



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΑΝΑΛΥΣΗ ΚΑΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΝΕΥΡΩΝΙΚΩΝ

ΔΙΚΤΥΩΝ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΟΥΝΤΑΙ ΣΤΗΝ

ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ ΕΙΚΟΝΩΝ

Διπλωματική Εργασία

Ευθυμίου Νικόλαος

Επιβλέπουσα: Τσαλαπάτα Χαρίκλεια

Βόλος έτος 2021



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΑΝΑΛΥΣΗ ΚΑΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΝΕΥΡΩΝΙΚΩΝ

ΔΙΚΤΥΩΝ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΟΥΝΤΑΙ ΣΤΗΝ

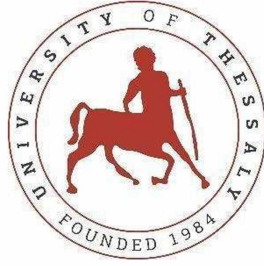
ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ ΕΙΚΟΝΩΝ

Διπλωματική Εργασία

Ευθυμίου Νικόλαος

Επιβλέπουσα: Τσαλαπάτα Χαρίκλεια

Βόλος έτος 2021



UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

**ANALYSIS AND OPTIMIZATION OF NEURAL
NETWORKS USED FOR IMAGE CLASSIFICATION**

Diploma Thesis

Nikolaos Efthymiou

Supervisor: Tsalapata Hariklia

Volos year 2021

Εγκρίνεται από την Επιτροπή Εξέτασης:

Επιβλέπουσα **Τσαλαπάτα Χαρίκλεια**

Ε.ΔΙ.Π., Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών
Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος **Δασκαλοπούλου Ασπασία**

ΕΠΙΚΟΥΡΗ ΚΑΘΗΓΗΤΡΙΑ, Τμήμα Ηλεκτρολόγων Μηχανικών και
Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος **Τουσίδου Ελένη**

Ε.ΔΙ.Π., Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών
Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Ημερομηνία έγκρισης: 09-02-2021

ΕΥΧΑΡΙΣΤΙΕΣ ή ΣΧΟΛΙΑ

Ευχαριστώ τη διδάσκουσα Τσαλαπάτα Χαρίκλεια για τη συμβολή της στην εκπόνηση της διπλωματικής μου εργασίας και τις διδάσκουσες Δασκαλοπούλου Ασπασία και Τουσίδου Ελένη για το ενδιαφέρον που έδειξαν όσον αφορά το θέμα που επέλεξα να ασχοληθώ.

Επιπλέον, ευχαριστίες αξίζει να αποδοθούν στους διδάσκοντες του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Ηλεκτρονικών Υπολογιστών του Πανεπιστημίου Θεσσαλίας για το ενδιαφέρον που δείχνουν καθημερινά στην ορθή διδασκαλία και επιστημονική κατάρτιση των φοιτητών του τμήματος.

Τέλος, ευχαριστώ την οικογένεια μου για τη στήριξη τους καθ όλη τη διάρκεια των σπουδών μου.

**ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΠΕΡΙ ΑΚΑΔΗΜΑΪΚΗΣ ΔΕΟΝΤΟΛΟΓΙΑΣ ΚΑΙ ΠΝΕΥΜΑΤΙΚΩΝ
ΔΙΚΑΙΩΜΑΤΩΝ**

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα διπλωματική εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας, αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο/Η Δηλών/ούσα

(Υπογραφή) 

Νικόλαος Ευθυμίου

Ημερομηνία 15/1/2021

ΠΕΡΙΛΗΨΗ

Η πτυχιακή αυτή έχει ως σκοπό να μελετήσει την συμπεριφορά των συνελκτικών νευρωνικών δικτύων σε πληθώρα καταστάσεων. Η εργασία αρχίζει με μια εισαγωγή στα νευρωνικά δίκτυα με σκοπό την ανάλυση του τρόπου λειτουργίας τους. Συνεχίζει με την ανάλυση συνελκτικών νευρωνικών δικτύων και τη βελτιστοποίηση αυτών. Και τέλος αναλύεται η μέθοδος “μεταφορά μάθησης” από ένα δίκτυο σε ένα άλλο, ενώ γίνεται αναφορά και χρήση του διάσημου δικτύου VGG-16.

Η εκπαίδευση νευρωνικών δικτύων είναι μια πολύπλοκη διαδικασία με πληθώρα παραμέτρων. Πολλές φορές η απόκτηση ορθής κρίσης όσον αφορά την επιλογή αυτών μπορεί να χρειαστεί χρόνια εμπειρίας. Η πτυχιακή αυτή στοχεύει στην ανάλυση των διαφόρων παραμέτρων με σκοπό να επιταχύνει την διαδικασία μάθησης των νέων επιστημών που θα ασχοληθούν με το αντικείμενο των νευρωνικών δικτύων.

ABSTRACT

The purpose of this work is to analyse and optimize Convolutional neural networks used for image classification. After introduction, the way a neural network works is explained. It continues with Convolutional neural network analysis and optimization. Then, the method “Transfer learning” and the way it can boost a CNN is examined. For this purpose, the popular CNN neural network VGG-16 is used.

Neural network training can be extremely complex with many parameters. It could take a lot of time for someone to wisely choose the right values. This work analyzes a variety of them with the ultimate goal being to boost the learning process of new scientists.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΕΡΙΛΗΨΗ	1
ABSTRACT	3
ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ	5
ΛΙΣΤΑ ΣΧΗΜΑΤΩΝ	7
ΚΕΦΑΛΑΙΟ 1	9
ΕΙΣΑΓΩΓΗ	
1.1 Τα νευρωνικά δίκτυα	9
1.2 Κατηγοριοποίηση εικόνας	10
1.3 Δομη εργασίας	10
ΚΕΦΑΛΑΙΟ 2	12
ΕΙΣΑΓΩΓΗ ΣΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ	
2.1 Πολυεπίπεδοι νευρώνες	12
2.2 Νευρώνας	14
2.3 Bias	14
2.4 Συναρτήσεις ενεργοποίησης	15
2.5 Δίκτυο πολλαπλών νευρώνων	16
ΚΕΦΑΛΑΙΟ 3	27
ΣΥΝΕΛΙΚΤΙΚΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ	
3.1 Συνελικτικά νευρωνικά δίκτυα	18
3.2 Στρώματα συνέλιξης	18
3.3 Στρώμα συσσώρευσης	20
ΚΕΦΑΛΑΙΟ 4	22
ΕΚΠΑΙΔΕΥΣΗ ΝΕΥΡΩΝΙΚΟΥ ΔΙΚΤΥΟΥ	
4.1 Αλγόριθμος εκπαίδευσης νευρωνικού δικτύου	22
4.2 Υπερμάθηση	24
4.3 Εποχές και παρτίδες δεδομένων	25
4.4 Dropout	25
4.5 Batch normalization	28
4.6 Δεδομένα	29

4.7 Κίνητρο χρήσης μεταφοράς μάθησης και δίκτυο VGG-16	32
4.8 Μεταφορά μάθησης	35
4.9 Τρόποι εφαρμογής μεταφοράς μάθησης	36
ΚΕΦΑΛΑΙΟ 5	38
<i>ΠΕΙΡΑΜΑΤΙΣΜΟΣ ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ ΣΥΝΕΛΙΚΤΙΚΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ</i>	
5.1 Βασική αρχιτεκτονική δικτύου και δεδομένα	38
5.2 Data augmentation	41
5.3 Μέγεθος παρτίδας	41
5.4 Dropout	42
5.5 Batch normalization	43
5.6 Χρήση της μεθόδου μεταφοράς μάθησης	44
ΣΥΜΠΕΡΑΣΜΑΤΑ	47
ΒΙΒΛΙΟΓΡΑΦΙΑ	48

ΛΙΣΤΑ ΣΧΗΜΑΤΩΝ

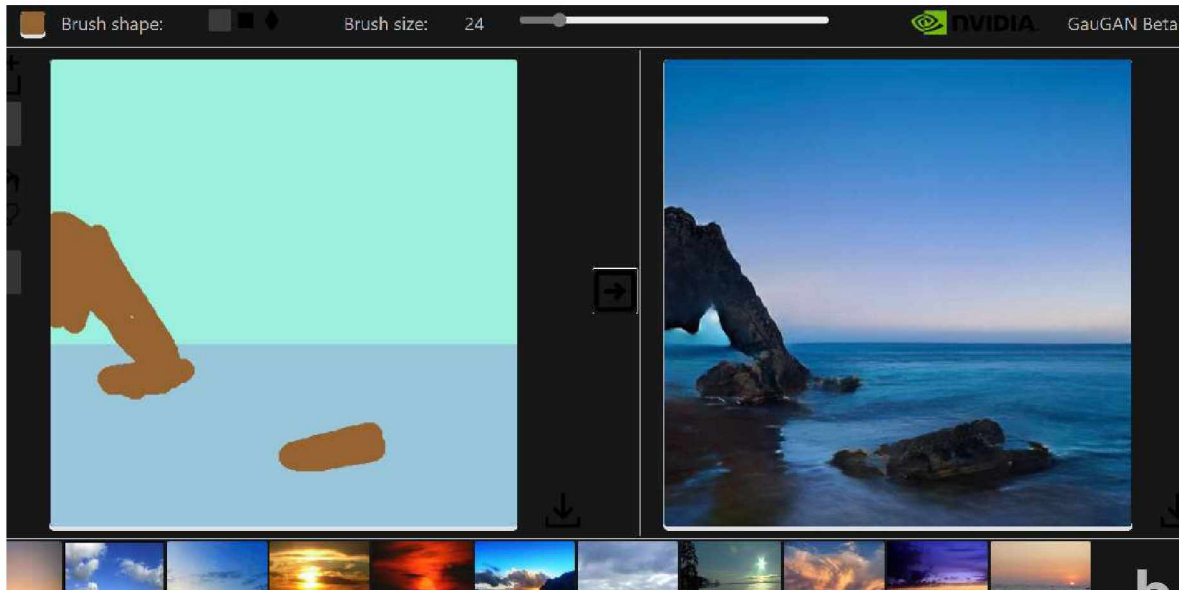
1.1 Βραχώδες τοπίο σε θάλασσα με τη χρήση GauGAN.	10
2.1 Πολυεπίπεδο δίκτυο νευρώνων σχεδιασμένο με τη χρήση NN-SVG.	12
2.2 Παράδειγμα νευρωνικού δικτύου που διαχωρίζει 2 είδη σε διαφορετικές κλάσεις.	13
2.3 Παράδειγμα νευρωνικού δικτύου που προβλέπει την τιμή του y άξονα με βάση την τιμή του x άξονα.	13
2.4 Παράδειγμα λειτουργίας ενός νευρώνα.	14
2.5 Παραδείγματα συναρτήσεων που μπορούν να προκύψουν με και χωρίς τη χρήση bias.	15
2.6 Παραδείγματα συναρτήσεων ενεργοποίησης νευρώνων.	16
2.7 Παράδειγμα εξόδου πολυεπίπεδου νευρωνικού δικτύου για μονοδιάστατα δεδομένα.	17
3.1 Παράδειγμα ενός απλοικού συνελκτικού δικτύου.	18
3.2 Παράδειγμα υπολογισμού φιλτραρισμένου στρώματος.	19
3.3 Παράδειγμα βήματος φίλτρου για διαφορετικές τιμές Stride.	20
3.4 Παράδειγμα υπολογισμού στρώματος συσσώρευσης.	21
4.1 Παράδειγμα νευρωνικού δικτύου για κατηγοριοποίηση ανθών.	23
4.2 Παράδειγμα εξόδου δικτύου και υπολογισμός σφάλματος.	23
4.3 Παράδειγμα δικτύου που έχει “υπερμάθει”.	24
4.4 Παράδειγμα εξόδου δικτύου όπου [1 1] η είσοδος και 1 η επιθυμητή έξοδος.	26
4.5 Παράδειγμα εξόδου δικτύου όπου ένας νευρώνας έχει παγώσει.	27
4.6 Παράδειγμα εξόδου δικτύου.	28
4.7 Παραδείγματα τυχαίων κατανομών εξόδου που μπορεί να προκύψουν σε κάποιο επίπεδο.	28
4.8 Παράδειγμα κανονικής κατανομής.	29
4.9 Παράδειγμα γεωμετρικής επεξεργασίας εικόνας.	31
4.10 Παράδειγμα χρωματικής επεξεργασίας εικόνας.	32
4.11 Αρχιτεκτονική VGG-16.	34
4.12 Παράδειγμα μεταφοράς εκπαιδευμένων παραμέτρων.	35
4.13 Παραλλαγές χρήσης μεταφοράς μάθησης.	37
5.1 Αρχιτεκτονική του βασικού δικτύου.	39
5.2 Πορτοκαλί γραμμή: δίκτυο που χρησιμοποιεί Data augmentation και μπλε είναι το δίκτυο που δεν χρησιμοποιήθηκε data augmentation.	41
5.3 Batch size 32: πορτοκαλί Batch size 16: κόκκινο.	42
5.4 Dropout comparison.	43
5.5 Batch normalization comparison.	44

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

1.1 Τα νευρωνικά δίκτυα

Τα νευρωνικά δίκτυα απασχολούν πληθώρα επιστημόνων στον 21ο αιώνα, καθώς υπάρχουν σε τεράστια ποικιλία εφαρμογών. Κύριο χαρακτηριστικό τους είναι ότι οι αρχές και οι βάσεις τους στηρίζονται στον τρόπο λειτουργίας του εγκεφάλου και έχουν άμεση συνάφεια με τον μαθηματικό τρόπο σκέψης. Συγκεκριμένα, στη βιολογία το νευρωνικό δίκτυο είναι στην ουσία ένα νευρικό σύστημα στους ζωντανούς οργανισμούς με το οποίο επιτελούν ορισμένες λειτουργίες όπως η αναγνώριση του εξωτερικού κόσμου, η εξάσκηση και μνήμη. Εμπνευσμένοι από αυτό το μοντέλο οι επιστήμονες εφηύραν στον κλάδο της πληροφορικής το -τεχνητό- νευρωνικό δίκτυο, το οποίο έχει τη δυνατότητα σε ηλεκτρονικό υπολογιστή να αναγνωρίζει πρότυπα σε μια βάση δεδομένων και να παράγει ένα μοντέλο για αυτά. Τα νευρωνικά δίκτυα έχουν σκοπό να αυτοματοποιούν μια πληθώρα διαδικασιών. Τεχνολογίες όπως αυτόματη οδήγηση αυτοκινήτων, αναγνώριση προσώπου, αυτόματη μετάφραση και εντοπισμός όγκων καρκινικού τύπου είναι μερικές από τις χιλιάδες εφαρμογές που βρίσκουν χρήση τα νευρωνικά δίκτυα βαθιάς μάθησης. Μεγάλο ενδιαφέρον υπάρχει ακόμη και στα νευρωνικά δίκτυα που χρησιμοποιούνται για καλλιτεχνικούς σκοπούς, όπως η παρακάτω εικόνα που δημιουργήθηκε με τη χρήση GauGAN, ένα project βαθιάς μάθησης, της εταιρίας Nvidia.



