

ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ
ΣΧΟΛΗ ΕΠΙΣΤΗΜΩΝ ΠΛΗΡΟΦΟΡΙΑΣ
ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΦΟΙΤΗΤΗΣ : ΜΑΡΙΝΟΣ ΜΑΡΙΟΣ

ΑΝΙΧΝΕΥΣΗ ΚΑΡΚΙΝΟΥ ΤΟΥ ΜΑΣΤΟΥ ΑΠΟ
ΕΙΚΟΝΕΣ ΜΕ ΧΡΗΣΗ ΒΑΘΕΙΩΝ ΝΕΥΡΩΝΙΚΩΝ
ΔΙΚΤΥΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ : ΙΩΑΝΝΗΣ ΡΕΦΑΝΙΔΗΣ

07/07/2020

ΕΥΧΑΡΙΣΤΙΕΣ

Στο σημείο αυτό θα ήθελα να ευχαριστήσω όλους τους ανθρώπους που συνέβαλαν στην εκπόνηση αυτής της πτυχιακής εργασίας. Αρχικά θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή της πτυχιακής κ. Ιωάννη Ρεφανίδη για την πολύτιμη βοήθεια και την υποστήριξη οπότε χρειάστηκα οτιδήποτε για την συγκεκριμένη πτυχιακή. Οι γνώσεις που πήρα από την εργασία αυτή είναι πολλές και τον ευχαριστώ που μου έδωσε την ευκαιρία να ασχοληθώ με κάτι που με ενδιαφέρει τόσο πολύ.

Επίσης θα ήθελα να ευχαριστήσω τους κοντινούς μου ανθρώπους, οικογένεια και φίλους που με υποστήριξαν για την εκπόνηση αυτής της πτυχιακής. Χωρίς αυτούς θα ήταν δυσκολότερο.

Περίληψη

Το αντικείμενο της παρούσας πτυχιακής εργασίας αφορά την ανάπτυξη και εφαρμογή 3^{ων} διαφορετικών μοντέλων συνελεκτικών νευρωνικών δικτύων σε συνδυασμό με νευρωνικά δίκτυα για κατηγοριοποιητή, στο πρόβλημα της δυαδικής κατηγοριοποίησης. Ο στόχος της παρούσας εργασίας ήταν η δημιουργία και έπειτα αξιολόγηση των διάφορων μοντέλων στην πρόβλεψη του καρκίνου του μαστού σε patch αρχικής εικόνας, ενώ ο σκοπός της ήταν καθαρά για να μπορούμε να πούμε με κάποια σιγουρία αν ο ασθενής έχει κάποιο ποσοστό καρκίνου ή όχι. Αυτό θα μπορούσε να χρησιμοποιηθεί σαν βοηθητικό εργαλείο από κάποιον γιατρό για να πάρει την τελική απόφαση όσον αφορά την διάγνωση του. Τα δεδομένα αντλήθηκαν από την πλατφόρμα Kaggle και συγκεκριμένα από το Dataset “Breast Histopathology Images” και αποτελεί ένα από τα λίγα σύνολα δεδομένων με ιατρικά δεδομένα σε μορφή εικόνων. Ως αποτέλεσμα των πειραμάτων που διεξήχθησαν για τον παραπάνω σκοπό, καταλήξαμε σε 3 μοντέλα που όλα είχαν την ίδια βάση συνελεκτικών νευρωνικών δικτύων, ενώ άλλαζαν στην παραμετροποίηση του νευρωνικού δικτύου (κατηγοριοποιητή) και των εξισώσεων ενεργοποίησης, σφάλματος και βελτιστοποίησης. Τέλος, τα αποτελέσματα έδειξαν ότι η καλύτερη μέθοδος ήταν με την χρήση αρκετά μεγάλου νευρωνικού δικτύου και εξίσωση βελτιστοποίησης ο Stochastic Gradient Descent με τελικό macro F1-Score ίσο με 0.87005 ενώ οι υπόλοιπες 2 αρχιτεκτονικές είχαν F1-Score περίπου 1% κάτω. Μελλοντικές βελτιώσεις αποτελεί η χρήση άλλων κατηγοριοποιητών σε συνδυασμό πάντα με τα συνελεκτικά νευρωνικά δίκτυα, ενώ και η «συναρμολόγηση» των patches για να εισαχθούν σε έναν μέτα-κατηγοριοποιητή με αποτέλεσμα να έχουμε ένα τελικό αποτέλεσμα για τον ασθενή.

Λέξεις κλειδιά: νευρωνικά δίκτυα, συνελεκτικά νευρωνικά δίκτυα, διάγνωση καρκίνου του μαστού, αύξηση δεδομένων

Abstract

The purpose of this thesis is to develop and implement 3 different models of convolutional neural networks in combination with neural networks as classifier, to the problem of binary classification. The aim of this current thesis is to create and afterwards appraise the different models on predicting breast cancer on the patches from the original images, whereas the aim was clearly to be able to predict if a patient has a probability of cancer or not. This could be used in conjunction with doctor in order to be able to make the decision easier as far as the diagnosis is concerned. The source dataset came from a Dataset on Kaggle called “Breast Histopathology Images” and it consists one of the few in size with medical information that relate to the subject. In consequence of the experiments that were done for the above reason, we ended up with 3 models that had the same base of convolutional neural networks, whereas it changed totally on the configuration of the neural network (classifier) and activation function, error function and optimization function. Finally, the results indicated that the best method was by using a kinda big neural network with optimization function the Stochastic Gradient Descent and final macro F1-Score equal to 0.87005 whereas the others architectures had 1% F1-Score below. Future improvements that could be used is the implementation of different classifiers than neural networks in combination with convolutional neural networks of course, and also the “assemblage” of patches in order to be introduced to a meta-classifier in order to have a total predict for the patient.

Keywords: Neural Networks, Convolutional Neural Networks, Diagnosis of breast cancer, Data Augmentation

Πίνακας Περιεχομένων

Περίληψη.....	1
Abstract.....	2
Κατάλογος Εικόνων.....	5
Ευρετήριο Πινάκων.....	7
1 Εισαγωγή.....	8
1.1 Πρόβλημα και σημαντικότητα του θέματος.....	8
1.2 Σκοπός και στόχοι Μελέτης.....	9
1.3 Διάρθρωση Μελέτης.....	10
2 Νευρωνικά Δίκτυα.....	11
2.1 Εισαγωγή.....	11
2.2 Βιολογικός Νευρώνας.....	12
2.3 Ιστορική Αναδρομή.....	13
2.4 Τεχνητό Νευρωνικό Δίκτυο.....	15
2.4.1 Συνηθείς Αρχιτεκτονικές Νευρωνικών Δικτύων.....	15
2.4.2 Συναρτήσεις Ενεργοποίησης.....	18
2.4.3 Αλγόριθμος Εκπαίδευσης Νευρωνικού Δικτύου – Οπισθοδιάδοση Σφάλματος....	22
2.4.4 Συναρτήσεις Σφάλματος.....	26
2.4.5 Εξισώσεις Βελτιστοποίησης.....	27
2.5 Προβλήματα Κατά Της Διάρκειας Της Εκπαίδευσης.....	29
3 Συνελεκτικά Νευρωνικά Δίκτυα.....	31
3.1 Συνελεκτικά Επίπεδα.....	31
3.2 Επίπεδα Pooling.....	34
4 Δεδομένα και Μεθοδολογία.....	36
4.1 Εισαγωγή.....	36
4.2 Μεθοδολογία.....	37
4.3 Γλώσσα Προγραμματισμού.....	37
4.4 Περιβάλλον Ανάπτυξης Κώδικα.....	37
4.5 Βιβλιοθήκες.....	38
4.6 Παράμετροι.....	40

4.7	Μετρικές.....	45
5	Πρόβλεψη με Συνελεκτικά Νευρωνικά Δίκτυα.....	50
5.1	Αρχιτεκτονικές Λύσεων και αποτελέσματα.....	51
6	Συμπεράσματα και μελλοντικά σχέδια.....	59
6.1	Μελλοντικές Επεκτάσεις.....	59
	References.....	60

Κατάλογος Εικόνων

- Εικόνα 2.1 : Βιολογικός Νευρώνας
Εικόνα 2.2 : Φυσικοί διασυνδεδεμένοι νευρώνες
Εικόνα 2.3: Πρώτος Τεχνητός Νευρώνας
Εικόνα 2.4 : The Perceptron
Εικόνα 2.5 : Δίκτυο ενός επίπεδου πρόσθιας τροφοδότησης.
Εικόνα 2.6 : Πολύ-επίπεδο δίκτυο πρόσθιας τροφοδότησης (βαθύ νευρωνικό δίκτυο)
Εικόνα 2.7 : Πολυεπίπεδο αναδρομικό δίκτυο (Με κόκκινο και ροζ φαίνεται η αναδρομή που ουσιαστικά δέχεται σαν είσοδο και τον ευατό του)
Εικόνα 2.8 : Βηματική συνάρτηση
Εικόνα 2.9 : Γραμμική Συνάρτηση ενεργοποίησης
Εικόνα 2.10 : Σιγμοειδής συνάρτηση για $b = 0.5, 1, 2.5$
Εικόνα 2.11 : Συνάρτηση ενεργοποίησης ReLU
Εικόνα 2.12 : Σύγκριση ReLU, Leaky ReLU, Parametric ReLU
Εικόνα 2.13 : Backpropagation Υπολογισμός προς τα εμπρός παράδειγμα
Εικόνα 2.14 : Κανόνας αλυσίδας
Εικόνα 2.15 : Χαμηλή, υψηλή και κατάλληλη τιμή για την τιμή του ρυθμού μάθησης
Εικόνα 2.16 : Αριστερά ο αλγόριθμος gradient descent χωρίς ορμή και δεξιά με ορμή
Εικόνα 2.17 : Γραφική αναπαράσταση του Underfitting/Overfitting
Εικόνα 3.1 : Μοντέλο συνελεκτικού νευρωνικού δικτύου
Εικόνα 3.2 : Φίλτρο για ανίχνευση καμπύλης σε εικόνα
Εικόνα 3.3 : Εφαρμογή φίλτρου στην εικόνα
Εικόνα 3.4 : εφαρμογή Stride=2 σε μία εικόνα
Εικόνα 3.5: Εφαρμογή Zero Padding
Εικόνα 3.6 : Παράδειγμα max/average pooling
Εικόνα 4.1 : Precision & Recall
Εικόνα 4.2 : Πίνακας Σύγχυσης (Confusion Matrix)
Εικόνα 4.3 : AUC-ROC Curve
Εικόνα 5.1 : 1^η αρχιτεκτονική μοντέλου
Εικόνα 5.2 : Classification Report για το 1^ο μοντέλο
Εικόνα 5.3 : Πίνακας Σύγχυσης (Confusion Matrix) 1^{ης} αρχιτεκτονικής
Εικόνα 5.4 : Καμπύλη ROC-AUC 1^{ης} αρχιτεκτονικής
Εικόνα 5.5 : 2^η αρχιτεκτονική μοντέλου
Εικόνα 5.6 : Classification Report για το 2^ο μοντέλο
Εικόνα 5.7 : Πίνακας Σύγχυσης (Confusion Matrix) 2^{ης} αρχιτεκτονικής
Εικόνα 5.8 : Καμπύλη ROC-AUC 2^{ης} αρχιτεκτονικής
Εικόνα 5.9 : 3^η αρχιτεκτονική μοντέλου
Εικόνα 5.10 : Classification Report για το 3^ο μοντέλο
Εικόνα 5.11 : Πίνακας Σύγχυσης (Confusion Matrix) 3^{ης} αρχιτεκτονικής
Εικόνα 5.12 : Καμπύλη ROC-AUC 3^{ης} αρχιτεκτονικής

Ευρετήριο Πινάκων

Πίνακας 4.1 : Συγκριτικός πίνακας τελικών αρχιτεκτονικών

1 Εισαγωγή

1.1 Πρόβλημα και σημαντικότητα του θέματος

Όπως είναι γνωστό σε όλους μας ο καρκίνος γενικότερα είναι αν όχι η σημαντικότερη, μία από τις σημαντικότερες ασθένειες που απασχολεί τους ιατρούς και τους ερευνητές τις τελευταίες δεκαετίες. Χωρίζεται σε καλοήγη και κακοήγη με τον καλοήγη καρκίνο να μην κάνει μετάσταση σε άλλα μέλη του σώματος ενώ από την άλλη σε όλα τα είδη του κακοήγη καρκίνου, τα κύτταρα του σώματός αρχίζουν να διαιρούνται χωρίς σταματημό, με αποτέλεσμα να εξαπλώνουν τον καρκίνο και στους περιβάλλοντες ιστούς. Γενικότερα, ο καρκίνος μπορεί να ξεκινήσει σχεδόν από οποιοδήποτε μέλος του σώματος.

Τα είδη του καρκίνου είναι πάνω από 100 με κάποια να είναι πιο συνηθισμένα όπως το καρκίνωμα που σχηματίζεται από επιθηλιακά κύτταρα, τα οποία είναι τα κύτταρα που καλύπτουν τις εσωτερικές και εξωτερικές επιφάνειες του σώματος [1]. Το 2018 τα νέα κρούσματα καρκίνου ήταν ίσα με 17 εκατομμύρια και στα δύο φύλλα με σχεδόν ισάριθμα κρούσματα, με τα πιο 3 συνηθισμένα να είναι: καρκίνος του πνεύμονα και καρκίνος του μαστού με 12.3% ενώ με 10.6% καρκίνος του εντέρου. Στους άντρες τα 3 επικρατέστερα είναι με 15.5% καρκίνος του πνεύμονα, ενώ με 14.5% και 11.4% καρκίνος του προστάτη και καρκίνος του εντέρου αντίστοιχα. Όσον αφορά τις γυναίκες, ο καρκίνος του μαστού φαίνεται να υπερέχει κατά πολύ έχοντας το 25.4% των κρουσμάτων με τον καρκίνο του εντέρου να έρχεται δεύτερος με μόλις 9.7% και τρίτο τον καρκίνο του πνεύμονα με 8.8% [2].

Όπως φαίνεται παραπάνω, ο καρκίνος του μαστού στις γυναίκες κυριαρχεί συντριπτικά οπότε προέκυψε η ανάγκη για έναν μηχανισμό βοήθειας για τους γιατρούς που είναι συγκριτικά πιο αργοί και κοστοβόροι όσον αφορά την διάγνωση ώστε να μπορούμε να διαγνώσουμε τον κακοήγη καρκίνο όσο πιο νωρίς γίνεται και με την μεγαλύτερη ακρίβεια, γνωρίζοντας ότι αν ο καρκίνος βρεθεί νωρίς μπορεί να αντιμετωπιστεί με μεγαλύτερη επιτυχία.

Τα τελευταία χρόνια, έχει αναπτυχθεί πολύ ο κλάδος της ανάλυσης δεδομένων και όλα τα κομμάτια του αλλά πιο συγκεκριμένα η μηχανική μάθηση η οποία αποτελεί κομμάτι της επιστήμης των υπολογιστών. Με τον όρο μηχανική μάθηση εννοούμε μία διαδικασία η οποία μέσω αλγορίθμων καταφέρνει να «εκπαιδεύσει» έναν υπολογιστή να κάνει μία δουλειά χωρίς να έχει προγραμματιστεί ρητά για αυτήν και αυτό το καταφέρνει δημιουργώντας ένα μοντέλο βασισμένο σε παλιότερα δεδομένα της ίδιας δουλειάς. Γενικότερα, η μηχανική μάθηση πλέον έχει εφαρμογή σε πληθώρα τομέων, όπως για παράδειγμα, στην αναζήτηση στον ιστό, στην δημιουργία self-driving αυτοκινήτων, στην δημιουργία bots όπως ή Siri της Apple και άλλες πολλές εφαρμογές. Οπότε είναι πασιφανές ότι όλες οι παραπάνω τεχνικές χρησιμοποιήθηκαν και χρησιμοποιούνται για την ανίχνευση του καρκίνου του μαστού (και όχι μόνο) με αξιοσημείωτα αποτελέσματα.

Υπάρχουν διάφοροι αλγόριθμοι μηχανικής μάθησης που χρησιμοποιούνται για την αναγνώριση καρκίνου σε εικόνες αφού γίνει πρώτα η απαραίτητη προεπεξεργασία στην εικόνα (πχ μετατροπή σε πίνακα αριθμών που αναπαριστούνται τα pixels), όπως ο Support Vector Machines, K-κοντινότερων γειτόνων, κλπ, γενικότερα μπορεί να χρησιμοποιηθεί οποιοσδήποτε κλασσικός αλγόριθμος κατηγοριοποίησης μηχανικής μάθησης, ωστόσο η μέθοδο που επικρατεί τα τελευταία χρόνια και έχει με διαφορά τα καλύτερα αποτελέσματα είναι η state of the art χρησιμοποιεί Νευρωνικά Δίκτυα και πιο συγκεκριμένα τα Συνελεκτικά Νευρωνικά Δίκτυα. Ακόμη μπορεί να χρησιμοποιηθεί και μέθοδος Συνελεκτικών Νευρωνικών Δικτύων μαζί με κάποιον κατηγοριοποιητή από τους γνωστούς (SVM, K-NN, κλπ.) και αφού εφαρμόσουμε τα Συνελεκτικά Νευρωνικά Δίκτυα στις εικόνες, αντί να έχουμε νευρωνικό δίκτυο για την κατηγοριοποίηση να έχουμε κάποιον από τους παραπάνω κατηγοριοποιητές.

Συγκεκριμένα, οι D. Bardou, K. Zhang και S. M. Ahmad στην μέλετη που διεξήγαν για την σύγκριση και ανάλυση διαφόρων τεχνικών μηχανικής μάθησης για την αναγνώριση του καρκίνου του μαστού σε ένα σχετικά μικρό σύνολο δεδομένων, κατέληξαν στο ότι η καλύτερη μέθοδος για την αναγνώριση καρκίνου του μαστού είναι τα Συνελεκτικά Νευρωνικά Δίκτυα μαζί με κατηγοριοποιητή Νευρωνικό Δίκτυο συγκρίνοντας τα αποτελέσματα με Συνελεκτικά Νευρωνικά Δίκτυα και κατηγοριοποιητή Support Vector Machines (SVM), όπου η ακρίβεια μειώθηκε κατά 5% περίπου (από 98% στο 93%). Αυτό οφείλεται στην γραμμικότητα του κατηγοριοποιητή SVM [3]. Ακόμη, σε μία ακόμα έρευνα από τους Araujo και Teresa, έχοντας ένα σύνολο δεδομένων με εικόνες υψηλής ανάλυσης (2040 x 1536 pixels) με περίπου 300 συνολικά εικόνες, χρησιμοποιώντας αρχικά μία μέθοδο κανονικοποίησης [4] και έπειτα μία τεχνική image-wise classification για να σπάσουν την εικόνα σε patches των 512x512 pixels ώστε εκεί να εφαρμοστούν τεχνικές augmentation για την αποφυγή overfitting. Τέλος, χρησιμοποιώντας ένα αρκετά μεγάλο Συνελεκτικό Νευρωνικό Δίκτυο με αρκετά επίπεδα, φτάσαν σε διαφορετικό συμπέρασμα από αυτό των D. Bardou, K. Zhang και S. M. Ahmad, όπου φαίνεται να αποδίδει καλύτερα στα δικά τους δεδομένα τα Συνελεκτικά Νευρωνικά Δίκτυα και στην έξοδο κατηγοριοποιητή SVM [5].

Συνεπώς, μπορούμε να πούμε ότι η εκάστοτε “βέλτιστη” λύση εξαρτάται, αν όχι πάντα τις περισσότερες φορές, από τις ιδιαιτερότητες του συνόλου δεδομένων που με την κατάλληλη προεπεξεργασία, τις σωστές αρχιτεκτονικές Συνελεκτικών Νευρωνικών Δικτύων και τον σωστό κατηγοριοποιητή έξοδου μπορούμε να έχουμε ικανοποιητικά αποτελέσματα για την ανίχνευση του καρκίνου του μαστού.

1.2 Σκοπός και στόχοι Μελέτης

Η παρούσα μελέτη εστιάζει στο πρόβλημα της ανίχνευσης του καρκίνου του μαστού με χρήση Νευρωνικών Δικτύων, και πιο συγκεκριμένα τον έλεγχο απόδοσης των Συνελεκτικών Νευρωνικών Δικτύων με βασικό κατηγοριοποιητή τα Νευρωνικά Δίκτυα. Στόχος είναι η

δημιουργία καλών μοντέλων με τις παραπάνω τεχνικές, η ανάλυση της αρχιτεκτονικής τους και η σύγκριση των αποδόσεων των διάφορων μοντέλων.

1.3 Διάρθρωση Μελέτης

Οι παραπάνω ενότητες αποτελούν μία εισαγωγή στο πρόβλημα και στην εργασία η οποία διαμορφώνεται στην συνέχεια ως εξής :

Στο δεύτερο κεφάλαιο γίνεται βιβλιογραφική εισαγωγή στην ιστορία των Νευρωνικών Δικτύων, στις έννοιες και τους μηχανισμούς που χρησιμοποιούν, στην αρχιτεκτονική καθώς και στην λειτουργία τους.

Στο τρίτο κεφάλαιο που ακολουθεί, εισάγεται η έννοια της Βαθείας μηχανικής μάθησης και των Συνελεκτικών Νευρωνικών Δικτύων ενώ και όλα τα συστατικά που αποτελούν τελικά ένα συνελεκτικό επίπεδο με τα φίλτρα, το padding, pooling και άλλα.

Στο τέταρτο κεφάλαιο παρουσιάζεται η μεθοδολογία με βάση την ρύθμιση των παραμέτρων των δημιουργηθέντων συνελεκτικών νευρωνικών δικτύων και των νευρωνικών δικτύων ενώ και τα εργαλεία που χρησιμοποιήθηκαν στην παρούσα εργασία όπως οι βιβλιοθήκες, γλώσσα προγραμματισμού κλπ. Τέλος, παρουσιάζονται αναλυτικά όλες οι αρχιτεκτονικές και οι παράμετροι που επιλέχθηκαν για τα τελικά μοντέλα μετά από πολλαπλά πειράματα.

Στο 5^ο κεφάλαιο περιγράφεται η τελική αρχιτεκτονική που ακολουθήθηκε των τελικών λύσεων, καθώς και τα πειραματικά αποτελέσματα που πήραμε από αυτές.

Στο 6^ο και τελευταίο κεφάλαιο παρουσιάζονται τα συμπεράσματα των προτεινόμενων τεχνικών, ενώ γίνεται και πρόταση μελλοντικών λύσεων που θα μπορούσαν να βελτιώσουν την απόδοση του συστήματος στα συγκεκριμένα δεδομένα.

