η ανάλυση κύριων συνιστωσών (PCA) ή παρόμοιοι μετασχηματισμοί. Γενικά, η κανονικοποίηση επιταχύνει την εκπαίδευση: αν για παράδειγμα, όλα τα pixels της εικόνας εισόδου έχουν θετικές τιμές, το ποσό της ανανέωσης από τον κανόνα μάθησης θα έχει ίδιο πρόσημο για όλα τα βάρη, με αποτέλεσμα να μετατοπίζονται όλα προς την ίδια κατεύθυνση. Με αυτόν τον τρόπο, τα βάρη θα αυξάνονται ή θα μειώνονται όλα μαζί και η εκπαίδευση θα γίνεται πάνω σε μία ευθεία. Γι' αυτό το λόγο, είναι πολύ σημαντικό να κεντράρονται τα δεδομένα ως προς τη μέση τιμή τους. Επίσης, είναι επιθυμητό τα παραδείγματα να είναι όσο το δυνατόν ασυσχέτιστα και να έχουν μικρή διασπορά εντός των κατηγοριών.
- **Αποκοπή δεδομένων:** Υπάρχει περίπτωση κάποια από τα δεδομένα να είναι ελλιπή, να περιέχουν πολύ ακραίες τιμές ή πολύ θόρυβο. Αν αυτά δε μπορούν να προεπεξεργαστούν, τότε θα χρειαστεί να εξαιρεθούν από την εκπαίδευση.

### 2.3.3.3 Συνάρτηση διεντροπίας και Softmax

Στη μέχρι τώρα ανάλυση λάβαμε υπόψιν μας ότι η συνάρτηση κόστους-σφάλματος ορίζεται ως το άθροισμα των τετραγώνων των αποστάσεων της εξόδου του δικτύου με το επιθυμητό διάνυσμα. Αυτό δεν είναι, βέβαια, απαραίτητο αφού στη βιβλιογραφία έχουν χρησιμοποιηθεί πολλές διαφορετικές συναρτήσεις. Μια από αυτές είναι η συνάρτηση διεντροπίας (cross entropy) που κερδίζει έδαφος τα τελευταία χρόνια, ειδικά στα βαθιά δίκτυα και ορίζεται ως εξής:

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

με  $n$  τον αριθμό των προτύπων εκπαίδευσης,  $\mathbf{x}$  το διάνυσμα εισόδου,  $\mathbf{y}$  το επιθυμητό διάνυσμα και  $\mathbf{a}$  την έξοδο των νευρώνων. Η συνάρτηση αυτή έχει την ιδιότητα να τείνει στο 0, όσο η απόσταση των  $\mathbf{y}$  και  $\mathbf{a}$  μικραίνει, και να είναι πάντα θετική, γεγονός που την καθιστά ιδανική για συνάρτηση κόστους. Έχει αποδειχθεί ότι η χρήση δεν αλλάζει σε μεγάλο βαθμό τις εξισώσεις του αλγορίθμου οπισθοδρομικής διάδοσης παρά μόνο στην τελική αναπροσαρμογή των βαρών, όπου έχουμε τη σχέση:

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

Έχει βρεθεί, παράλληλα, μετά από δοκιμές ότι η συνάρτηση διεντροπίας σε συνδυασμό με τη συνάρτηση ενεργοποίησης softmax επιταχύνει σε μεγάλο βαθμό την εκπαίδευση, ειδικά στα συνελκτικά δίκτυα. Η softmax είναι μια συνάρτηση ειδικής μορφής και αρκετά διαφορετική από τις υπόλοιπες συναρτήσεις ενεργοποίησης, αφού κάθε έξοδος εξαρτάται από όλες τις υπόλοιπες:

$$a_j^L = \frac{e^{z_j^L}}{\sum_k e^{z_k^L}},$$

Για να γίνει αντιληπτό πώς λειτουργεί η softmax είναι αναγκαίο να επισημανθεί πως στις πραγματικές εφαρμογές αναγνώρισης σε  $n$  κλάσεις το επιθυμητό διάνυσμα εξόδου είναι της μορφής  $[0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0]$  (one hot encoding). Αυτό σημαίνει πως το διάνυσμα εξόδου στη θέση της κλάσης που ανήκει η είσοδος παίρνει τη τιμή 1, ενώ στις υπόλοιπες την τιμή 0. Η συνάρτηση softmax, που στην ουσία προέρχεται από τη

γενικευμένη Bernouli συνάρτηση πιθανότητας, απλά μετατρέπει την έξοδο του νευρώνα σε πιθανότητα, την κανονικοποιεί με άλλα λόγια στο διάστημα  $[0,1]$ , για να είναι συμβατή με την κωδικοποίηση που προαναφέρθηκε.

### 3 Συνελκτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks)

Τα Συνελκτικά Νευρωνικά Δίκτυα (ΣΝΔ) ή στα Αγγλικά Convolutional Neural Networks (CNN), χωρίζονται σε δυο μεγάλες κατηγορίες τα Αβαθή Νευρωνικά Δίκτυα (Shallow Neural Networks) και τα Βαθιά Νευρωνικά Δίκτυα (Deep Neural Networks). Ένα συνελκτικό επίπεδο (convolutional layer) είναι ουσιαστικά ένα σύνολο από νευρώνες που εκτελούν συνέλιξη των φίλτρων που έχουν προκαθοριστεί, με την εικόνα-διάνυσμα που δέχονται στην είσοδο. Κάθε επίπεδο μπορεί να περιλαμβάνει νευρώνες που εκτελούν συνέλιξη, διαδικασίες pooling, εισαγωγή μη γραμμικότητας ή ακόμη και κανονικοποίηση, ενώ έχει διακριτές εισόδους και εξόδους. Οι διαστάσεις των φίλτρων που περιλαμβάνουν, ο αριθμός τους και το βάθος τους (αριθμός καναλιών) μπορεί να διαφέρει σημαντικά ανάλογα με το πρόβλημα.