Σχήμα 1.1 Βραχώδες τοπίο σε θάλασσα με τη χρήση GauGAN.

1.2 Κατηγοριοποίηση εικόνας

Η συγκεκριμένη εφαρμογή των νευρωνικών δικτύων για την οποία θα γίνει εκτεταμένη αναφορά στην παρούσα πτυχιακή είναι η κατηγοριοποίηση εικόνων. Δηλαδή, χρήση νευρωνικών δικτύων για την εκπαίδευση ενός μοντέλου που έχει την ικανότητα να αναγνωρίζει εικόνες και να τις τοποθετεί σε κλάσεις. Τα πεδία που καλύπτει η κατηγοριοποίηση εικόνων μπορεί να είναι η ιατρική (π.χ. ανίχνευση παθήσεων όπως ο καρκίνος σε ακτινογραφίες), τα συστήματα ασφαλείας (π.χ. αναγνώριση προσώπου) ακόμα και η πλοήγηση (π.χ. αναγνώριση αντικειμένων στο χώρο). Άλλες ενδιαφέρουσες εφαρμογές της κατηγοριοποίησης εικόνων εντοπίζονται στην αναγνώριση και στις προτάσεις παρόμοιων αντικειμένων από φωτογραφία μέσα από τις δημοφιλείς μηχανές αναζήτησης.

1.3 Δομη εργασίας

Τα κεφάλαια έχουν οργανωθεί με τρόπο, ώστε να καλυφθεί με ευρύτητα η θεματική της παρούσας διπλωματικής εργασίας. Μετά την εισαγωγή, στο 2ο κεφάλαιο της διπλωματικής αναλύεται η λειτουργία των απλών γραμμικών νευρωνικών δικτύων προκειμένου να κατανοηθούν καλύτερα στα επόμενα κεφάλαια τα συνελκτικά δίκτυα. Στο 3ο κεφάλαιο επεξηγείται ο τρόπος λειτουργίας των συνελκτικών δικτύων όπου υπάρχουν δύο νέα στρώματα νευρώνων το συνελκτικό και το συσσωρευτικό. Στο 4ο

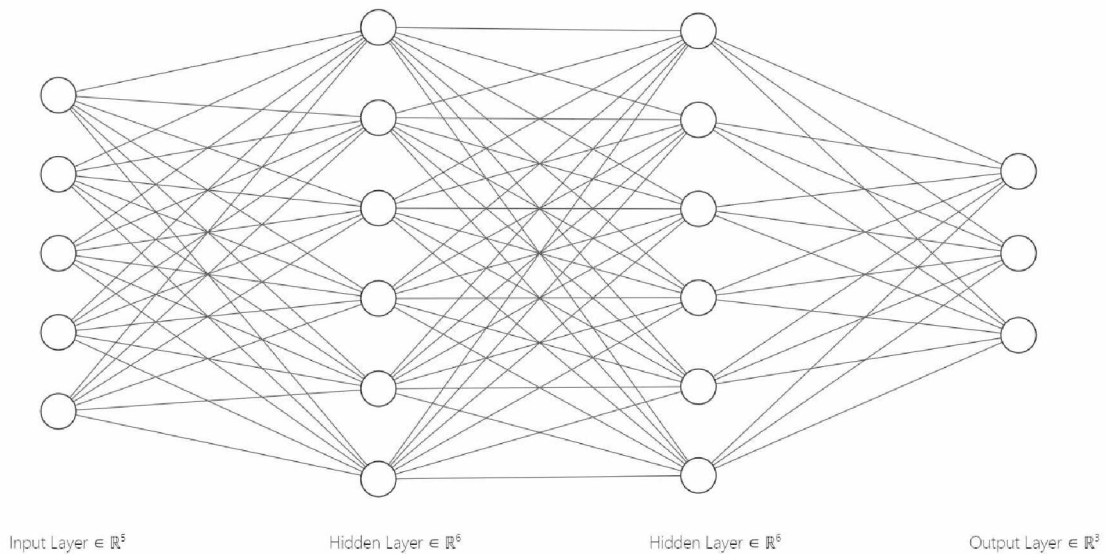
κεφάλαιο γίνεται αναφορά και επεξήγηση διαφόρων πρακτικών που εφαρμόζονται σε νευρωνικά δίκτυα, ενώ παρακάτω θα χρησιμοποιηθούν για να μετρηθεί η αποτελεσματικότητα τους στα συνελκτικά δίκτυα. Στο 5ο κεφάλαιο παρουσιάζεται η κατασκευή ενός νευρωνικού δικτύου σαν “βάση”, με δεδομένα που αφορούν σκύλους και γάτες, πάνω στο οποίο εφαρμόζονται οι διάφορες πρακτικές που αναφέρονται στο 4ο κεφάλαιο και γίνονται μετρήσεις για την αποτελεσματικότητα αυτών.

ΚΕΦΑΛΑΙΟ 2

ΕΙΣΑΓΩΓΗ ΣΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

2.1 Πολυεπίπεδοι νευρώνες

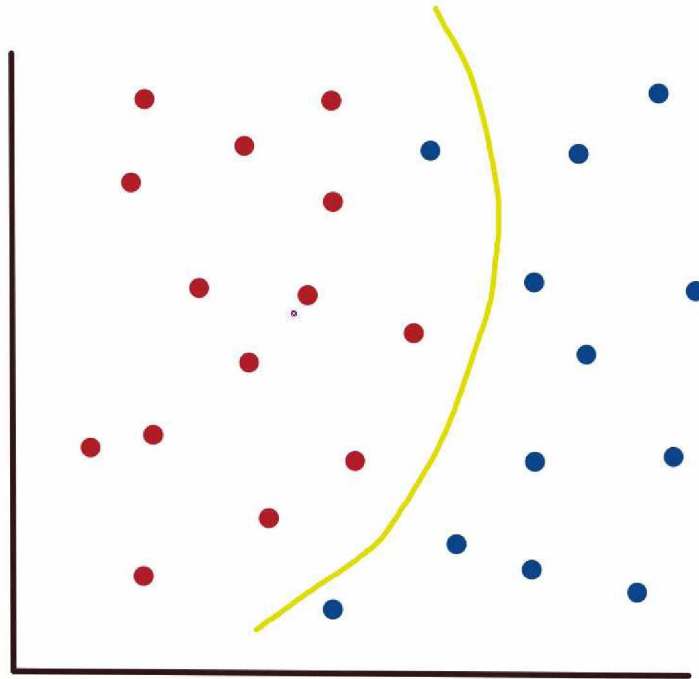
Ένα νευρωνικό δίκτυο πολυεπίπεδων νευρώνων αποτελείται από πολλαπλές στιβάδες νευρώνων ενωμένες μεταξύ τους [1] Σχήμα 2.1 [2].



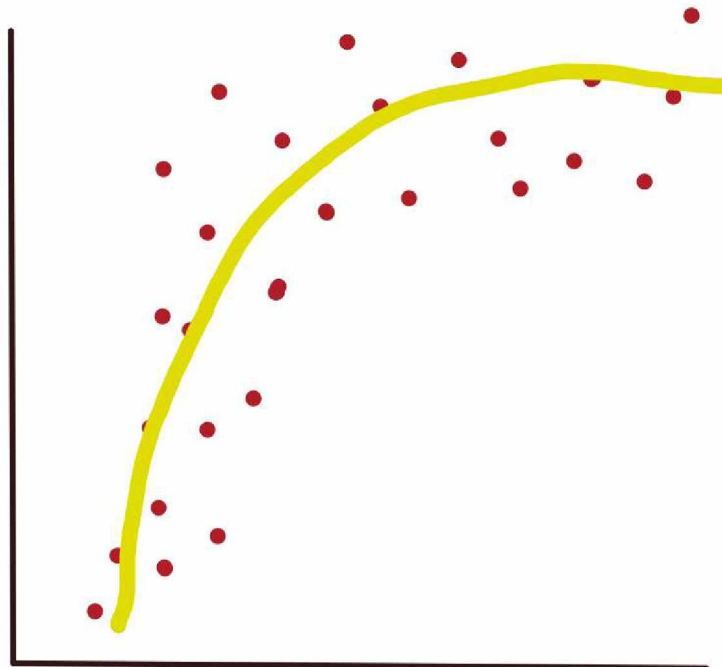
Σχήμα 2.1 Πολυεπίπεδο δίκτυο νευρώνων σχεδιασμένο με τη χρήση NN-SVG.

Τα δίκτυα πολυεπίπεδων νευρώνων μπορούν να χρησιμοποιηθούν με διάφορους τρόπους όπως:

- Ταξινόμηση δεδομένων σε διαφορετικές κλάσεις Σχήμα 2.2.
- Προβλήματα παλινδρόμησης Σχήμα 2.3.



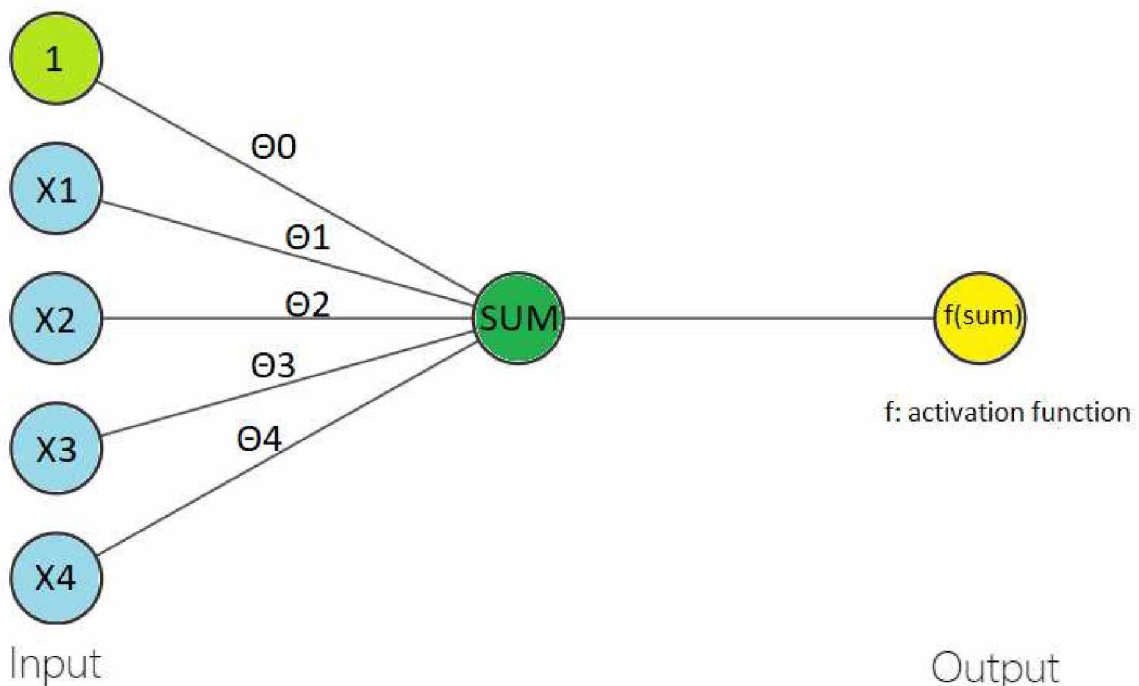
Σχήμα 2.2 Παράδειγμα νευρωνικού δικτύου που διαχωρίζει 2 είδη σε διαφορετικές κλάσεις.



Σχήμα 2.3 Παράδειγμα νευρωνικού δικτύου που προβλέπει την τιμή του y άξονα με βάση την τιμή του x άξονα.

2.2 Νευρώνας

Ο κάθε νευρώνας ξεχωριστά μπορεί να έχει πολλαπλές εισόδους ανάλογα με την αρχιτεκτονική του δικτύου που θα αποφασιστεί από τον προγραμματιστή, οι εισοδοι μπορούν να έχουν οποιαδήποτε τιμή, ενώ το καθένα από αυτά πολλαπλασιάζεται με ένα βάρος Θ το οποίο ονομάζεται και παράμετρος. Στη συνέχεια, οι εισοδοι πολλαπλασιάζονται με τα βάρη που αντιστοιχούν σε κάθε μια και στη συνέχεια τα αποτελέσματα τους προστίθενται. Το αποτέλεσμα του αθροίσματος το χρησιμοποιούμε σαν παράμετρο σε μια συνάρτηση της επιλογής μας(ανάλογα με την αρχιτεκτονική του δικτύου) όπου και αποκτάμε την τελική έξοδο του νευρώνα σχήμα 2.4.

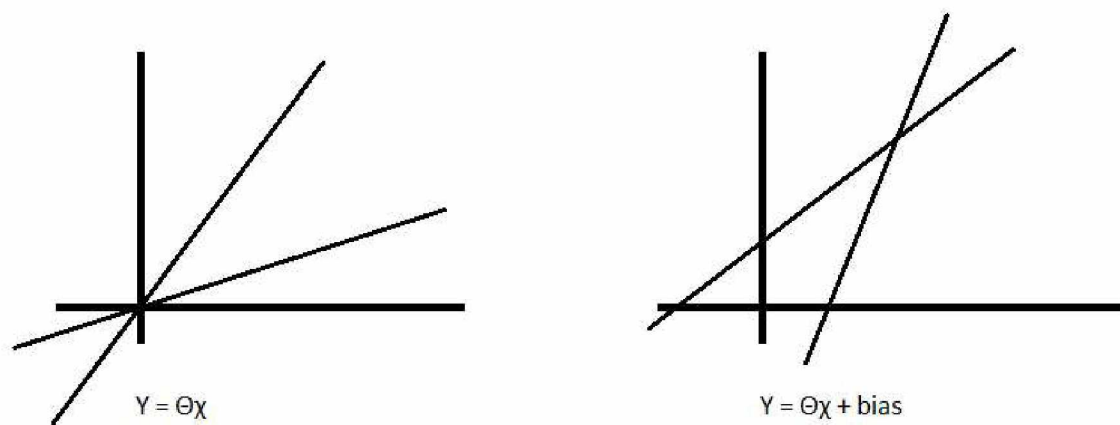


Σχήμα 2.4 Παράδειγμα λειτουργίας ενός νευρώνα.