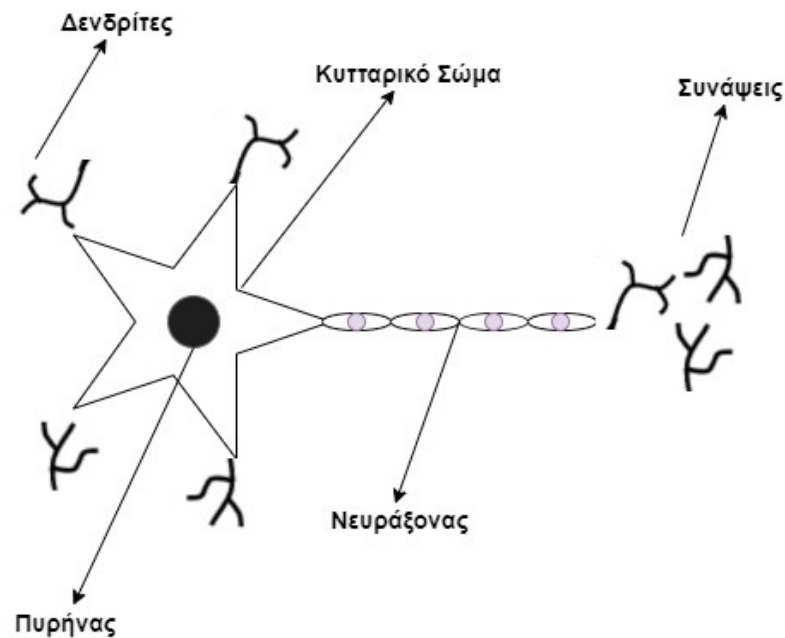
2 Νευρωνικά Δίκτυα

2.1 Εισαγωγή

Τον τελευταίο αιώνα, οι υπολογιστές έχουν αναπτυχθεί τόσο πολύ που μπορούν να κάνουν πράγματα που είναι πολύ επίπονα και χρονοβόρα για τον άνθρωπο και μάλιστα σε πολύ λίγο χρόνο. Πιο συγκεκριμένα, αναφέρομαι στους μαθηματικούς υπολογισμούς που ήταν και ο αρχικός υπαίτιος για την δημιουργία των ηλεκτρονικών υπολογιστών και αργότερα με τα χρόνια άρχισε και η χρήση των ηλεκτρονικών υπολογιστών και για άλλους σκοπούς, όπως ψυχαγωγία, επικοινωνία από απόσταση, ενημέρωση, κλπ.. Γενικότερα, χρειαζόμασταν ένα εργαλείο που θα είναι και γρήγορο όσον αφορά τους μαθηματικούς υπολογισμούς αλλά και αξιόπιστο όσον αφορά το κομμάτι των μαθηματικών ώστε να μπορούμε να πάρουμε σωστές αποφάσεις. Ωστόσο, ο ανθρώπινος εγκέφαλος έχει απίστευτη ικανότητα στην διαχείριση και εξαγωγή συμπερασμάτων από πολυδιάστατες πηγές δεδομένων, χρησιμοποιώντας την εμπειρία που έχει αποκτήσει για να συγκρίνει καταστάσεις οι οποίες έχουν κοινά χαρακτηριστικά αλλά και την πιθανοτική σκέψη του ανθρώπου οδηγούν στο συμπέρασμα ότι είναι αδύνατο ένας υπολογιστής να μπορέσει να προσομοιώσει ή να αντικαταστήσει πλήρως τις λειτουργίες και τις αποφάσεις που μπορεί να πάρει ένας άνθρωπος.

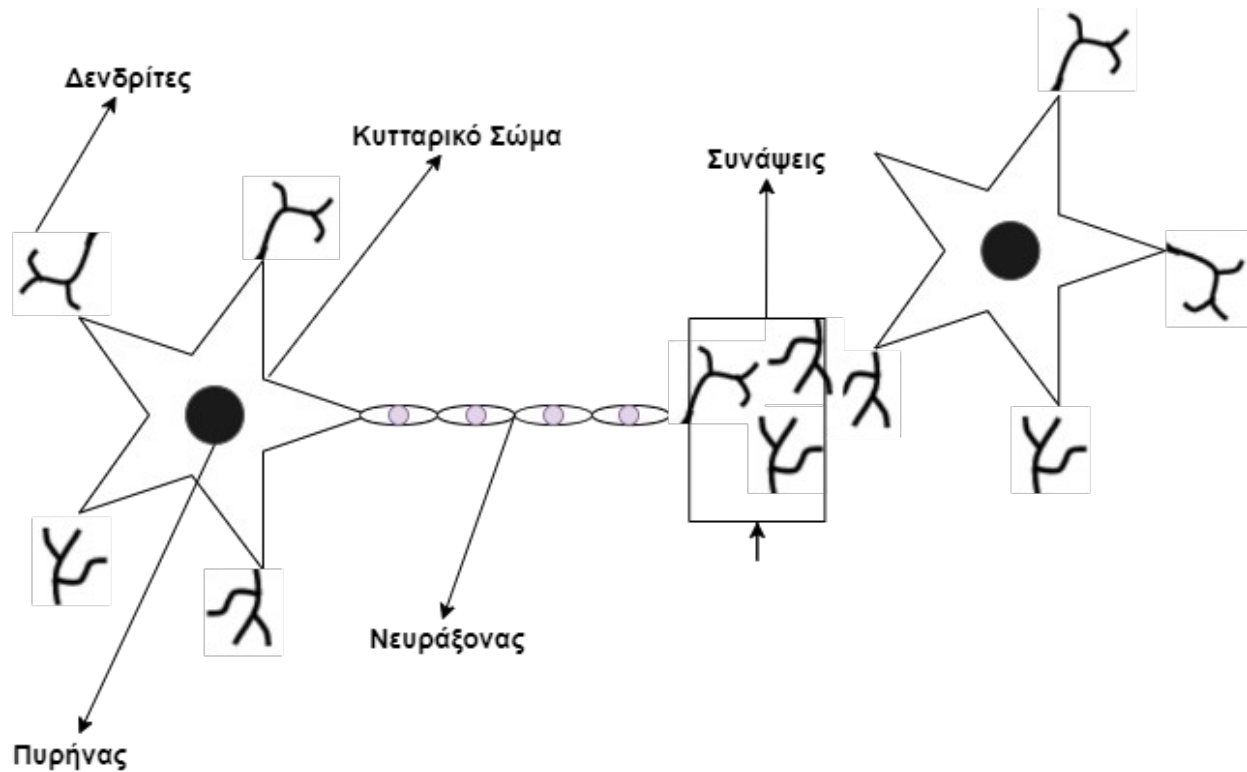
Οπότε, σκέφτηκα αφού στο παρελθόν έχουν γίνει πολλές ανακαλύψεις που πηγάζουν από την φύση όπως τα πουλιά που πετούσαν και ενέμπνευσαν τα αεροπλάνα, γιατί να μην ψάξουν βαθύτερα πώς δουλεύει ο εγκέφαλος για να τον προσομοιώσουν σε μία μηχανή; Αυτή είναι η βασική ιδέα που ενέμπνευσε τα Τεχνητά Νευρωνικά Δίκτυα. Προφανώς όμως δεν μπορεί να προσομοιωθεί πλήρως η λειτουργία του εγκεφάλου και δεν «κληρονομεί» όλα τα χαρακτηριστικά ενός βιολογικού νευρώνα. Συνεπώς, πρώτου δούμε την ιστορία και το πώς εξελίχθηκαν τα Τεχνητά Νευρωνικά Δίκτυα σε αυτό που είναι σήμερα ας δούμε την βασική δομή του βιολογικού νευρώνα.

2.2 Βιολογικός Νευρώνας



Εικόνα 2.1 : Βιολογικός Νευρώνας

Ένας βιολογικός νευρώνας, αποτελείται από 3 κύρια τμήματα (βλέπε Εικόνα 2.2.1.) και αυτά είναι, οι δενδρίτες οι οποίοι λειτουργούν ως κανάλια εισόδου για τον νευρώνα, το κύριως κυτταρικό σώμα, και τον άξονα του κυττάρου-νευροάξονα, που συνδέει έναν νευρώνα με άλλους νευρώνες. Ο άξονας του ενός νευρώνα μεταφέρει σήματα στους δενδρίτες γειτονικών νευρώνων μέσω του σημείου ένωσης που ονομάζεται νευροαξονική απόληξη ή σύναψη (synapse). Ένας νευρώνας μπορεί να λάβει σήματα από ένα σύνολο γειτονικών νευρώνων μέσω των δενδριτών, να τα επεξεργαστεί και να τροφοδοτήσει την έξοδό του μέσω του άξονα προς ένα άλλο σύνολο γειτονικών νευρώνων (βλέπε Εικόνα 2.2.2). Τα σήματα που έρχονται μέσω των δενδριτών «ζυγίζονται» και τα αποτελέσματα αθροίζονται. Όταν το άθροισμα ξεπεράσει το οριακό επίπεδο (τιμή κατωφλίου), ο νευρώνας δημιουργεί μια έξοδο (με τη μορφή νευρικής ώσης ή ηλεκτρικού σήματος) στον άξονά του, η οποία εν συνεχεία μέσω των συνάψεων θα μεταφερθεί στους γειτονικούς νευρώνες [6].



Εικόνα 2.2 : Φυσικοί διασυνδεδεμένοι νευρώνες

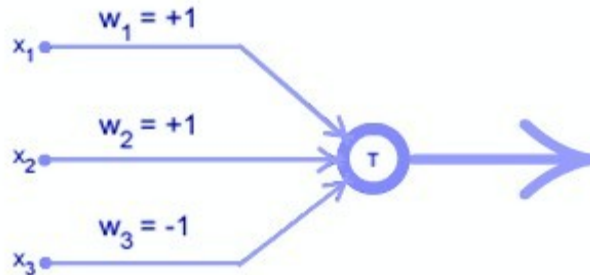
2.3 Ιστορική Αναδρομή

Η πρώτη προσπάθεια που έγινε για την «μετάβαση» από βιολογικό σε τεχνητό νευρώνα έγινε από τους McCulloch ο οποίος ήταν νευροεπιστήμονας και τον μαθηματικό, Pitts το 1943. Σύμφωνα με την εικόνα (βλέπε Εικόνα 2.3.1) ο τεχνητός νευρώνας χωρίζεται σε 2 μέρη. Στο πρώτο μέρος g παίρνει μία εισροή εισόδων όπου μπορεί να είναι είτε διεγερτική ή ανασταλτική το οποίο ορίζεται από τα βάρη, πραγματοποιεί μία συνάθροιση και με βάση την συνολική αξία το δεύτερο μέρος f παίρνει μία απόφαση. Ο μαθηματικός τύπος που το εκφράζει είναι ο εξής:

$$g(x_1, x_2, x_3, \dots, x_n) = g(x) = \sum_{i=1}^n x_i w_i(1)$$

$$y = f(g(x)) = \begin{cases} 1 & \text{if } g(x) \geq \theta \\ 0 & \text{if } g(x) < \theta \end{cases}$$

Όπου τα $x_1, x_2, x_3, \dots, x_n$ κλπ είναι οι εισόδους, w_i τα αντίστοιχα βάρη και τέλος το y είναι η τελική έξοδος του νευρώνα (στην εικόνα αναπαριστάται ως τ) όπου θα μας δίνει την τελική απόφαση βάσει του θ που αποτελεί μία παράμετρος «κατώφλι» όπου ορίζουμε εμείς [7].

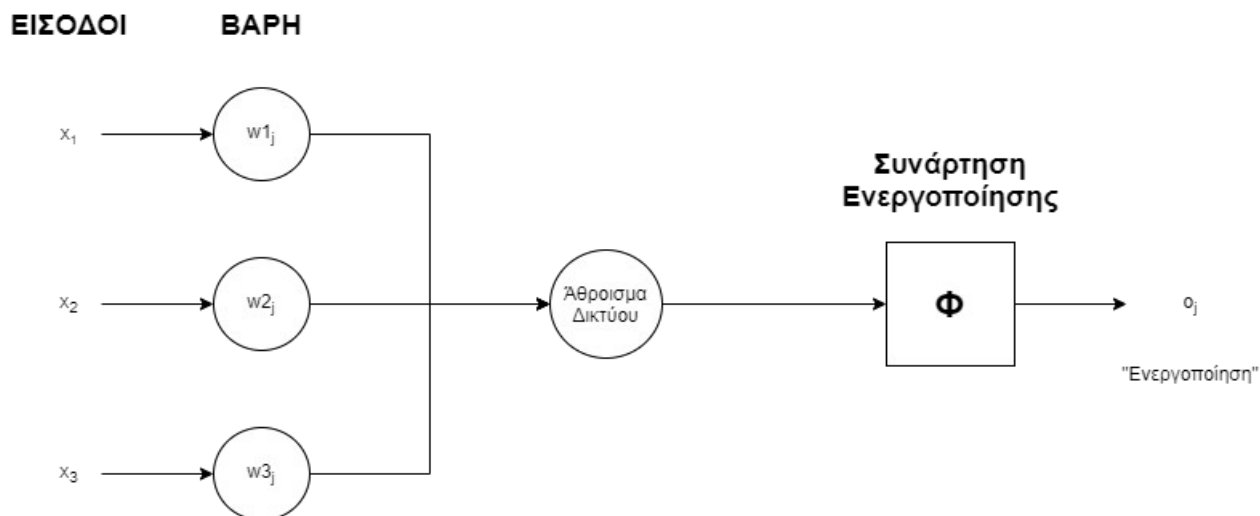


Εικόνα 2.3: Πρώτος Τεχνητός Νευρώνας

Αυτή ήταν και η αρχή που πυροδότησε αργότερα, το 1957, τον Rosenblatt να αναπτύξει τον Perceptron. Η κύρια διαφορά του με αυτό που πρότειναν οι McCulloch και Pitts, ήταν ότι είχαμε και μία συνάρτηση ενεργοποίησης στην έξοδο αμέσως μετά την άθροιση του τύπου 1. Η συνάρτηση που επικράτησε κυρίως σαν συνάρτηση ενεργοποίησης ενώ υπήρχαν και άλλες φυσικά, για τον Perceptron ήταν η βηματική συνάρτηση [8] (βλέπε Εικόνα 2.3.2). Η συνάρτηση αυτή είχε την εξής εξίσωση :

βηματική

Επίσης, κάτι καινούριο που προστέθηκε και δεν φαίνεται στην παρακάτω εικόνα είναι το γεγονός ότι προστέθηκε μία είσοδο που ονομάζεται bias feature με συνηθής τιμή $I_0=1$. Αυτό έγινε για λόγους καλύτερης προσαρμογής του Perceptron στα δεδομένα μιας και σου επιτρέπει να κινείς την συνάρτηση ενεργοποίησης προς τα εκεί που θέλεις «πειράζοντας» το βάρος του bias term το οποίο βοηθάει στην διαδικασία μάθησης.



Εικόνα 2.4 : The Perceptron

Οι Perceptrons φάνηκε να μπορούν να λύσουν απλά δυαδικά προβλήματα κατηγοριοποίησης ικανοποιητικά. Ωστόσο, το 1969 οι Marvin Minsky και Seymour Papert στο βιβλίο τους Perceptrons δείξαν κάποια μειονεκτήματα που έχει ο Perceptron. Πιο συγκεκριμένα, απέδειξαν ότι ο Perceptron δεν μπορεί να λύσει προβλήματα των οποίων τα δεδομένα είναι μη-γραμμικά διαχωρίσιμα και το μείζον παράδειγμα τους για αυτό ήταν η πολύ απλή λογική πύλη XOR. Το γεγονός αυτό απογοήτευσε πολλούς ερευνητές και τους οδήγησε στο να σταματήσουν την έρευνα. Βέβαια, αργότερα αποδείχθηκε ότι το πρόβλημα αυτό μπορούσε να λυθεί χρησιμοποιώντας παραπάνω από έναν Perceptron.

Αργότερα, το 1982 τα Νευρωνικά Δίκτυα άρχισαν να αποκτούν ενδιαφέρον για τους ερευνητές με την εισαγωγή του μοντέλου του Hopfield, το οποίο μπορούσε να χρησιμοποιηθεί ως αποθηκευτικός χώρος. Επιπλέον, σημαντικό ήταν το γεγονός ότι μπορούσε να αποκαταστατήσει την μορφή της εισόδου ακόμα και αν ήταν ημιτελής ή είχε κάποιο ποσοστό θορύβου.

Εκεί που ήρθε η μεγάλη επανάσταση στον χώρο των νευρωνικών δικτύων ήταν το 1986, με τους McClelland, Rumelhart και την ερευνητική ομάδα της PDP εισάγοντας μία μέθοδο για παράλληλη εκπαίδευση νευρωνικών δικτύων με πολλαπλά επίπεδα η οποία λεγόταν Backpropagation η οποία έδωσε την δυνατότητα όχι μόνο να ξεπεράσει το πρόβλημα της μη-γραμμικότητας αλλά και της εκπαίδευσης μεγαλύτερων δικτύων.

Έχοντας ξεπεράσει το κύριο πρόβλημα που ήταν αυτό της εκπαίδευσης των νευρωνικών δικτύων η έρευνα πάνω σε αυτά έφερε πολλά σημαντικά αντικείμενα στο προσκήνιο, όπως τις μηχανές Boltzman, δίκτυα Elman, Recurrent Neural Networks, LSTM, auto-encoders, Convolutional Neural Networks και πολλά άλλα. Συνεπώς, πλέον τα νευρωνικά δίκτυα μπορούν να δώσουν πολύ ικανοποιητικές λύσεις σε προβλήματα που αντιμετωπίζονται και από τους υπόλοιπους αλγορίθμους μηχανικής μάθησης και αποτελούν τεχνολογία αιχμής σε πολλά κομμάτια όπως η κατηγοριοποίηση κειμένου, εικόνων κλπ και αναμένεται να έχουμε μεγαλύτερες ερευνητικές εξελίξεις με την άνοδο της τεχνητής νοημοσύνης.

2.4 Τεχνητό Νευρωνικό Δίκτυο

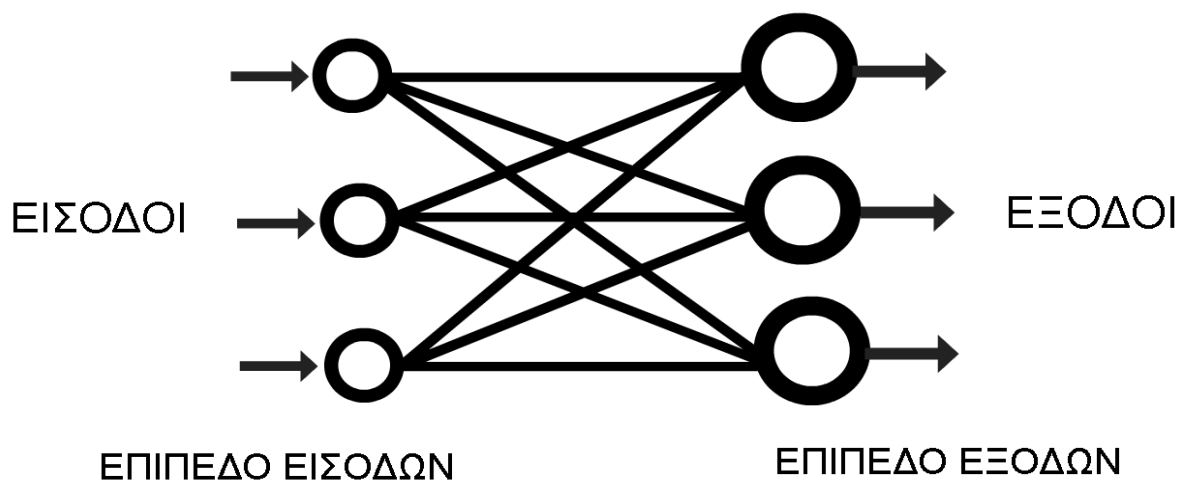
Όπως μπορούμε λοιπόν να συμπαιράνουμε από όλα τα παραπάνω, ο απλός νευρώνας των McCulloch και Pitts και ο Perceptron του Rosenblatt δεν είναι ικανοί να λύσουν προβλήματα μη-γραμμικά και η απόδοση τους γενικότερα είναι πολύ περιορισμένη. Όταν έχοντας μία πολύ καλή μέθοδο εκπαίδευσης (backpropagation) πολλαπλών επιπέδων νευρωνικών δικτύων οδηγηθήκαμε στο να δημιουργούμε σύνολα από τέτοιους νευρώνες, οργανωμένα σε επίπεδα και συνδεδεμένα μεταξύ τους κατάλληλα ώστε να έχουμε τα επιθυμητά αποτελέσματα.

2.4.1 Συνηθείς Αρχιτεκτονικές Νευρωνικών Δικτύων

Ίσως το σημαντικότερο πράγμα που πρέπει να αποφασίσουμε σε ένα δίκτυο είναι η αρχιτεκτονική του, η οποία κατ'επέκταση επηρεάζει την λειτουργία και την διαδικασία μάθησης του νευρωνικού μας δικτύου. Παρακάτω θα παρουσιάσουμε κάποιες πολύ βασικές αρχιτεκτονικές των νευρωνικών δικτύων :

1. Δίκτυα ενός επιπέδου, πρόσθιας τροφοδότησης ή αλλιώς δίκτυα feed forward

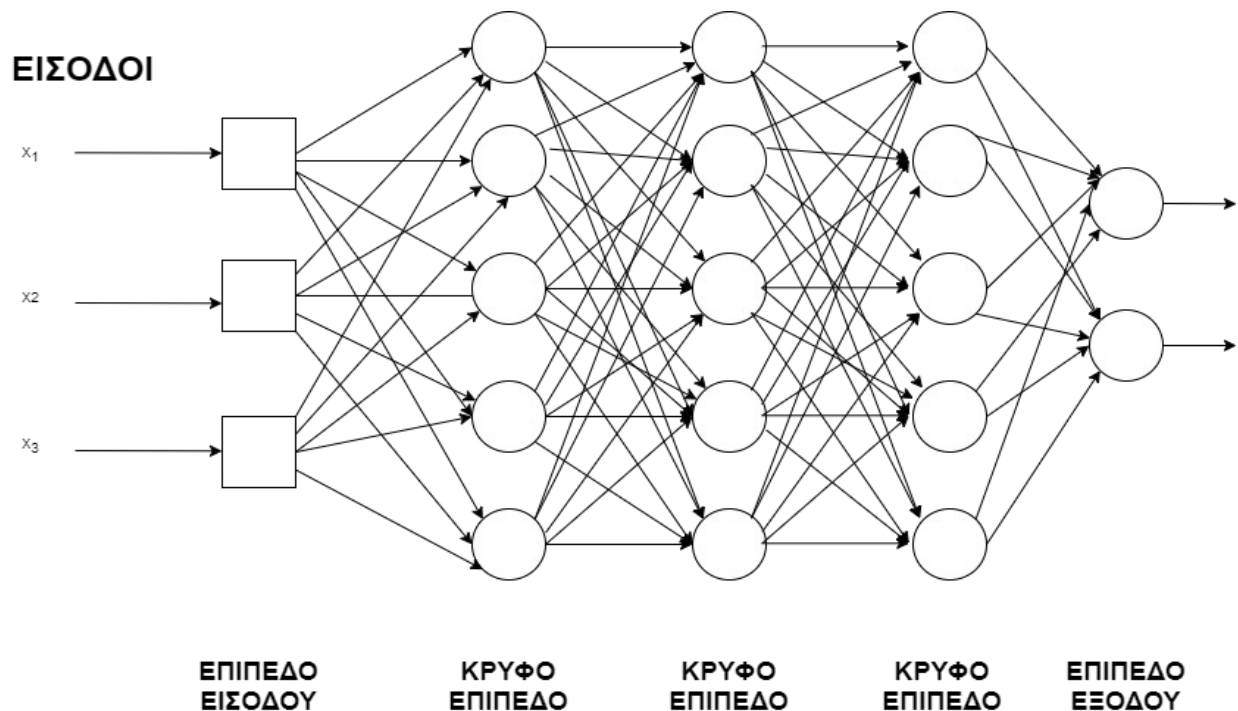
Αποτελούνται από ένα επίπεδο εισόδων το οποίο συνδέεται απευθείας με το επόμενο και μοναδικό επίπεδων νευρώνων εξόδων. Το επίπεδο εισόδου στα δίκτυα δεν προσμετράται στο συνολικό άθροισμα επιπέδων, καθώς σε αυτό δεν εκτελούνται υπολογισμοί. Η συγκεκριμένη αρχιτεκτονική νευρωνικών δικτύων αποτελεί και την απλούστερη που μπορεί να συναντήσει κάποιος η οποία είναι και full-connected, δηλαδή πλήρως συνδεόμενη μιας και όλες οι εισόδους του δικτύου συνδέονται με όλους τους νευρώνες στην έξοδο [9].



Εικόνα 2.5 : Δίκτυο ενός επιπέδου πρόσθιας τροφοδότησης.