Στα παραδοσιακά νευρωνικά δίκτυα κάθε νεύρο στο επίπεδο εισόδου συνδέεται σε ένα νεύρο στο επίπεδο εξόδου και γι' αυτό λέμε ότι τα δίκτυα αυτά είναι πλήρως συνδεδεμένα. Παρόλα αυτά στα συνελκτικά νευρωνικά δίκτυα δεν χρησιμοποιούμε επίπεδα πλήρους διασύνδεσης μέχρι να φτάσουμε στα τελευταία επίπεδα. Συνεπώς μπορούμε να ορίσουμε ένα συνελκτικό νευρωνικό δίκτυο (Convolutional Neural Network-CNN) ως ένα νευρωνικό δίκτυο το οποίο χρησιμοποιεί ένα εξειδικευμένο συνελκτικό επίπεδο αντί για το πλήρως συνδεδεμένο επίπεδο για τουλάχιστον ένα επίπεδο του δικτύου.

Κάθε Νευρωνικό Δίκτυο, όπως είπαμε και πιο πάνω, αποτελείται από διάφορα επίπεδα τα οποία περιλαμβάνουν τις αντίστοιχες εισόδους και εξόδους. Εφόσον μετά από κάθε τέτοιο επίπεδο η έξοδος αποτελεί μια αναπαράσταση της εισόδου με διαφορετικές (συνήθως μικρότερες) διαστάσεις, μπορούμε να θεωρήσουμε κάθε έξοδο ενός συνελκτικού νευρώνα σαν χαρακτηριστικό διάνυσμα που περιγράφει την αντίστοιχη είσοδο. Το ίδιο μπορεί να υποθεθεί για την έξοδο ενός pooling νευρώνα, η οποία αποτελεί την προέκταση μιας διαδρομής από την είσοδο μιας εικόνας προς το κρυφό επίπεδο συνέλιξης το οποίο εξετάζουμε. Από τη θεωρία γνωρίζουμε ότι τα βάρη στις εισόδους κάθε κρυμμένου επιπέδου πρέπει οπωσδήποτε να είναι τυχαία και επομένως για κάθε είσοδο, η έξοδος ενός συγκεκριμένου νευρώνα θα είναι μοναδική για κάθε εικόνα. Τα χαρακτηριστικά διανύσματα δηλαδή που προκύπτουν ακόμη και από το ίδιο κρυφό επίπεδο, διαφέρουν μεταξύ τους.





*Εικόνα 12-Shallow Neural Network 3 επιπέδων.*

Όπως προαναφέρθηκε τα νευρωνικά δίκτυα χωρίζονται σε Shallow και Deep Neural Networks. Στην περίπτωση των Deep NN, ο αριθμός των συνελίξεων μπορεί να είναι σημαντικά αυξημένος και τα χαρακτηριστικά διανύσματα που προκύπτουν να έχουν ελαττωμένες τις διαστάσεις σε μεγάλο ποσοστό, χωρίς όμως να χάνουν την ουσία της πληροφορίας που περιλαμβάνουν. Γενικά, όσο πιο “βαθύ” είναι το επίπεδο το οποίο εξετάζουμε, τόσο πιο “γενικά” (global) είναι τα χαρακτηριστικά και πιο αναλλοίωτα. Παρατηρούμε ότι στο τέλος του Δικτύου περιλαμβάνεται πάντα ένας ταξινομητής. Στην επόμενη εικόνα φαίνεται η γενική εικόνα μιας Deep Architecture, συγκρινόμενη με την προηγούμενη εικόνα ως προς τον αριθμό των επιπέδων.



*Εικόνα 13-Deep Neural Network N επιπέδων.*

Μια μη γραμμική συνάρτηση όπως η ReLU εφαρμόζεται στη συνέχεια στην έξοδο αυτών των συνελίξεων και η διαδικασία της συνέλιξης-ενεργοποίησης συνεχίζεται μέχρι να φτάσουμε στο τέλος του δικτύου και εφαρμόζοντας ένα ή δυο επίπεδα πλήρους διασύνδεσης μπορούμε να αποκτήσουμε τις τελικές ταξινομήσεις.

Σε κάθε επίπεδο CNN εφαρμόζεται ένα διαφορετικό σύνολο φίλτρων, τυπικά ή εκατοντάδες χιλιάδες από αυτά, και συνδυάζει τα αποτελέσματα στέλνοντας την έξοδο στο επόμενο επίπεδο του δικτύου. Κατά τη διάρκεια της εκμάθησης ένα δίκτυο CNN αυτόματα μαθαίνει τις τιμές για αυτά τα φίλτρα.

Με όρους ομαδοποίησης εικόνας ένα δίκτυο CNN μπορεί να μάθει τα εξής:

- Ανίχνευση ακμών από καθαρά δεδομένα pixels στο πρώτο επίπεδο
- Χρησιμοποίηση αυτών των ακμών για ανίχνευση σχημάτων (π.χ “blobs”) στο επόμενο επίπεδο
- Χρησιμοποίηση αυτών των σχημάτων για ανίχνευση χαρακτηριστικών όπως δομές προσώπου, τμήματα ενός αυτοκινήτου κ.λ.π. στα πιο υψηλά επίπεδα του δικτύου

Το πρώτο επίπεδο ενός CNN χρησιμοποιεί αυτά τα χαρακτηριστικά ενός υψηλότερου επιπέδου για να κάνει προβλέψεις που αφορούν τα περιεχόμενα της εικόνας . Στη πράξη τα CNN έχουν δύο βασικά πλεονεκτήματα: Τοπική αμεταβλητότητα (Local invariance) και Συνθετικότητα (compositionality).

Η αρχή της τοπικής αμεταβλητότητας μας επιτρέπει να κατηγοριοποιήσουμε μια εικόνα ώστε να περιέχει ένα συγκεκριμένο αντικείμενο ανεξάρτητα από το αν το αντικείμενο εμφανίζεται στην εικόνα ή όχι. Αποκτούμε αυτή τη τοπική αμεταβλητότητα μέσω της χρήσης “στρωμάτων συγκέντρωσης” (pooling layers) τα οποία μας επιτρέπουν να αναγνωρίζουμε περιοχές των εισόδων μας με μεγάλη ανταπόκριση σε ένα συγκεκριμένο φίλτρο.