Η συνάρτηση υπολογισμού εξόδου ενός νευρώνα είναι: $f\left(\sum_{i=1}^n x_i \Theta_i\right)$.

2.3 Bias

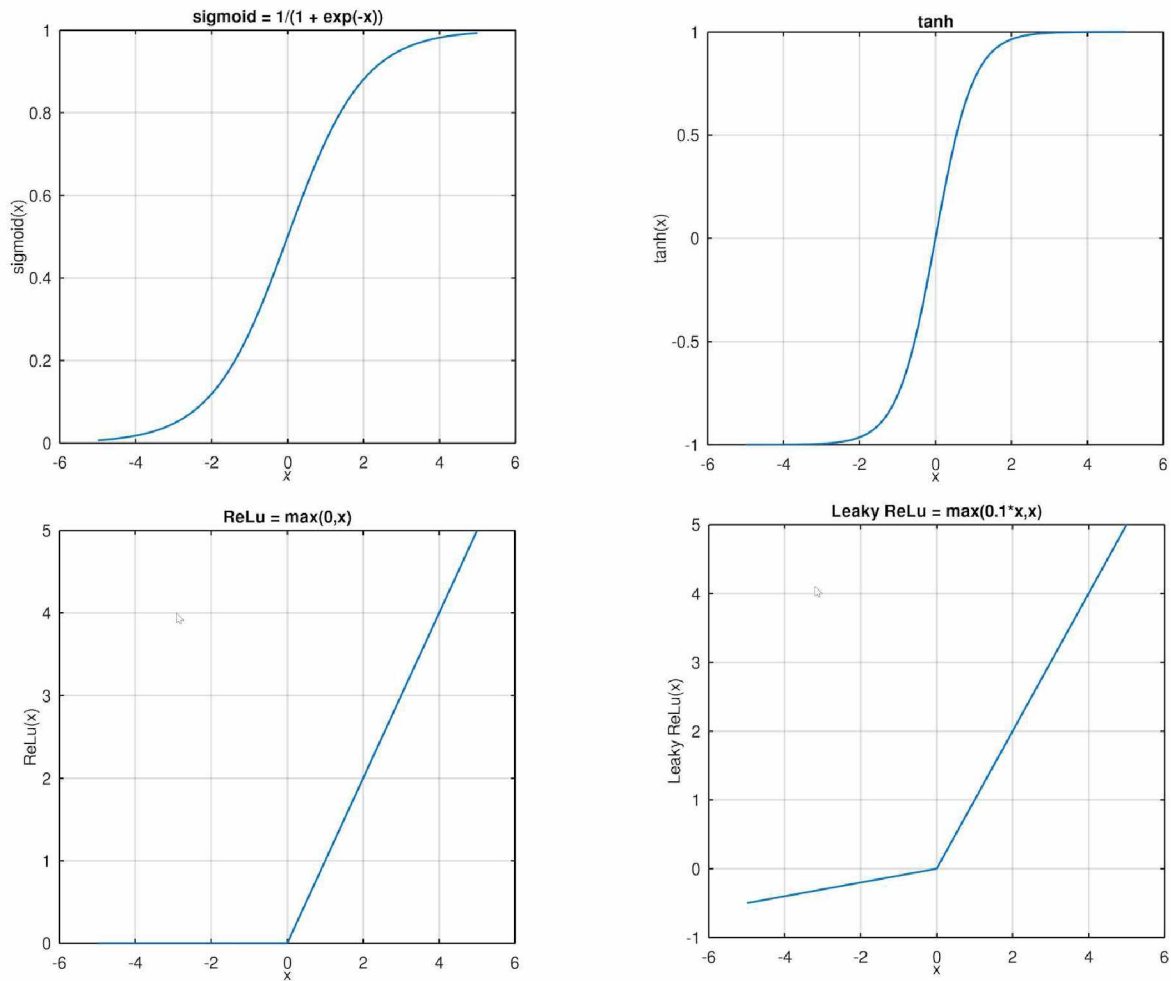
Πέραν της μη ντετερμινιστικής εισόδου, τις περισσότερες φορές, εκ κατασκευής υπάρχει μια επιπλέον είσοδος τιμής 1 πολλαπλασιασμένη με την ανάλογη παράμετρο Θ , όπως φαίνεται στο σχήμα 2.4. Η χρήση αυτού δίνει τη δυνατότητα στον νευρώνα να αυξάνει την πολυπλοκότητα των εξόδων. Παράδειγμα στο σχήμα 2.5.



Σχήμα 2.5 Παραδείγματα συναρτήσεων που μπορούν να προκύψουν με και χωρίς τη χρήση bias.

2.4 Συναρτήσεις ενεργοποίησης

Όπως η είσοδος bias, έτσι και η συνάρτηση ενεργοποίησης χρησιμοποιείται για να αυξήσει την πολυπλοκότητα της εξόδου. Παραδείγματα συναρτήσεων ενεργοποίησης φαίνονται στο σχήμα 2.6 [3].

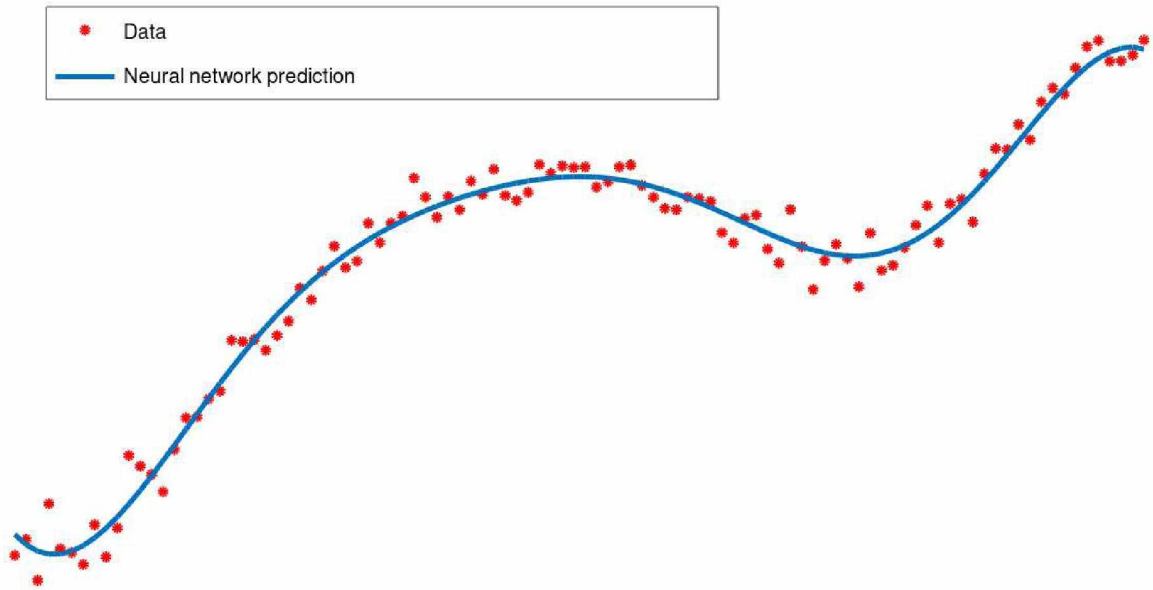


Σχήμα 2.6 Παραδείγματα συναρτήσεων ενεργοποίησης νευρώνα.

Η επιλογή συνάρτησης ενεργοποίησης έχει να κάνει με τη φύση του προβλήματος και το επίπεδο των νευρώνων. Συνήθως συναρτήσεις όπως η ReLU επιλέγονται για τα ενδιάμεσα επίπεδα, ενώ συναρτήσεις που μετασχηματίζουν την έξοδο στο διάστημα $[0,1]$ όπως η σιγμοειδής, είναι ιδιαίτερα χρήσιμες και πολλές φορές επιλέγονται για το τελικό επίπεδο, διότι οι έξοδοι που παράγουν μπορούν να χρησιμοποιηθούν και ως "πιθανότητα".

2.5 Δίκτυο πολλαπλών νευρώνων

Πολλαπλοί νευρώνες ενωμένοι μεταξύ τους έχουν την δυνατότητα δημιουργίας σύνθετης εξόδου, όπως φαίνεται στο σχήμα 2.7.



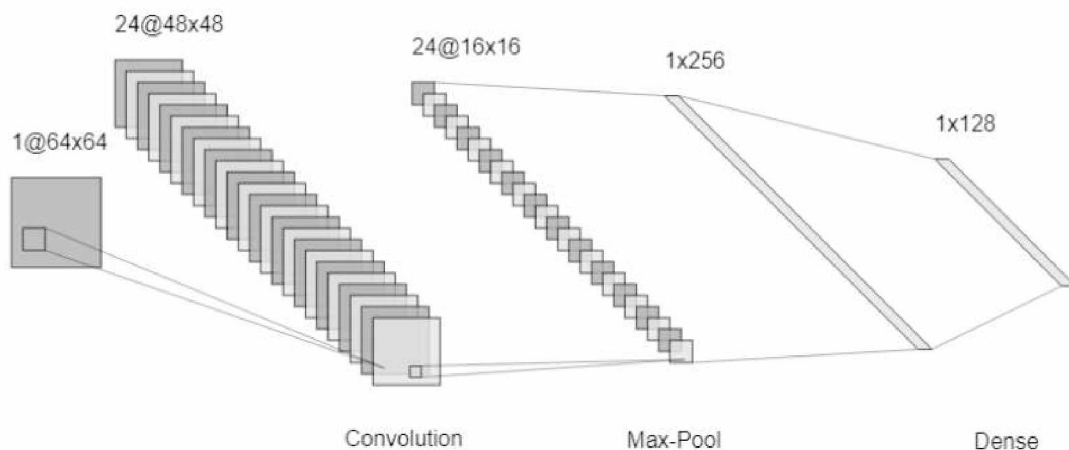
Σχήμα 2.7 Παράδειγμα εξόδου πολυεπίπεδου νευρωνικού δικτύου για μονοδιάστατα δεδομένα.

ΚΕΦΑΛΑΙΟ 3

ΣΥΝΕΛΙΚΤΙΚΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

3.1 Συνελικτικά νευρωνικά δίκτυα

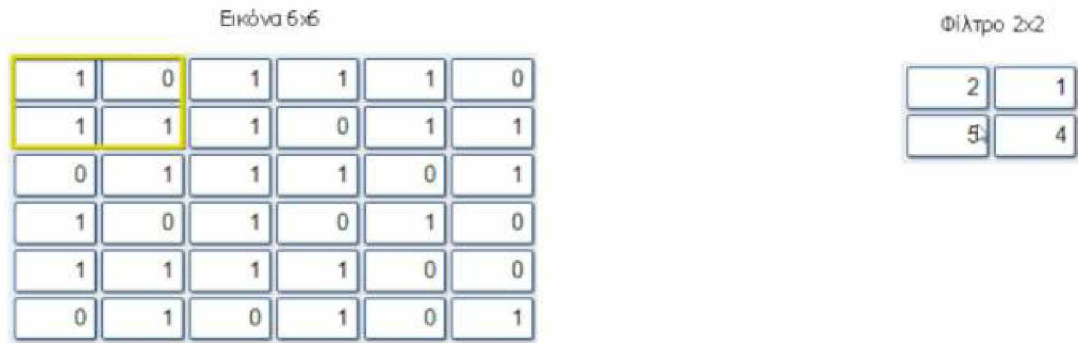
Τα συνελικτικά νευρωνικά δίκτυα [σχήμα 3.1] είναι μια διαφοροποίηση των απλοϊκών πολυεπίπεδων νευρωνικών δικτύων και χρησιμοποιούνται κατα κόρον όταν τα δεδομένα είναι τύπου εικόνας ή βίντεο [4][5]. Στα συνελικτικά νευρωνικά δίκτυα χρησιμοποιούνται στρώματα συνέλιξης προκειμένου το δίκτυο να “μάθει” τα χαρακτηριστικά μιας εικόνας και στο τέλος προστίθενται γραμμικά επίπεδα με σκοπό το δίκτυο να “μάθει” σε τι είδους εικόνα ανήκουν αυτά τα χαρακτηριστικά.



Σχήμα 3.1 Παράδειγμα ενός απλοϊκού συνελικτικού δικτύου.

3.2 Στρώματα συνέλιξης

Το στρώμα συνέλιξης (convolution layer) δημιουργείται με τη βοήθεια φίλτρων. Φίλτρο είναι ένας πίνακας $k * n$, όπου k ο αριθμός των γραμμών του πίνακα και n ο αριθμός των στηλών. Οι τιμές του πίνακα αυτού είναι τα αντίστοιχα βάρη θ . Το φίλτρο διαπερνά ολόκληρη την εικόνα πολλαπλασιάζοντας τον εαυτό του με έναν υποπίνακα της εικόνας κάθε φορά σαν γινόμενο Hadamard, ενώ τα στοιχεία του πίνακα που έχει σαν αποτέλεσμα, προστίθενται μεταξύ τους και καταχωρούνται στο τελικό στρώμα συνέλιξης [6][7], όπως φαίνεται στο σχήμα 3.2.