2. Πολύ-επίπεδα δίκτυα πρόσθιας τροφοδότησης

Τα δίκτυα αυτά είναι από τις πιο συνηθισμένες αρχιτεκτονικές και αποτελούν την βάση των περισσότερων αρχιτεκτονικών και αποτελείται από 3 μέρη. Το επίπεδο εισόδου, που τροφοδοτεί στο υπόλοιπο δίκτυο τις αριθμητικές τιμές στην μορφή διάνυσματος (διάνυσμα εισόδου), το επίπεδο εξόδου, το οποίο είναι σημαντικό όσον αφορά το τι θέλουμε να προβλέψουμε, και ανάλογα με τις τιμές που έχουν φτάσει θα εκτελέσει την άθροιση, την συνάρτηση ενεργοποίησης και τέλος θα δώσει σαν έξοδο την υπολογισμένη τιμή. Τέλος το κρυφό επίπεδο που αποτελεί ίσως το σημαντικότερο κομμάτι της αρχιτεκτονικής, μπορεί να αποτελείται από οσοδήποτε πολλά επίπεδα νευρώνων και το κάθε ένα επίπεδο μπορεί να έχει όσους νευρώνες εμείς επιλέξουμε. Ένα νευρωνικό δίκτυο με πολλά κρυφά επίπεδα ονομάζεται βαθύ νευρωνικό δίκτυο ενώ πιο απλές αρχιτεκτονικές ονομάζονται τεχνητά νευρωνικά δίκτυα. Ονομάζεται κρυφό καθώς η λειτουργία του δεν είναι άμεσα ορατή στην είσοδο ή στην έξοδο [10]. Ωστόσο, η λειτουργία του κρυφού επιπέδου είναι η σημαντικότερη μετασχηματίζοντας κατάλληλα την είσοδο για την καλύτερη δυνατή πρόβλεψη. Πρακτικά, αυτό που συμβαίνει στους κρυφούς νευρώνες είναι το εξής: Αρχικά, δέχεται τις εισόδους του δικτύου αν μιλάμε για το πρώτο κρυφό επίπεδο αλλιώς δέχεται τις εξόδους του προηγούμενο κρυφού επιπέδου, τις πολλαπλασιάζει με τα αντίστοιχα βάρη τους και έπειτα ο κάθε νευρώνας αθροίζει και περνάει αυτό το άθροισμα από μία συνάρτηση ενεργοποίησης που έχει οριστεί. Οι συναρτήσεις ενεργοποίησης συνήθως είναι οι ίδιες σε όλα τα κρυφά επίπεδα και διαφέρει στο επίπεδο εξόδου ανάλογα με το αντίστοιχο πρόβλημα που έχουμε να αντιμετωπίσουμε. Η συνάρτηση ενεργοποίησης είναι ένα πολύ σημαντικό κομμάτι των αρχιτεκτονικών και θα εξηγηθούν σχολαστικά παρακάτω στην εργασία.



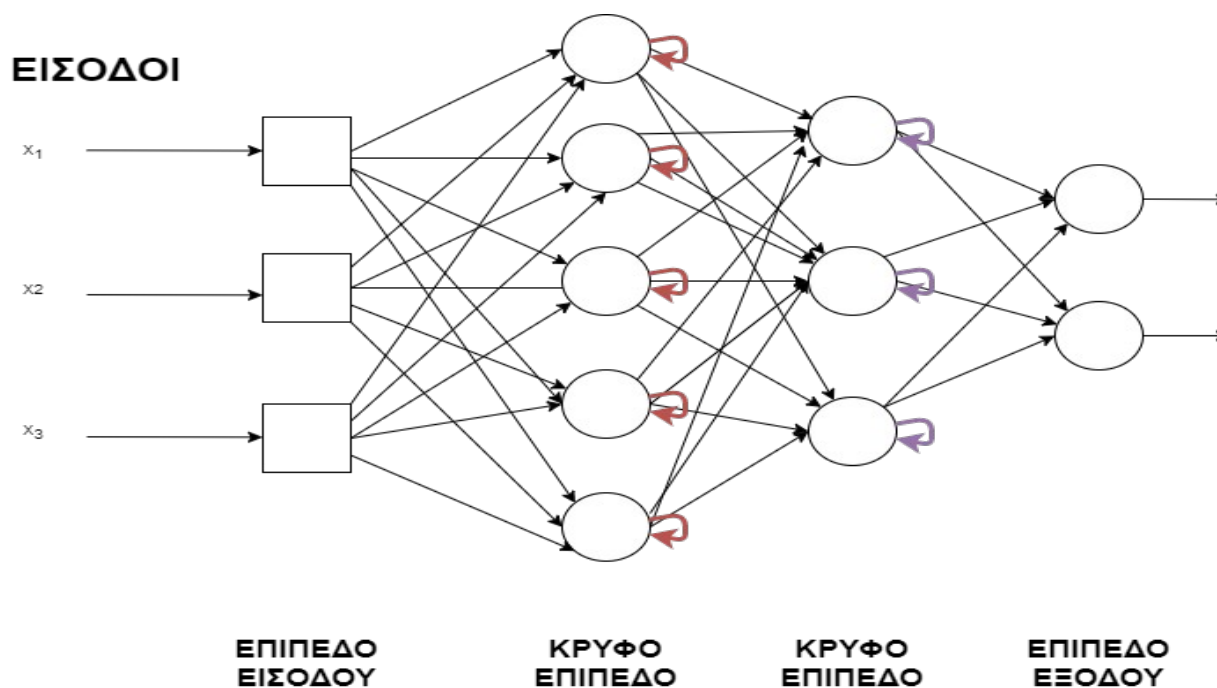
Εικόνα 2.6 : Πολύ-επίπεδο δίκτυο πρόσθιας τροφοδότησης (βαθύ νευρωνικό δίκτυο)

Το νευρωνικό δίκτυο της παραπάνω εικόνας είναι ένα βαθύ νευρωνικό δίκτυο μιας και έχει πολλά κρυφά επίπεδα. Ονομάζεται πλήρως συνδεδεμένο για τον λόγο ότι κάθε κόμβος του προηγούμενο επιπέδου συνδέεται με όλους τους κόμβους του επόμενου επιπέδου με κατεύθυνση

προς τα εμπρός. Αξιοσημείωτο είναι ότι υπάρχουν δίκτυα που δεν ακολουθούν την παραπάνω συμπεριφορά της πλήρης συνδεσιμότητας τα οποία δουλεύουν πολύ καλά για την αντιμετώπιση του overfitting. Η μέθοδος αυτή ονομάζεται dropout η οποία «απενεργοποιεί» κάποιους νευρώνες. Θα συζητηθούν και τα δύο παρακάτω στην εργασία.

3. Αναδρομικά δίκτυα

Αυτό που διαφοροποιεί τα αναδρομικά δίκτυα από τα υπόλοιπα είναι το γεγονός ότι διαθέτουν τουλάχιστον μία σύνδεση, αν όχι παραπάνω με τον ίδιο τους τον εαυτό. Σύνηθες είναι επίσης να έχουν συνδέσεις και με προηγούμενα επίπεδα πέρα από τον ευατό τους. Ακόμη υπάρχουν και τα LSTM επαναληπτικά δίκτυα που έχουν την δυνατότητα να «θυμούνται» παλιές καταστάσεις. Έπόμενο είναι τα δίκτυα αυτά να έχουν διαφορετική συμπεριφορά όσον αφορά την διαδικασία μάθησης και φυσικά της απόδοσης τους.

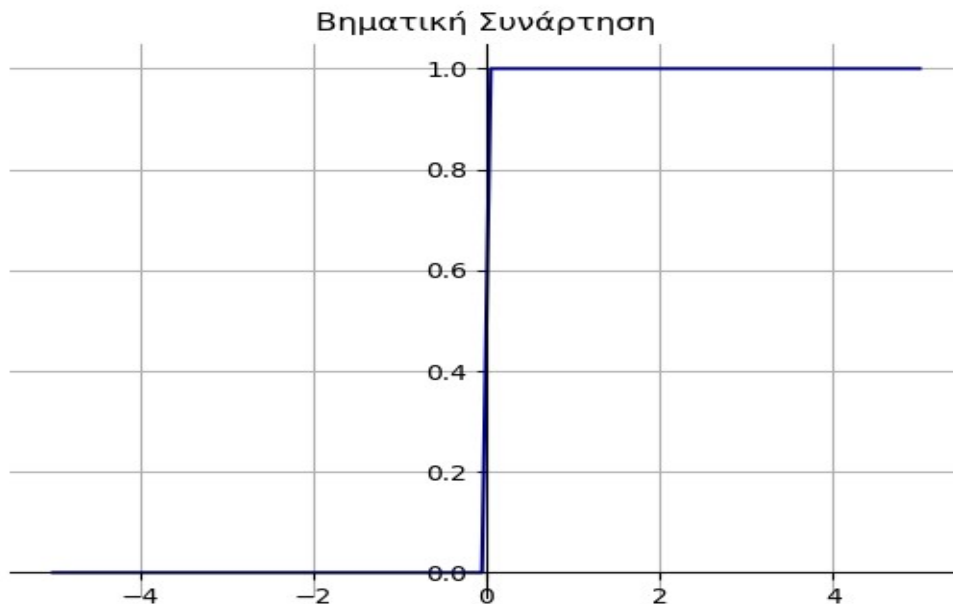


Εικόνα 2.7 : Πολυεπίεδο αναδρομικό δίκτυο (Με κόκκινο και ροζ φαίνεται η αναδρομή που ουσιαστικά δέχεται σαν είσοδο και τον ευατό του)

2.4.2 Συναρτήσεις Ενεργοποίησης

Οι εξισώσεις ενεργοποίησης όπως προαναφέρθηκε παίζουν μείζον ρόλο στην συμπεριφορά του δικτύου και όσον αφορά την προσπάθεια της εκπαίδευσής τους. Για τον λόγο αυτό έχουμε κατασκευάσει αρκετές διαφορετικές συναρτήσεις ενεργοποίησης και πρέπει να είμαστε προσεκτικοί στην επιλογή τους ώστε να μην έχουμε προβλήματα. Συνηθίζεται να έχουμε διαφορετική συνάρτηση ενεργοποίησης στο επίπεδο εξόδου με το κρυφό επίπεδο μιας και το επίπεδο εξόδου πρέπει να έχει κατάλληλη συνάρτηση ενεργοποίησης για να εξαχθεί η σωστή «απόφαση». Παρακάτω θα δούμε κάποιες από τις βασικότερες συναρτήσεις ενεργοποίησης ξεκινώντας με :

- Βηματική συνάρτηση ενεργοποίησης : $f(x) = \begin{cases} 1, & \text{αν } x \geq 0 \\ 0, & \text{αν } x < 0 \end{cases}$

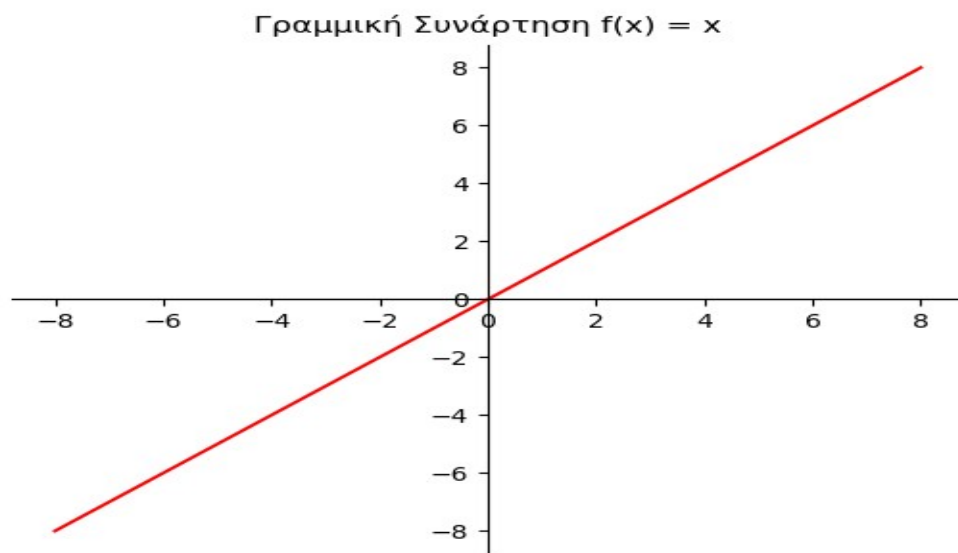


Εικόνα 2.8 : Βηματική συνάρτηση

Η βηματική συνάρτηση ήταν και η πρώτη που χρησιμοποιήθηκε σαν συνάρτηση ενεργοποίησης και ήταν και αυτή που είχε δημοσιεύσει ο Rosenblatt στην κατασκευή του Perceptron. Ωστόσο, έχει πολλά μειονεκτήματα με πρώτο και πιο σημαντικό ότι είναι εξαιρετικά δύσκολο να κάνει πρόβλεψη για περισσότερες από δύο κλάσεις ή αλλιώς δουλεύει μόνο για δυαδικά προβλήματα κατηγοριοποίησης (binary classification). Ένα ακόμη πρόβλημα είναι το γεγονός ότι η συγκεκριμένη συνάρτηση δεν είναι παραγωγίσιμη στο 0 το οποίο είναι θέμα όσον αφορά την εκπαίδευση μέσω της μεθόδου backpropagation που θα αναλυθεί παρακάτω στην εργασία. Επίσης μία αντίστοιχη συνάρτηση που απλά την αναφέρω σαν επέκταση της βηματικής είναι η συνάρτηση προσήμου που η διαφορά είναι ότι η προσήμου δίνει έξοδο -1 όπου $x < 0$ και έχει τα ίδια προβλήματα με την βηματική.

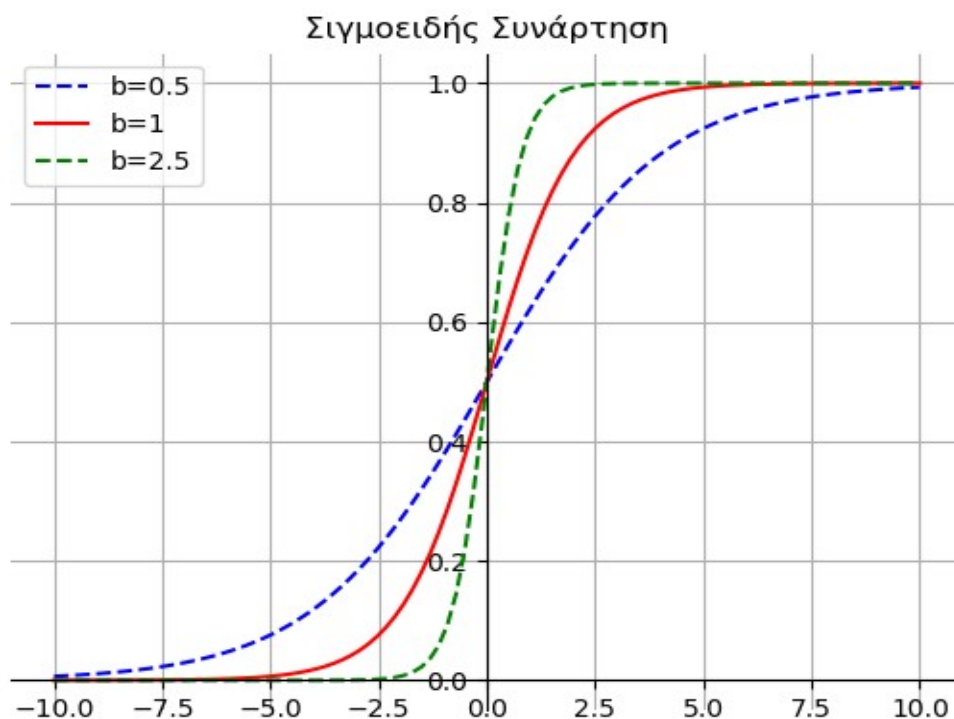
- Γραμμική συνάρτηση ενεργοποίησης :

Από την μία η γραμμική συνάρτηση λύνει το πρόβλημα της δυαδικής κατηγοριοποίησης αφού γνωρίζουμε ποιοί νευρώνες «ενεργοποιούνται» μπορούμε να πάρουμε την σωστή απόφαση βάσει αυτού και να έχουμε και πολλαπλή κατηγοριοποίηση. Το πρόβλημα που προκύπτει με την μέθοδο αυτή είναι ότι η παράγωγος της είναι μία σταθερά και ουσιαστικά πάλι αντιμετωπίζουμε πρόβλημα στην εκπαίδευση του δικτύου όταν έχουμε σφάλμα μιας και πάντα θα αλλάζουμε το σφάλμα με την σταθερά. Επίσης, υπάρχει το πρόβλημα της γραμμικότητας που θέλουμε να αποφύγουμε για προφανής λόγους.



Εικόνα 2.9 : Γραμμική Συνάρτηση ενεργοποίησης

- Σιγμοειδής συνάρτηση : $f(x) = \frac{1}{1+e^{-bx}}$

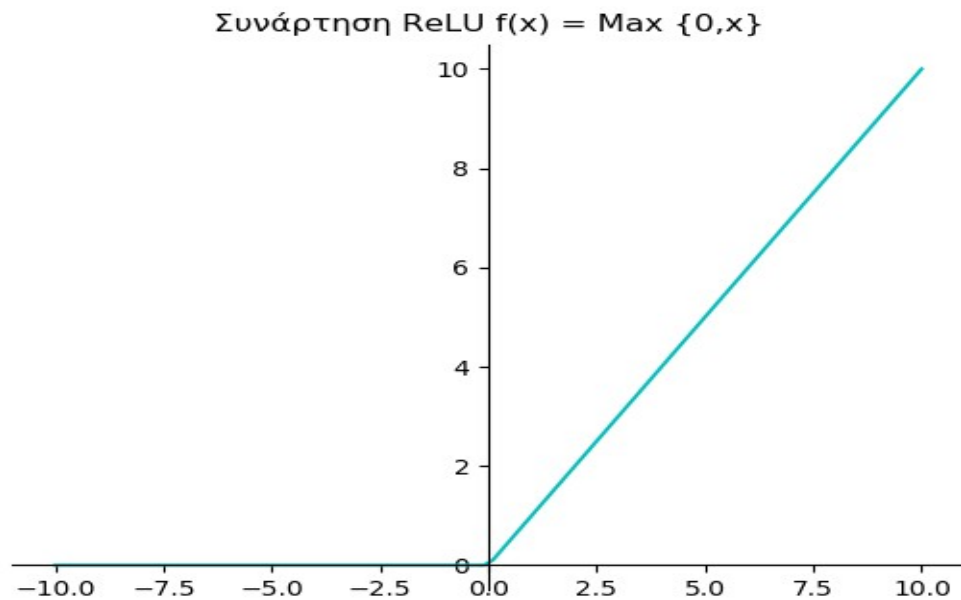


Εικόνα 2.10 : Σιγμοειδής συνάρτηση για $b = 0.5, 1, 2.5$

Αρχικά, η σιγμοειδής συνάρτηση λύνει τα περισσότερα προβλήματα από τις δύο προηγούμενες συναρτήσεις που παρουσιάσαμε παραπάνω. Πρώτον, δεν είναι γραμμική συνάρτηση. Επίσης, μπορούμε να έχουμε και μη δυαδική κατηγοριοποίηση μιας και δίνει αναλογικές εξόδους ενεργοποίησης και όχι μόνο 0 ή 1 όπως την βηματική. Οι σιγμοειδής νευρώνες χρησιμοποιούνται μέχρι και σήμερα και είναι μία από τις πιο χρησιμοποιούμενες συναρτήσεις ενεργοποίησης. Ωστόσο, έχει ένα μειονέκτημα. Το γεγονός ότι όταν φτάσει το η έξοδος να είναι πολύ κοντά στο 1 ή στο 0, δηλαδή στα σημεία που οριζοντιώνεται η συνάρτηση η κλίση τείνει προς το 0 ή γίνεται 0 και αυτό οδηγεί σε δυσκολία της εκπαίδευσης του δικτύου.

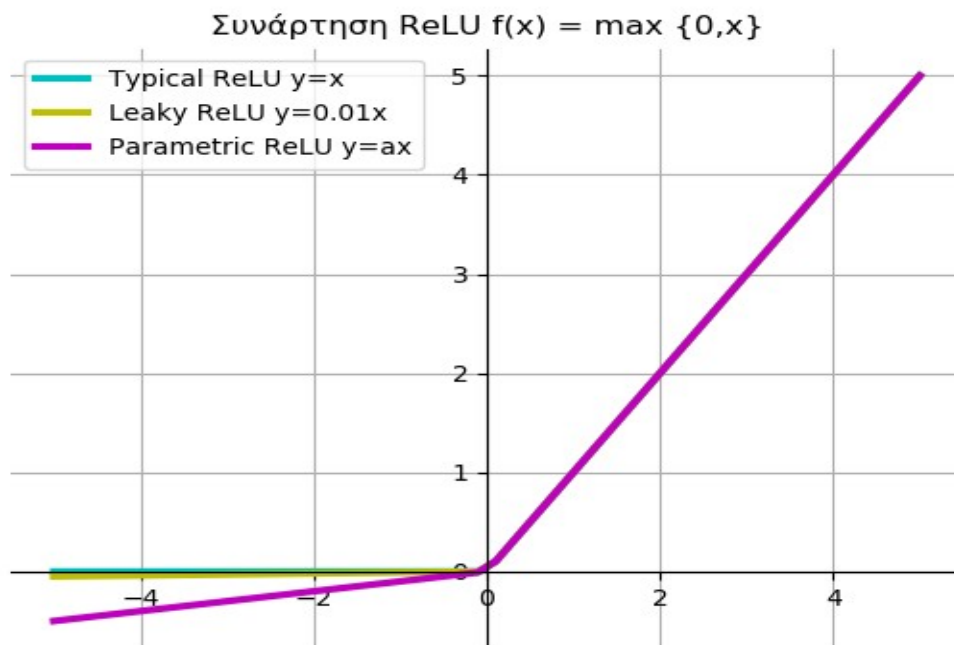
- Γραμμική μονάδα ανόρθωσης (ReLU):

Τέλος, η συνάρτηση ReLU η οποία αποτελεί την πλέον πιο διαδεδομένη συνάρτηση ενεργοποίησης ειδικά στα βαθιά νευρωνικά δίκτυα. Ο λόγος της μεγάλης επιτυχίας της συνάρτησης έγκειται στην απλότητα και την αποτελεσματικότητά της, αφού τα βαθιά νευρωνικά δίκτυα επιτυγχάνουν γρηγορότερη σύγκλιση σε σχέση με άλλες συναρτήσεις. Η ReLU είναι αρκετά παρόμοια με την γραμμική συνάρτηση, με την διαφορά ότι η έξοδος δίνει μηδέν για όλες τις μη θετικές εισόδους. Έτσι οι μερικές παράγωγοί της, παραμένουν μεγάλες και συνεχείς, με αποτέλεσμα να δύνανται να «κυκλοφορήσουν» στο δίκτυο [11].



Εικόνα 2.11 : Συνάρτηση ενεργοποίησης ReLU

Παρατηρώντας την παραπάνω συνάρτηση μπορούμε εύκολα να εντοπίσουμε το πρόβλημα που έχει και αυτή η συνάρτηση ενεργοποίησης. Στις αρνητικές τιμές δίνει έξοδο 0 με αποτέλεσμα να οδηγούμαστε σε παρεμφερεί προβλήματα των υπόλοιπων συναρτήσεων, όπως η κλίση να είναι 0 και επίσης των «νεκρών νευρώνων», δηλαδή τον μηδενισμό της εξόδου του νευρώνα με αποτέλεσμα να μην αλλάζουν τα βάρη αυτών. Για την αντιμετώπιση αυτών των προβλημάτων δημιουργήθηκαν διαφορετικές εκδοχές τις ReLU όπως οι Leaky ReLU, Swish ReLU, Parametric ReLU οι οποίες έχουν την ίδια λειτουργία για $x > 0$ ενώ για τις αρνητικές δίνουν κάποια μικρή αλλά μη-μηδενική τιμή στην έξοδο .



Εικόνα 2.12 : Σύγκριση ReLU, Leaky ReLU, Parametric ReLU

2.4.3 Αλγόριθμος Εκπαίδευσης Νευρωνικού Δικτύου – Οπισθοδιάδοση Σφάλματος

Αρχικά, ο αλγόριθμος της εκπαίδευσης νευρωνικού δικτύου, οπισθοδιάδοση σφάλματος, ανήκει στην κατηγορία αλγορίθμων επιβλεπόμενης μάθησης. Γενικότερα, υπάρχουν 3 κατηγορίες-τομείς της μηχανικής μάθησης και είναι η εξής : Επιβλεπόμενη μάθηση, μη-επιβλεπόμενη μάθηση και ενισχυτική μάθηση, ωστόσο στην παρακάτω εργασία θα μας απασχολήσει μόνο ο αλγόριθμος ο οπισθοδιάδοσης σφάλματος και συνεπώς η επιβλεπόμενη μάθηση. Οπότε με τον όρο επιβλεπόμενη μάθηση, εννοούμε την εκπαίδευση κάποιου μοντέλου με κάποιον αλγόριθμο που για το συγκεκριμένο σύνολο δεδομένων γνωρίζουμε τις αντίστοιχες πραγματικές τιμές εξόδου που αντιστοιχούν στο κάθε παράδειγμα.