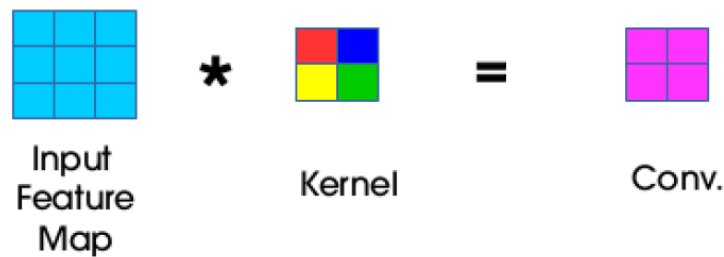
Το δεύτερο πλεονέκτημα είναι η συνθετικότητα. Κάθε φίλτρο συνθέτει ένα τμήμα τοπικών χαρακτηριστικών κατώτερου επιπέδου σε μια αναπαράσταση υψηλότερου επιπέδου παρόμοια με αυτή που έχουμε συνθέσει ένα σύνολο μαθηματικών συναρτήσεων για να κατασκευάσουμε την έξοδο προηγούμενων συναρτήσεων:  $f(g(h(x)))$ . Αυτή η σύνδεση επιτρέπει στο δίκτυο να μάθει μερικά πράγματα βαθύτερα.

### 3.1 Επίπεδο συνέλιξης

Το επίπεδο συνέλιξης περιλαμβάνει ένα ή περισσότερα φίλτρα/kernel με τιμές που αντιπροσωπεύουν τα εκπαιδευόμενα βάρη. Κάθε φίλτρο είναι μικρό χωρικά (οι τυπικές διαστάσεις των φίλτρων είναι 3x3, 5x5 ή 7x7) αλλά εκτείνεται σε όλο το φάσμα της εισόδου. Στο εμπρόσθιο πέρασμα κάθε φίλτρο συνελίσσεται με την είσοδο, δηλαδή υπολογίζονται τα εσωτερικά γινόμενα της εισόδου με το φίλτρο και παράγεται ένα διάγραμμα εξόδου που ονομάζεται **χάρτης χαρακτηριστικών** (feature map) σύμφωνα με τον τύπο:

$$\begin{aligned}
O(x, y) &= w_{x,y}^l * a_{x,y}^l + b_{x,y} \\
O(x, y) &= \sum_{x'} \sum_{y'} w_{x',y'}^l a_{x-x',y-y'}^l + b_{x,y} \\
&= \sum_{x'} \sum_{y'} w_{x',y'}^l f(o_{x-x',y-y'}^{l-1}) + b_{x,y}
\end{aligned}$$

Τα  $w_{x,y}$  αντιστοιχούν στους συντελεστές του φίλτρου, τα  $a_{x,y}$  στην είσοδο, το  $b$  είναι μια σταθερή τιμή (bias), ενώ το διάνυσμα  $O(x,y)$  είναι ο χάρτης των χαρακτηριστικών. Σχηματικά συμβαίνει το ακόλουθο:



*Εικόνα 14- Συνέλιξη*

Διαισθητικά καταλαβαίνουμε ότι το συνελκτικό δίκτυο θα μάθει τα φίλτρα που ενεργοποιούνται όταν εντοπίσουν κάποιο βασικό χαρακτηριστικό της εικόνας, όπως ακμές ή κορυφές.

Από μια άλλη οπτική γωνία μπορούμε να αναπαραστήσουμε κάθε pixel της εισόδου και της εξόδου με ένα νευρώνα και τα φίλτρα ως τα βάρη των συνδέσεων που ενώνουν τους νευρώνες. Τότε καταλαβαίνουμε ότι έχουμε μερική και τοπική συνδεσιμότητα των νευρώνων, σε αντίθεση με τα κλασσικά νευρωνικά δίκτυα, αφού κάθε έξοδος βλέπει μόνο μια μικρή περιοχή της εισόδου.

Πριν προχωρήσουμε στα επόμενα επίπεδα θα ήταν χρήσιμο να αναφερθούν περιληπτικά κάποιες βασικές αρχές που διέπουν τέτοιου είδους δίκτυα:

- **Τοπική συνδεσιμότητα**

Όταν ασχολούμαστε με εισόδους πολλών διαστάσεων δεν είναι πρακτικό να συνδεθούν όλοι οι νευρώνες του επιπέδου με όλους του προηγούμενου, αφού θα προκύψει πολύ μεγάλος αριθμός παραμέτρων. Για το λόγο αυτό, όπως αναφέρθηκε ήδη, συνδέουμε κάθε νευρώνα με μια περιοχή της εισόδου. Έτσι, προκύπτει μια νέα παράμετρος

που ονομάζεται οπτικό πεδίο νευρώνων (receptive field)  $F$  και καθορίζει πόσα εικονοστοιχεία (pixels) βλέπει ο κάθε νευρώνας. Είναι, δηλαδή, η διάσταση των φίλτρων του δικτύου. Όσο μεγαλύτερη είναι η είσοδος, τόσο μεγαλύτερο είναι συνήθως το οπτικό πεδίο.

- **Χωρική διάταξη και υπερπαράμετροι**

**1. Το βάθος (depth)  $D$**  του χάρτη χαρακτηριστικών αντιπροσωπεύει τον αριθμό των καναλιών της εξόδου και σχετίζεται άμεσα με τον αριθμό των φίλτρων

**2. Βήμα φιλτραρίσματος (stride)  $S$ :** καθορίζει κάθε πόσα εικονοστοιχεία θα γίνεται φιλτράρισμα, δηλαδή καθορίζει την επικάλυψη των οπτικών πεδίων

Ενώ έχουμε εξηγήσει τη συνδεσιμότητα των νευρώνων δεν είναι ακόμα πλήρως κατανοητό το πώς αυτοί οργανώνονται στο χώρο. Για το λόγο αυτό υπάρχουν 3 παράμετροι:

**3. Επέκταση (padding) της εισόδου  $P$ :** μας βοηθάει να επιμηκύνουμε την είσοδο, κυρίως με μηδενικά, για να καθορίσουμε τις διαστάσεις μήκους και όγκου της εξόδου.

Τώρα μπορούμε εύκολα να βρούμε τις διαστάσεις του διανύσματος του χάρτη χαρακτηριστικών που δίνονται από τον τύπο:

$$(W - F + 2P) / S + 1$$

όπου  $W$  η διάσταση του διανύσματος εισόδου.

Συνοπτικά το επίπεδο συνέλιξης χαρακτηρίζεται από τα εξής στοιχεία:

- **Είσοδος: όγκος διαστάσεων  $W1 \times H1 \times D1$**  (πολυκαναλική εικόνα) .
- **Υπερπαράμετροι:** 4 παράμετροι που καθορίζουν τη συνέλιξη : ο αριθμός των φίλτρων  $K$ , το μέγεθος των φίλτρων  $F$ , το βήμα του φιλτραρίσματος  $S$  και ο αριθμός των pixels για την επέκταση της εικόνας  $P$ .