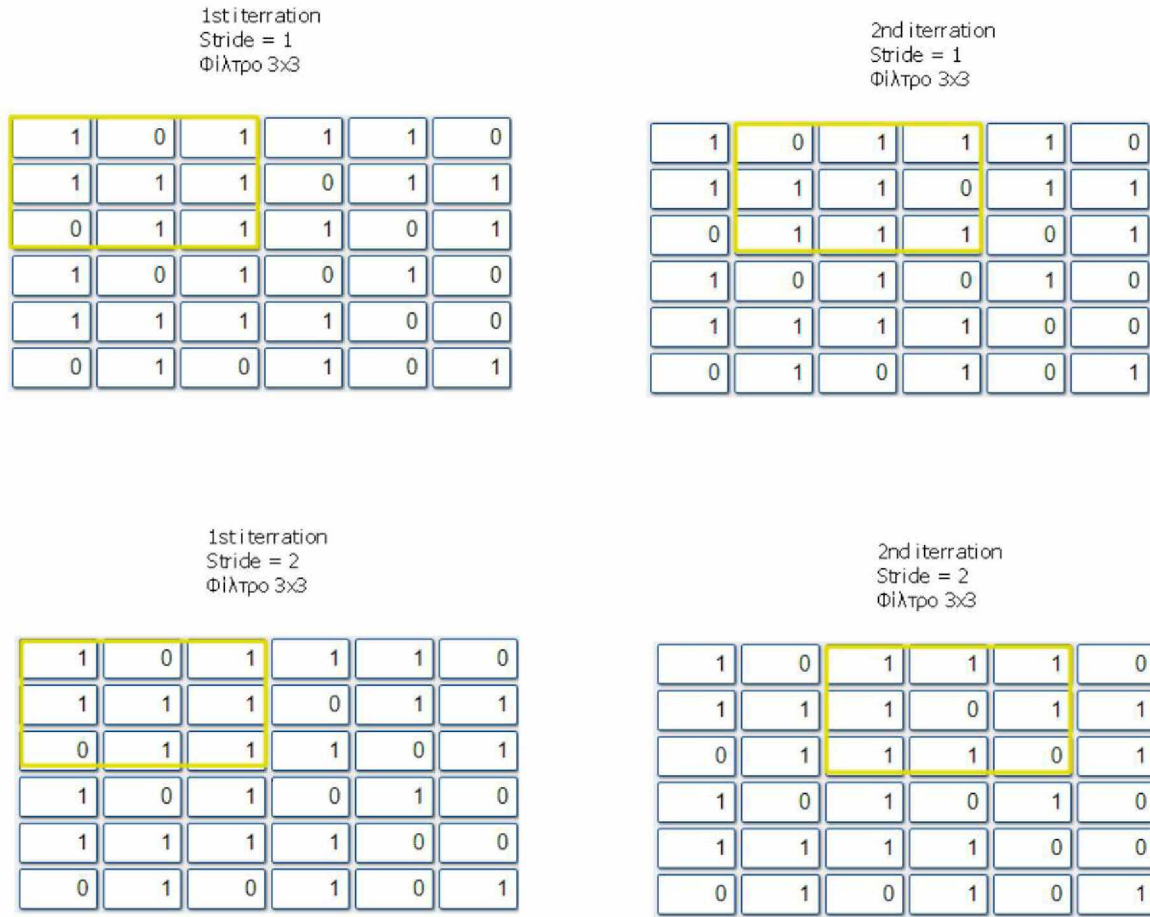


Γινόμενο Hadamard



Σχήμα 3.2 Παράδειγμα υπολογισμού φιλτραρισμένου στρώματος.

Στην κατασκευή του τελικού πίνακα σημαντικό ρόλο έχει η παράμετρος stride (Βήμα). Η τιμή της παραμέτρου, ορίζει πόσες θέσεις μετακινούμαστε κάθε φορά στην εικόνα προκειμένου να πολλαπλασιάσουμε τον υποπίνακα με το φίλτρο, όπως φαίνεται στο σχήμα 3.3.



Σχήμα 3.3 Παράδειγμα βήματος φίλτρου για διαφορετικές τιμές Stride.

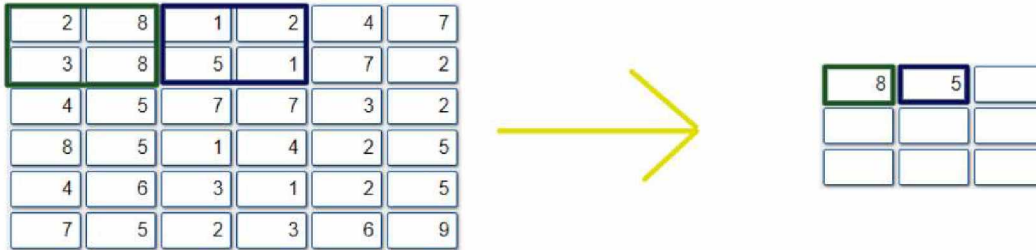
Πέραν του μεγέθους του φίλτρου και του βήματος (stride), ο κατασκευαστής του νευρωνικού δικτύου έχει την ευχέρεια να επιλέξει τον αριθμό των φίλτρων και τη συνάρτηση ενεργοποίησης του κάθε φίλτρου που εφαρμόζεται στο τελικό του αποτέλεσμα. Η συνήθης συνάρτηση ενεργοποίησης του φίλτρου είναι η ReLU και οι αρχικές τιμές ενός φίλτρου αρχικοποιούνται τυχαία, ενώ αργότερα εκπαιδεύονται.

3.3 Στρώμα συσσώρευσης

Συνήθως μετά από κάθε στρώμα φίλτρων υπάρχει και ένα στρώμα συσσώρευσης (Max pool), το οποίο “συνοψίζει” τον τελικό πίνακα κάθε φίλτρου. Όπως στην περίπτωση των στρωμάτων συνέλιξης έτσι και στα στρώματα συσσώρευσης ο πίνακας εισόδου διαπερνάται εξ ολοκλήρου από έναν πίνακα(περιοχή) μεγέθους $k * n$ της επιλογής του κατασκευαστή του δικτύου. Η παράμετρος stride υπάρχει και στα στρώματα συσσώρευσης με ακριβώς την ίδια λειτουργία. Συνήθως στον τελικό πίνακα εξόδου του στρώματος συσσώρευσης εισάγεται το μεγαλύτερο στοιχείο κάθε περιοχής ή ο μέσος

όρος αυτών, όπως φαίνεται στο σχήμα 3.4. Ο κύριος λόγος χρήσης των στρωμάτων συσσώρευσης είναι η σμίκρυνση των δεδομένων [8][9].

Max pooling 2x2
Stride = 2



Σχήμα 3.4 Παράδειγμα υπολογισμού στρώματος συσσώρευσης.

ΚΕΦΑΛΑΙΟ 4

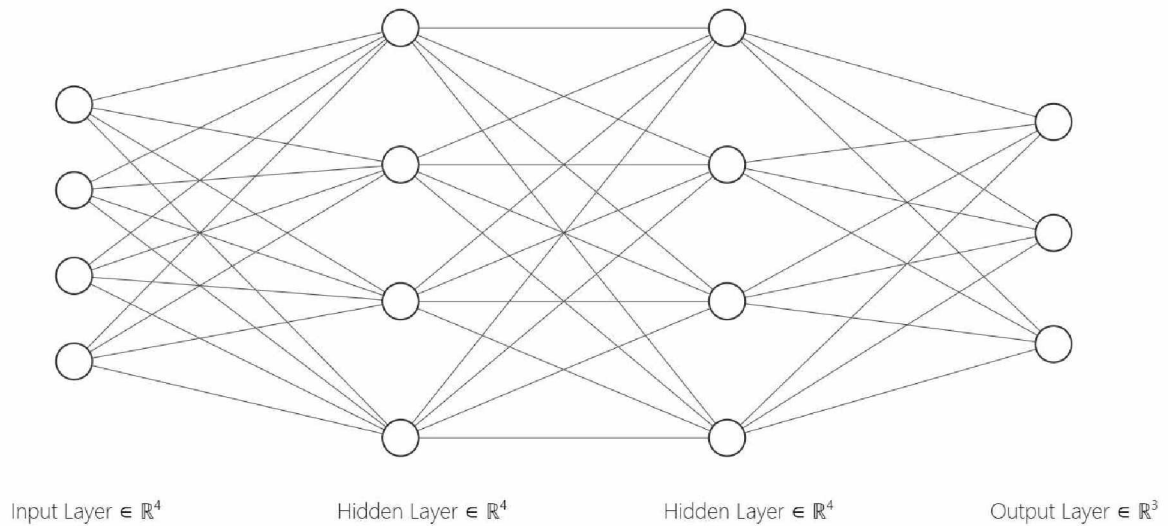
ΕΚΠΑΙΔΕΥΣΗ ΝΕΥΡΩΝΙΚΟΥ ΔΙΚΤΥΟΥ

4.1 Αλγόριθμος εκπαίδευσης νευρωνικού δικτύου

Τα νευρωνικά δίκτυα έχουν την δυνατότητα να προβλέπουν καινούριες τιμές για καινούρια δεδομένα. Αυτό το πετυχαίνουν, διότι έχουν εκπαιδευτεί με προηγούμενα δεδομένα. Ο συνήθης τρόπος εκπαίδευσης ενός δικτύου είναι με οπισθοδιάδοση του σφάλματος, μια διαδικασία κατά την οποία τα βάρη του κάθε νευρώνα διορθώνονται, ώστε να μειωθεί το σφάλμα, ενώ περισσότερες πληροφορίες για τον αλγόριθμο μπορούν να βρεθούν εδώ [10][11].

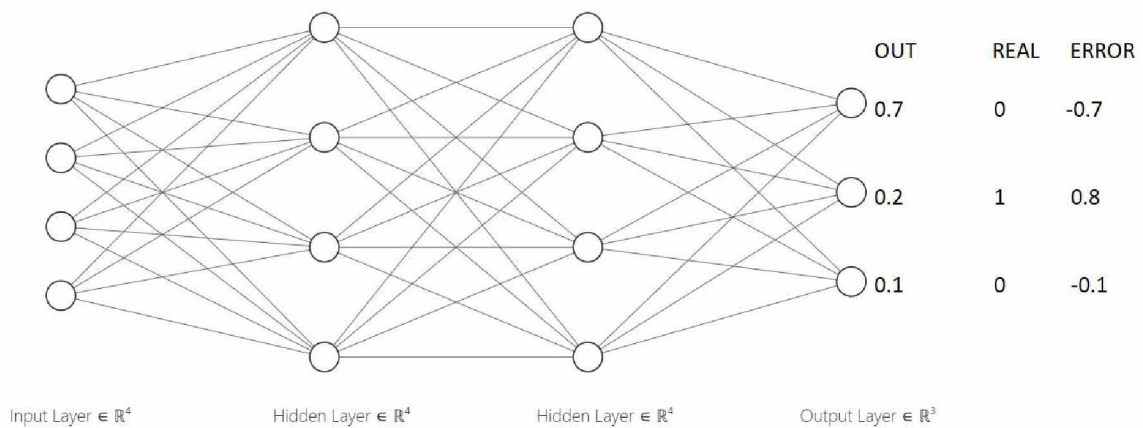
Η εργασία αυτή δεν στοχεύει στα μαθηματικά και στην ανάλυση αυτών στην εκπαίδευση ενός νευρωνικού δικτύου, παρόλα αυτά παρακάτω γίνεται μια συνοπτική και επιφανειακή ανάλυση για τον τρόπο λειτουργίας τους.

Εστω ότι υπάρχει μια βάση δεδομένων που αφορά την κατηγοριοποίηση ανθών έχοντας 4 χαρακτηριστικά τους όπως: μήκος φύλλων, πλάτος φύλλων, χρώμα και πάχος. Έστω επίσης πως πρέπει να τα κατηγοριοποιήσουμε σε 3 διαφορετικές κατηγορίες. Κάτι τέτοιο απαιτεί την κατασκευή ενός νευρωνικού δικτύου τεσσάρων εισόδων και τριών εξόδων όπου κάθε έξοδος είναι η πιθανότητα να ανήκει το άνθος σε κάποια κατηγορία [Σχήμα 4.1].



Σχήμα 4.1 Παράδειγμα νευρωνικού δικτύου για κατηγοριοποίηση ανθών.

Έστω ότι γνωρίζουμε ότι ένα άνθος ανήκει στην κατηγορία 2 και η έξοδος του δικτύου πρέπει να είναι $[0,1,0]$ αλλά το αποτέλεσμα του δικτύου είναι $[0.7,0.2,0.1]$. Τότε υπάρχει το λεγόμενο σφάλμα το οποίο είναι η έξοδος του δικτύου αφαιρούμενη από την πραγματική τιμή. Δηλαδή $[0,1,0] - [0.7,0.2,0.1] = [-0.7,0.8,-0.1]$ Σχήμα 4.2.



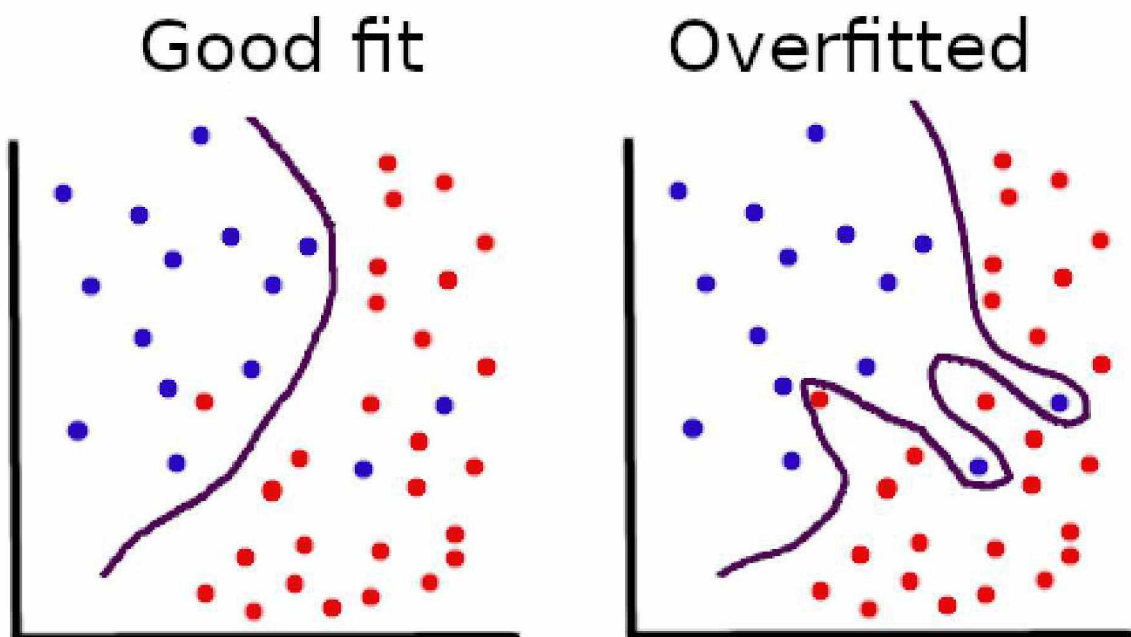
Σχήμα 4.2 Παράδειγμα εξόδου δικτύου και υπολογισμός σφάλματος.