Στην συγκεκριμένη περίπτωση χρησιμοποιώντας τον αλγόριθμο οπισθοδιάδοσης σφάλματος, έχουμε ένα διάνυσμα τιμών n διαστάσεων το οποίο προβάλλουμε σε ένα διάνυσμα εξόδου m διαστάσεων. Συνοπτικά αυτό που κάνει ο αλγόριθμος χωρίζεται σε δύο μέρη, υπολογισμός προς τα εμπρός και υπολογισμός προς τα πίσω. Αρχικά ο αλγόριθμος για ένα συγκεκριμένο παράδειγμα εκπαίδευσης δέχοντας την αντίστοιχη είσοδο και μεταφέρει τα «σήματα» μέσω του κρυφού επιπέδου στο επίπεδο εξόδου όπου παράγεται το αντίστοιχο διάνυσμα εξόδου που είναι η εκτίμηση του δικτύου για το αντίστοιχο παράδειγμα. Στο δεύτερο μέρος, υπολογίζεται η τιμή του σφάλματος ως η επιθυμητή έξοδος – εκτιμώμενη έξοδος δικτύου και σύμφωνα με τον κανόνα διόρθωσης του λάθους αλλάζουν τα βάρη στο δίκτυο έτσι ώστε την επόμενη φορά για

το συγκεκριμένο παράδειγμα η εκτιμώμενη έξοδος του δικτύου να συγκλίνει προς την επιθυμητή έξοδο.

Παρακάτω θα παρουσιάσω πιο σχολαστικά τα 2 μέρη του αλγορίθμου :

1. Υπολογισμός προς τα εμπρός

Έστω ότι έχουμε είσοδο ένα παράδειγμα εκπαίδευσης $[x, d]$ όπου x το διάνυσμα εισόδου του παραδείγματος και d το επιθυμητό διάνυσμα εξόδου. Στην συνέχεια υπολόγισε τις τιμές των νευρώνων και τα διαδοχικά σήματα του δικτύου με κατεύθυνση προς τα εμπρός, επίπεδο προς επίπεδο. Η τιμή του νευρώνα j του L επιπέδου είναι:

$$z_j^{(L)} = w_{ji}^{(L-1)} a_i^{(L-1)} + b_j^{(L-1)} (1)$$

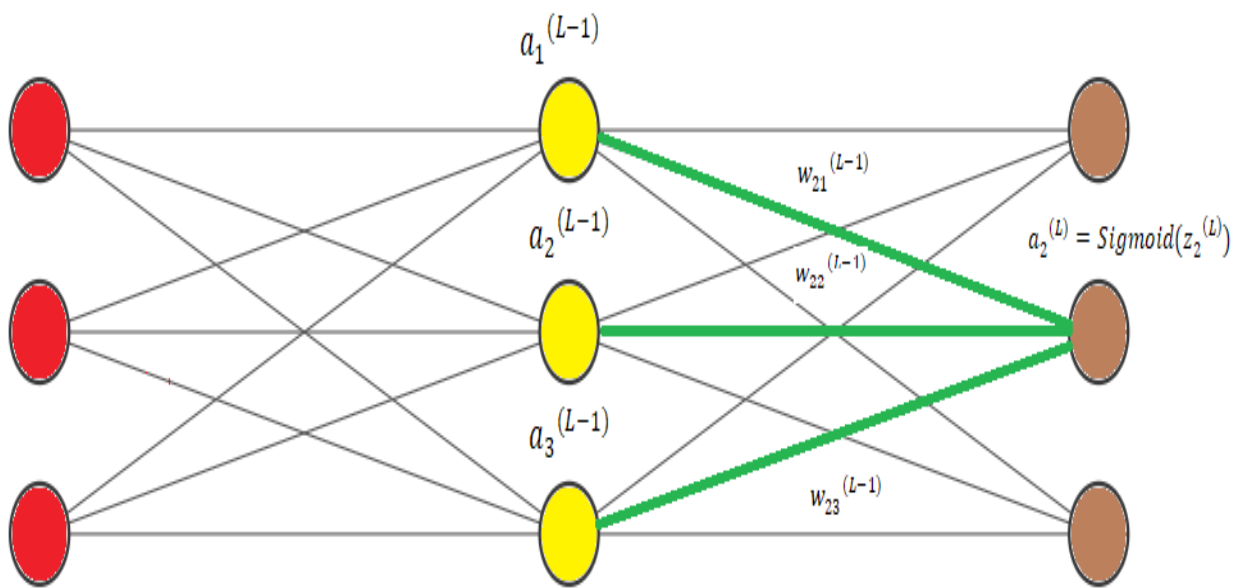
Όπου $a_i^{(L-1)}$ η τιμή του σήματος του νευρώνα i του προηγούμενου επιπέδου, $w_{ji}^{(L-1)}$ το βάρος του νευρώνα j του επιπέδου L που μεταδίδεται από τον νευρώνα i του επιπέδου $L-1$ και $b_j^{(L-1)}$ μία σταθερά bias του αντίστοιχου νευρώνα. Για λόγους απλότητας υποθέτουμε ότι χρησιμοποιείται σιγμοειδής συνάρτηση ενεργοποίησης οπότε η έξοδος του νευρώνα γίνεται :

$$a_i^{(L)} = \sigma(z_j^{(L)})$$

Τέλος, υπολογίζεται η τιμή του σφάλματος με το ην εξίσωση του μέσου τετραγωνικού σφάλματος που είναι η επιθυμητή έξοδος d_j εκτιμώμενη έξοδος o_j στο τετράγωνο.

$$Err_j = (d_j - o_j)^2$$

Παρακάτω στην εικόνα 2.13 φαίνεται πώς δουλεύει ο τύπος (1) για τον νευρώνα $a_2^{(L)}$. N_{L-1} είναι ίσο με 3 δηλαδή στο προηγούμενο επίπεδο του j νευρώνα υπάρχουν 3 νευρώνες οπότε για αυτούς τους 3 θα πάρουμε το άθροισμα αμέσως αφού πολλαπλασιάσουμε τα αντίστοιχα βάρη με τις σημάτων των νευρών και προσθέσουμε την σταθερά bias (η οποία δεν φαίνεται στο παρακάτω σχήμα για λόγους απλότητας). Τέλος, αυτό το άθροισμα το περνάμε από μία συνάρτηση ενεργοποίησης και έχουμε την τελική τιμή του σήματος του νευρώνα $a_2^{(L)}$.



Εικόνα 2.13 : Backpropagation Υπολογισμός προς τα εμπρός παράδειγμα

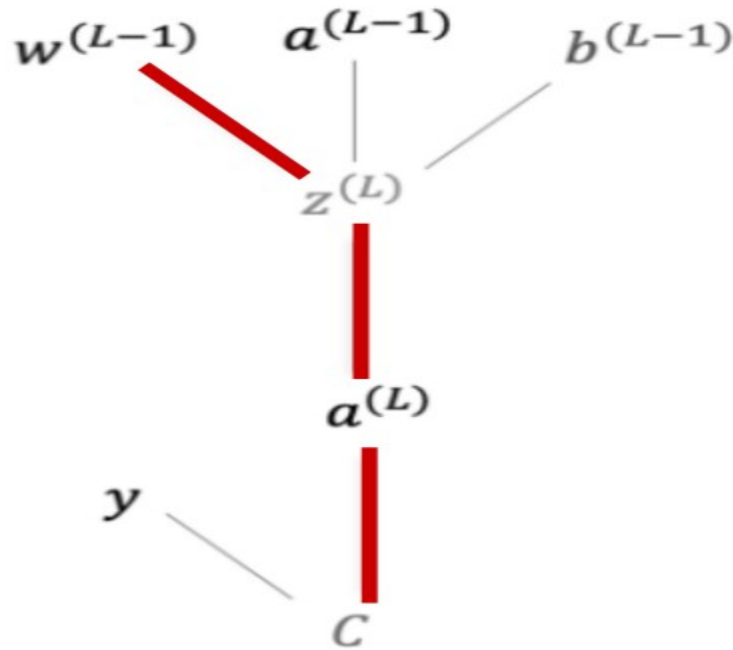
2. Υπολογισμός προς τα πίσω

Συνεχίζοντας με τον υπολογισμό προς τα πίσω (backpropagation) ο κύριος σκοπός μας είναι να ελαχιστοποιήσουμε το σφάλμα (μέσω τετραγωνικό σφάλμα). Αυτό γίνεται με τον γνωστό σε όλους μας «κανόνα της αλυσίδας» από τα μαθηματικά και τις παραγώγους. Βρίσκουμε πόσο θα αλλάξει το σφάλμα αν αλλάξουμε το βάρος στο επίπεδο L-1. Στην εικόνα 2.14 φαίνεται ως C το τελικό σφάλμα στο τελικό επίπεδο, ενώ το z^L και a^L είναι τα αντίστοιχα σήματα στον νευρώνα πριν και μετά την εφαρμογή της συνάρτησης ενεργοποίησης και τέλος έχουμε το βάρος, το σήμα εξόδου και τον bias term του προηγούμενου επιπέδου. Οπότε για να βρούμε πώς επηρεάζεται το συνολικό σφάλμα από το βάρος του νευρώνα για να το αλλάξουμε προς την κατεύθυνση που ελαχιστοποιείται το σφάλμα έχουμε :

$$\delta_j^L = \frac{\partial C}{\partial w_{ji}^{(L-1)}} = \frac{\partial z_j^L}{\partial w_{ji}^{(L-1)}} \frac{\partial a_j^L}{\partial z_j^L} \frac{\partial C}{\partial a_j^L}$$

παραγωγίζοντας όλα τα παραπάνω από το πρώτο μέρος του αλγορίθμου έχουμε

$$\delta_j^L = w_{ji}^{(L-1)} \sigma'(z_j^L) 2(a_j^L - o_j)$$



Εικόνα 2.14 : Κανόνας αλυσίδας

Έπειτα για τον υπολογισμό του νεού βάρους στην επανάληψη n του νευρώνα χρησιμοποιείται ο γενικευμένος κανόνας Δέλτα για κάθε επίπεδο L .

$$w_{ji}^{L(n+1)} = w_{ji}^{L(n)} + m \left[w_{ji}^{L(n-1)} \right] + d \delta_j^L(n) a_i^{(L-1)}(n)$$

Όπου d ο ρυθμός μάθησης και m ο συντελεστής της ορμής.

Ο Αλγόριθμος οπισθοδιάδοσης του σφάλματος είναι ο πλέον δημοφιλής αλγόριθμος μάθησης νευρωνικών δικτύων, ωστόσο υπάρχουν και οι περιπτώσεις που ο αλγόριθμος δεν παράγει ικανοποιητικά αποτελέσματα. Αυτό μπορεί να οφείλεται σε διάφορους λόγους, και πρέπει εμείς να κάνουμε διάφορες παραμετροποιήσεις στο δίκτυο μέχρι να έχουμε τα επιθυμητά αποτελέσματα. Ακόμη ένα μείζον πρόβλημα είναι το γεγονός ότι μπορεί το δίκτυο να συγκλίνει σε κάποιο τοπικό ελάχιστο και να μην μπορεί να φύγει από αυτό σε κάποιο καλύτερο ή ολικό ελάχιστο με αποτέλεσμα οι μεταβολές των βαρών να είναι μηδαμινές και συνεπώς σταματάει η εκπαίδευση. Για να αντιμετωπίσουμε το παραπάνω θέμα υπάρχουν διάφορες τεχνικές όπως η σωστή αρχικοποίηση των βαρών, η εναλλαγή των τιμών του ρυθμού μάθησης και της ορμής ή και ακόμη διάφορες συναρτηρήσεις βελτιστοποίησης που τροποποιούν δυναμικά τα δύο παραπάνω μεγέθη.

2.4.4 Συναρτήσεις Σφάλματος

Όπως είδαμε μέχρι στιγμής, για την εφαρμογή του αλγορίθμου χρειάζεται μία μετρική σφάλματος, στην περίπτωση μας χρησιμοποιήσαμε το μέσο τετραγωνικό σφάλμα. Γενικότερα, πρέπει να επιλέξουμε την κατάλληλη μετρική σφάλματος από την πληθώρα μετρικών σύμφωνα με το αντίστοιχο πρόβλημα που έχουμε να αντιμετωπίσουμε σε όλες τις τεχνικές μηχανικής μάθησης συνεπώς και στα νευρωνικά δίκτυα. Έχουμε δύο κύριες κατηγορίες προβλημάτων οι οποίες χωρίζονται σε προβλήματα κατηγοριοποίησης και προβλήματα παλινδρόμησης.

Στα προβλήματα παλινδρόμησης σκοπός είναι η εφαρμογή μιας γραμμικής εξίσωσης σε ένα σύνολο δεδομένων για την πρόβλεψη συνεχών τιμών. Τέτοια προβλήματα μπορεί να είναι η πρόβλεψη του Ακαθάριστου Εγχώριου Προϊόντος μίας χώρας ή η πρόβλεψη τιμής σπιτιών. Για τέτοιου είδους προβλημάτων φαίνεται ότι η χρήση της μετρικής του μέσου τετραγωνικού σφάλματος να επιφέρει πολύ καλά αποτελέσματα. Η εξίσωση έχει ως εξής :

$$MSE = \frac{1}{n} \sum_{i=1}^n (d_j - o_j)^2$$

Ενώ για προβλήματα κατηγοριοποίησης, δηλαδή την πιθανοκρατική πρόβλεψη της κατηγορίας μίας εισόδου, η χρήση της διασταυρωμένης εντροπίας έχει πολύ καλές αποδόσεις. Η διασταυρωμένη εντροπία ουσιαστικά μετράει την αταξία ενός συστήματος, δηλαδή πόσο δύσκολα μπορεί να προβλέψει την κατάσταση αυτού σε μία ορισμένη χρονική στιγμή. Η διασταυρωμένη εντροπία για k αριθμό κλάσεων, ορίζεται ως εξής :

$$H(p, q) = - \sum_{i=1}^k p(c_i) \log(q(c_i))$$

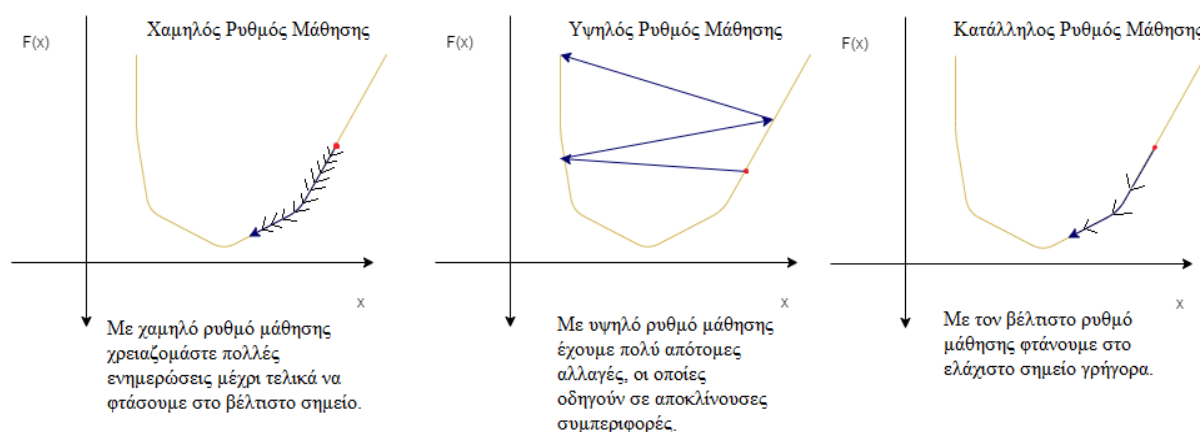
Όπου p η πραγματική κατανομή, που αποδίδει πιθανότητα 1 σε μια κλάση και 0 στις υπόλοιπες και q η κατανομή που προκύπτει από την έξοδο του δικτύου (η/οι νευρώνες/ες εξόδου πρέπει αν χρησιμοποιούν την κατάλληλη συνάρτηση ενεργοποίησης για την κατηγορία του προβλήματος). Παραλλαγές της εντροπίας εμφανίζονται ανάλογα τον αριθμό των κλάσεων του προβλήματος ή τη αρχιτεκτονική του δικτύου. Γνωστές μετρικές σφάλματος είναι η binary cross-entropy για ακριβώς δύο κλάσεις και για προβλήματα με περισσότερες από 2 κλάσεις χρησιμοποιούμε το τύπο σφάλματος categorical cross-entropy. Για να εφαρμοστεί αυτός, η έξοδος του δικτύου απαιτεί κωδικοποίηση και συνήθως γίνεται ένα διάνυσμα διαστάσεων m, όπου m είναι ο αριθμός των κλάσεων. Η τιμές αυτού του διανύσματος μπορεί να είναι είτε δυαδικές, με το 1 να συμβολίζει τον δείκτη της κλάσης που έχει κατηγοριοποιηθεί το παράδειγμα και 0 τους δείκτες των υπόλοιπων κλάσεων, είτε να ανήκουν [0,1] με συνολικό άθροισμα 1 και να παρουσιάζουν την κατανομή των πιθανοτήτων των κλάσεων που ανήκει το παράδειγμα.

2.4.5 Εξισώσεις Βελτιστοποίησης

Παραπάνω αναφέραμε την έννοια της ελαχιστοποίησης του σφάλματος του νευρωνικού δικτύου η οποία επιτυγχάνεται με την μέθοδο του gradient descent ο οποίος υπολογίζει την κλίση με την χρήση μερικών παργώνων της συνάρτησης του κόστους ως προς τα βάρη του δικτύου. Ο συγκεκριμένος αλγόριθμος που είναι και ο θεμελιώδης, τροποποιεί κατάλληλα τα βάρη του δικτύου με κατεύθυνση αντίθετη από αυτήν την συνάρτηση σφάλματος.

Έχουν αναφερθεί κάποια από τα προβλήματα του συγκεκριμένου βελτιστοποιητή παραπάνω, αλλά το μείζον πρόβλημα του είναι ότι συνήθως ένα νευρωνικό δίκτυο έχει πολλαπλά τοπικά βέλτιστα σημεία από τα οποία μόνο ένα είναι το ολικό. Αυτό καθιστά πολύ δύσκολη και σχεδόν αδύνατη την εύρεσή του σε λογικό χρόνο. Συνεπώς, για να αντιμετωπιστεί αυτό το πρόβλημα, προσπαθούμε να εστιάσουμε στην εύρεση τοπικών ελαχίστων με αποδεκτή τιμή σφάλματος. Με αυτόν τρόπο έχουμε καλύτερους χρόνους εκπαίδευσης και συνήθως ικανοποιητικές λύσεις που αγγίζουν και το βέλτιστο.

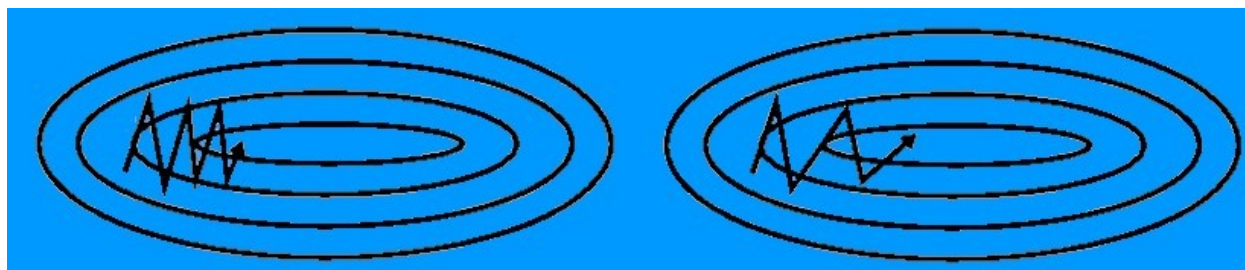
Ο αλγόριθμος gradient descent έχει δύο σημαντικές παραμέτρους οι οποίες είναι ο ρυθμός μάθησης d και η ορμή m αντίστοιχα. Η πρώτη παράμετρος εκφράζει το πόσο θα αλλάξει το βάρος κατά την εκπαίδευση, με μεγάλο ρυθμό μάθησης έχουμε μεγάλες αλλαγές βάρη ενώ με μικρό τα βάρη αλλάζουν ελάχιστα. Η τιμή του ρυθμού μάθησης είναι ένας πολύ σημαντικός παράγοντας και επηρεάζει την εκπαίδευση του δικτύου, καθώς ένας πολύ μικρός ρυθμός μπορεί να επιβραδύνει το δίκτυο αλλά επίσης μπορεί να εγγυηθεί καλύτερη σύγκλιση προς το ολικό ελάχιστο, ενώ η χρήση μεγάλου ρυθμού μάθησης επιταχύνει την εκπαίδευση του δικτύου αλλά επηρεάζει την απόδοση του, αφού το σημείο μπορεί να «περιστρέφεται» γύρω από ένα τοπικό ελάχιστο χωρίς να μπορεί να συγκλίνει σε αυτό κάνοντας το δίκτυο να έχει μεγαλύτερο σφάλμα.



Εικόνα 2.15 : Χαμηλή, υψηλή και κατάλληλη τιμή για την τιμή του ρυθμού μάθησης

Από την άλλη η ορμή είναι μία μέθοδος για να μπορέσουμε να κρατήσουμε σωστή πορεία όσον αφορά τις αλλαγές των βαρών του αλγορίθμου. Δηλαδή αν οι τιμές κινούνται προς κάποια κατεύθυνση η ορμή προσπαθεί να κρατήσει αυτήν την πορεία για κάποιες επόμενες τιμές οδηγώντας γρηγορότερα στο ελάχιστο. Η ορμή προστέθηκε αργότερα ως βοήθημα στην

αντικειμενική συνάρτηση αλλά πλέον είναι ένας από τους πιο χρήσιμους μηχανισμούς για την επιτυχία εύρεσης ελαχίστου στα νευρωνικά δίκτυα. Οι τιμές της ανήκουν στο διάστημα $(0,1)$ και όσο μεγαλύτερες είναι τόσο μειώνεται ο αριθμός των βημάτων για την σύγκλιση σε ολικό ελάχιστο και αυξάνεται η δυνατότητα υπέρβασης τοπικών ελαχίστων. Αν η τιμή της ορμής είναι μεγάλη προτείνεται ο ρυθμός μάθησης να παραμένει σχετικά μικρός, σε άλλη περίπτωση το ολικό μέγιστο μπορεί να ξεπεραστεί με ένα μεγάλο βήμα. Τέλος, συνήθως η καταλληλότερη τιμή για την ορμή πρέπει να βρεθεί μέσω δοκιμών στο αντίστοιχο σύνολο δεδομένων.



Εικόνα 2.16 : Αριστερά ο αλγόριθμος gradient descent χωρίς ορμή και δεξιά με ορμή

Πέρα από τον βασικό αλγόριθμο gradient descent αναπτύχθηκαν πολλές παραλλαγές βελτιστοποιητών κατά καιρούς με κάποιους να είναι από τους πλέον πιο δημοφιλείς και ισχυρούς τους οποίους θα δούμε παρακάτω.

1. Stochastic Gradient Descent

Η βασική διαφορά αυτού του αλγορίθμου από τον κλασσικό είναι ότι ανανεώνονται οι παραμέτροι για κάθε παράδειγμα εκπαίδευσης αντί για ένα πέρασμα ολόκληρου του συνόλου δεδομένων. Το καλό με αυτό είναι ότι οι μεταβολές των βαρών έχουν μεγάλη διακύμανση με αποτέλεσμα το ίδιο να συμβαίνει και για την συνάρτηση σφάλματος το οποίο βοηθάει στην ανακάλυψη νέας και πιθανώς καλύτερα τοπικά ελάχιστα ενώ από την άλλη ο βασικός αλγόριθμος συγκλίνει μόνο προς κάποιο ελάχιστο. Από την άλλη λόγω των συχνών ενημερώσεων των βαρών και των διακυμάνσεων περιπλέκει την σύγκλιση προς το ολικό ελάχιστο και συνήθως «εγκλωβίζεται» στα ελάχιστα των συχνών διακυμάνσεων το οποίο αποτελεί και το κύριο μειονέκτημα της μεθόδου. Μία παραλλαγή και ταυτόχρονα συνδυασμός του βασικού αλγορίθμου και του **Stochastic Gradient Descent** είναι ο **Mini Batch Gradient Descent** ο οποίος ενημερώνει τα βάρη μετά από έναν συγκεκριμένο αριθμό παραδειγμάτων n (Batch) με πάρα πολύ καλά αποτελέσματα και είναι ένας από τους δημοφιλείς αλγορίθμους εκπαίδευσης βαθιών νευρωνικών δικτύων.