- **Έξοδος:** όγκος διαστάσεων  $W2 \times H2 \times D2$ , για τον οποίο ισχύουν οι σχέσεις:

$$- W2 = (W1 - F + 2P)/S + 1,$$

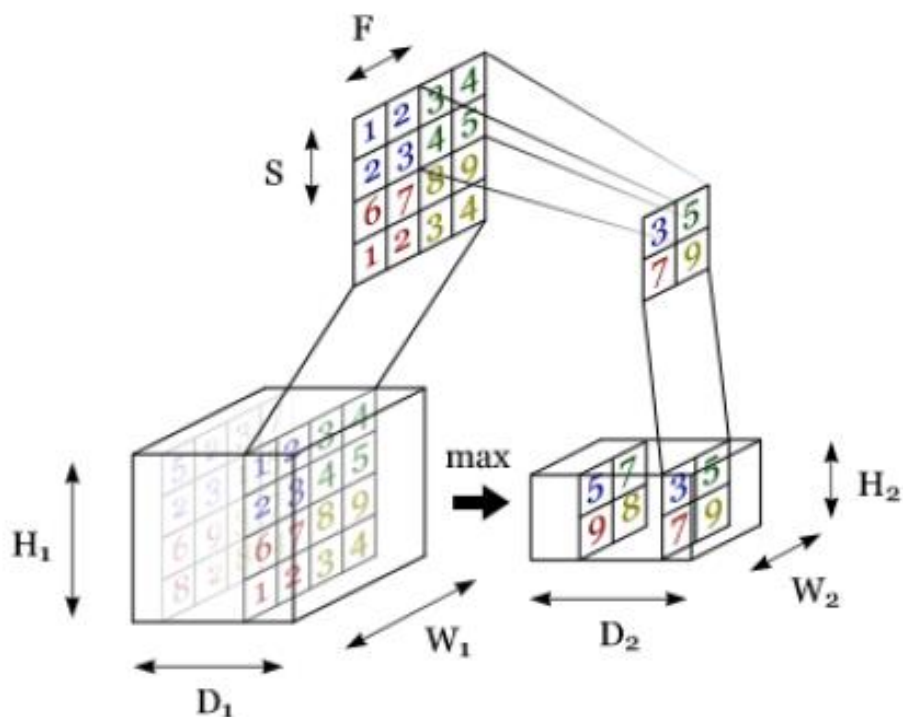
$$- H2 = (H1 - F + 2P)/S + 1,$$

$$- D2 = K$$

**Βάρη:** εισάγει γενικά  $(W2 * H2 * D2) * (F * F * D1 + 1)$  βάρη και κατώφλια (biases).

### 3.2 Επίπεδο συγκέντρωσης

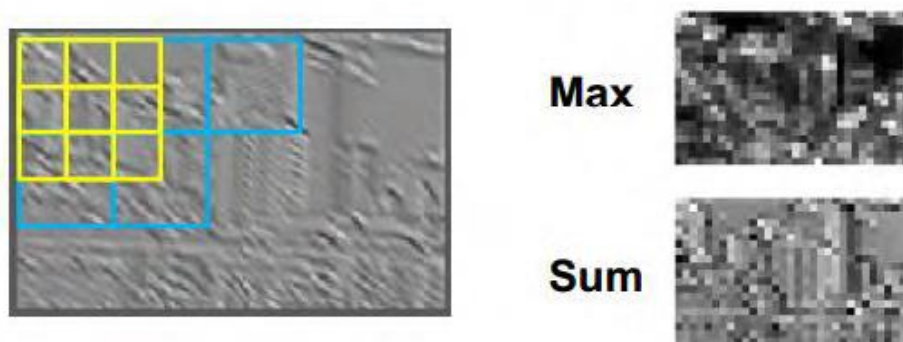
Το επίπεδο συγκέντρωσης (pooling layer) μειώνει τις διαστάσεις του χάρτη χαρακτηριστικών και κατ' επέκταση το πλήθος των παραμέτρων του συνελκτικού δικτύου. Στην ουσία κάνει μια δειγματοληψία των τιμών στις διαστάσεις μήκους και πλάτους, χωρίς να επηρεάζει το βάθος του χάρτη. Είθισται μετά από κάθε συνελκτικό επίπεδο να υπάρχει και ένα επίπεδο συγκέντρωσης για να απλοποιείται το πλήθος των υπολογισμών. Η δειγματοληψία γίνεται με δύο βασικούς τρόπους: συγκέντρωση μέσου όρου ή μέγιστου (average or max pooling). Στη συγκέντρωση μέσου όρου ορίζεται ένα τετραγωνικό φίλτρο (συνήθως  $2 \times 2$ ) που σαρώνει την εικόνα με βήμα  $S$  και βρίσκει το μέγιστο από κάθε "γειτονιά" του χάρτη χαρακτηριστικών. Από την άλλη, η συγκέντρωση μέσου όρου υπολογίζει το μέσο όρο των τιμών κάθε "γειτονιάς". Γραφικά η διαδικασία απεικονίζεται έτσι:



Εικόνα 15-Επίπεδο Συγκέντρωσης

Τα επίπεδα συγκέντρωσης (pooling layers) στα CNNs, συνοψίζουν τις εξόδους γειτονικών γκρουπ νευρώνων εντός ενός παραθύρου (patch) με μια αντιπροσωπευτική τιμή, ενώ συνήθως τα γειτονικά παράθυρα δεν επικαλύπτονται. Πρόκειται ουσιαστικά για μια διαδικασία υπο-δειγματοληψίας των δεδομένων ενώ για καλύτερη κατανόηση της διαδικασίας μπορούμε να φανταστούμε ένα επίπεδο pooling σαν ένα “πλέγμα” pooling νευρώνων τοποθετημένων σε απόσταση  $s$  pixels, καθένας από τους οποίους συνοψίζει μια περιοχή  $z \times z$  με κέντρο τον ίδιο τον νευρώνα. Θέτοντας  $s=z$  λαμβάνουμε τα κλασικά αποτελέσματα του κοινώς χρησιμοποιούμενου, μη επικαλυπτόμενων παραθύρων pooling, ενώ για  $s < z$  λαμβάνουμε τιμές από επικαλυπτόμενα παράθυρα. Το pooling αποτελεί μια πολύ βασική λειτουργία για κάθε CNN, αφού απλοποιεί πολύ τη διαδικασία λόγω της σημαντικής μείωσης των δεδομένων κι επομένως του αριθμού των απαιτούμενων πράξεων. Οι επικρατέστερες κατηγορίες του pooling είναι το max, sum και average pooling, ενώ μπορεί τα παράθυρα που χρησιμοποιούνται να επικαλύπτονται ή και όχι ανάλογα με τις ανάγκες του προβλήματος. Η διαδικασία του pooling, εκτός απ τη μείωση του μεγέθους των δεδομένων μας δίνει τη δυνατότητα προσθήκης

περισσότερης πληροφορίας στην αρχική εικόνα μέσω των αρχικών διαστάσεων ενώ είναι ανεξάρτητο μικρών μετασχηματισμών.