Εφόσον υπολογιστεί το σφάλμα, το δίκτυο προσπαθεί να προσαρμόσει τις τιμές των παραμέτρων του, ώστε να πλησιάσει την πραγματική έξοδο. Στην συγκεκριμένη περίπτωση αυξάνεται το “βάρος” των συνδέσεων που αυξάνουν την τιμή της δεύτερης εξόδου και μειώνεται το “βάρος” των συνδέσεων που αυξάνουν την τιμή των άλλων δύο εξόδων για την συγκεκριμένη είσοδο. Σε αυτή τη διαδικασία υπάρχει η μεταβλητή

“learning rate”, η οποία σκοπό έχει να επηρεάσει την επίδραση που έχει κάθε είσοδος στην εκπαίδευση του δικτύου. Παραδείγματος χάρη με learning rate = 0.01 , η κάθε είσοδος αλλάζει τις παραμέτρους του δικτύου κατά 1%. Στο σχήμα 3.2 το σφάλμα του δικτύου για τον δεύτερο νευρώνα της εξόδου είναι 0.8, που σημαίνει ότι το δίκτυο θα ενημερώσει τις παραμέτρους του με σκοπό η έξοδος του νευρώνα αυτού να αυξηθεί κατά 0.8. Με learning rate = 0.01 όμως, το δίκτυο θα ενημερώσει τις παραμέτρους του, ώστε η έξοδος να αυξηθεί κατά $0.8 * 0.01 = 0.008$. Αυτό είναι χρήσιμο, καθώς κάθε είσοδος έχει μικρή επίδραση στο δίκτυο, οπότε η εκπαίδευση μοιράζεται σε όλες τις εισόδους.

4.2 Υπερμάθηση

Η υπερμάθηση (overfitting) σε ένα δίκτυο συμβαίνει όταν το δίκτυο ανταποκρίνεται πολύ καλά στις προβλέψεις έχοντας σαν είσοδο τα δεδομένα που έχει εκπαιδευτεί, αλλά δεν έχει τα ίδια καλά αποτελέσματα σε νέες προβλέψεις [12]. Στο σχήμα 4.3 φαίνεται μια αναπαράσταση τέτοιου δικτύου.



Σχήμα 4.3 Παράδειγμα δικτύου που έχει “υπερμάθει”.

4.3 Εποχές και παρτίδες δεδομένων

Κάθε φορά που το νευρωνικό δίκτυο διαπερνά μια φορά όλα τα δεδομένα που διαθέτουμε σαν είσοδο και εκπαιδεύεται πάνω σε αυτά, λέμε ότι έχουμε μια εποχή. Το νευρωνικό δίκτυο εκπαιδεύεται πολλαπλές φορές με τα ίδια δεδομένα και οι εποχές μπορεί να είναι από δέκα μέχρι μερικές χιλιάδες [13].

Τα δεδομένα μπορούν να χωριστούν σε παρτίδες δεδομένων. Οι παράμετροι του δικτύου δεν χρειάζεται να ανανεώνονται μετά από κάθε είσοδο κατά την εκπαίδευση. Χρησιμοποιώντας παρτίδες δεδομένων, υπάρχει η δυνατότητα χρήσης του μέσου όρου των σφαλμάτων από τις εισόδους που ανήκουν στην παρτίδα για την ενημέρωση των παραμέτρων του δικτύου. Η χρήση παρτίδας δεδομένων συμβαίνει κυρίως για τους εξής λόγους:

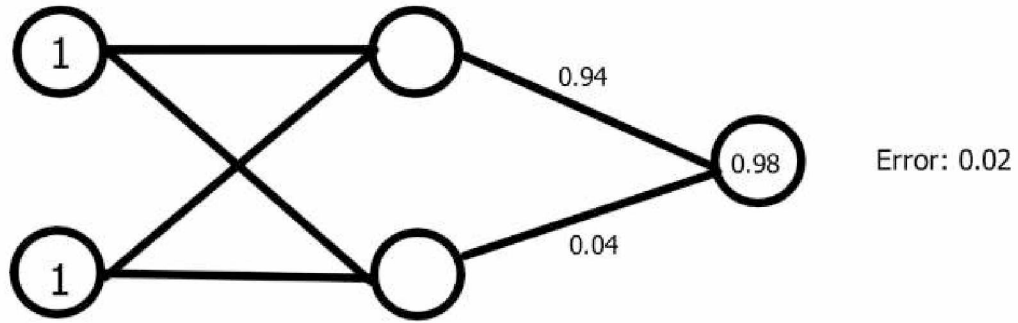
- 1) Ταχύτερη εκπαίδευση, καθώς το δίκτυο ενημερώνει τις παραμέτρους του μετά από μερικές δεκάδες εισόδους αντί κάθε εισόδου.
- 2) Αποφυγή υπερμάθησης καθώς οι ακραίες περιπτώσεις εισόδου έχουν λιγότερη επίδραση στην λανθασμένη εκπαίδευση του δικτύου λόγω της χρήσης του μέσου όρου.

4.4 Dropout

Το “Dropout” είναι μια τεχνική κατά την εκπαίδευση που στοχεύει στην αντιμετώπιση της υπερμάθησης. Ουσιαστικά η λειτουργία του είναι να “παγώνει” τυχαίους νευρώνες του δικτύου με πιθανότητα p με αποτέλεσμα να μικραίνουν οι διαστάσεις του δικτύου. Με αυτό πετυχαίνει τον διαμοιρασμό της εξόδου σε πολλά υποδίκτυα αντί για ένα μοναδικό που μπορεί να έτυχε να “υπερμάθει” κάποιο χαρακτηριστικό [14].

Παρακάτω εξηγείται η λειτουργία του Dropout με ένα απλοϊκό εικονογραφημένο παράδειγμα.

[1 1] : 1

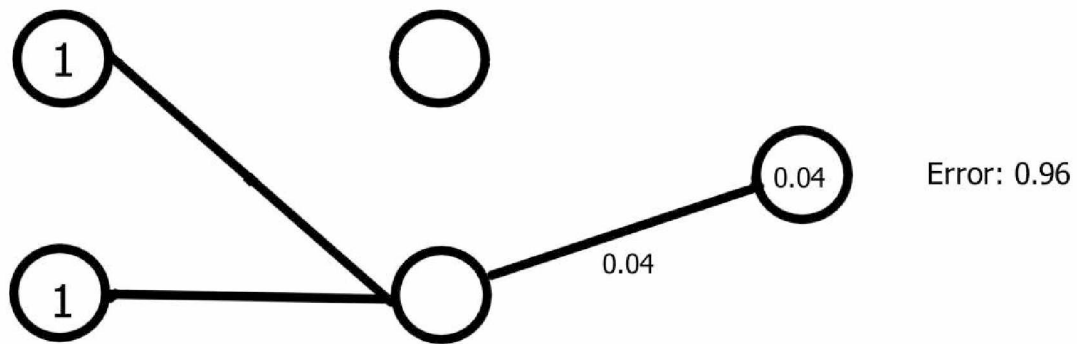


[1 1] : 1

Σχήμα 4.4 Παράδειγμα εξόδου δικτύου όπου [1 1] η είσοδος και 1 η επιθυμητή έξοδος.

Στο σχήμα 4.4 το σφάλμα ναί μεν είναι μικρό, όμως μόνο ένας νευρώνας συμβάλλει σημαντικά στην τελική έξοδο του συστήματος. Αυτό πολλές φορές σημαίνει ότι το δίκτυο μπορεί να έχει "υπερμάθει". Παγώνοντας τυχαία τον νευρώνα 1 έχουμε το δίκτυο του σχήματος 4.5.

$$[1 \ 1] : 1$$

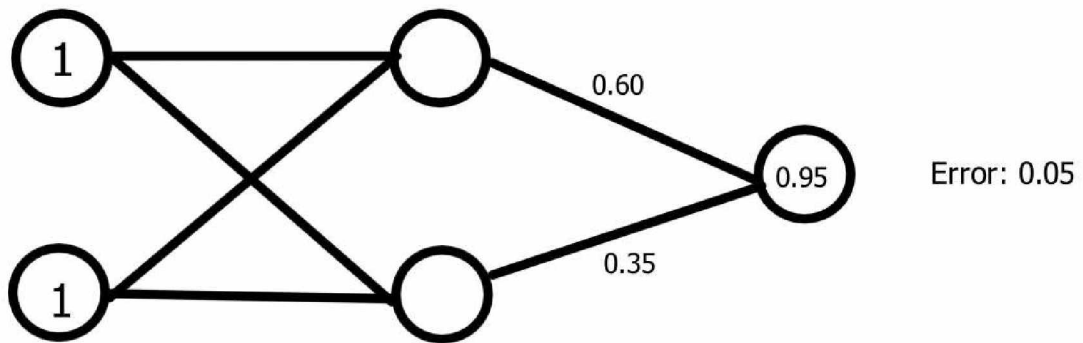


Σχήμα 4.5: Παράδειγμα εξόδου δικτύου όπου ένας νευρώνας έχει παγώσει.

Αυτό δημιουργεί ένα υποδίκτυο με μεγάλο σφάλμα, όμως τώρα ο δεύτερος νευρώνας μπορεί να αρχίσει και αυτός να συμμετέχει στην έξοδο του δικτύου, καθώς μετά από μια τέτοια συμπεριφορά το βάρος του νευρώνα 2 προς την έξοδο αυξάνεται.

Μετά από πληθώρα επαναλήψεων και dropout το δίκτυο μπορεί να φτάσει σε μια επιθυμητή κατάσταση όπου η έξοδος είναι μοιρασμένη με ποιο ομοιόμορφο τρόπο στους νευρώνες όπως στο σχήμα 4.6.

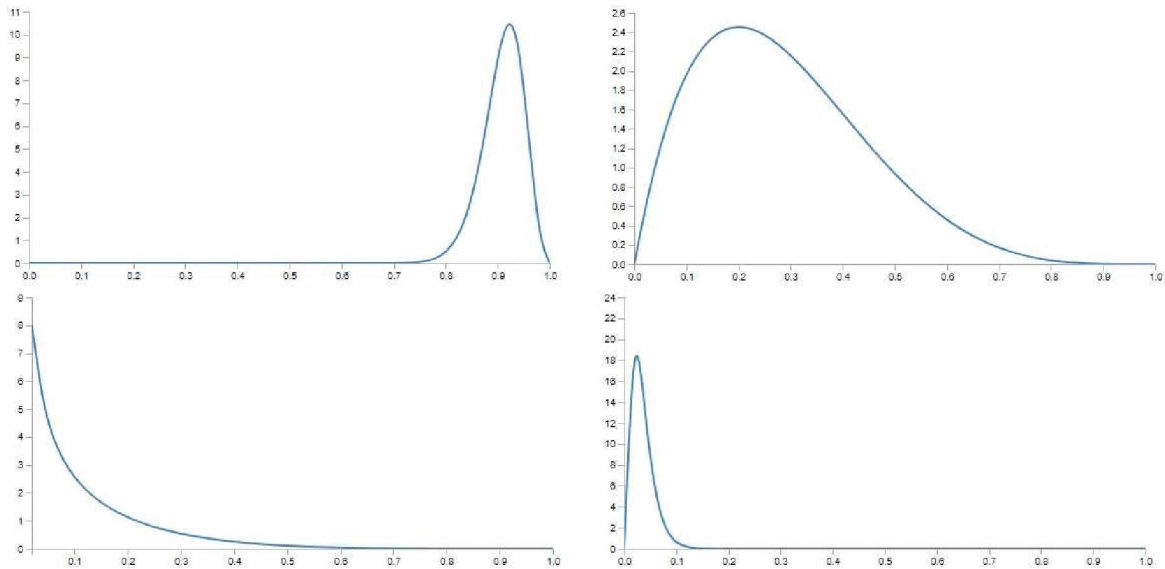
$$[1 \ 1] : 1$$



Σχήμα 4.6: Παράδειγμα εξόδου δικτύου.

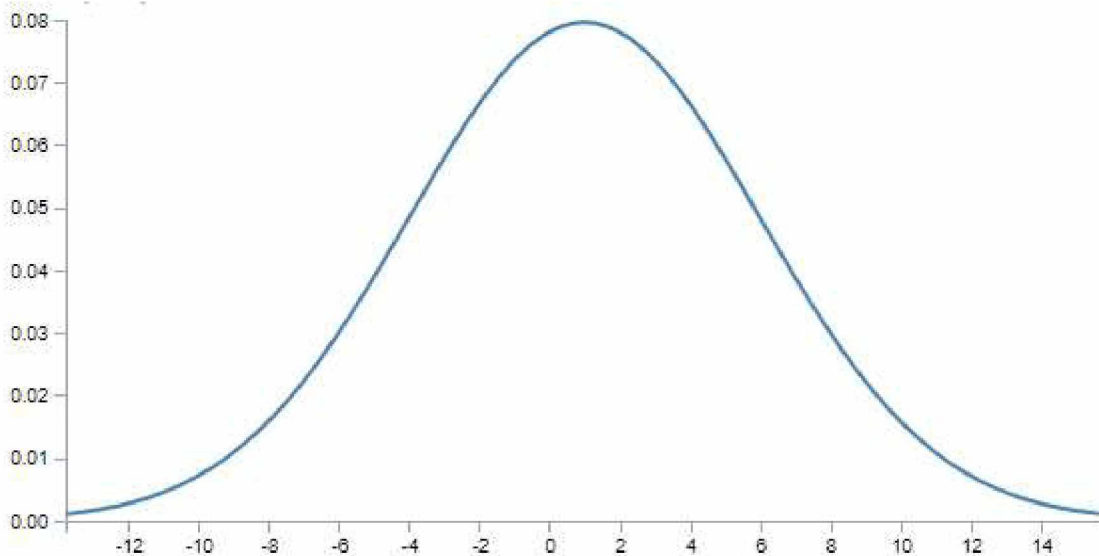
4.5 Batch normalization

Σε ένα νευρωνικό δίκτυο μετά από κάθε επίπεδο, οι έξοδοι μπορεί να έχουν διαφόρων ειδών κατανομές, όπως στο σχήμα 4.7.



Σχήμα 4.7 Παραδείγματα τυχαίων κατανομών εξόδου που μπορεί να προκύψουν σε κάποιο επίπεδο.

Οι κατανομές αυτές μπορεί να αλλάζουν από εποχή σε εποχή σε κάθε επίπεδο. Πειραματικά, κάποια δίκτυα είχαν καλύτερα αποτελέσματα όταν η κατανομή των εξόδων κάθε επιπέδου προερχόταν από κανονική κατανομή. Εξαιτίας αυτού πολλές φορές προσθέτουμε στην εκπαίδευση του νευρωνικού δικτύου την μετατροπή των εξόδων σε κανονική κατανομή [15] [Σχήμα 4.8].



Σχήμα 4.8 Παράδειγμα κανονικής κατανομής.