2. AdaGrad

Η τεχνική AdaGrad προσαρμόζει τον ρυμό μάθησης σύμφωνα με τα αντίστοιχα βάρη κατά την διάρκεια της εκπαίδευσης του δικτύου. Για παράδειγμα βάρη τα οποία αλλάζουν πολύ αργά έχουν μεγαλύτερο ρυθμό μάθησης ενώ από την άλλη βάρη που αλλάζουν γρήγορα έχουν μικρό ρυθμό μάθησης. Το θετικό της μεθόδου αυτής είναι το γεγονός ότι δεν χρειάζεται να επέμβουμε στην διαχείριση του ρυθμού μάθησης οι ίδιοι ενώ το αρνητικό είναι ότι συνεχώς θα φθίνει ο ρυθμός μάθησης με αποτέλεσμα να εκπαιδευτεί πολύ αργά και εν τέλει να σταματήσει.

3. AdaDelta

Αποτελεί μία επέκταση του αλγορίθμου AdaGrad η οποία αντιμετωπίζει το πρόβλημα του φθίνοντος ρυθμού μάθησης μιας και ο αλγόριθμος δίνει βάση στις w τελευταίες τιμές των υπολογισμένων κλίσεων.

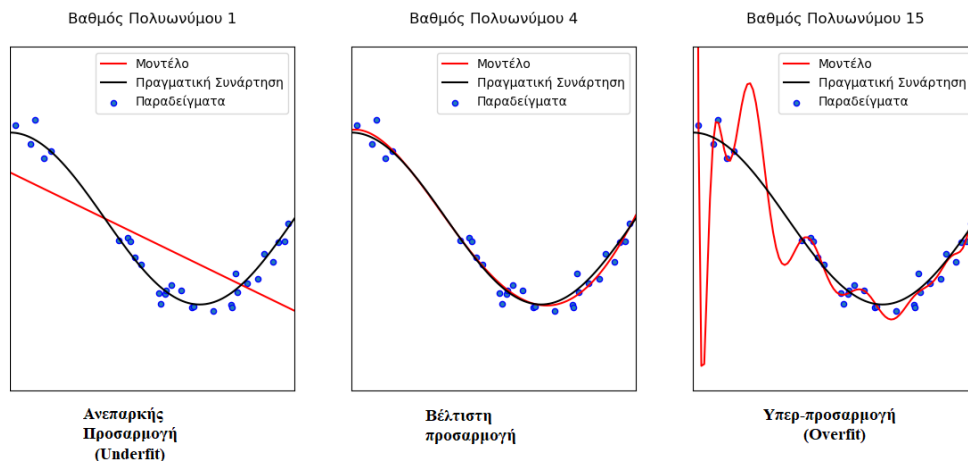
4. Adam (Adaptive Moment Estimator)

Τέλος, ίσως ο πιο διαδεδομένος αλγόριθμος βελτιστοποίησης βαθιών νευρωνικών δικτύων ο οποίος είναι ένας συνδυασμός του AdaDelta και της ορμής. Αφού εκτός από τον υπολογισμό των δυναμικών ρυθμών μάθησης για κάθε παράμετρο, υπολογίζει και τον φθίνοντα μέσο όρο των προηγούμενων κλίσεων υποδεικνύοντας μια γενική κατεύθυνση της παραμέτρου, κάτι παρόμοιο με την χρήση ορμής. Η τεχνική αυτή φαίνεται να λύνει τα περισσότερα αν όχι όλα προβλήματα της μάθησης βαθιών νευρωνικών δικτύων όπως την «εξαφάνιση» του ρυθμού μάθησης και την χαμηλή ή πολύ υψηλή διακύμανση στην ενημέρωση των παραμέτρων. Τέλος στην πράξη ο αλγόριθμος Adam ξεπερνά όλες τις προσαρμοστικές μεθόδους βελτιστοποίησης πόσο μάλλον τις πιο απλές [12].

2.5 Προβλήματα Κατά Της Διάρκειας Της Εκπαίδευσης

Κατά την διάρκεια εκπαίδευσης όλων των αλγορίθμων μηχανικής μάθησης στην επιβλεπόμενη μάθηση κυρίως, υπάρχει το φαινόμενο υπερπροσαρμογής στα δεδομένα ή στην διεθνή βιβλιογραφία γνωστό ως Overfitting. Με τον όρο αυτόν εννοούμε την υπερπροσαρμογή στα δεδομένα κάτι που καθιστά το μοντέλο μας πολύ καλό στο να προβλέπει τα δεδομένα τα οποία έχει εκπαιδευτεί αλλά σε νέα δεδομένα δεν προβλέπει σωστά οπότε το μοντέλο μας δεν έχει γενικευτεί σωστά για το συγκεκριμένο πρόβλημα. Επίσης, κάτι λιγότερο σύνηθες είναι ο όρος του Underfitting που ουσιαστικά είναι το αντίθετο του Overfitting. Ο αλγόριθμος δεν μπορεί να προσαρμοστεί σωστά στα δεδομένα εκπαίδευσης και συνεπώς οι προβλέψεις είναι κακές.

Για τον εντοπισμό του των παραπάνων προβλημάτων εισάγουμε τις έννοιες bias(προτίμηση) και variance(διακύμανση). Ο όρος bias μας δείχνει το επίπεδο των γενικεύσεων που έχει κάνει ο αλγόριθμος βασισμένος σε ένα σύνολο δεδομένων. Οι αλγόριθμοι που παρουσιάζουν υψηλό bias, έχουν καλή απόδοση όσον αφορά την ταχύτητα αλλά από την άλλη έχουν πολλά λάθη στα δεδομένα εκπαίδευσης με αποτέλεσμα οι προβλέψεις να είναι κακές. Ενώ ο όρος variance (διακύμανση) μας φανερώνει την απόκλιση των προβλέψεων ενός μοντέλου. Δηλαδή το πόσο το μοντέλο μας αλλάζει με την εφαρμογή νέων δεδομένων. Αν το μοντέλο μας έχει υψηλή διακύμανση τότε εξαρτάται πολύ από τα δεδομένα εκπαίδευσης, ενώ όταν έχει χαμηλή, αυτά, το επηρεάζουν ελάχιστα και συνεπώς έχουμε ένα γενικευμένο μοντέλο.



Εικόνα 2.17 : Γραφική αναπαράσταση του Underfitting/Overfitting

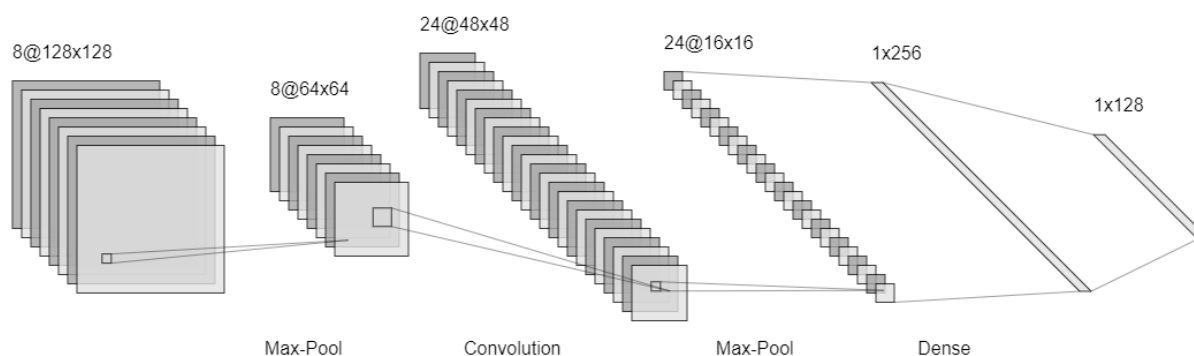
Στην εικόνα 2.17 διακρίνονται τα δύο προβλήματα σε σχέση με τους εισαγόμενους όρους bias/variance όπως αναφέρθηκαν και παραπάνω, ενώ στην μέση βλέπουμε το ιδανικό μοντέλο που συνδυάζει ακρίβεια και γρήγορη εκπαίδευση.

Αφού δημιουργήσουμε το μοντέλο μας και κάνουμε τις απαραίτητες ενέργειες για να εντοπίσουμε τέτοια προβλήματα πρέπει να τα αντιμετωπίσουμε ώστε να βελτιωθεί το τελικό μας μοντέλο. Στην περίπτωση του Underfitting που δεν είναι τόσο συνηθισμένη και είναι και πιο απλή η πιο σύνηθες λύση είναι να αυξάνουμε την πολυπλοκότητα του δικτύου με εισαγωγή μεταβλητών $2^{\text{ης}}$, $3^{\text{ης}}$ τάξης κ.ο.κ έτσι ώστε να προσθέσουμε την μη-γραμμικότητα στο μοντέλο μας. Μία ακόμη προσέγγιση είναι η προσθήκη δεδομένων εκπαίδευσης κάτι που συχνά δεν είναι δυνατό για ευνόητους λόγους όποτε τείνουμε προς την $1^{\text{η}}$ προσέγγιση.

Από την άλλη για καταπολέμηση του Overfitting υπάρχουν πληθώρα επιλογών μιας και είναι το κύριο πρόβλημα που απασχολεί τα τελευταία χρόνια τους ερευνητές. Αυτό μπορεί να επιτευχθεί με την εφαρμογή τεχνικών κανονικοποίησης των δεδομένων που μειώνουν διάφορες τιμές σε συγκεκριμένο διάστημα π.χ. $[0,1]$ με την ελάχιστη τιμή να παίρνει τιμή 0 και την μέγιστη το 1 και οι υπόλοιπες μεταβλητές κατανέμονται ενδιάμεσα ομοιόρφα. Μία άλλη τεχνική είναι η εφαρμογή του early stopping όταν εκπαιδευόμαστε ένα νευρωνικό δίκτυο επαναληπτικά, σταματάτε την εκπαίδευση αν δούμε ότι στα δεδομένα επαλήθευσης η κύρια μετρική μας (πχ. accuracy /ακρίβεια) δεν αυξάνεται καθόλου ή ακόμη χειρότερα μειώνεται ενώ στα δεδομένα εκπαίδευσης συνεχίζει την αυξητική πορεία οπότε διακόπτουμε την εκπαίδευση μετά από κάποιες επαναλήψεις του παραπάνω φαινομένου. Τέλος, μία επίσης πολύ σημαντική μέθοδος και παρούσα σχεδόν σε όλα τα μοντέλα των τελευταίων χρόνων είναι η εφαρμογή dropout όταν πρόκειται για νευρωνικά δίκτυα, όπου ανά εποχή εκπαίδευσης διαλέγουμε ένα τυχαίο ποσοστό νευρώνων να είναι «νεκροί» εξαναγκάζοντας το δίκτυο να μην επηρεάζεται από συγκεκριμένα παραδείγματα.

3 Συνελεκτικά Νευρωνικά Δίκτυα

Στην παρούσα εργασία, το κύριο θέμα είναι η εφαρμογή των βαθιών νευρωνικών δικτύων και πιο συγκεκριμένα των συνελεκτικών νευρωνικών δικτύων για την ανίχνευση καρκίνου του μαστού από εικόνες. Όπως αναφέρθηκε και στο Κεφάλαιο 1 τα συνελεκτικά νευρωνικά δίκτυα είναι η state of the art τεχνική για την επεξεργασία εικόνας στον τομέα της μηχανικής μάθησης και της τεχνητής νοημοσύνης. Παρακάτω θα δούμε αναλυτικά από τι αποτελούνται τα συνελεκτικά νευρωνικά δίκτυα και πώς μπορούμε να εξάγουμε πολύ εύκολα χαρακτηριστικά από τις εικόνες και για ποιόν λόγο τελικά είναι η καλύτερη τεχνική.



Εικόνα 3.18 : Μοντέλο συνελεκτικού νευρωνικού δικτύου

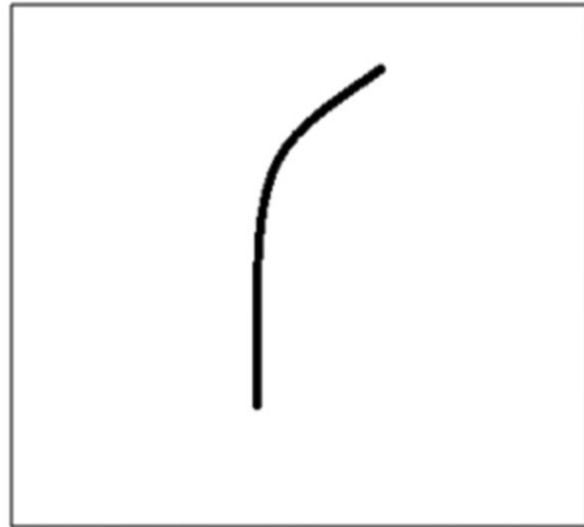
Στην Εικόνα 3.1 φαίνεται η δομή ενός συνελεκτικού νευρωνικού δικτύου. Αρχικά, η εικόνα στην είσοδο εισάγεται σαν πίνακας (συνήθως 3 πίνακες αφού έχουμε μοντέλο RGB χρωμάτων) στο συλνελεκτικό επίπεδο που θα εξηγήσουμε παρακάτω αναλυτικά τι είναι. Έπειτα εφαρμόζουμε την συνάρτηση ReLU στην οποία έχουμε αναφερθεί παραπάνω και τέλος εφαρμόζουμε pooling στο οποίο και πάλι θα αναφερθούμε παρακάτω. Μπορούμε να προσθέσουμε όσεςδήποτε τέτοια επίπεδα θέλουμε με στόχο να μειώσουμε την εικόνα στα επιθυμητά pixels, κρατώντας τα σημαντικά χαρακτηριστικά της εικόνας ενώ μειώνουμε κατά πολύ την εικόνα ώστε να μην χρειάζεται να επεξεργαστούμε τεράστιες εικόνες, το οποίο δεν το θέλουμε γιατί οδηγεί σε πολύ μεγάλους χρόνους εκπαίδευσης. Τέλος μετατρέπουμε τον δισδιάστατο πίνακα σε διάνυσμα για να το εισάγουμε σε κάποιον κατηγοριοποιητή και πιο συγκεκριμένα σε ένα νευρωνικό δίκτυο.

3.1 Συνελεκτικά Επίπεδα

Αρχικά τα συνελεκτικά επίπεδα είναι ένας μηχανισμός εξαγωγής χαρακτηριστικών από τις εικόνες με την εφαρμογή διάφορων φίλτρων ή αλλιώς πυρήνων που αναπαριστούν ένα συγκεκριμένο χαρακτηριστικό π.χ. μία καμπύλη ή ακμές. Στην Εικόνα 3.2 φαίνεται ένα φίλτρο για την συγκεκριμένη καμπύλη που αν εφαρμοστεί σε μία εικόνα μέσω του convolutional θα πάρουμε ως αποτέλεσμα τα σημεία που επικρατεί αυτό το χαρακτηριστικό.

0	0	0	0	0	50	0
0	0	0	0	50	0	0
0	0	0	50	0	0	0
0	0	0	50	0	0	0
0	0	0	50	0	0	0
0	0	0	50	0	0	0
0	0	0	0	0	0	0

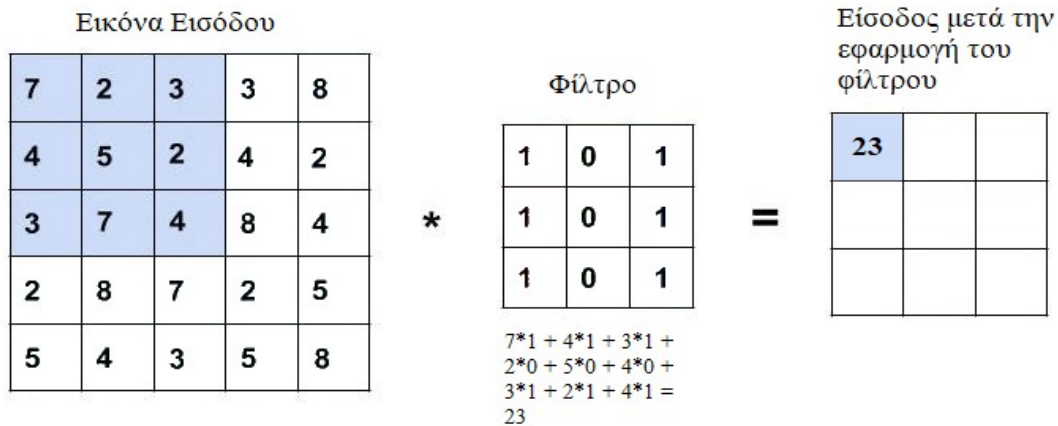
ΑΝΑΠΑΡΑΣΤΑΣΗ ΣΕ PIXEL ΤΟΥ ΦΙΛΤΡΟΥ



ΑΝΑΠΑΡΑΣΤΑΣΗ ΤΟΥ ΦΙΛΤΡΟΥ ΠΟΥ ΑΝΙΧΝΕΥΕΙ ΚΑΜΠΥΛΕΣ

Εικόνα 3.19 : Φίλτρο για ανίχνευση καμπύλης σε εικόνα

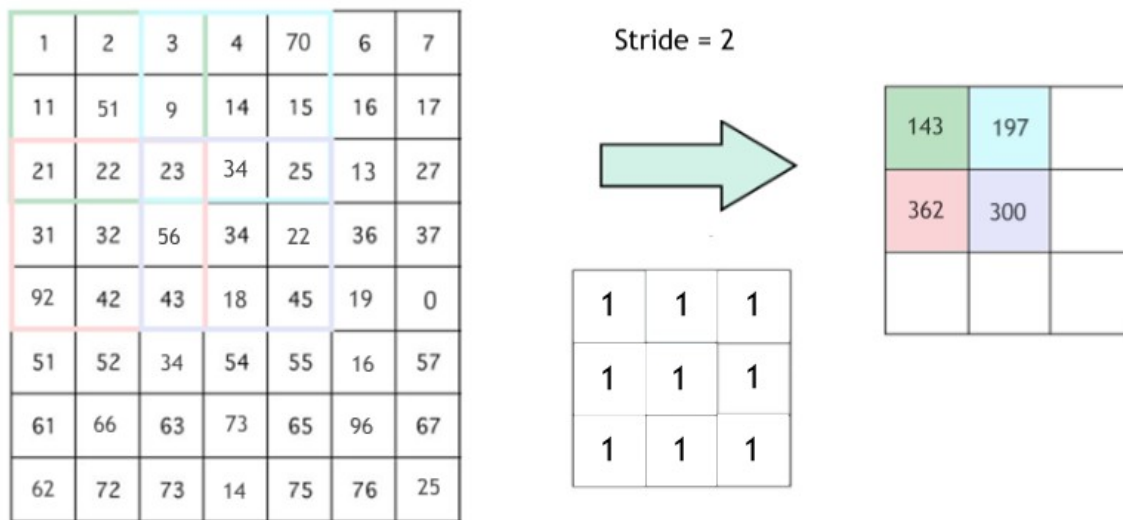
Τώρα όσον αφορά το convolutional είναι μία πράξη που γίνεται σε έναν συγκεκριμένο πίνακα(τον πίνακα της εικόνας) με χρήση άλλου πίνακα(τον πίνακα φίλτρου). Αυτό που γίνεται ουσιαστικά είναι να πολλαπλασιάζουμε την τιμή ενός κελιού της εικόνας με την αντίστοιχη του κελιού του φίλτρου. Αυτό το κάνουμε για όλα τα κελιά που αντιστοιχούν στο διάστημα του φίλτρου και έπειτα τα προσθέτουμε και αυτή είναι η τελική τιμή.



Εικόνα 3.20 : Εφαρμογή φίλτρου στην εικόνα

Στην Εικόνα 3.3 φαίνεται ακριβώς ποιά διαδικασία πραγματοποιείται ώστε να βγει το τελικό αποτέλεσμα. Αρχικά, το πρώτο κομμάτι 3x3 της εικόνας πολλαπλασιάζεται με το φίλτρο με τον εξής τρόπο που φαίνεται παραπάνω $7*1 + 4*1 + 3*1$ κοκ. Αμέσως μετά, εφαρμόζεται και η

συνάρτηση ReLU στην συγκεκριμένη έξοδο. Αυτή η διαδικασία επαναλαμβάνεται μέχρι να εφαρμόσουμε το φίλτρο σε όλη την εικόνα. Όσον αφορά τώρα το πώς κινούμαστε πάνω στην εικόνα, δηλαδή από το πρώτο 3x3 «κουτί» της εικόνας πώς πάμε στο επόμενο ή αλλιώς το πόσα pixels θα κινηθούμε υπάρχει το λεγόμενο Stride. Αν το θέσουμε 1 πάμε, κινούμαστε κατά μία στήλη δεξιά, αν το θέσουμε 2 πάμε 2 κοκ. Παρακάτω φαίνεται στην Εικόνα εφαρμογή κατά 1 stride που είναι και η πιο σύνηθες τιμή για το Stride.



Εικόνα 3.21 : εφαρμογή Stride=2 σε μία εικόνα

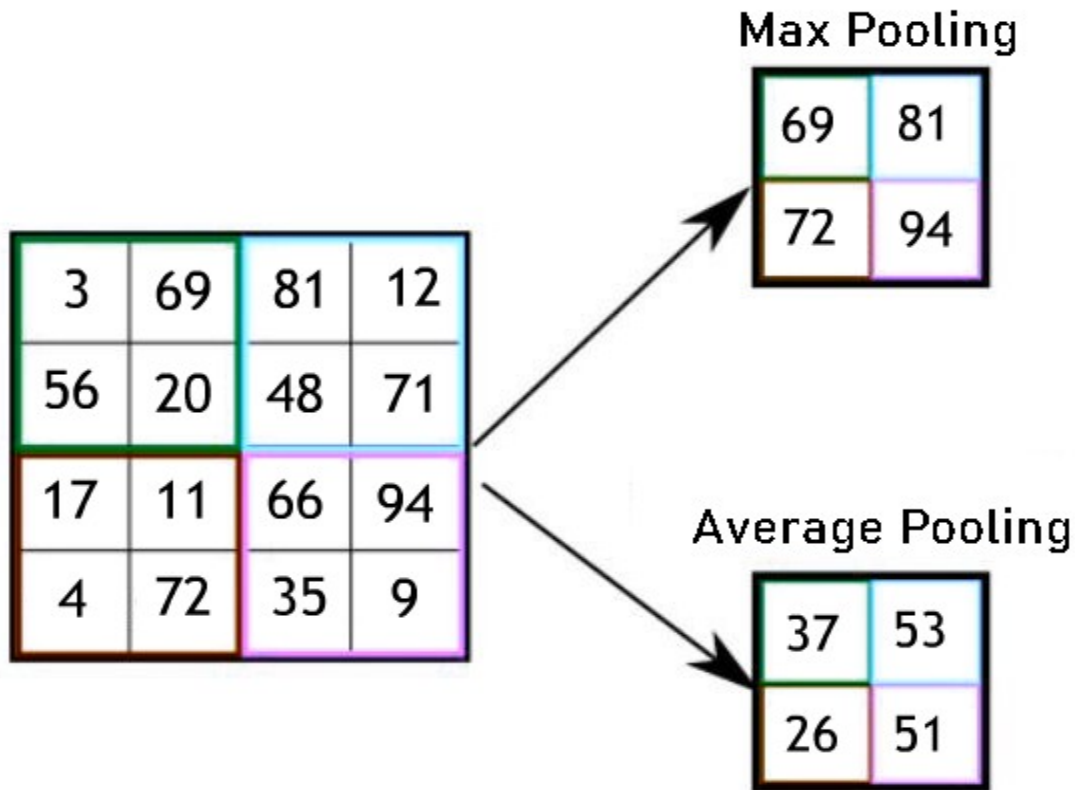
Τέλος, μπορούμε να εφαρμόσουμε zero padding στην αρχική εικόνα(πίνακα) και να προσθέσουμε γύρω γύρω μηδενικά όπως φαίνεται στην παρακάτω εικόνα. Ο σκοπός αυτού του μηχανισμού είναι να δώσει παραπάνω βάρος στα γύρω γύρω pixels και στις γωνίες.