Εικόνα 16. Αποτέλεσμα εφαρμογής *max pooling* και *sum pooling* πάνω σε εικόνα.

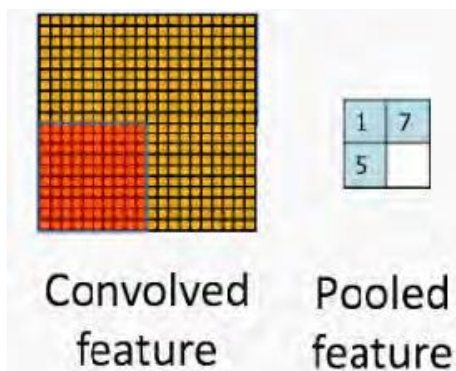
Όταν έχουμε πλέον λάβει τα χαρακτηριστικά διανύσματα που περιγράψαμε νωρίτερα, αποφασίζουμε για το μέγεθος των παραθύρων, ας πούμε  $m \times n$  που θα κάνουμε pool τα δεδομένα. Τότε διαιρούμε τα χαρακτηριστικά μας σε  $m \times n$  περιοχές και παίρνουμε το μέγιστο (*max*) ή το μέσο όρο (*mean*) του παραθύρου, το οποίο αποτελεί το νέο χαρακτηριστικό μας. Αυτές οι *pooled* περιοχές μπορούν πλέον να χρησιμοποιηθούν για ταξινόμηση.

Μετά την διαδικασία του *pooling* και αφού έχουμε λάβει τα χαρακτηριστικά, συνήθως θέλουμε να τα χρησιμοποιήσουμε για ταξινόμηση. Στη θεωρία, μπορούμε να χρησιμοποιήσουμε όλα τα χαρακτηριστικά σε έναν ταξινομητή όπως ο *softmax classifier*, ωστόσο αυτό είναι υπολογιστικά αδύνατο. Ας αναλογιστούμε την περίπτωση εικόνων  $96 \times 96$  pixels και 400 χαρακτηριστικά από  $8 \times 8$  inputs. Κάθε συνέλιξη φέρει ως αποτέλεσμα μια έξοδο μεγέθους  $(96-8+1) \times (96-8+1)=7921$ , και εφόσον έχουμε 400 χαρακτηριστικά, προκύπτει ένα διάνυσμα  $89^2 \times 400=3.168.400$  χαρακτηριστικά ανά δείγμα. Η εκμάθηση ενός ταξινομητή με 3+ εκατομμύρια χαρακτηριστικά, μπορεί να γίνει πολύ επίπονη και υπέρ-εξειδικευμένη (*over-fitting*).

Προκειμένου να το διευθετήσουμε, ας ανακαλέσουμε το γεγονός ότι αποφασίσαμε να λαμβάνουμε *convolved* εικόνες επειδή έχουν την ιδιότητα της “στατικότητας”, δηλαδή ότι τα χαρακτηριστικά που είναι χρήσιμα σε μια περιοχή της εικόνας, είναι χρήσιμα σε κάθε άλλη περιοχή της. Έτσι, προκειμένου να περιγράψουμε μια μεγάλη εικόνα θα λάβουμε ένα μίγμα στατιστικών των χαρακτηριστικών των παραπάνω εικόνων. Για παράδειγμα, θα μπορούσαμε να λάβουμε έναν μέσο όρο ή τη μέγιστη τιμή ενός συγκεκριμένου χαρακτηριστικού για ένα *patch* της εικόνας. Έτσι, ελαττώνονται

σημαντικά οι διαστάσεις των δεδομένων και πετυχαίνουμε μικρότερο over-fitting. Ονομάζουμε αυτή τη μέθοδο “pooling”, ή πιο συγκεκριμένα “mean pooling” και “max pooling” ανάλογα τη μέθοδο που χρησιμοποιούμε.

Στην επόμενη εικόνα φαίνεται η διαδικασία του pooling:



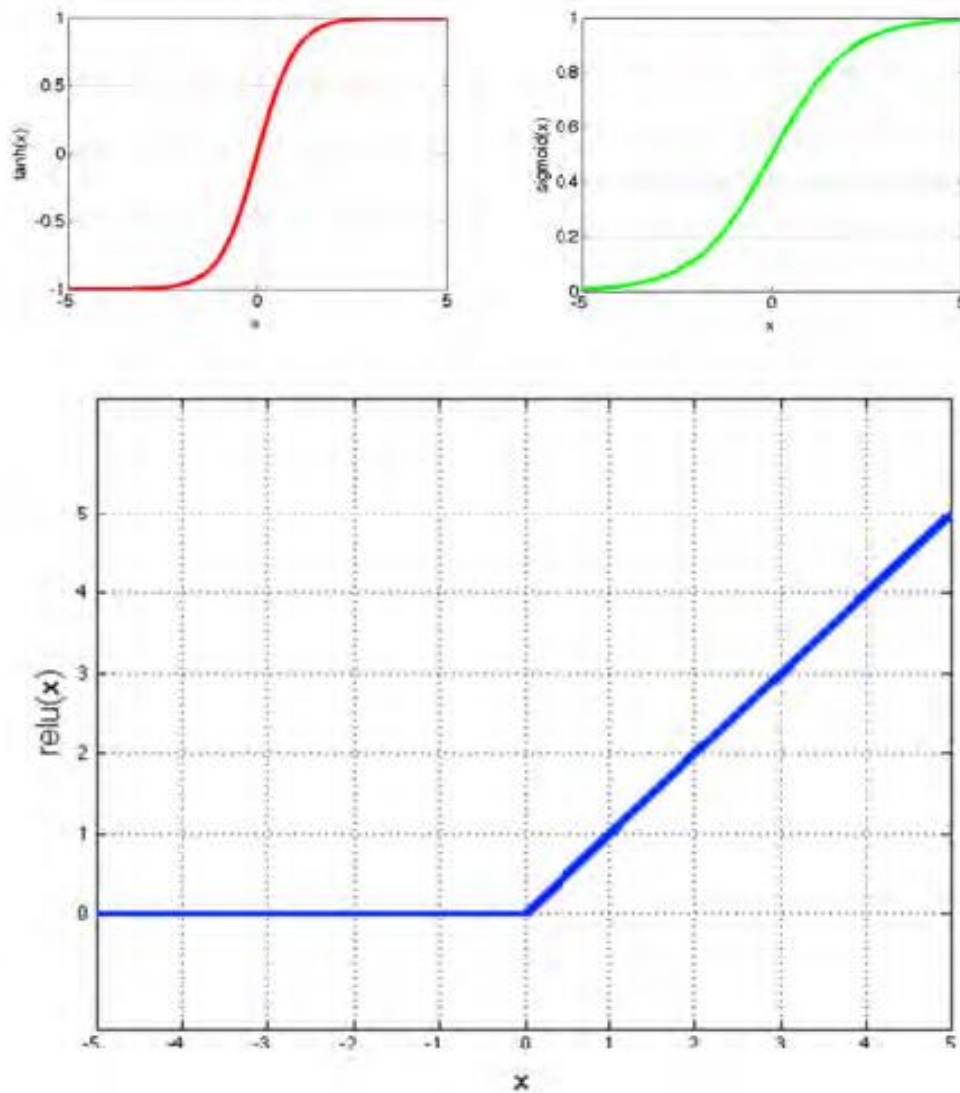
*Εικόνα 17 Η διαδικασία του pooling.*

Εάν κάποιος επιλέξει οι περιοχές που γίνεται το pooling να είναι συνεχόμενες στην εικόνα και λαμβάνει χαρακτηριστικά μόνο από τους ίδιους κρυφούς νευρώνες, τότε αυτοί οι pooling νευρώνες θα γίνουν “translation invariant” ή αλλιώς ανεξάρτητα μετατοπίσεων της εικόνας. Αυτό σημαίνει ότι το ίδιο χαρακτηριστικό θα ενεργοποιείται ακόμα και όταν η εικόνα υπόκειται μικρές μετατοπίσεις. Συνήθως για αυτό το λόγο επιδιώκεται τα χαρακτηριστικά μας να είναι ανεξάρτητα μετατοπίσεων, όπως για παράδειγμα στην ανίχνευση αντικειμένων ή την αναγνώριση ήχου.

### 3.3 Μη γραμμικότητα

Η εισαγωγή της μη γραμμικότητας (non-linearity) δίνει σοβαρό πλεονέκτημα στα CNN έναντι άλλων γνωστών μεθόδων αντιμετώπισης πολλών προβλημάτων. Σαν παράδειγμα αναφέρουμε την περίπτωση ενός μη γραμμικού συστήματος και την προσπάθεια πρόβλεψης αυτού, όπου οι γραμμικές μέθοδοι αδυνατούν να δώσουν καλά αποτελέσματα, ιδιαίτερα όταν το σύστημα εμφανίζει χαοτική συμπεριφορά. Έτσι, γίνεται αμέσως αντιληπτό το πλεονέκτημα των μη γραμμικών ΝΔ έναντι άλλων γνωστών γραμμικών μεθόδων. Μερικά παραδείγματα μη γραμμικότητας αποτελούν η , η sigmoid και η ευρέως χρησιμοποιούμενη Rectified Linear Unit (ReLU), οι μορφές των οποίων φαίνονται στο πιο κάτω σχήμα:





Εικόνα 18 Συναρτήσεις εισαγωγής μη γραμμικότητας. Με κόκκινη συμβολίζεται η  $\tanh$ , με πράσινο η  $\text{sigmoid}$  και με μπλε η  $\text{ReLU}$ .

Πιο κατάλληλη θεωρείται η  $\text{ReLU}$ , για το λόγο ότι απλοποιεί τον αλγόριθμο  $\text{back-propagation}$  και τον επιταχύνει, εφόσον οι πράξεις που χρειάζεται να εκτελεστούν είναι πολύ λιγότερες.

### 3.4 Dropout

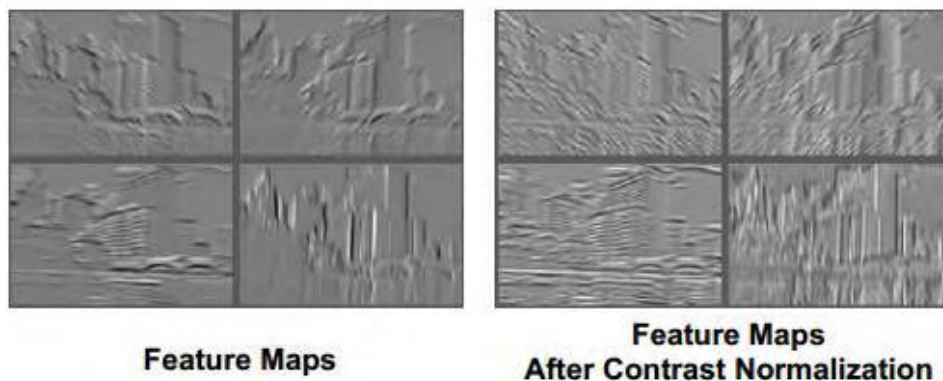
Υπάρχουν διάφορες μέθοδοι για να μειώσουμε τα σφάλματα εκπαίδευσης ενός Νευρωνικού Δικτύου, όπως για παράδειγμα να συγκρίνουμε τις προβλέψεις πολλών και διαφορετικών μοντέλων. Για μεγάλα Νευρωνικά Δίκτυα των οποίων η εκπαίδευση μπορεί να κρατήσει αρκετές ημέρες αυτή η μέθοδος είναι χρονοβόρα και σε ορισμένες περιπτώσεις ίσως και αδύνατη. Προς αντιμετώπιση αυτού του προβλήματος έχει ανα-