4.6 Δεδομένα

Πρωτεύων ρόλο στην εκπαίδευση ενός δικτύου έχουν τα δεδομένα που διαθέτουμε, άλλωστε ένας από τους λόγους που τα νευρωνικά δίκτυα έχουν αναπτυχθεί είναι ότι λόγω του διαδικτύου υπάρχει πρόσβαση σε τεράστιο όγκο δεδομένων. Τις περισσότερες φορές στα συνελκτικά νευρωνικά δίκτυα, χρησιμοποιείται το 80% των εικόνων ως δεδομένα εκπαίδευσης και αντίστοιχα το 20% των εικόνων ως δεδομένα εκτίμησης. Με τα δεδομένα εκπαίδευσης θα εκπαιδευτεί το δίκτυο, ενώ τα δεδομένα εκτίμησης δίνουν τη δυνατότητα στον κατασκευαστή να βγάλει συμπεράσματα για το πόσο αποτελεσματικό είναι το δίκτυο στην αξιολόγηση καινούργιων δεδομένων. Πολλές φορές βέβαια ένας μεγάλος αριθμός και ποικιλία δεδομένων μπορεί να αυξήσει σημαντικά την αποτελεσματικότητα του δικτύου.

Ένα ακόμα χρήσιμο εργαλείο όσον αφορά τα δεδομένα είναι το ονομαζόμενο “Data augmentation” [16][17] με το οποίο αυξάνουμε το πλήθος των δεδομένων επεξεργάζοντας τις εικόνες με διάφορους τρόπους όπως στο σχήμα 4.9 και σχήμα 4.10:

Γεωμετρική επεξεργασία εικόνων όπως:

- Περιστροφή
- Καθρεφτισμός
- Περικοπή
- Μεγέθυνση

Χρωματική επεξεργασία εικόνων όπως:

- Αυξομείωση φωτεινότητας
- Αυξομείωση αντίθεσης
- Αλλαγή απόχρωσης
- Αυξομείωση κορεσμού

Original



Περιστροφή



Καθρεφτισμός



Περικοπή



Μεγέθυνση



Σχήμα 4.9 Παράδειγμα γεωμετρικής επεξεργασίας εικόνας.

Original



Αυξομείωση φωτεινότητας



Αυξομείωση αντίθεσης



Αλλαγή απόχρωσης



Αυξομείωση κορεσμου



Σχήμα 4.10 Παράδειγμα χρωματικής επεξεργασίας εικόνας.

Συνήθως οι εικόνες επεξεργάζονται με τυχαίο τρόπο, ενώ εφαρμόζονται πολλαπλοί τρόποι ταυτόχρονα, όπως για παράδειγμα μεγέθυνση, καθρεφτισμός, αύξηση φωτεινότητας.

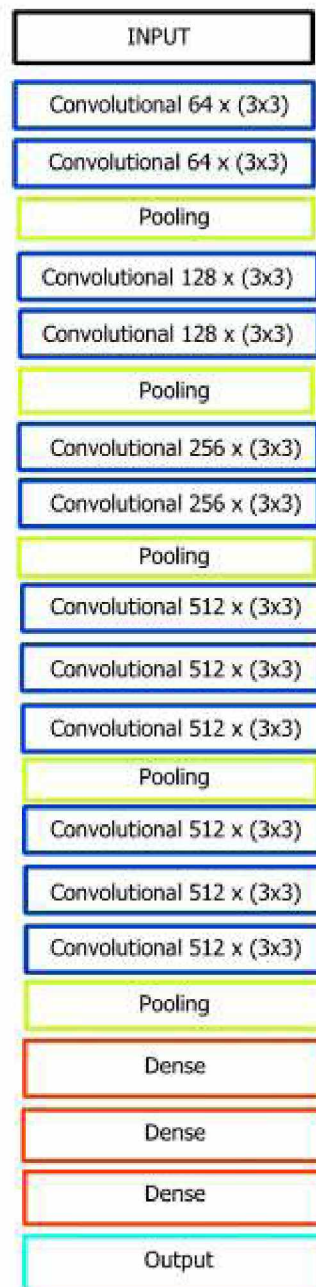
4.7 Κίνητρο χρήσης μεταφοράς μάθησης και δίκτυο VGG-16

Η κατασκευή ενός συνελκτικού δικτύου από την αρχή είναι μια χρονοβόρα διαδικασία όσον αφορά την αρχιτεκτονική που θα ακολουθήσει το δίκτυο, ενώ χρειάζεται ακόμα περισσότερο χρόνο να εκπαιδευτεί. Πλέον όμως προκατασκευασμένα και προεκπαιδευμένα νευρωνικά δίκτυα βρίσκονται ελεύθερα προς χρήση και με ελάχιστες αλλαγές, έτσι ώστε ο κατασκευαστής μπορεί να έχει ποιοτικά αποτελέσματα για τις ανάγκες του δικτύου.

Το συνελκτικό δίκτυο VGG-16 κατασκευάστηκε από τους Karen Simonyan και Andrew Zisserman και συμμετείχε στον διαγωνισμό 2014 ILSVRC challenge, όπου και κατέκτησε την πρώτη θέση όσον αφορά την ανίχνευση αντικειμένων και την δεύτερη θέση σε κατηγοριοποίηση εικόνων [18].

Το δίκτυο VGG-16 θα χρησιμοποιηθεί για τις ανάγκες της εργασίας. Περαν αυτού η βιβλιοθήκη Keras έχει διαθέσιμα προς χρήση και άλλα γνωστά νευρωνικά δίκτυα για την μεταφορά μάθησης όπως:

- Xception
- VGG16
- VGG19
- ResNet50
- InceptionV3
- InceptionResNetV2
- MobileNet
- MobileNetV2
- DenseNet
- NASNet



Σχήμα 4.11 Αρχιτεκτονική VGG-16.

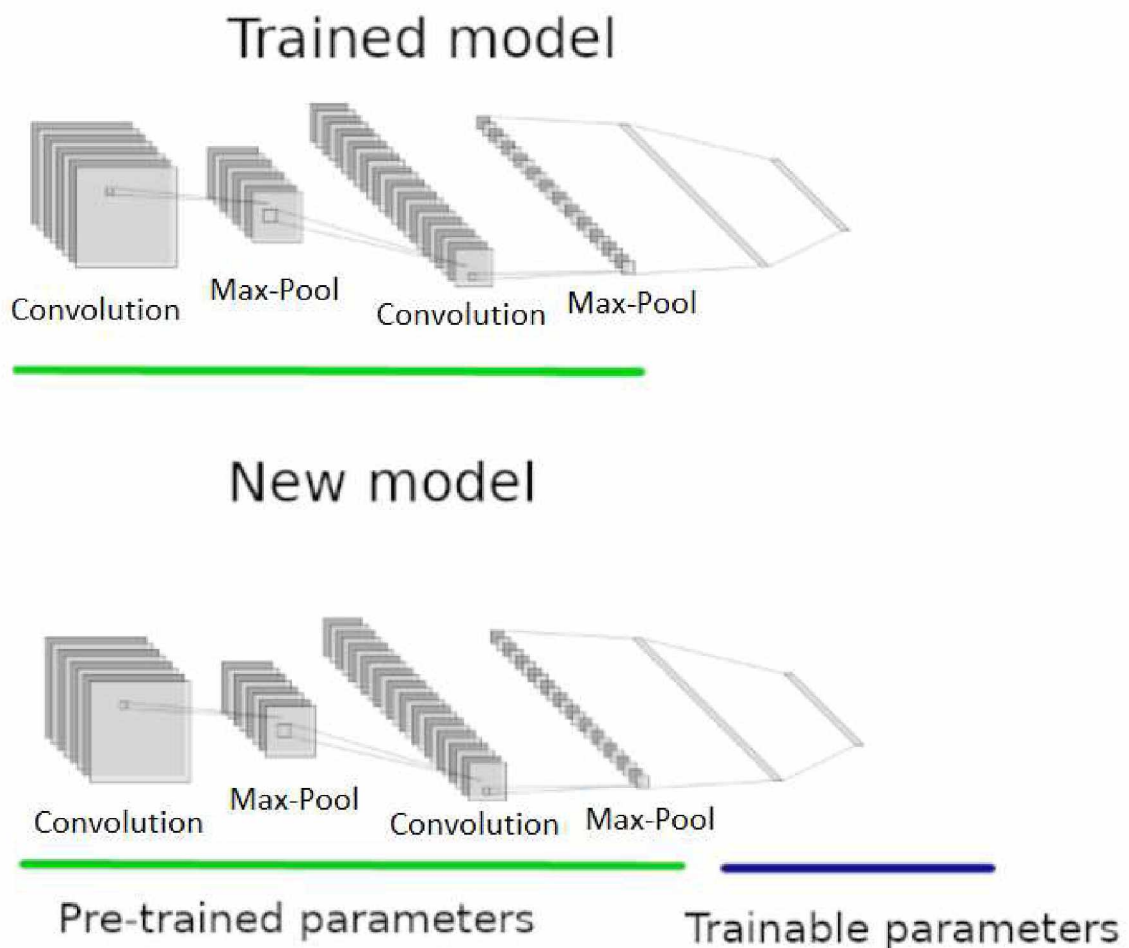
Στο σχήμα 4.11 φαίνεται η αρχιτεκτονική του νευρωνικού δικτύου VGG-16. Για την εκπαίδευση του χρησιμοποιήθηκε βάση δεδομένων δεκατεσσάρων εκατομμυρίων εικόνων χιλίων κλάσεων. Χρειάστηκε χρόνος τριών εβδομάδων, για την εκπαίδευση του και για τον σκοπό αυτό χρησιμοποιήθηκε η κάρτα γραφικών Nvidia Titan. Στο τέλος της εκπαίδευσης, το δίκτυο είχε ακρίβεια 92.7%.

Για διάφορους λόγους, παραδείγματος χάρη, έλλειψη χρόνου, μικρά project, έλλειψη hardware, οι κατασκευαστές ενός νέου δικτύου για νέα δεδομένα επιλέγουν την μέθοδο

μεταφοράς μάθησης αντί της κατασκευής και εκπαίδευσης του νευρωνικού δικτύου από την αρχή. Η μέθοδος μεταφοράς μάθησης αναλύεται στην επόμενη ενότητα.

4.8 Μεταφορά μάθησης

Μεταφορά μάθησης είναι μια διαδικασία κατά την οποία αντιγράφουμε από ένα ήδη εκπαιδευμένο δίκτυο μέρος των παραμέτρων σε ένα καινούριο δίκτυο. Συνήθως οι αντιγεγραμμένες παράμετροι μένουν σταθερές ή αλλιώς “παγώνουν” και η εκπαίδευση γίνεται στο υπόλοιπο δίκτυο, όπως φαίνεται στο σχήμα 4.12.



Σχήμα 4.12 Παράδειγμα μεταφοράς εκπαιδευμένων παραμέτρων.

Οι έτοιμες παράμετροι που αντιγράφονται από το ήδη εκπαιδευμένο δίκτυο είναι συνήθως αυτές των συνελκτικών και συσσωρευτικών στρωμάτων, καθώς τα στρώματα αυτά είναι υπεύθυνα για την αναγνώριση χαρακτηριστικών σε μια εικόνα, ενώ τα

γραμμικά στρώματα λειτουργούν περισσότερο ως κατηγοριοποιητές των χαρακτηριστικών αυτών.

Η μέθοδος εκπαίδευσης νευρωνικού δικτύου με μεταφορά μάθησης είναι αποδοτική όταν στο εκπαιδευμένο νευρωνικό δίκτυο από το οποίο έχουν αντιγραφεί οι παράμετροι, έχουν χρησιμοποιηθεί παρόμοια δεδομένα με το καινούριο δίκτυο προς εκπαίδευση. Ένα παράδειγμα ορθής χρήσης της μεθόδου είναι η χρήση των παραμέτρων ενός νευρωνικού δικτύου που έχει εκπαιδευτεί για την αναγνώριση και κατηγοριοποίηση οικόσιτων ζώων με σκοπό την κατασκευή ενός καινούριου δικτύου που κατηγοριοποιεί τις ράτσες το σκυλιών. Ένα τέτοιο δίκτυο μπορεί να είναι αποτελεσματικό, καθώς τα συνελκτικά στρώματα του προηγούμενου δικτύου έχουν “μάθει” πληθώρα χαρακτηριστικών ενός ζώου [19].

4.9 Τρόποι εφαρμογής μεταφοράς μάθησης

Υπάρχει ευελιξία στον τρόπο εφαρμογής της μεθόδου μεταφοράς μάθησης και η επιλογή συνήθως γίνεται με βάση της συνθήκες του κάθε προβλήματος. Η συνήθης τεχνική όταν τα δεδομένα και των δύο δικτύων ,εκπαιδευμένου και εκπαιδευόμενου, έχουν μεγάλες ομοιότητες είναι το “πάγωμα” των παραμέτρων των συνελκτικών στρωμάτων και εκπαίδευση των γραμμικών στρωμάτων, τα οποία χρησιμοποιούνται για κατηγοριοποίηση. Όταν τα δεδομένα έχουν διαφορές , υπάρχει περίπτωση να βελτιωθεί η απόδοση του δικτύου αν επανεκπαιδευτούν κάποια από τα επιμέρους στρώματα νευρώνων , η μέθοδος αυτή λέγεται fine tuning. Στο σχήμα 4.13 παρουσιάζονται διάφορες παραλλαγές fine tuning που μπορεί να προκύψουν.



Σχήμα 4.13 Παραλλαγές χρήσης μεταφοράς μάθησης.

ΚΕΦΑΛΑΙΟ 5

ΠΕΙΡΑΜΑΤΙΣΜΟΣ ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ ΣΥΝΕΛΙΚΤΙΚΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ

5.1 Βασική αρχιτεκτονική δικτύου και δεδομένα

Τα δεδομένα που χρησιμοποιήθηκαν για αυτή την εργασία προέρχονται από τη βάση δεδομένων του kaggle και από το dataset Dogs vs Cats [20]. 21000 εικόνες χρησιμοποιήθηκαν για την εκπαίδευση, ενώ 4000 εικόνες χρησιμοποιήθηκαν για την αξιολόγηση.

Προκειμένου να διεξαχθεί ο πειραματισμός κατασκευάστηκε ένα βασικό δίκτυο σε γλώσσα προγραμματισμού python 3 και χρησιμοποιήθηκε η βιβλιοθήκη tensorflow [21] της google, ενώ πολλά από τα διαγράμματα έχουν παραχθεί με τη χρήση tensorboard.