0	0	0	0	0	0
0	9	18	32	11	0
0	57	13	8	26	0
0	41	6	33	75	0
0	62	54	47	58	0
0	0	0	0	0	0

Εικόνα 3.22: Εφαρμογή Zero Padding

3.2 Επίπεδα Pooling

Ο σκοπός των pooling επιπέδων είναι να εξάγει μία συγκεκριμένη τιμή από ένα σύνολο τιμών, συνήθως την μεγαλύτερη τιμή(max pooling) ή την μέση τιμή όλων των τιμών(average pooling). Αυτό έχει ως σκοπό την ελαχιστοποίηση της εικόνας εξόδου και συνεπώς πιο γρήγορους υπολογισμούς. Για παράδειγμα, αν πάρουμε την μεγαλύτερη τιμή από όλες, ουσιαστικά παίρνουμε την παρουσία του πιο δυνατού χαρακτηριστικού στο συγκεκριμένο κομμάτι. Έτσι καταφέρνουμε να διώξουμε την άχρηστη πληροφορία και ταυτόχρονα να κρατήσουμε την ακριβής θέση του χαρακτηριστικού. Συνήθως εισάγουμε ένα pooling layer αμέσως μετά την διαδικασία του συνελεκτικού επιπέδου και της ReLU και αυτό το κάνουμε κυρίως για να μειώσουμε την εικόνα και συνεπώς τις παραμέτρους του δικτύου για να έχουμε πιο γρήγορη εκπαίδευση τελικά. Όπως φαίνεται από την Εικόνα 3.6 έχουμε και εδώ χρήση Stride για το πώς θα μετακινηθούμε πάνω στην εικόνα και εδώ είναι ίσο με 2 και φαίνεται τι αποτελέσματα θα έχουμε με την χρήση των 2 τεχνικών πάνω στον συγκεκριμένο πίνακα.



Εικόνα 3.23 : Παράδειγμα max/average pooling

Οι παραπάνω τεχνικές όλες μαζί αποτελούν ένα συνελεκτικό block, δηλαδή το συνελεκτικό επίπεδο, αμέσως μετά η συνάρτηση ReLU(ή οποιαδήποτε άλλη) και τέλος κάποια τεχνική pooling με αυτή που επικρατεί να είναι η max pooling τεχνική. Γενικότερα, βάζοντας πολλά τέτοια επίπεδα έχουμε μία πλήρης αρχιτεκτονική συνελεκτικού νευρωνικού δικτύου που εκτελεί εξαγωγή χαρακτηριστικών σε διάφορα επίπεδα, και συνεπώς μείωση της αρχικής είσοδου (εικόνας) στο επιθυμητό μέγεθος. Τέλος, αφού πραγματοποιηθεί η διαδικασία αυτή όσες φορές χρειαζόμαστε, μετατρέπουμε τον τελικό πίνακα σε διάνυσμα (Flatten) για να τον δώουμε ως είσοδο σε ένα πλήρως συνδεδεμένο νευρωνικό δίκτυο που εν τέλει αυτό θα κάνει την κατηγοριοποίηση της συγκεκριμένης εικόνας με την ακριβώς ίδια διαδικασία που είδαμε στο κεφάλαιο 2.

4 Δεδομένα και Μεθοδολογία

4.1 Εισαγωγή

Στα προβλήματα μηχανικής μάθησης το σημαντικότερο στοιχείο για την αποτελεσματικότητα του τελικού μοντέλου είναι τα δεδομένα. Τα στοιχεία που εκπαιδεύεται κάποιος αλγόριθμος ώστε να διαμορφώσουμε το τελικό μοντέλο είναι αυτά που μας δίνουν και την καθοδήγηση για την χρήση διάφορων τεχνικών ώστε να καταλήξουμε σε ένα αποδοτικό και αποδεκτό τελικά μοντέλο όπου θα κάνει την τελική πρόβλεψη σε νέα δεδομένα.

Συνεπώς, και στο πρόβλημα ανίχνευσης καρκίνου του μαστού πρέπει να έχουμε ένα μεγάλο σύνολο δεδομένων ώστε να μπορέσουμε να κάνουμε ένα καλό μοντέλο που θα αποδίδει ικανοποιητικές προβλέψεις. Το πρόβλημα που αντιμετωπίζουμε λοιπόν είναι η έλλειψη τέτοιων δεδομένων μιας και τα ιατρικά δεδομένα είναι απόρρητα και προσωπικά δεδομένα. Ωστόσο, με πλήρης ανωνυμία μπορείς να τα δημοσιεύσεις για έρευνες σαν και αυτή αλλά και πάλι υπάρχει το πρόβλημα ότι δεν υπάρχουν αρκετά δεδομένα σε αυτόν τον τομέα, δηλαδή στον τομέα της υγείας. Επίσης, ένα ακόμα σοβαρό πρόβλημα που αντιμετωπίζουμε είναι η ακεραιότητα των δεδομένων. Ο κλάδος της ιατρικής και συγκεκριμένα η αναγνώριση καρκίνου είναι δύσκολη διαδικασία ακόμα και για έναν γιατρό. Με αυτό εννοώ ότι μπορεί να έχουμε αρκετά δεδομένα στα οποία έχει γίνει λάθος διάγνωση και δεν μπορούμε να κάνουμε κάτι για αυτό.

Το σύνολο δεδομένων που απασχολεί την παρούσα πτυχιακή εργασία υπάρχει διαθέσιμο στο Kaggle και έχει το όνομα Breast Histopathology Images. Αποτελείται από ανώνυμα «κομμάτια» εικόνων μεγέθους 50x50 pixels που έχουν παρθεί από το αρχικό σύνολο δεδομένων 162 εικόνων σκαναρισμένων από μικροσκόπιο 40x. Συνολικά το σύνολο παραδειγμάτων, δηλαδή τα «κομμάτια» που πάρθηκαν είναι σε σύνολο 277.524 εκ των οποίων τα 78.786 από αυτά είναι καρκινικά, δηλαδή περίπου το 29% των δεδομένων. Αυτό καθιστά το σύνολο δεδομένων μας ανισόροπο κάτι που συνηθίζεται όταν πρόκειται για δεδομένα που αφορούν την ιατρική και πιο συγκεκριμένα τον καρκίνο. Ακόμη, στα «κομμάτια» αυτά σαν όνομα κάθε κομματιού αναγράφεται το ID του ασθενή, οι συντεταγμένες του συγκεκριμένου 50x50 κομματιού και τέλος το αν είναι καρκινικό ή όχι το οποίο αναπαριστάται με 1 και 0 αντίστοιχα. Τέλος, παρατηρείται ότι τα «κομμάτια» των εικόνων ανά ασθενή έχουν διαφορά στην κατανομή τους, δηλαδή κάποιοι έχουν περισσότερα «κομμάτια» από άλλους ασθενείς, που μας οδηγεί στο ερώτημα αν όλες οι εικόνες έχουν την ίδια ανάλυση από τους ιστούς των κυττάρων ή αυτό ποικίλει ανάμεσα στους ασθενείς.

Πέρα από το Kaggle όπου μπορείς να βρεις διάφορα σύνολα δεδομένων για τέτοιου είδους έρευνες, υπάρχουν και άλλοι οργανισμοί που παρέχουν δεδομένα τέτοιου είδους όπως το National Cancer Institute και πιο συγκεκριμένα στην ιστοσελίδα The Cancer Imaging Archive η οποία αποτελεί μία μεγάλη βάση δεδομένων με εικόνες ιατρικής για έρευνα καρκίνου. Ο σκοπός κυρίως της συγκεκριμένης βάσης δεδομένων είναι να βοηθήσει στην ανάπτυξη μεθόδων στο να διαγνώσκουμε μέσω υπολογιστών.

4.2 Μεθοδολογία

Για την παρούσα πτυχιακή εργασία έγινε χρήση διάφορων τεχνικών και εργαλείων όπου θα αναλυθούν παρακάτω. Σε αυτά περιλαμβάνονται η προγραμματιστική γλώσσα που χρησιμοποιήθηκε, το περιβάλλον, οι βιβλιοθήκες που χρησιμοποιήθηκαν για όλη την διαδικασία ανάπτυξης μιας λύσης, από την εισαγωγή, την οπτικοποίηση, των χειρισμό των δεδομένων μέχρι την ανάπτυξη των δικτύων και την αξιολόγηση τους. Επίσης, οι παραμέτροι στις αρχιτεκτονικές των δικτύων που έπρεπε να υπολογιστούν και έπειτα να τροποποιηθούν για την βελτίωση της απόδοσης του δικτύου. Ενδεικτικά κάποιες παραμέτροι είναι η εξίσωση κόστους, τα επίπεδα ενός δικτύου, οι συναρτήσεις ενεργοποίησης κ.α. Τέλος, θα αναλυθούν οι τρόποι αξιολόγησης της απόδοσης σε προβλήματα κατηγοριοποίησης αλλά και μετρικές που εν τέλει χρησιμοποιήθηκαν ως κύριες βάσει την φύση του συγκεκριμένου συνόλου δεδομένων.

4.3 Γλώσσα Προγραμματισμού

Η γλώσσα προγραμματισμού που επιλέχτηκε για την εκτέλεση των πειραμάτων είναι η Python. Η Python είναι μία γλώσσα υψηλού επιπέδου, γενικής χρήσης, αντικειμενοστραφής και δομημένη που είναι πολύ φιλική προς τον χρήστη με όσο πιο κατανοητή σύνταξη γίνεται. Πρωτοεμφανίστηκε το 1990 από τον Guido van Rossum στο εθνικό κέντρο ερευνών για μαθηματικά και πληροφορική στην Ολλανδία. Η πρώτη έκδοση που είχε καινούρια χαρακτηριστικά ήταν η Python 2 που εμφανίστηκε το 2000 ενώ η επόμενη έκδοση ήταν η Python 3 με ριζικές αλλαγές και πολλά πλεονεκτήματα οι οποίες αλλάξαν την γλώσσα ριζικά ενώ μετά τον Ιανουάριο του 2020 σταμάτησε και η υποστήριξη της από τους δημιουργούς [13]. Η Python τα τελευταία χρόνια έχει τρομερή ανταπόκριση από τον κόσμο, κάτι το οποίο την καθιστά μία από τις πιο χρησιμοποιούμενες γλώσσες τα τελευταία χρόνια [14]. Τέλος, η Python προσφέρει πληθώρα από βιβλιοθήκες μηχανικής μάθησης, βαθιάς μάθησης και γενικότερα για ανάλυση δεδομένων κάτι που την καθιστά πολύ δημοφιλής στον χώρο.

4.4 Περιβάλλον Ανάπτυξης Κώδικα

Για την ανάπτυξη τέτοιων προγραμμάτων που αφορούν χρήση τεχνικών βαθιάς μάθησης χρειάζεται αυξημένη υπολογιστική ισχύ που έρχεται με την χρήση κάρτας γραφικών, μεγάλη χωρητικότητα και τέλος ευκολία στην διαδικασία εξόδου και εισόδου των δεδομένων μας. Ένας ατομικός υπολογιστής προϋποθέτει πολλά χρήματα ενώ πολλές απομακρυσμένες υπηρεσίες που προσφέρουν τέτοια εργαλεία κοστίζουν. Από την άλλη δύο από τις πιο διαδεδομένες πλατφόρμες που παρέχουν τέτοιες δυνατότητες με μηδενικό κόστος είναι το Google Colaboratory και τα shells scripts του Kaggle.

Στην παρούσα πτυχιακή χρησιμοποιήθηκε αποκλειστικά η χρήση του Google Colaboratory λόγω του όγκου των δεδομένων μας. Η πλατφόρμα του Google Colaboratory προσφέρει την δυνατότητα δημιουργίας notebook και την χρήση αυτών μέσα στο Google Colaboratory με την

χρήση του Jupyter Notebook. Το Jupyter Notebook είναι μία πλατφόρμα που επιτρέπει στον χρήστη να γράφει κομμάτια σε blocks που μπορούν να εκτελεστούν ανεξάρτητα ενώ δίνει και την δυνατότητα να το εκτελέσεις και ακουλουθικά. Ένα μεγάλο πλεονέκτημα του Jupyter Notebook είναι ότι μπορείς να εκτελέσεις τα blocks ανεξάρτητα και να πάρεις είτε διαγράμματα είτε γενικότερα αποτελέσματα διαδραστικά. Το Google Colaboratory υποστηρίζει μόνο την γλώσσα Python ενώ το Kaggle υποστηρίζει Python και R, κάτι που καθιστά και τις δύο πλατφόρμες ιδανικές για εκτέλεση αλγορίθμων μηχανικής μάθησης και γενικότερα για ανάλυση δεδομένων.

Για την εισαγωγή των δεδομένων μπορεί να γίνει είτε χρήση του Google Drive και έπειτα η μεταφόρτωση τους στην πλατφόρμα είτε να τα «κατεβάζεις» τοπικά στον απομακρυσμένο υπολογιστή. Κάθε script έχει την δυνατότητα να τρέχει συνεχόμενα για 12 ώρες σε ένα συγκεκριμένο μηχάνημα, ο αποθηκευτικός χώρος είναι ίσος με 69 Gb, προσωρινή μνήμη (RAM) η οποία διαφέρει (περίπου 12Gb) ώστε να προσαρμόζει κατάλληλα με βάσει τους χρήστες στο Google Colaboratory και τέλος παρέχει τον επεξεργαστή Intel (R) Xeon (R) CPU 2.00GHz. Τέλος το πιο σημαντικό είναι η παροχή κάρτας γραφικών χωρητικότητας 12 Gb για την δυνατότητα εκπαίδευσης μεγάλων νευρωνικών δικτύων και γενικότερα για εκτέλεση αλγορίθμων μηχανικής μάθησης που χρειάζονται πολύ υπολογιστική ισχύ [15]. Ένα πολύ σημαντικό στην χρήση της πλατφόρμας Google Colaboratory είναι ότι δεν υπάρχει κάποιος περιορισμός στην χρήση των καρτών γραφικών πέρα από την 12ωρη συνεχόμενη χρήση. Συνεπώς, η πλατφόρμα είναι ικανοποιητική όσον αφορά την ταχύτητα και τους πόρους που παρέχει για την εκτέλεση πειραμάτων για την παρούσα πτυχιακή. Ωστόσο, μπορεί να χρησιμοποιηθεί και για άλλους σκοπούς όπως να διευρύνεις τις ικανότητες στις δεξιότητες στον τομέα της ανάλυσης δεδομένων αφού παρέχει τους πόρους.

4.5 Βιβλιοθήκες

Όπως αναφέρθηκε παραπάνω, η Python φημίζεται για την ευκολία στην εκμάθηση και για την φιλικότητα προς τον χρήστη, ωστόσο το πιο δυνατό της σημείο είναι η ποικιλία βιβλιοθηκών που υποστηρίζει. Πολλές βιβλιοθήκες από αυτές χρησιμοποιήθηκαν στα πειράματα και θα αναλυθούν παρακάτω.

- **TensorFlow**

Αρχικά, η βιβλιοθήκη Tensorflow αποτελεί μία πλατφόρμα για την δημιουργία προγραμμάτων μηχανικής μάθησης και βαθιών νευρωνικών δικτύων και η ανάπτυξή της ξεκίνησε από την ομάδα Google Brain με σκοπό την έρευνα πάνω στην μηχανική μάθηση και στα βαθιά νευρωνικά δίκτυα. Παρέχει API για τις γλώσσες C++ και Python, ενώ μπορεί να συμβάλλει ο κάθε προγραμματιστής αφού είναι λογισμικό ανοικτού κώδικα.

Οι υπολογισμοί στην βιβλιοθήκη γίνονται με την χρήση ροών δεδομένων όπου αποτυπώνει την επεξεργασία των δεδομένων με συνδεδεμένους γράφους που ο καθένας αποτελεί κάποιον υπολογισμό. Μπορούμε να αναπαραστήσουμε γραφικά αυτήν την επεξεργασία χρησιμοποιώντας το TensorBoard όπου εμφανίζει τις ροές δεδομένων σαν ευθείες και τα

σημεία που υπάρχει επεξεργασία σαν κόμβους. Μέσω αυτού μπορούμε να κατανοήσουμε καλύτερα την διαδικασία της εκπαίδευσης και συνεπώς να τροποποιηθεί καταλλήλως. Τέλος, μέσω της πλατφόρμας CUDA (Compute Unified Device Architecture) δίνεται η δυνατότητα στην βιβλιοθήκη να κάνει παράλληλη επεξεργασία των ροών με την χρήση υπολογιστικής ισχύς από κάρτες γραφικών ενώ δίνεται και η δυνατότητα χρήσης TPU's (Tensor Processing Unit). Αυτό επηρεάζει πολύ θετικά τον χρόνο εκπαίδευσης των μοντέλων μας [16].

- **Keras**

Η βιβλιοθήκη Keras αποτελεί ένα υψηλού επιπέδου API που αναπτύχθηκε για εφαρμογές νευρωνικών δικτύων. Είναι γραμμένο σε Python και μπορεί να χρησιμοποιηθεί μαζί με το TensorFlow, CNTK ή το Theano. Πλέον, το TensorFlow έχει εντάξει το Keras και τις λειτουργίες του μέσα στην ίδια την βιβλιοθήκη και μπορεί να χρησιμοποιηθεί από εκεί. Οι λειτουργίες του Keras είναι πολλές και ποικίλουν από την φόρτωση δεδομένων στο σύστημα μέχρι και τους αλγορίθμους που χρησιμοποιούνται για την εκτέλεση πειραμάτων, χρησιμοποιώντας συνελεκτικά νευρωνικά δίκτυα, αναδρομικά δίκτυα και προφανώς νευρωνικά δίκτυα. Όλα αυτά μπορούν να εκτελεστούν είτε με την χρήση CPU, GPU αλλά και TPU's μιας και συνεργάζεται με το Tensorflow το οποίο δίνει αυτήν την δυνατότητα.

Το Keras έχει σαν προτεραιότητα την ευκολία του χρήστη και να έχει αρθρωτή μορφή, δηλαδή κάθε λειτουργία να μπορεί να λειτουργήσει ανεξάρτητα ή και σε συνδυασμό με άλλες, να είναι εύκολα επεκτάσιμο και τέλος πλήρως συμβατό με την Python.

Η δομική μονάδα που χρησιμοποιείται είναι το `model`, το οποίο συγκροτείται από `layers` και αποτελεί έναν τρόπο οργάνωσης των `layers`. Μπορούμε να παραμετροποιήσουμε κάθε `layer` ξεχωριστά ανάλογα με τον τύπο του, την εξίσωση ενεργοποίησης, διαστάσεις, συνάρτησεις ενεργοποιήσεων και λοιπές παραμέτρους. Τέλος, επιλέγεται ο αλγόριθμος εκπαίδευσης του μοντέλου καθώς επίσης και η μορφή της πρόβλεψης [17].

- **Matplotlib**

Η βιβλιοθήκη Matplotlib δημιουργήθηκε από τον John Hunter και έχει ως κύριο σκοπό την οπτικοποίηση δεδομένων. Περιέχει πληθώρα γραφημάτων όπως ιστογράμματα, θηκογράμματα, διαγράμματα διασποράς και γενικότερα οτιδήποτε γραφήματα άφορούν την οπτικοποίηση δεδομένων για εξαγωγή συμπερασμάτων. Η πρόσβαση σε αυτά είναι εύκολη και ο κώδικας λίγος το οποίο καθιστά την βιβλιοθήκη πολύ φιλική προς τον χρήστη [18]. Στην συγκεκριμένη πτυχιακή χρησιμοποιήθηκε εκτεταμένα για την καλύτερη κατανόηση των δεδομένων με την χρήση διάφορων διαγραμμάτων.

- **Scikit-Learn**

Αποτελεί ίσως την ισχυρότερη βιβλιοθήκη όσων αφορά τους αλγορίθμους μηχανικής μάθησης. Περιέχει τεράστια ποικιλία αλγορίθμων κατηγοριοποίησης, όπως Support Vector Machines(SVM), logistic regression, random forest, decisions trees και άλλους, αλγόριθμους παλινδρόμησης όπως linear regression, lasso και άλλους ενώ και αλγορίθμους συσταδοποίησης όπως k-mean, DBSCAN και άλλους. Πέραν όμως από τον σύνολο αλγορίθμων που παρέχει, δίνει την δυνατότητα στον χρήστη να κανονικοποιήσει τα δεδομένα με αλγορίθμους όπως ο PCA,

LDA. Επίσης, μπορείς να διαχωρίσεις τα δεδομένα σου σε σύνολο ελέγχου, επαλήθευσης και ελέγχου με κάποιο ποσοστό, ενώ ακόμη μπορείς να χρησιμοποιήσεις και την μέθοδο κ-αναδιυλώσεων με επανατοποθέτηση. Τέλος μπορείς χρησιμοποιώντας την βιβλιοθήκη για να αξιολογήσεις τις επιδόσεις των μοντέλων με τις μετρικές που παρέχονται. Στην συγκεκριμένη πτυχιακή χρησιμοποιήθηκε κυρίως για την αξιολόγηση των μοντέλων με διάφορες μετρικές [19].

- **Seaborn**

Είναι μία βιβλιοθήκη βασισμένη πάνω στην matplotlib. Σκοπός είναι να προσφέρει μία υψηλού επιπέδου διεπαφή για την δημιουργία «ελκυστικών» και χρήσιμων στατιστικών γραφημάτων. Παρέχει περισσότερα γραφήματα και πιο προσγμένα όσον αφορά την οπτικοποίηση των δεδομένων, ενώ τυπικά θεωρείται πιο εύκολη στην χρήση της [20].

- **Tensorflow-Addons**

Η βιβλιοθήκη Tensorflow-Addons είναι ουσιαστικά μία «βοηθητική» βιβλιοθήκη σε επέκταση της αρχικής η οποία είναι ανοιχτή στο να συμβάλλει οποιοσδήποτε θέλει. Γενικότερα, ενώ η Tensorflow παρέχει όλα τα απαραίτητα όπως επίπεδα, βελτιστοποιητές, μετρικές και άλλα. Σε έναν κλαδό όπως αυτόν της μηχανικής μάθησης που αναπτύσσεται ραγδαία τα τελευταία χρόνια προκύπτουν συνεχώς νέες εφευρέσεις που δεν είναι δυνατόν να τις περνάμε απευθείας στην βασική βιβλιοθήκη οπότε έχουμε αυτό το Addons για τον σκοπό αυτό της δοκιμής. Στην συγκεκριμένη εργασία χρησιμοποιήθηκε για την χρήση της κύριας μετρικής F1_Score που θα περιγραφεί παρακάτω μιας και το Tensorflow αρχικά υπήρχε αλλά υπολογιζόταν σε κάθε batch ξεχωριστά, κάτι το οποίο δεν είναι σωστό για αυτήν την μετρική οπότε προστέθηκε η συγκεκριμένη λειτουργία στα Addons που υπολογίζει βάσει της εποχής.

- **Split Folders**

Η βιβλιοθήκη Split Folders είναι μία βιβλιοθήκη για τον διαχωρισμό των δεδομένων σε δεδομένα εκπαίδευσης, επαλήθευσης και ελέγχου και τα τοποθετεί αντίστοιχα σε φακέλους σύμφωνα με την τιμή κλάση τους και τι δεδομένα είναι [21]. Στην συγκεκριμένη εργασία χρησιμοποιήθηκε για την διαχώριση των δεδομένων σε φακέλους ώστε μετέπειτα να τροφοδοτηθούν στο πρόγραμμα μέσω της βιβλιοθήκης Keras με την μέθοδο `flow_from_directory` η οποία φορτώνει τα δεδομένα από φάκελο.

- **tqdm**

Η Tqdm είναι μία βιβλιοθήκη που δημιουργεί μία μπάρα προόδου για κάποια εργασία κατά την εκτέλεσή της. Έχει χαμηλό overhead και μέσω αλγορίθμου υπολογίζει τον χρόνο που απομένει [22]. Χρησιμοποιήθηκε εκτεταμένα στην εργασία για την εμφάνιση των χρόνων των εργασιών.