πτυχθεί κάποια μέθοδος σύγκρισης μοντέλων αρκετά αποτελεσματική, το υπολογιστικό κόστος της οποίας είναι πολύ χαμηλό. Η μέθοδος αυτή, η οποία ονομάζεται “Dropout” συνίσταται στην την ανάθεση ως “0” της εξόδου κάθε κρυφού νευρώνα με πιθανότητα 0.5. Οι νευρώνες που συμμετέχουν με αυτό τον τρόπο στο “Dropout” δεν συνεισφέρουν στη διάδοση προς τα εμπρός των σημάτων εκπαίδευσης και δεν συμμετέχουν στην διαδικασία του back-propagation. Έτσι, κάθε φορά που μια είσοδος παρουσιάζεται στο Δίκτυο, εκείνο χρησιμοποιεί διαφορετική αρχιτεκτονική αλλά όλες αυτές οι αρχιτεκτονικές μοιράζονται τα ίδια βάρη. Αυτή η τεχνική μειώνει τις περίπλοκες συν-προσαρμογές των νευρώνων κι έτσι το Δίκτυο αποκτά τη δυνατότητα εκμάθησης ισχυρότερων χαρακτηριστικών.

Κατά τη διαδικασία της επαλήθευσης (test) χρησιμοποιούμε όλους τους νευρώνες πολλαπλασιάζοντας όμως τις εξόδους τους με 0.5 προκειμένου να πάρουμε τον γεωμετρικό μέσο των κατανομών πρόβλεψης. Τέλος, αναφέρουμε ότι η χρήση του Dropout σχεδόν διπλασιάζει τον αριθμό των επαναλήψεων που απαιτούνται για σύγκλιση.

### 3.5 Κανονικοποίηση

Η κανονικοποίηση (Normalization), μπορεί να εφαρμοσθεί εντός ενός χαρακτηριστικού ή και μεταξύ ενός συνόλου χαρακτηριστικών και προσφέρει μικρότερη διακύμανση στα δεδομένα. Επίσης μπορεί να εφαρμοσθεί πριν ή μετά το pooling χωρίς ιδιαίτερη διαφορά μεταξύ των περιπτώσεων, ενώ δεν είναι απαραίτητη πάντοτε.



*Εικόνα 19-Εφαρμογή Κανονικοποίησης σε convolved Εικόνα*

### 3.6 Προ-εκπαίδευση

Κατά την εκπαίδευση ενός Συνελκτικού Νευρωνικού Δικτύου, η διαδικασία της οπισθο-διάδοσης σφάλματος συνίσταται στη διάδοση κλίσεων στα προηγούμενα επίπεδα του Δικτύου για τη διόρθωση των βαρών των φίλτρων που έχουν υπολογιστεί. Μερικά από τα πρώτα επίπεδα ενός πολύ-επίπεδου Νευρωνικού Δικτύου αλλάζουν ελάχιστα την τιμή τους λόγω της μειωμένης διάδοσης της κλίσης (εκπαιδεύονται ανεπαρκώς) και αυτό μπορεί να αποτελέσει σημαντικό πρόβλημα. Η λύση έρχεται από τη διάθεση ενός μεγάλου αριθμού δειγμάτων, τουλάχιστον 5000 δείγματα ανά κλάση και μετά από αρκετό χρόνο εκπαίδευσης το δίκτυο καταφέρνει να αποκτήσει την τελική και αποδοτική του μορφή.

Όταν τα δεδομένα εκπαίδευσης είναι ανεπαρκή, απαιτείται προ-εκπαίδευση και η τεχνική που ακολουθείται είναι η παροχή στο Δίκτυο διανυσμάτων-εικόνων μεγάλης βάσης δεδομένων από τομέα σχετικό με την εφαρμογή που θέλουμε να δημιουργήσουμε, ώστε να επέλθει σύγκλιση των παραμέτρων του Δικτύου. Κατά την εκπαίδευση, δίνονται ως είσοδοι εικόνες-διανύσματα από τα δεδομένα που έχουμε στη διάθεσή μας. Στην περίπτωση των εικόνων, ένα επιπλέον βήμα που μπορεί να ακολουθηθεί, είναι η επέκταση της βάσης δεδομένων μέσα από επεξεργασία των εικόνων εκμεταλλευόμενοι του στοιχείου της “τοπικότητας” των Συνελκτικών Νευρωνικών Δικτύων. Έτσι, για κάθε εικόνα μπορούμε να δημιουργήσουμε μερικά ελαφρώς παραλλαγμένα αντίγραφα, όπου έχουμε αλλάξει τις τιμές αριθμού τυχαίων pixel πάνω στην εικόνα, πχ ανάθεση σε ορισμένα pixel της τιμής “0” ή “1”, περιστροφές των εικόνων για ελάχιστες μοίρες (-10 έως +10 μοίρες) και μετατόπιση των εικόνων σε δυο διευθύνσεις με βήμα ένα μικρό αριθμό pixels (translation). Με αυτό τον τρόπο, δίνουμε στο Δίκτυο μια άλλη “οπτική” της ίδιας πληροφορίας, όπως συμβαίνει και στις εικόνες RGB όπου το Δίκτυο έχει στη διάθεσή του τρεις διαφορετικές εκδοχές της ίδιας εικόνας, με αποτέλεσμα η εκπαίδευσή του να γίνεται πιο αποτελεσματική.

Γενικά η προ-εκπαίδευση:

- Κάνει την βελτιστοποίηση του Δικτύου πιο εύκολη
- Μειώνει την υπερ-προσαρμογή (overfitting)

Σε άλλες μορφές νευρωνικών δικτύων, για παράδειγμα στους Autoencoders και τις RBMs, η προ-εκπαίδευση είναι αρκετά σημαντική αφού ένα από τα θεμελιώδη ζητήματα για την ορθή εκπαίδευσή τους αποτελεί η ανάθεση κατάλληλων αρχικών τιμών των βαρών τους.