Ο κώδικας ξεκινά με την φόρτωση των δεδομένων, όπου:

- Οι εικόνες μετατρέπονται σε μέγεθος 128 x 128 pixels, ώστε να είναι όλες το ίδιο μέγεθος
- Όλα τα χρώματα προσαρμόζονται στην κλίμακα 0-1
- Τα δεδομένα χωρίζονται σε παρτίδες των 32
- Υποβάλλονται σε μια τυχαία γεωμετρική επεξεργασία

```
# Preprocessing the Training set
train_datagen = ImageDataGenerator(rescale = 1./255, shear_range =
0.2, zoom_range = 0.2, horizontal_flip = True)

training_set = train_datagen.flow_from_directory('train', shuffle
= True, target_size = (128, 128), batch_size = 32, class_mode =
'binary')

# Preprocessing the Test set
test_datagen = ImageDataGenerator(rescale = 1./255)

test_set = test_datagen.flow_from_directory('validation',
target_size = (128, 128), batch_size = 32, class_mode = 'binary')
```

Η αρχιτεκτονική του δικτύου είναι αυτή του σχήματος 5.1.

Layer (type)	Output Shape	Param #
2D_Convolutioal_A (Conv2D)	(None, 125, 125, 32)	1568
2D_Convolutioal_B (Conv2D)	(None, 122, 122, 32)	16416
ReLU (Activation)	(None, 122, 122, 32)	0
2D_Pooling_A (MaxPooling2D)	(None, 61, 61, 32)	0
2D_Convolutioal_C (Conv2D)	(None, 58, 58, 64)	32832
2D_Convolutioal_D (Conv2D)	(None, 56, 56, 64)	36928
ReLU_A (Activation)	(None, 56, 56, 64)	0
2D_Pooling_B (MaxPooling2D)	(None, 28, 28, 64)	0
2D_Convolutioal_E (Conv2D)	(None, 26, 26, 128)	73856
2D_Convolutioal_F (Conv2D)	(None, 24, 24, 128)	147584
ReLU_B (Activation)	(None, 24, 24, 128)	0
2D_Pooling_C (MaxPooling2D)	(None, 12, 12, 128)	0
flatten_2 (Flatten)	(None, 18432)	0
dense_6 (Dense)	(None, 256)	4718848
ReLU_C (Activation)	(None, 256)	0
dropout_4 (Dropout)	(None, 256)	0
dense_7 (Dense)	(None, 256)	65792
ReLU_D (Activation)	(None, 256)	0
dropout_5 (Dropout)	(None, 256)	0
dense_8 (Dense)	(None, 1)	257
Sigmoid (Activation)	(None, 1)	0
=====		
Total params: 5,094,081		
Trainable params: 5,094,081		
Non-trainable params: 0		

Σχήμα 5.1 Αρχιτεκτονική του βασικού δικτύου.

Ο κώδικας κατασκευής του φαίνεται παρακάτω:

```
model = Sequential()

model.add(Conv2D(32, (4, 4),
```



```

input_shape=[128,128,3],name='2D_Convolutional_A'))
model.add(Conv2D(32, (4, 4),name='2D_Convolutional_B'))
model.add(Activation('relu',name='ReLU'))
model.add(MaxPooling2D(pool_size=(2, 2),name='2D_Pooling_A'))

model.add(Conv2D(64, (4, 4),name='2D_Convolutional_C'))
model.add(Conv2D(64, (3, 3),name='2D_Convolutional_D'))
model.add(Activation('relu',name='ReLU_A'))
model.add(MaxPooling2D(pool_size=(2, 2),name='2D_Pooling_B'))

model.add(Conv2D(128, (3, 3),name='2D_Convolutional_E'))
model.add(Conv2D(128, (3, 3),name='2D_Convolutional_F'))
model.add(Activation('relu',name='ReLU_B'))
model.add(MaxPooling2D(pool_size=(2, 2),name='2D_Pooling_C'))

model.add(Flatten())

model.add(Dense(256))
model.add(Activation('relu',name='ReLU_C'))
model.add(Dropout(0.5))

model.add(Dense(256))
model.add(Activation('relu',name='ReLU_D'))
model.add(Dropout(0.5))

model.add(Dense(1))
model.add(Activation('sigmoid',name='Sigmoid'))

```

Για την εκπαίδευση του δικτύου χρησιμοποιήθηκε ο αλγόριθμος “RMSprop” με learning rate = 0.00001 [22].

```

model.compile(loss='binary_crossentropy',
optimizer=tf.keras.optimizers.RMSprop(learning_rate=0.0001,center
ed = True),metrics=['accuracy'])

```

Το δίκτυο εκπαιδεύεται για 30 εποχές όπου κάθε εποχή χρησιμοποιεί ένα τυχαίο δείγμα 8.192 εικόνων.

```

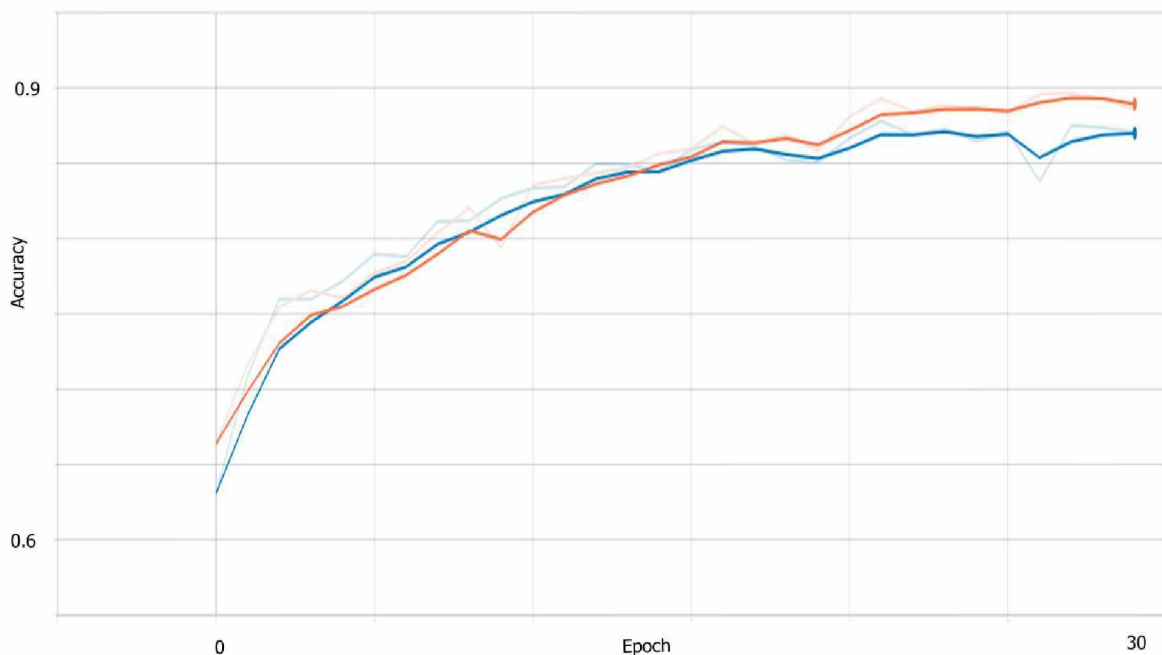
model.fit(x = training_set, validation_data = test_set, epochs =
30 , steps_per_epoch = 256,callbacks=[tensorboard_callback])

```


Στις επόμενες ενότητες εκπαιδεύονται διάφορα παρόμοια νευρωνικά δίκτυα με πρόσθεση ή αλλαγή κάποιων παραμέτρων με σκοπό την εξαγωγή συμπερασμάτων για την συσχέτιση του δικτύου με την αποτελεσματικότητα του όσον αφορά την ακρίβεια. Τα δίκτυα έχουν εκπαιδευτεί πολλαπλές φορές με σκοπό την αναγνώριση αν τα αποτελέσματα ήταν ρεαλιστικά ή αν προήλθαν λόγω τυχαιότητας. Τα δίκτυα εκπαιδεύτηκαν με την χρήση κάρτας γραφικών gtx 1660 ti η οποία διαθέτει 1.536 cuda cores.

5.2 Data augmentation

Στο σχήμα 5.2 φαίνονται τα αποτελέσματα του δικτύου που χρησιμοποίησε data augmentation έναντι του δικτύου που δεν χρησιμοποίησε

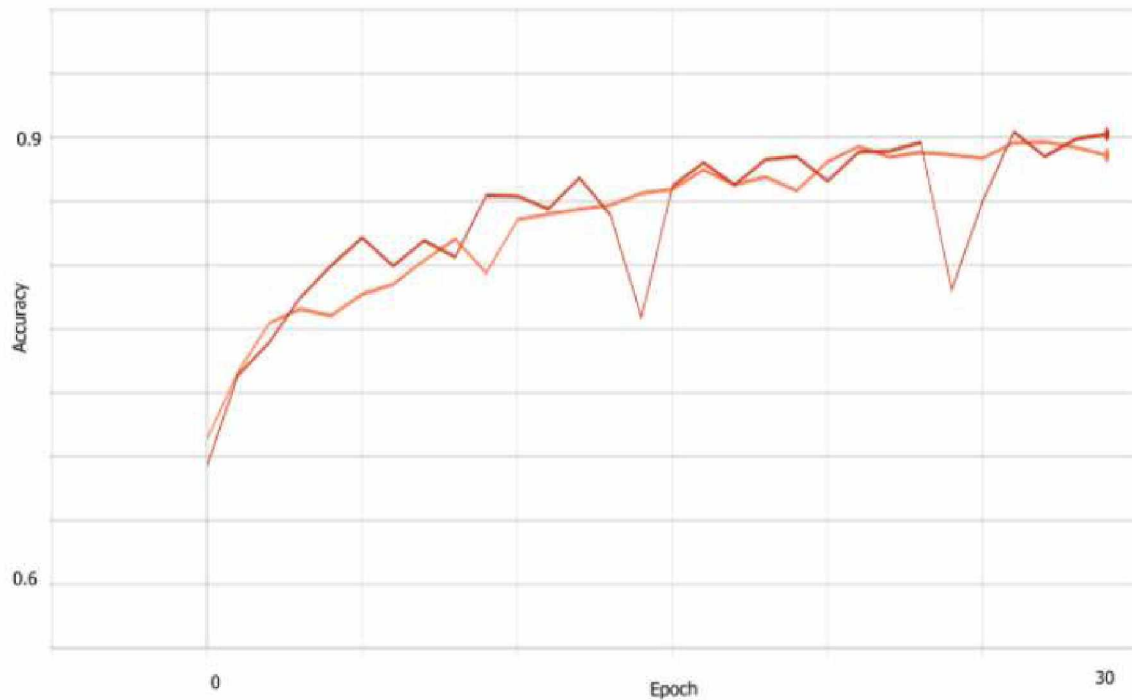


Σχήμα 5.2 Πορτοκαλί γραμμή: δίκτυο που χρησιμοποιεί data augmentation και μπλε είναι το δίκτυο που δεν χρησιμοποιήθηκε data augmentation. Με τη χρήση data augmentation επιτεύχθηκε 2% (89% vs 87%) μεγαλύτερη ακρίβεια, κάτι που ήταν αναμενόμενο καθώς δημιουργούνται πολλαπλά καινούρια “ψευδοδεδομένα”. Παρόλα αυτά, το data augmentation αποδείχτηκε χρονοβόρο, αφού χρειάστηκε 23 λεπτά για την ολοκλήρωσή του, ενώ η μη χρήση αυτού, χρειάστηκε μόλις 13 λεπτά. Σε μικρής κλίμακας δίκτυα με λίγα δεδομένα μπορεί να φανεί ιδιαίτερα χρήσιμο, ωστόσο χρειάζεται ιδιαίτερη προσοχή ο τρόπος εφαρμογής σε μεγάλης κλίμακας δίκτυα, διότι μπορεί να αυξήσει δραματικά τον χρόνο εκπαίδευσης χωρίς ουσιαστικό κέρδος.

Για τις επόμενες ενότητες θα χρησιμοποιηθεί η μέθοδος data augmentation σε όλα τα δίκτυα.

5.3 Μέγεθος παρτίδας

Στο σχήμα 5.3 φαίνονται 2 δίκτυα που εκπαιδεύτηκαν για διαφορετική τιμή παρτίδας.

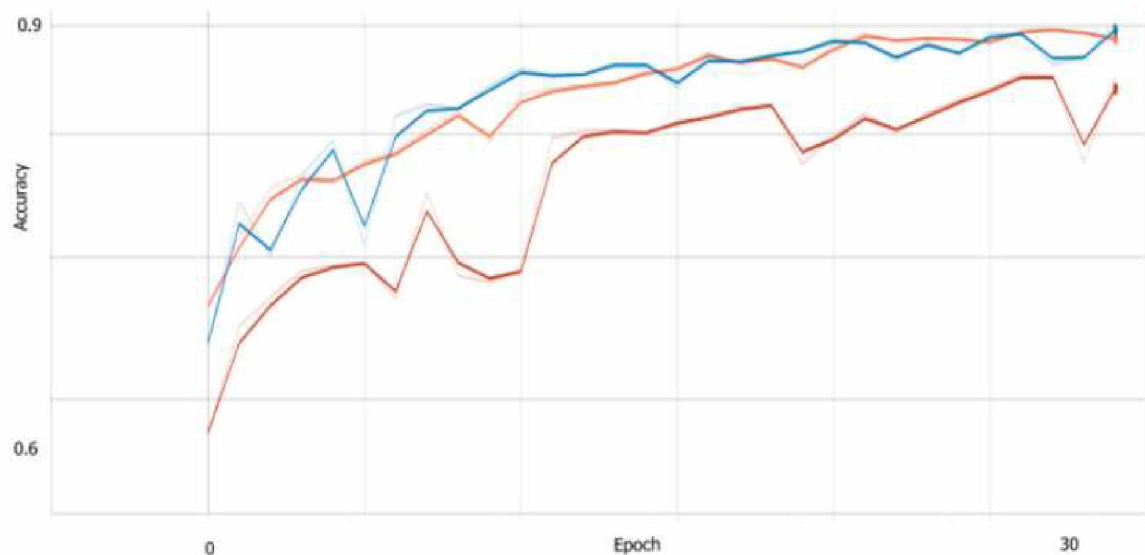


Σχήμα 5.3 Batch size 32: πορτοκαλί, Batch size 16: κόκκινο.