4.6 Παράμετροι

Στην παρακάτω ενότητα, παρουσιάζεται ο τρόπος επιλογής όλων των παραμέτρων που αφορούν τα συνελεκτικά νευρωνικά δίκτυα που δημιουργήθηκαν. Η διαδικασία επιλογής όλων των

παραμέτρων των μεμονομένων μοντέλων δεν ακολουθεί κανόνες και είναι κατά βάση εμπειρική, δηλαδή μέσω δοκιμής και σύγκρισης αποτελεσμάτων πρέπει να καταφέρουμε να φτάσουμε σε έναν συνδυασμό των παραμέτρων που ικανοποιούν τους σκοπούς μας. Κανείς δεν εγγυάται όμως ότι οι συνδυασμοί μας θα είναι και οι βέλτιστοι.

Αρχιτεκτονική Μοντέλων

Η προσέγγιση που ακολουθήθηκε στην παρούσα πτυχιακή εργασία, αφορούσε αποκλειστικά την χρήση συνελεκτικών νευρωνικών δικτύων που είναι και η καλύτερη μέθοδος όσον αφορά την επεξεργασία εικόνας.

Η πρόκληση που έχουμε να αντιμετωπίσουμε όταν χρειαστεί να δημιουργήσουμε συνελεκτικά νευρωνικά δίκτυα είναι το μεγάλο πλήθος παραμέτρων και τεχνικών που έχουμε στην διάθεση μας ώστε να καταλήξουμε σε κάποια αρχιτεκτονική. Αρχικά πρέπει να επιλεχτεί πόσα επίπεδα συνελεκτικά θα έχουμε και έπειτα το μέγεθος των φίλτρων (3x3, 5x5, κ.ο.κ) και το πόσα φίλτρα θα έχουμε σε κάθε ένα ξεχωριστά. Επίσης, πρέπει να αποφασίσουμε αν θα χρησιμοποιήσουμε padding ή όχι σε κάθε επίπεδο (συνήθως επιλέγεται same padding που με αυτό εννοούμε την διατήρηση των διαστάσεων της αρχικής εικόνας μετά την εφαρμογή του φίλτρου.) Έπειτα, έχουμε όλες τις παραμέτρους των νευρωνικών δικτύων, δηλαδή το πόσα επίπεδα και με πόσους νευρώνες θα είναι το καθένα. Οπότε κάνοντας χρήση διάφορων μετρικών σφάλματος και προσπαθώντας να αποφύγουμε το φαινόμενο της υπεπροσαρμογής προσπαθούμε να καταλήξουμε στα καλύτερα αποτελέσματα.

Ακολουθώντας αυτήν την διαδικασία και μετά από πολλά πειράματα, καταλήξαμε στην χρήση τριών διαφορετικών αρχιτεκτονικών συνελεκτικών νευρωνικών δικτύων.

Το πρώτο μοντέλο αποτελείται αρχικά, από 4 συνελεκτικά επίπεδα με 32,64,128,256 φίλτρα αντίστοιχα, μεγέθους 3x3, ακολουθούμενα από max pooling επίπεδο για όλα εκτός του τελευταίου, για την σμίκρυνση της εικόνας κρατώντας τα σημαντικά χαρακτηριστικά. Έπειτα έχουμε ένα δίκτυο τύπου feedforward με 3 επίπεδα που έχουν 128,64,1 νευρώνες αντίστοιχα με μετάδοση σφάλματος και οπισθοδιάδοση, πλήρως συνδεδεμένα μεταξύ τους και ενδιάμεσα στους 128 και 64 νευρώνες επίπεδο 0.2 dropout, ενώ μετά υπήρχε και Batch Normalization αμέσως μετά τα συνελεκτικά επίπεδα και τα feedforward.

Το δεύτερο μοντέλο αποτελείται και αυτό από τα ίδια 4 συνελεκτικά επίπεδα. Ωστόσο, εκεί που αλλάζει είναι στο πλήρως συνδεδεμένο νευρωνικό δίκτυο αμέσως μετά τα συνελεκτικά επίπεδα. Στο δίκτυο τύπου feedforward έχουμε 4 επίπεδα με διαδοχικά 128,256,512,1 νευρώνες αντίστοιχα και ενδιάμεσα dropout 0.2 ενώ και εδώ έχουμε Batch Normalization ενδιάμεσα.

Τέλος, και το τρίτο μοντέλο έχει τα ίδια ακριβώς 4 συνελεκτικά επίπεδα και πάλι αλλάζει στο πλήρως συνδεδεμένο δίκτυο. Αποτελείται από 4 επίπεδα με 256,512,1024,1 νευρώνες και πάλι ενδιάμεσα dropout με ποσοστό 0.15.

	Αρχιτεκτονική 1	Αρχιτεκτονική 2	Αρχιτεκτονική 3
Συνελεκτικά Επίπεδα	4	4	4
Διάσταση Φίλτρων	3 x 3	3x3	3x3
Φίλτρα	32,64,128,256	32,64,128,256	32,64,128,256
Padding	Same	Same	Same
Max Pooling	NAI	NAI	NAI
Επίπεδα	3	4	4
Νευρώνες Ανά Επίπεδο	128-64-1	128-256-512-1	256-512-1024-1
Dropout	NAI	NAI	NAI
Batch Normalization	NAI	NAI	NAI
Ποσοστό Dropout	0.2-0	0.2-0.2-0	0.2-0.2-0
Feed-Forward	NAI	NAI	NAI
Back-Propagation	NAI	NAI	NAI
Εξίσωση Βελτιστοποίησης	Adam	Adam	SGD
Εξίσωση Ενεργοποίησης	ReLU	ReLU	ReLU
Εξίσωση Σφάλματος	B_Crossentropy	B_Crossentropy	B_Crossentropy
Batch Size	64	64	64
Εποχές Εκπαίδευσης	18	7	39

Πίνακας 4.1 : Συγκριτικός πίνακας τελικών αρχιτεκτονικών

Παραπάνω στον Πίνακα φαίνονται περιεκτικά οι 3 τελικές αρχιτεκτονικές των συνελεκτικών νευρωνικών δικτύων, και έπειτα του νευρωνικού δικτύου που χρησιμοποιήθηκαν στην παρούσα πτυχιακή εργασία.

Συναρτήσεις ενεργοποίησης

Η επιλογή της συνάρτησης ενεργοποίησης είναι ίσως ένα από τα πιο σημαντικά κομμάτια του δικτύου καθώς είναι υπεύθυνο για την μετατροπή της εξόδου κάθε νευρώνα, ο οποίος μεταφέρετε στο δίκτυο. Ανάλογα το τι θέλουμε να κάνει το δίκτυο και την μορφή των δεδομένων επιλέγουμε την κατάλληλη συνάρτηση για τα κρυφά επίπεδα και το επίπεδο εξόδου, που συνήθως είναι διαφορετικές.

Για τις αρχιτεκτονικές της εργασίας επιλέξαμε την χρήση, για όλα τα κρυφά επίπεδα των νευρώνων, την εξίσωση ReLU που έχει περιγραφεί παραπάνω ενώ δοκιμάστηκε και η Leaky ReLU με διάφορες τιμές του α . Η ReLU επιταχύνει σημαντικά την διαδικασία μάθησης στα βαθιά νευρωνικά δίκτυα κυρίως, χωρίς να επηρεάζεται η απόδοση. Τέλος, για το επίπεδο εξόδου, χρησιμοποιήθηκε η σιγμοειδής συνάρτησης αφού έχουμε πρόβλημα δυαδικής κατηγοριοποίησης και μας ενδιαφέρει στην έξοδο να έχουμε τις τιμές 0 και 1, ή πιθανότητες αυτών.

Συναρτήσεις Σφάλματος

Οι συναρτήσεις σφάλματος όπως αναφέρθηκε παραπάνω προσπαθούν για τον καλύτερο υπολογισμό του σφάλματος της εξόδου. Όπως και σε όλα και εδώ, υπάρχει πληθώρα συναρτήσεων σφαλμάτων που αποφασίζεται βάσει του σκοπού του δικτύου αλλά και από την συνάρτηση ενεργοποίησης εξόδου.

Στην παρούσα εργασία, ο σκοπός του δικτύου μας ήταν να κατηγοριοποιήσει μία εικόνα σε καρκινική ή μη καρκινική το οποίο αποτελεί δυαδική κατηγοριοποίηση όπως προαναφέρθηκε. Ως εξίσωση σφάλματος λοιπόν, χρησιμοποιήθηκε η δυαδική διασταυρωμένη εντροπία (binary cross-entropy) που αναλύθηκε στο Κεφάλαιο 2.

Συναρτήσεις βελτιστοποίησης

Η συνάρτηση βελτιστοποίησης είναι υπεύθυνη για την μείωση του σφάλματος που έχει υπολογιστεί σύμφωνα με την συνάρτηση σφάλματος. Και εδώ, ανάλογα με τον σκοπό του δικτύου, όπως στις περισσότερες παραμέτρους του δικτύου, ορίζεται και η συνάρτηση βελτιστοποίησης η οποία προκύπτει με δοκιμές και σύγκριση.

Στην εργασία χρησιμοποιήθηκαν δύο συναρτήσεις βελτιστοποίησης που έχουν αναλυθεί στο Κεφάλαιο 2. Η πρώτη είναι η Adam και η δεύτερη η SGD που ήταν διαθέσιμες από την βιβλιοθήκη keras. Η συνάρτηση Adam είναι ο state-of-the-art αλγόριθμος βελτιστοποίησης στα βαθιά νευρωνικά δίκτυα και είναι εύκολος στην χρήση του καθώς δεν χρειάζεται να ορίσουμε την ορμή και την μεταβολή του ρυθμού μάθησης, και με αυτόν τρόπο επιταχύνεται η εκπαίδευση. Ωστόσο, δεν δίνει πάντα τα καλύτερα αποτελέσματα γιατί και πάλι η συνάρτηση βελτιστοποίησης αφορά το πρόβλημα. Η συνάρτηση SGD από την άλλη εφαρμόστηκε με μεταβλητό ρυθμό μάθησης που αυξομειώνεται κατάλληλα μετά από έναν ορισμένο κύκλο μάθησης και την χρήση σταθερής ορμής ίση με 0.9 ή 90%.

Εποχές εκπαίδευσης

Μία εποχή εκπαίδευσης ορίζεται ως η εισαγωγή όλων των δεδομένων εκπαίδευσης στο νευρωνικό δίκτυο και εν τέλει, ο υπολογισμός της τελικής εξόδου και τέλος, η μεταβολή των βαρών για την ελαχιστοποίηση του σφάλματος. Δηλαδή πρόκειται για ένα πέρασμα προς τα μπροστά των εισόδων(διανυσμάτων) του κάθε επιπέδου, τον υπολογισμό της τελικής εξόδου και στην συνέχεια τον υπολογισμό του συνολικού σφάλματος της εξόδου και την μετάδοση του προς τα πίσω (οπισθοδιάδοση) και αλλαγής των εκάστοτε βαρών. Η εποχή εκπαίδευσης μπορεί να χωριστεί σε batches ώστε το δίκτυο να αλλάζει τα βάρη του μετά από ένα συγκεκριμένο batch size, μιας και αν έχουμε πολλά δεδομένα θέλουμε να μην περιμένουμε να αλλάξουν μία φορά στο τέλος του περάσματος.

Η επιλογή των εποχών πάλι δεν διέπεται σε κανόνες και πρέπει να εκτελέσουμε πειράματα για να ορίσουμε πόσες εποχές χρειάζεται εκπαίδευση. Παράγοντες που το επηρεάζουν είναι ο το μέγεθος του δικτύου, οι συναρτήσεις βελτιστοποίησης και σφάλματος, τα δεδομένα και άλλα. Για τον καθορισμό των εποχών υπάρχουν εργαλεία της βιβλιοθήκης Tensorflow όπως το callback earlystopping και tensorboard που μπορούν να βοηθήσουν εμφανίζοντας το σφάλμα και τις μετρικές κάθε τέλος εποχής και βοηθάνε στην αποφυγή του overfitting.

Τα μοντέλα που δημιουργήσαμε εκπαιδεύτηκαν σε πλήθος εποχών 7, 18, και 39.

Batch Size

Αποτελούν τον αριθμό παραδειγμάτων εκπαίδευσης για επεξεργασία πριν την ανανέωση των βαρών του δικτύου. Το σύνολο δεδομένων χωρίζεται σε n batches και ξεκινώντας από το 1^ο θα υπολογιστεί το σφάλμα και έπειτα θα ανανεώσει τις παραμέτρους του δικτύου, έπειτα το δεύτερο κ.ο.κ., αυτή η διαδικασία συνεχίζεται μέχρι να ανανεώσει τα βάρη n φορές, δηλαδή όσο τα υποσύνολα που έχουμε δημιουργήσει.

Αν έχουμε n ίσο με ένα η ανανέωση των βαρών θα γίνει μία φορά μετά το πέρασμα όλων των παραδειγμάτων και είναι κάτι που δεν θέλουμε γιατί αυξάνει τον χρόνο εκπαίδευσης και επίσης, κάνει το μοντέλο πιο επιρρεπές στο φαινόμενο της υπερπροσαρμογής που έχει αναλυθεί παραπάνω. Από την άλλη μπορούμε να χρησιμοποιήσουμε batch size ίσο με όσα είναι τα παραδείγματά μας και μετά από κάθε παράδειγμα να τροποποιούμε τα βάρη το οποίο κάνει ο αλγόριθμος SDG (Stochastic Gradient Descent) αν δεν τροποποιηθεί. Με αυτό κερδίζουμε χρόνο εκπαίδευσης αλλά έχουμε ταλαντώσεις στην εύρεση του βέλτιστου μιας και συνεχώς αλλάζουμε βάρη.

Και στον ορισμό του batch size δεν έχουμε κάποιον κανόνα και εξαρτάται από όλα τα υπόλοιπα συν τον αριθμό των παραδειγμάτων που έχουμε. Συνήθως επιλέγονται δυνάμεις του 2 (32,64,128,...,1024 κ.ο.κ.) ως batch size.

Στην παρούσα εργασίες έπειτα από πολλές δοκιμές βρέθηκε ότι η καταλληλότερη τιμή είναι το 64.

Φίλτρα Συνελεκτικών Νευρωνικών Δικτύων

Τα φίλτρα των συνελεκτικών νευρωνικών δικτύων, είναι ίσως η σημαντικότερη παράμετρος όσον αφορά την επιλογή τους για την καλύτερη απόδοση του συνολικού μας δικτύου. Η δουλειά των φίλτρων είναι να εξάγει χαρακτηριστικά από την εικόνα, όπως αν είναι θολή, ανίχνευση άκρων,

και άλλα. Πρέπει να επιλέξουμε αρχικά το μέγεθος των φίλτρων (σύννηθες μεγέθη είναι 3x3, 5x5) και έπειτα να επιλέξουμε πόσα φίλτρα θα εφαρμόσουμε σε κάθε ξεχωριστό συνελεκτικό επίπεδο. Στο πόσα θα είναι συνήθως εφαρμόζουμε δυνάμεις του 2 φίλτρα, και αυξητικά, δηλαδή ξεκινάμε με 16,32 και συνεχίζουμε διπλασιάζοντας τα (1^ο επίπεδο 32 φίλτρα, 2^ο 64 φίλτρα, κ.ο.κ.) ώστε να εξαγάγουμε όλο και περισσότερα χαρακτηριστικά πηγαίνοντας βαθύτερα. Στην εργασία δοκιμάστηκαν διάφορες τεχνικές και για τα δύο και αυτά που δημιουργήσαμε, έχουν όλα φίλτρα 3x3 σε μέγεθος ενώ διαφέρουν το πόσα.

4.7 Μετρικές

Ο ρόλος των μετρικών στην διαδικασία εκπαίδευσης ενός αλγορίθμου μηχανικής μάθησης και συγκεκριμένα ενός νευρωνικού δικτύου, είναι η αξιολόγηση των τελικών εξόδων μετά την εκπαίδευση. Με την χρήση μετρικών σκοπός μας είναι να παρατηρήσουμε κατά πόσο ο αλγόριθμος (νευρωνικό δίκτυο) επιτυγχάνει τον αντικειμενικό σκοπό, και αν βάσει της κρίσης μας δεν το κάνει καλά, να συνεχίσουμε τους πειραματισμούς για να πάρουμε καλύτερα αποτελέσματα, ενδεχομένως τροποιώντας κατάλληλα τις παραμέτρους που παρουσιάστηκαν στο κεφάλαιο 4.6. Τέλος, οι μετρικές παρέχουν πληροφορίες καίριας σημασίας όσον αφορά τα προβλήματα που μπορεί να αντιμετωπίζει ο αλγόριθμός και ο τρόπος εκπαίδευσης του.

Υπάρχουν πληθώρα μετρικών που μπορούν να χρησιμοποιηθούν ανάλογα με την φύση του κάθε προβλήματος, με πιο σύννηθες την μετρική της ακρίβειας (Accuracy). Στην συγκεκριμένη πτυχιακή εργασία χρησιμοποιήθηκαν συνολικά 4 μετρικές, με κύρια μετρική την F1-Score ενώ οι άλλες 3 σαν δευτερεύουσες στο στάδιο της ανάπτυξης των τελικών μοντέλων. Ο μείζων σκοπός της χρήσης τους ήταν η βελτίωση του εκάστοτε νευρωνικού δικτύου για την διαδικασία εκπαίδευσης και τα τυχόν προβλήματα που μπορούν να υπάρχουν.

Τα δεδομένα είχαν χωριστεί σε αναλογία 80-10-10 τοις εκατό σε δεδομένα εκπαίδευσης, δεδομένα επαλήθευσης και τέλος δεδομένα ελέγχου αντίστοιχα. Η τελική αξιολόγηση των λύσεων, γινόταν πάνω στο 10% δεδομένων ελέγχου χρησιμοποιώντας την κύρια μετρική F1-Score.

Κύρια Μετρική:

Η κύρια μετρική που επιλέξαμε, ήταν η μετρική F1-Score και πιο συγκεκριμένα η macro F1-Score που συνοπολογίζει τις 2 κλάσσες ίσα. Αρχικά, η μετρική F1-Score είναι ο αρμονικός μέσος του Precision και του Recall αντίστοιχα δίνοντας ίση βαρύτητα και στις 2 ποσότητες, ενώ υπάρχουν και F-Scores που μπορεί να παραμετροποιηθούν αντίστοιχα για να δώσεις παραπάνω βαρύτητα σε ένα από τα δύο. Τώρα, το Precision είναι ο λόγος των σωστών προβλέψεων ως προς το πόσα συνολικά προβλέψαμε ως σωστά, ενώ το Recall είναι ο λόγος των σωστών προβλέψεων ως προς το σύνολο των σωστών παραδειγμάτων.

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN}$$

Τα δύο αυτά μεγέθη στέκονται και μόνα τους, με το Recall να είναι πολύ σημαντικό και να έχει ζωτικό ρόλο στην ανάπτυξη του συστήματος, μιας και υπολογίζει το ποσοστό των πόσων

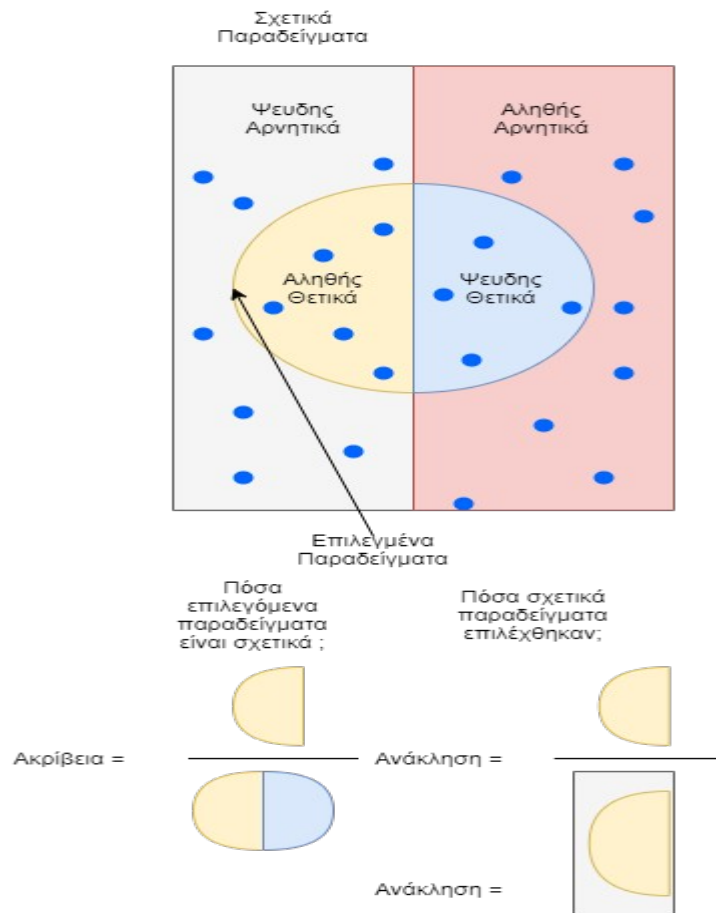
σωστών παραδειγμάτων καταφέραμε να προβλέψουμε σωστά, ενώ είναι αρκετά σημαντικό και το Precision, ώστε να κρατάμε τα ποσοστά των συνολικών ‘σωστών’ προβλέψεων σε καλό ποσοστό.

Τώρα όσον αφορά την μετρική F1-Score υπάρχουν διάφορες εκδοχές της όσον αφορά τον υπολογισμό της σε προβλήματα κατηγοριοποίησης. Οι δύο πιο συνηθισμένες είναι οι τεχνικές micro και macro οι οποίες διαφοροποιούνται στο πόσο βαρύτητα δίνουν στα αντίστοιχα Precision, Recall κάθε διαφορετικής κλάσης. Αυτό θα γίνει πιο κατανοητό χρησιμοποιώντας ένα παράδειγμα. Έστω ότι έχουμε ένα πρόβλημα δυαδικής κατηγοριοποίησης και η κλάση A είναι το 40% των δεδομένων ενώ το άλλο 60% είναι της κλάσης B, και έστω έχουμε 80% F1-Score για την κλάση A ενώ για την κλάση B έχουμε 20%. Θα υπολογίσουμε τα 2 αντίστοιχα μεγέθη.

$micro\ F1-Score = 0.8 * 40\% + 0.2 * 60\% = 0.44$ ή 44%.

$macro\ F1-Score = \frac{(0.8+0.2)}{2} = 0.5$ ή 50%

Οι δύο αυτές μέθοδοι αλλάζουν στο που δίνουν βαρύτητα. Για παράδειγμα η micro F1-Score δίνει ίδια βαρύτητα σε κάθε παράδειγμα το οποίο δεν θέλουμε όταν έχουμε προβλήματα κατηγοριοποίησης στα οποία υπερτερεί κάποια κλάση γιατί έτσι θα έχουμε «εξαφανισμό» της άλλης κλάσης και καθόλου συμμετοχή της στο τελικό F1-Score. Αυτό το κάνει η macro F1-Score δίνοντας την δυνατότητα να μην μετράει καθόλου το πόσα παραδείγματα έχουμε από κάθε κλάση, ώστε να αποφύγουμε αυτό το φαινόμενο και συνεπώς να έχουμε καλύτερη οπτική όσον αφορά το τι πρέπει να κάνουμε για να βελτιώσουμε το σύστημά μας.



Εικόνα 4.24 : Precision & Recall

Δευτερεύουσες Μετρικές:

Όσον αφορά τις δευτερεύουσες μετρικές χρησιμοποιήθηκαν αρκετές, με πιο συνηθισμένη το Accuracy (Ακρίβεια). Είναι αρκετά εύκολη ως προς την κατανόησή της, και εκφράζεται ως ο λόγος των σωστών παραδειγμάτων ως προς το σύνολο τους. Η μετρική αυτή είναι επαρκής για διάφορα προβλήματα, ωστόσο παρουσιάζει σημαντικό πρόβλημα όταν το ποσοστό των κλάσεων των δεδομένων είναι άνιση. Για παράδειγμα, αν έχουμε να κάνουμε πρόβλεψη για το αν θα χιονίσει μέσα σε έναν χρόνο, μπορούμε κάλλιστα να απαντάμε σε όλα τα παραδείγματα όχι και θα έχουμε ποσοστό ακρίβειας 99% ή και 100% χρησιμοποιώντας την συγκεκριμένη μετρική. Ωστόσο, μας νοιάζει να μπορούμε να προβλέψουμε το ποτέ δεν θα χιονίσει που είναι και το ασυνήθιστο. Αυτό συμβαίνει και στην περίπτωση μας όπου έχουμε ανισοροπία ανάμεσα στα δεδομένα. Πράγμα που όμως σημαίνει ότι δεν βοηθάει ουσιαστικά στην ανάπτυξη του δικτύου μας. Στην εργασία χρησιμοποιήθηκε ως ένας γρήγορος προσεγγιστικός τρόπος για την παρακολούθηση της εκπαίδευσης των δικτύων, αλλά στο τέλος η εκπαίδευση και τα δίκτυα κρίνονταν από την κύρια μετρική.