### 3.7 Πλήρως διασυνδεδεμένο επίπεδο

Στο τέλος κάθε συνελκτικού δικτύου υπάρχουν ένα ή περισσότερα πλήρως διασυνδεδεμένα επίπεδα που εκτελούν τη βασική λειτουργία του δικτύου (πχ ταξινόμηση), σε αντίθεση με τα προηγούμενα επίπεδα που απλά κάνουν εξαγωγή χαρακτηριστικών. Εκτός από τη ταξινόμηση η πρόσθεσή τους αποτελεί ένα φθηνό τρόπο εκμάθησης μη γραμμικών συνδυασμών των χαρακτηριστικών. Τα χαρακτηριστικά των συνελκτικών και των επιπέδων συγκέντρωσης μπορεί να είναι καλά για την εργασία ταξινόμησης, αλλά οι συνδυασμοί αυτών των χαρακτηριστικών μπορεί να είναι ακόμη καλύτεροι. Με περισσότερη ανάλυση μπορεί κάποιος να καταλήξει στο συμπέρασμα ότι τα πλήρως συνδεδεμένα επίπεδα μπορούν να θεωρηθούν συνελκτικά επίπεδα που υλοποιούν συνελίξεις  $1 \times 1$ , διευκολύνοντας έτσι την αλγοριθμική υλοποίησή τους.

### 3.8 Εκπαίδευση

Για την εκπαίδευση των συνελκτικών δικτύων χρησιμοποιείται ο αλγόριθμος οπισθοδρομικής διάδοσης του σφάλματος όπως και στα κλασικά νευρωνικά δίκτυα με μερικές παραλλαγές. Ισχύουν, δηλαδή, στη γενική τους μορφή τα βήματα της ενότητας 2.3. Επειδή η δομή του συνελκτικού νευρωνικού δικτύου διαφέρει, επηρεάζονται και οι εξισώσεις του αλγόριθμου, χωρίς όμως να αλλοιώνονται οι βασικές του ιδιότητες. Ως συνάρτηση σφάλματος ορίζεται πάλι το άθροισμα των τετραγώνων της απόκλισης των εξόδων του συνελκτικού νευρωνικού δικτύου από την επιθυμητή έξοδο

$$E = \sum_{p=1}^P \left( \frac{1}{2} (t_p - a_p^L)^2 \right)$$

ενώ οι εξισώσεις του αλγόριθμου διαμορφώνονται με τον παρακάτω τρόπο:

$$\begin{aligned}
\frac{\partial E}{\partial w_{x,y}^l} &= \sum_{x'} \sum_{y'} \delta_{x',y'}^l f \left( o_{x'-x,y'-y}^{l-1} \right) \\
&= \delta_{x,y}^l * f \left( o_{-x,-y}^{l-1} \right) \\
&= \delta_{x,y}^l * f \left( \text{rot}_{180^\circ} \left( o_{x,y}^{l-1} \right) \right)
\end{aligned}$$

Παρατηρούμε ότι στην ουσία πρόκειται για την ίδια σχέση, απλά εκφράζεται με τη μορφή συνέλιξης, αφού έχουν σαν είσοδο στο δίκτυο εικόνες και όχι μονοδιάστατα διανύσματα.

$$\delta_{x,y}^l = \frac{\partial E}{\partial o_{x,y}^l}$$

Όσον αφορά τα  $\delta_{x,y}^l$  ορίζονται ως ο ρυθμός μεταβολής της συνάρτησης σφάλματος ως προς τις εξόδους του επιπέδου

Και αποδεικνύεται εύκολα ότι τελικά ισχύει

$$\begin{aligned}
\frac{\partial E}{\partial o_{x,y}^l} &= \sum_{x'} \sum_{y'} \delta_{x',y'}^{l+1} w_{x'-x,y'-y}^{l+1} f' \left( o_{x,y}^l \right) \\
&= \delta_{x,y}^{l+1} * w_{-x,-y}^{l+1} f' \left( o_{x,y}^l \right) \\
&= \delta_{x,y}^{l+1} * \text{rot}_{180^\circ} \left( w_{x,y}^{l+1} \right) f' \left( o_{x,y}^l \right)
\end{aligned}$$

Τα παραπάνω είναι εμφανές ότι αφορούν την οπισθοδρομική διάδοση στα συνεκτικά επίπεδα καθώς στα πλήρως συνδεδεμένα επίπεδα ισχύουν οι εξισώσεις της ενότητας 2.3. Στα επίπεδα συγκέντρωσης δε λαμβάνει μέρος κάποια εκμάθηση, αλλά ακολουθείται η αντίστροφη διαδικασία από αυτή της δειγματοληψίας. Το σφάλμα υπολογίζεται από την τιμή της "νικήτριας μονάδας". Για να είναι διαθέσιμη, κατά την εμπρόσθια διάδοση αποθηκεύεται ο δείκτης της θέσης της, ο οποίος χρησιμοποιείται κατά την οπισθοδρομική. Στη συγκέντρωση μεγίστου το σφάλμα ανατίθεται στη μονάδα που συνεισέφερε στον αλγόριθμο ("νικήτρια μονάδα") και περιείχε το μέγιστο της "γειτονιάς" της. Αντίθετα, στη συγκέντρωση μέσου όρου το σφάλμα πολλαπλασιάζεται με τον όρο  $1/(N*N)$  και ανατίθεται σε όλες της μονάδες της "γειτονιάς". Κάθε μονάδα, δηλαδή, αποκτάει την ίδια τιμή.

## **4 Μια εφαρμογή της μηχανικής μάθησης στη αιματολογική διάγνωση**

### **4.1 Εκπαίδευση**

Η μηχανική μάθηση έχει υποστεί σημαντική ανάπτυξη την τελευταία δεκαετία και χρησιμοποιείται με επιτυχία σε πολλές έξυπνες εφαρμογές που καλύπτουν ένα ευρύ φάσμα προβλημάτων που σχετίζονται με τα δεδομένα. Ένα από τα πιο ενδιαφέροντα ερωτήματα είναι αν η μηχανική μάθηση μπορεί να εφαρμοστεί με επιτυχία στον τομέα της ιατρικής διαγνωστικής. Επιπλέον, υπάρχει ένα ερώτημα σχετικά με το είδος των δεδομένων που χρειάζονται. Αρκετά παραδείγματα επιτυχημένων εφαρμογών των μεθόδων μηχανικής μάθησης σε εξειδικευμένους ιατρικούς τομείς υπάρχουν.

## **5 Βιβλιογραφία**

1. A. Dar, D. R. (2018). A COMPREHENSIVE STUDY ON CLOUD COMPUTING.
2. US-CERT. (2012). The Basics of Cloud Computing.
3. Y. LeCun, Y. B. (2015). Deep Learning. Nature.