Όπως φαίνεται στο σχήμα, μεγαλύτερο μέγεθος παρτίδας συμβάλλει στην πιο ομαλή εκπαίδευση ανά εποχή, καθώς οι ακραίες περιπτώσεις δεν επιδρούν σε μεγάλο βαθμό στην εκπαίδευση του δικτύου. Υπάρχουν όμως και αρνητικές επιπτώσεις, γιατί μεγάλο μέγεθος παρτίδας εξασθενεί την διαφορετικότητα. Αυτό έχει σαν αποτέλεσμα κατά περιπτώσεις το δίκτυο να καταλήγει σε καλύτερα αποτελέσματα. Στο ως άνω παράδειγμα η παρτίδα μεγέθους 16 είχε 1%(89% vs 90%) καλύτερη ακρίβεια από αυτή της παρτίδας των 32. Η αστάθεια όμως της παρτίδας των 16 χρειάζεται ιδιαίτερη προσοχή και χειρισμό, διότι το δίκτυο χρειάζεται συχνή αποθήκευση και επιλογή κάποιας έκδοσης με καλή ακρίβεια.

5.4 Dropout

Στο σχήμα 5.4 έχουν εκπαιδευτεί 3 διαφορετικά νευρωνικά δίκτυα. Το δίκτυο της μπλε γραμμής δεν διαθέτει dropout επίπεδα, το δίκτυο της πορτοκαλί γραμμής διαθέτει dropout δίκτυα μόνο στα τελευταία γραμμικά επίπεδα, ενώ στο δίκτυο της κόκκινης έχει χρησιμοποιηθεί η μέθοδος dropout ακόμη και στα συνελκτικά στρώματα.



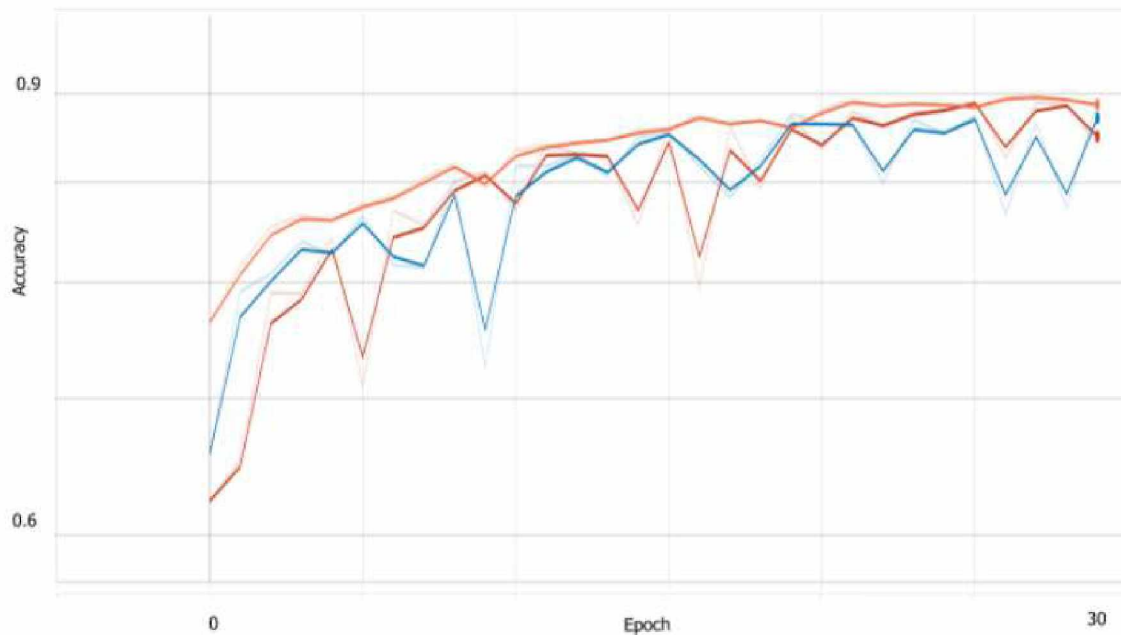
Σχήμα 5.4 Dropout comparison.

Η εφαρμογή dropout στα συνελκτικά στρώματα φαίνεται να έχει αρνητικά επίπεδα στην αποτελεσματικότητα του δικτύου 4%(89% vs 85%).

Η εφαρμογή dropout στα τελευταία γραμμικά επίπεδα είχε σαν αποτέλεσμα μια ελαφρώς πιο ομαλή εκπαίδευση ανά εποχή, αλλά το τελικό αποτέλεσμα δεν είχε σημαντική διαφορά, αφού και τα δύο δίκτυα είχαν παρόμοια ακρίβεια.

5.5 Batch normalization

Στο σχήμα 5.5 έχουν εκπαιδευτεί 3 διαφορετικά νευρωνικά δίκτυα. Το πρώτο, το δίκτυο της πορτοκαλί γραμμής δεν διαθέτει Batch normalization επίπεδα, το δεύτερο, το δίκτυο της κόκκινης γραμμής διαθέτει Batch normalization δίκτυα μόνο στα τελευταία γραμμικά επίπεδα, ενώ στο τρίτο δίκτυο της μπλε έχει χρησιμοποιηθεί η μέθοδος Batch normalization ακόμη και στα συνελκτικά στρώματα.



Σχήμα 5.5 Batch normalization comparison.

Όπως συμπεραίνεται από τα αποτελέσματα των δικτύων, η μέθοδος batch normalization δεν ωφέλησε ιδιαίτερα την εκπαίδευση του συνελικτικού δικτύου του παραδείγματος, αντιθέτως είχε αρνητικά αποτελέσματα, καθώς έκανε την εκπαίδευση του δικτύου πιο “ανώμαλη”.

5.6 Χρήση της μεθόδου μεταφοράς μάθησης

Στην παρούσα ενότητα με τη χρήση της βιβλιοθήκης tensorflow-gpu και της κάρτας γραφικών GeForce GTX 1660 Ti, η οποία διαθέτει 1,536 CUDA cores, επιτεύχθηκε ακρίβεια νευρωνικού δικτύου 98% στην κατηγοριοποίηση σκύλων και γατών με χρόνο εκπαίδευσης 4 λεπτών, λόγω της χρήσης μεταφοράς μάθησης.

Ορισμός μεγέθους εικόνας που θα έχει το δίκτυο σαν είσοδο και φόρτωση του VGG-16 χωρίς τα τελευταία στρώματα με το attitude(include_top = False).

```
input_shape = (300, 300, 3)
base_model = VGG16(weights='imagenet', include_top=False,
input_shape=input_shape)
```

Δημιουργία νέων γραμμικών στρωμάτων για κατηγοριοποίηση.

```
x=base_model.output
x=GlobalAveragePooling2D()(x)
x=Dense(1024,activation='relu')(x)
x=Dense(1024,activation='relu')(x)
x=Dense(512,activation='relu')(x)
preds=Dense(2,activation='softmax')(x)

model=Model(inputs=base_model.input,outputs=preds)
```

Πάγωμα των συνελκτικικών στρωμάτων

```
for layer in model.layers[:20]:
    layer.trainable=False
for layer in model.layers[20:]:
    layer.trainable=True
```

Εισαγωγή δεδομένων. Όπου `train_generator` οι εικόνες που χρησιμοποιεί το δίκτυο προς εκπαίδευση και `val` οι εικόνες που χρησιμοποιεί το δίκτυο προς αξιολόγηση.

```
datagen=ImageDataGenerator()

train_generator=datagen.flow_from_directory('./trainingset/',
target_size=input_shape[:2],color_mode='rgb',batch_size=32,
class_mode='categorical',shuffle=True)

val=datagen.flow_from_directory('./testset/',target_size=input_shape[:2],color_mode='rgb',batch_size=32,
class_mode='categorical')
```

Και τελικά η εκπαίδευση του δικτύου όπως απεικονίζεται παρακάτω

```
model.compile(optimizer='Adam',loss='categorical_crossentropy',metrics=['accuracy'])
```

Σε μόλις 4 λεπτά και 3 εποχές μπορεί να επιτευχθεί 98% ακρίβεια, σημαντικά αυξημένη σε σύγκριση με την 90% ακρίβεια που είχε το προηγούμενο δίκτυο του κεφαλαίου 5, το οποίο χρειάστηκε 25 λεπτά εκπαίδευσης. Αξιοσημείωτο είναι το μέγεθος του VGG16 που είναι πολλαπλάσιο με 138 εκατομμύρια παραμέτρους έναντι των πέντε εκατομμυρίων παραμέτρων που είχε το δίκτυο στο κεφάλαιο 5.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Συνοπτικά, στην παραπάνω εργασία αναλύθηκε ο τρόπος λειτουργίας των γραμμικών νευρωνικών δικτύων, έπειτα δόθηκε έμφαση στα συνελκτικά νευρωνικά που χρησιμοποιούνται για την κατηγοριοποίηση εικόνων όπου και αναλύθηκαν διάφορες τεχνικές για τη βελτιστοποίησή τους. Παρουσιάστηκε στη συνέχεια πώς μπορούν διάφορες επιλογές στην αρχιτεκτονική του δικτύου να επηρεάσουν την ακρίβειά του χωρίς όμως απαραίτητα τα αποτελέσματα να “μεταφέρονται” σε όλες τις διαφορετικές βάσεις δεδομένων. Με λίγα λόγια διαφορετικά δεδομένα χρήζουν διαφορετικής μεταχείρισης. Αναδεικνύεται μάλιστα ως ένα σημαντικό εργαλείο απόκτησης εμπειρίας και αντίληψης μιας γενικότερης εικόνας όσον αφορά τα συνελκτικά νευρωνικά δίκτυα.

Η πιο ενδιαφέρουσα πρακτική στη παρούσα εργασία ήταν η μεταφορά μάθησης αφού βελτίωσε την απόδοση του δικτύου κατά 8% και έπειτα η τεχνική data augmentation με βελτίωση 2%. Η αλλαγή στο μέγεθος πατρίδας και χρήση dropout δεν είχαν σημαντική βελτίωση στην απόδοση του δικτύου, παρόλα αυτά με το σωστό ορισμό των παραμέτρων τους είχαν σαν αποτέλεσμα μια πιο “ομαλή” εκπαίδευση. Τέλος η τεχνική batch normalization δεν είχε θετικά αποτελέσματα για το δίκτυο καθώς μείωσε την αποδοτικότητα του.

Υπο το πρίσμα αυτής της έρευνας εντοπίζεται μια αντίστοιχη του Song Han[23] με θεματική τη μάθηση βαρών και συνδέσεων με σκοπό τη βελτίωση των νευρωνικών δικτύων η οποία επικεντρώνεται σε αλλαγές που μπορεί να συμβούν στην αρχιτεκτονική ενός δικτύου, με σκοπό την μείωση του μεγέθους του, που ως αποτέλεσμα έχει την χρήση λιγότερης μνήμης και απαιτητικότητας σε υλικό υπολογιστή(hardware).

Τέλος, θα πρέπει να μελετηθούν και οι επόμενες “πρακτικές” και τεχνολογίες που αφορούν τα συνελκτικά νευρωνικά δίκτυα, όπως για παράδειγμα η πρακτική “residual” η οποία αυξάνει την αποδοτικότητα ενός δικτύου αυξάνοντας το βάθος του[24].

BIBΛΙΟΓΡΑΦΙΑ

- [1] Sankar k. Pal, "Multilayer Perceptron, Fuzzy Sets, and Classification", IEEE Transactions on neural networks, v 3, n 50, p 683, Sep 1992.
- [2] Lenail, A., "NN-SVG: Publication-Ready Neural Network Architecture Schematics", The Journal of Open Source Software 2019.
- [3] Jadon S., "Introduction to different activation functions for deep learning", Medium, Augmenting Humanity, v 16, Mar 2018.
- [4] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. 2014.
- [5] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. "Large-scale video classification with convolutional neural networks." IEEE Conference on Computer Vision and Pattern Recognition, 2014
- [6] P. Sermanet, S. Chintala, and Y. LeCun, "Convolutional neural networks applied to house numbers digit classification," IEEE International Conference on Pattern Recognition (ICPR), n 21, pp. 3288–3291, Apr. 2012.
- [7] P. Sermanet, D. Eigen, "Overfeat: Integrated recognition, localization and detection using convolutional networks" Feb. 2013
- [8] K. He, X. Zhang, "Spatial pyramid pooling in deep convolutional networks for visual recognition," European Conference on Computer Vision,, pp. 346–361, 2014.
- [9] L. Li, K. Jamieson, "Hyperband: A novel bandit-based approach to hyperparameter optimization," Mar. 2016
- [10] Buntine, W. L. and A. S. Weigend, "Bayesian back-propagation", Complex Systems, v 5, pp 603-643, 1991.
- [11] Johansson, E. M., F. U. Dowla, and D. M. Goodman, " Backpropagation learning for multilayer feedforward neural networks using the conjugate gradient method.", International Journal of Neural Systems 1992.

- [12] Tetko, I. V., Livingstone, D. J., and Luik, A. I. "Neural network studies. 1. comparison of overfitting and overtraining", Computer science 1995.
- [13] J. Brownlee, "What is the Difference Between a Batch and an Epoch in a Neural Network?", Machine Learning Mastery, Apr. 2018.
- [14] N. Srivastava, G. E. Hinton et al., "Dropout: a simple way to prevent neural networks from overfitting." Journal of Machine Learning Research, v 15, n 1, pp. 1929–1958, 2014
- [15] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv, Feb 2015.
- [16] Jun D., Bo C., Hongwei L.; Mengyuan H., "Convolutional Neural Network With Data Augmentation for SAR Target Recognition", IEEE Geoscience and Remote Sensing Letters , v 13, Mar 2016)
- [17] R. Wu, S. Yan, Y. Shan, Qingqing Dang, Gang Sun, "Deep image: Scaling up image recognition", arXiv, Jul. 2015.
- [18] C. Alippi, S. Disabato and M. Roveri, "Moving convolutional neural networks to embedded systems: the alexnet and VGG-16 case.", International Conference on Information Processing in Sensor Networks I, Apr. 2018.
- [19] Pedro M. "Transfer learning from pre-trained models", Towards data science, Oct 2018
- [20] <https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition/data?select=train.zip> Feb. 2021
- [21] https://www.tensorflow.org/api_docs/python/tf Feb. 2021
- [22] Y. N. Dauphin, H. de Vries, J. Chung, and Y. Bengio. "Rmsprop and equilibrated adaptive learning rates for non-convex optimization. ", CoRR, 2015.
- [23] Han, S., Pool, J., Tran, J., and Dally, W. "Learning both weights and connections for efficient neural network.", Advances in Neural Information Processing Systems, pp. 1135–1143, 2015.
- [24] K He, X Zhang, S Ren, J Sun, "Deep residual learning for image recognition", Proceedings of the IEEE conference on computer 2016