Πίνακας Σύγχυσης (Confusion Matrix) : Αποτελεί την βάση για τον υπολογισμό του Precision και του Recall, και συνεπώς και της κύριας μετρικής F1-Score. Η αναπαράσταση του για πρόβλημα 2 κλάσεων, έχει ως εξής : Στις γραμμές έχουμε τις προβλέψεις, την γραμμή των

θετικών προβλέψεων και αρνητικών, κάθε γραμμή έχει δύο στήλες, η μία αφορά τις προβλέψεις που έχουν δοθεί σωστά, δηλαδή τις πραγματικά θετικές (True Positive) και τις πραγματικά αρνητικές (True Negative), και η άλλη τις προβλέψεις που έχουν δοθεί λάθος, δηλαδή τις ψευδώς θετικές (False Positive) που αποτελούν το στατιστικό λάθος τύπου I, και τις ψευδώς αρνητικές (False Negative) που αναφέρονται αλλιώς ως στατιστικό λάθος τύπου II. Μέσω αυτού του πίνακα εξετάσαμε την πραγματική απόδοση του μοντέλου μας, στην πρόβλεψη των κλάσεων μας κατά τα διάφορα πειράματα.

		Πραγματική Κλάση	
		Θ	A
Προβλεπόμενη Κλάση	Θ	Αληθής Θετικά (ΑΘ)	Ψευδής Θετικά (ΨΘ)
	A	Ψευδής Αρνητικά (ΨΑ)	Αληθής Αρνητικά (ΑΝ)

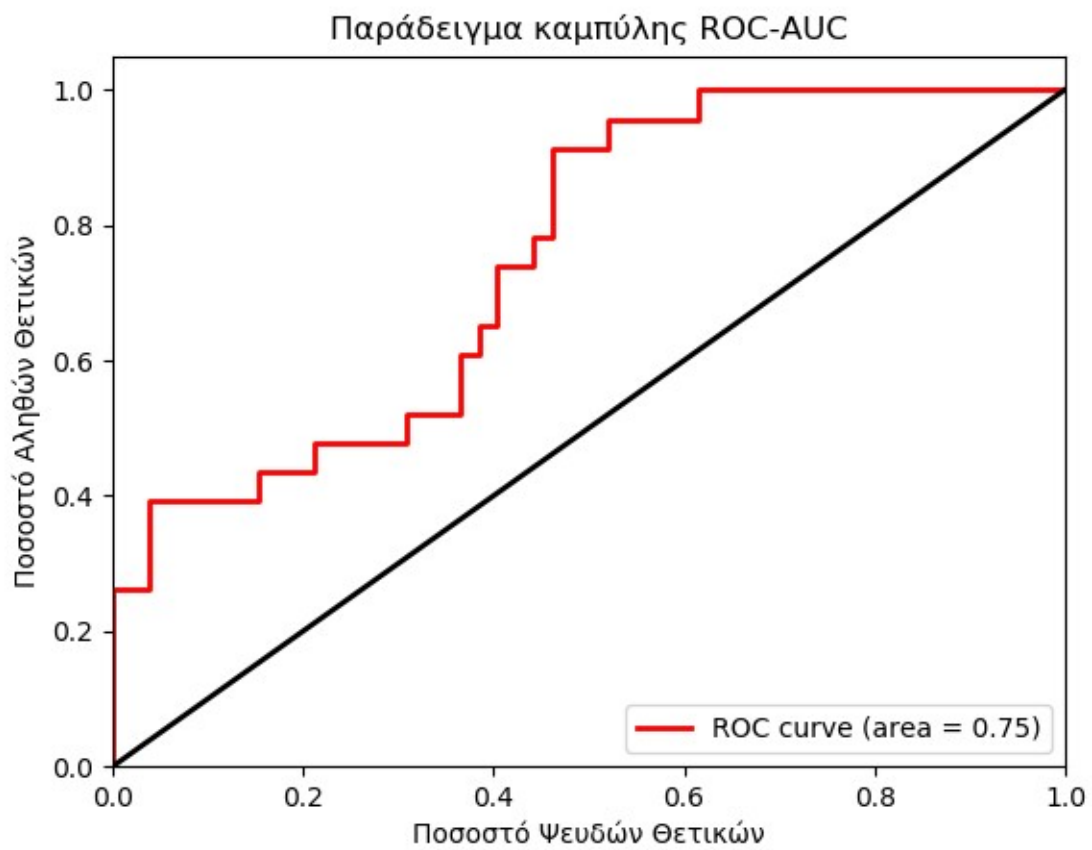
Εικόνα 4.25 : Πίνακας Σύγχυσης (Confusion Matrix)

AUC-ROC: Τελευταία μετρική που χρησιμοποιήθηκε ήταν το AUC-ROC που είναι το εμβαδόν που περιλαμβάνει η καμπύλη ROC. Το ιδανικό για έναν κατηγοριοποιητή είναι όσο πιο κοντά στο 1, ενώ στο 0 ο κατηγοριοποιητής δεν είναι αποτελεσματικός. Η τιμή 0.5 θεωρείται ως τιμή που κατηγοριοποιεί όλα τα παραδείγματα στην τύχη 50-50%. Με αυτό τον τρόπο μπορεί εύκολα και αποτελεσματικά να αξιολογηθεί ένας αλγόριθμος για κατηγοριοποίηση καθώς δεν επηρεάζεται ακόμα και αν η ποσοστόση των κλάσεων είναι εξαιρετικά άνιση.

$$TPR(\text{True Positive Rate ή Recall}) = \frac{TP}{TP + FN}$$

$$FPR(\text{False Positive Rate ή Fall out}) = \frac{FP}{FP + TN}$$

Τα TP, TN, FP, FN αντίστοιχα έχουν συζητηθεί παραπάνω όπως και το Recall. Το FPR ή Fall Out από την άλλη είναι το ποσοστό των πραγματικά λάθος παραδειγμάτων που προβλέψαμε ως σωστά δια το πλήθος των πραγματικά λάθος παραδειγμάτων που τα προβλέψαμε ως σωστά και αυτά που ήταν λάθος και τα προβλέψαμε λάθος (True Negatives).



Εικόνα 4.26 : AUC-ROC Curve

5 Πρόβλεψη με Συνελεκτικά Νευρωνικά Δίκτυα

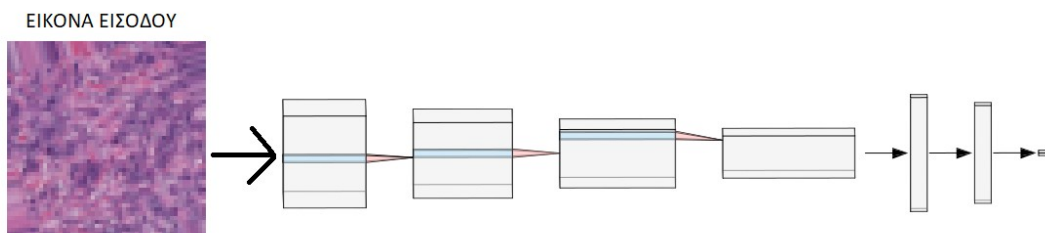
Η γενική προσέγγιση που ακολουθήθηκε για την λύση του συγκεκριμένου προβλήματος ήταν καθαρά με την χρήση συνελεκτικών νευρωνικών δικτύων. Η τεχνική αυτή είναι η state of the art μέθοδος που ακολουθήθηκε κυρίως σε προβλήματα κατηγοριοποίησης εικόνων και ότι έχει να κάνει με εικόνα (π.χ. βίντεο, εντόπιση αντικειμένου/αντικειμένων σε εικόνα, κλπ). Η τεχνική αυτή όπως έχει αναφερθεί παραπάνω σχολαστικά, ξεκινάει με την επεξεργασία της αρχικής εικόνας (εισόδου) με σκοπό να εξάγει τα σημαντικά χαρακτηριστικά από αυτήν. Αυτό γίνεται με την εφαρμογή πολλαπλών συνελεκτικών επιπέδων και επιπέδων pooling. Έπειτα, αφού φτάσουμε στο επιθυμητό μέγεθος της αρχικής εικόνας και την έχουμε μειώσει σημαντικά όσον αφορά τις διαστάσεις κρατώντας τα σημαντικότερα χαρακτηριστικά, μπορούμε να μετατρέψουμε την εικόνα από πίνακα pixels σε διάνυσμα pixels για να μπορέσουμε να εφαρμόσουμε οποιονδήποτε αλγόριθμο μηχανικής μάθησης για την κατηγοριοποίηση της αρχικής εικόνας. Στην παρούσα πτυχιακή εργασία ως κατηγοριοποιητής χρησιμοποιήθηκε αποκλειστικά νευρωνικό δίκτυο μιας και απέδιδε καλύτερα.

Τα πειράματα που πραγματοποιήθηκαν βασίστηκαν αποκλειστικά στην παραπάνω προσέγγιση. Υπάρχουν και άλλες προσεγγίσεις που θα μπορούσαν να εφαρμοστούν όπως συνδυασμός Συνελεκτικών Νευρωνικών Δικτύων με νευρωνικά δίκτυα στα οποία θα είχαμε επίπεδο/επίπεδα LSTM το οποίο δεν έχει συζητηθεί. Περιεκτικά, τα επίπεδα LSTM είναι ένα είδος μνήμης παραδειγμάτων το οποίο χρησιμοποιήθηκε κυρίως σε συνδυασμό με τα αναδρομικά νευρωνικά δίκτυα, ώστε να θυμούνται μακροπρόθεσμα χαρακτηριστικά των παραδειγμάτων. Ωστόσο, αυτό που δοκιμάστηκε ήταν η τεχνική Data augmentation σε συνδυασμό με τα Συνελεκτικά Νευρωνικά Δίκτυα. Αυτό που κάνει η τεχνική αυτή είναι να «αλλάζει» τα παραδείγματα με διάφορους τρόπους (π.χ. μπορούμε να κάνουμε περιστροφή 90-180-270° μοίρες στην εικόνα, zoom-in, zoom-out, αύξηση/μείωση φωτεινότητας κλπ) για να αυξήσουμε την ποικιλία των παραδειγμάτων που διαθέτουμε. Η μέθοδος αυτή χρησιμοποιείται για την καταπολέμηση του φαινομένου που αναφέρθηκε στο Κεφάλαιο 2.5 της υπερπροσαρμογής ή αλλιώς Overfitting. Δοκιμάστηκε στην παρούσα εργασία, αλλά εν τέλει δεν συμπεριλήφθηκε στα τελικά μοντέλα επειδή παρατηρήσαμε ότι δεν βοηθάει ιδιαίτερα, μιας και οι εικόνες που είχαμε ήταν αρκετές χιλιάδες ώστε να έχουμε καλή εκπαίδευση του δικτύου και συνεπώς, αποφυγή του φαινομένου της υπερπροσαρμογής.

Τέλος, τα πειράματα που έγιναν ήταν πάρα πολλά που ποικίλουν στις παραμέτρους, στις αρχιτεκτονικές και προφανώς στα αποτελέσματα. Το να παρουσιάζαμε όλα τα μοντέλα δεν θα είχε νοήμα, οπότε κρίνοντας βάσει της κύριας μετρικής F1-Score στο 10% δεδομένων ελέγχου καταλήξαμε στα 3 καλύτερα μοντέλα που είχαν και τις καλύτερες τιμές για την παραπάνω μετρική.

5.1 Αρχιτεκτονικές Λύσεων και αποτελέσματα

Για όλα τα μοντέλα που επιλέχθηκαν έγινε πληθώρα δοκιμών και πειραμάτων ώστε να καταλήξουμε στις καλύτερες δυνατές μετρικές και συνεπώς, στα καλύτερα αποτελέσματα. Το πρώτο μοντέλο αποτελούταν συνολικά από 4 συνελεκτικά επίπεδα σε συνδυασμό με επίπεδα pooling εκτός του τελευταίου συνελεκτικού επιπέδου γιατί μείωνε υπερβολικά την εικόνα και χανόταν πολλή πληροφορία. Οι παραμέτροι των συνελεκτικών νευρωνικών δικτύων και του νευρωνικού δικτύου στην έξοδο, παρουσιάζονται στο Κεφάλαιο 4.6 αναλυτικά. Αρχικά, το συνελεκτικό νευρωνικό δίκτυο δεχόταν ως αρχική είσοδο μία εικόνα 50x50x3 διάστασης, και έπειτα από την προεπεξεργασία πάνω σε αυτήν κατέληγε να έχει εξάγει τα πιο σημαντικά χαρακτηριστικά και να έχει ελαττώσει τις διαστάσεις. Έπειτα, ο κατηγοριοποιητής δεχόταν όλα αυτά τα pixels σαν διάνυσμα τιμών, και εισερχόταν σε ένα άλλο νευρωνικό δίκτυο ώστε να κατηγοριοποιήσει την εικόνα σε καρκινική ή μη. Από τα πολλαπλά πειράματα που έγιναν, φάνηκε ο συνδυασμός των 4 συνελεκτικών επιπέδων με εφαρμογή φίλτρων 3x3 (32,64,128,256 αντίστοιχα) χωρίς max pooling στο τελευταίο επίπεδο και άρα εισαγωγή 6x6 εικόνα στον κατηγοριοποιητή, να είναι ο αποδοτικότερος οπότε τα 3 τελικά μοντέλα έχουν τα ίδια συνελεκτικά ωστόσο και εκεί που διαφέρουν σημαντικά είναι στο νευρωνικό δίκτυο εξόδου, όπου αλλάζουν οι εξισώσεις βελτιστοποίησης, οι νευρώνες, τα επίπεδα και γενικά, οι περισσότερες παράμετροι. Όλα τα μοντέλα που θα παρουσιαστούν παρακάτω, έχουν εκπαιδευτεί σωστά όσον αφορά την αντιμετώπιση του φαινομένου της υπερπροσαρμογής (overfitting).



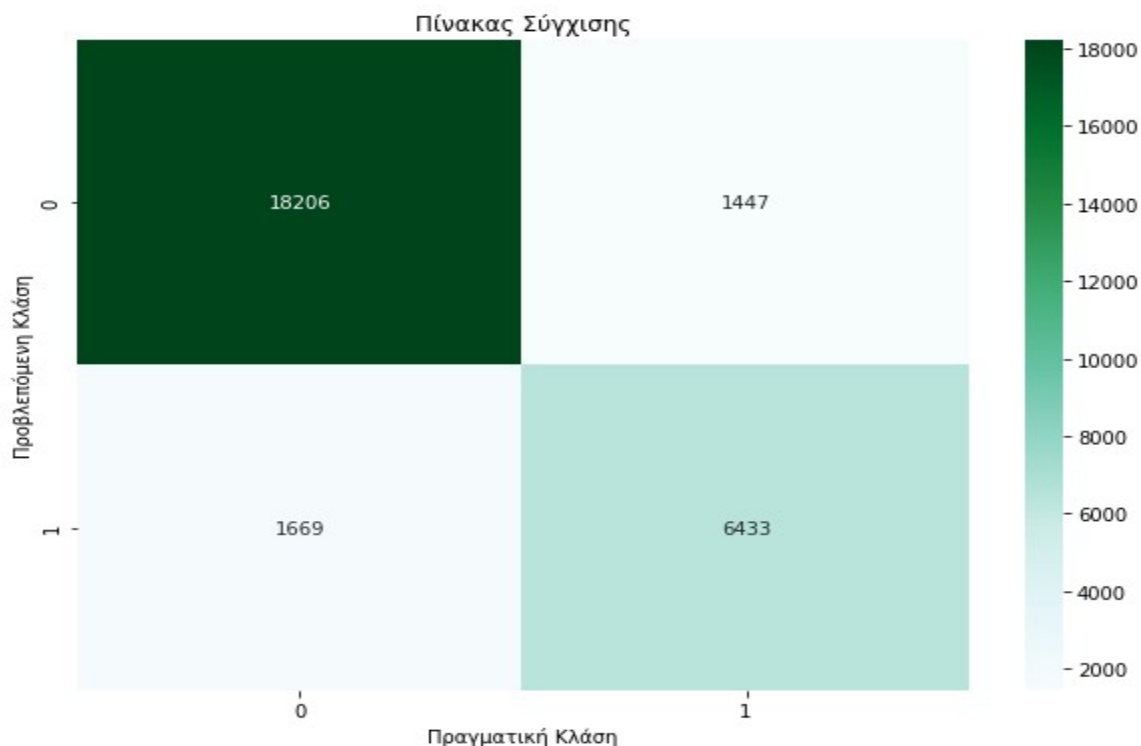
Εικόνα 5.27 : 1^η αρχιτεκτονική μοντέλου

1. Τα αποτελέσματα που πάρθηκαν με την 1^η αρχιτεκτονική που παρουσιάζεται στην εικόνα 5.1, και στο Κεφάλαιο 4.6 αναλυτικότερα ήταν ικανοποιητικά. Αρχικά, η μετρική F1-Score average ‘macro’ η οποία συζητήθηκε νωρίτερα, για το 10% των δεδομένων ελέγχου ήταν ίση με 86.31%. Οι υπόλοιπες μετρικές, του macro Recall και Precision είναι ίσες με 0.87 και 0.86 αντίστοιχα. Παρακάτω φαίνεται και ένα στιγμιότυπο που δείχνει πιο αναλυτικά τις μετρικές του συστήματος χρησιμοποιώντας την βιβλιοθήκη scikit-learn της Python.

	precision	recall	f1-score	support
0	0.92637	0.91603	0.92117	19875
1	0.79400	0.81637	0.80503	7880
accuracy			0.88773	27755
macro avg	0.86019	0.86620	0.86310	27755
weighted avg	0.88879	0.88773	0.88820	27755

Εικόνα 5.28 : Classification Report για το 1^ο μοντέλο

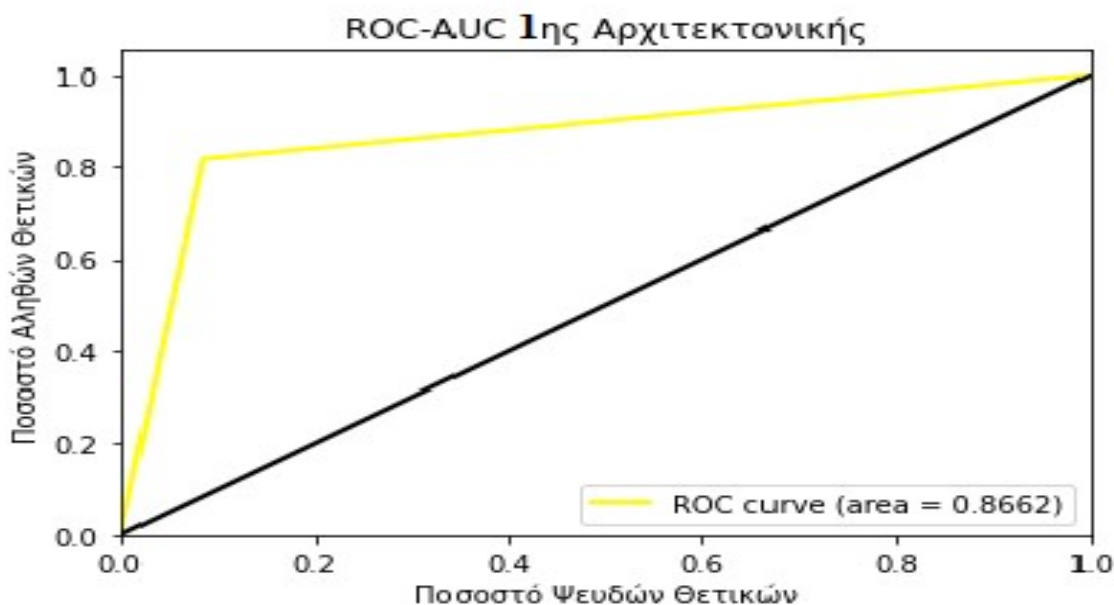
Επίσης, μία ακόμη σημαντική μετρική είναι ο πίνακας σύγκρισης ή αλλιώς Confusion matrix το οποίο μελετήθηκε παραπάνω στην εργασία και ποιός είναι ο ρόλος του. Ο παρακάτω πίνακας είναι ο πίνακας σύγκρισης του συγκεκριμένου μοντέλου στα δεδομένα ελέγχου που δημιουργήθηκε με την βοήθεια της βιβλιοθήκης scikit-learn και seaborn της Python. Το 1 αναπαριστά τα καρκινικά παραδείγματα ενώ το 0 δείχνει τα μη καρκινικά τα True Positive, True Negative, False Positive και τέλος, False Negative στον παρακάτω πίνακα.



Εικόνα 5.29 : Πίνακας Σύγκρισης (Confusion Matrix) 1^{ης} αρχιτεκτονικής

Παρατηρώντας την Εικόνα 5.3 του πίνακα σύγκρισης βλέπουμε ότι το μοντέλο μας καταφέρνει να κατηγοριοποιήσει σωστά 18.206 παραδείγματα ως μη καρκινικά και 6433 σωστά ως καρκινικά. Ωστόσο, έχουμε μεγάλες τιμές στα Ψευδείς Αρνητικά και στα Ψευδείς Θετικά το οποίο σημαίνει ότι κατηγοριοποιούμε 1669 ως καρκινικά ενώ είναι μη-καρκινικά, ενώ το πιο σημαντικό «λάθος» τύπου II είναι το να κατηγοριοποιούμε παραδείγματα ως μη-καρκινικά ενώ είναι καρκινικά και αυτά είναι ίσα με 1447.

Τέλος, μία ακόμα χρήσιμη μετρική που χρησιμοποιήθηκε ήταν αυτής της ROC-AUC που όπως αναφέρθηκε στο κεφάλαιο 4.7 είναι το εμβαδόν που περιλαμβάνει η καμπύλη ROC. Παρακάτω φαίνεται η καμπύλη ROC-AUC για την 1^η αρχιτεκτονική που παρουσιάσαμε. Δημιουργήθηκε με την βοήθεια της βιβλιοθήκης scikit-learn και matplotlib.

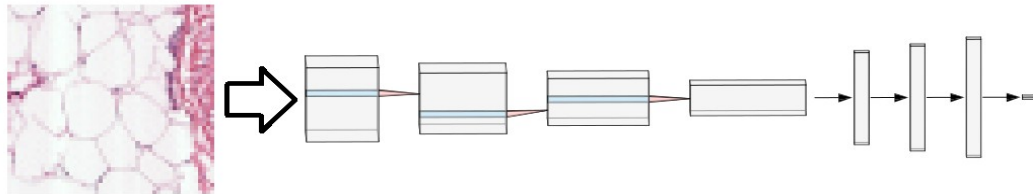


Εικόνα 5.30 : Καμπύλη ROC-AUC 1^{ης} αρχιτεκτονικής

Από την παραπάνω καμπύλη, μπορούμε να διακρίνουμε ότι το 1^ο μοντέλο μας καταφέρνει αρκετά καλά να διαχωρίζει μεταξύ των δύο κλάσεων κάτι το οποίο συμφωνεί και με τις μετρικές που παρουσιάστηκαν στην Εικόνα 5.2.

2. Η 2^η αρχιτεκτονική που ακολουθήθηκε διέφερε μόνο στην επιλογή των πλήρως συνδεδεμένων επιπέδων και στο πόσους νευρώνες είχαμε το οποίο φαίνεται και στην Εικόνα 5.5, ενώ στα υπόλοιπα όπως οι συναρτήσεις ενεργοποίησης, σφάλματος, κλπ ήταν οι ίδιες. Τα αποτελέσματα που πήραμε ήταν λίγο χειρότερα, ωστόσο πάρθηκαν πιο γρήγορα μιας και το μοντέλο αυτό εκπαιδεύτηκε σε 7 εποχές ενώ το άλλο σε 18. Παρακάτω θα παρουσιαστούν το μοντέλο, οι διάφορες μετρικές και ο πίνακας σύγκρισης με την μορφή εικόνων.

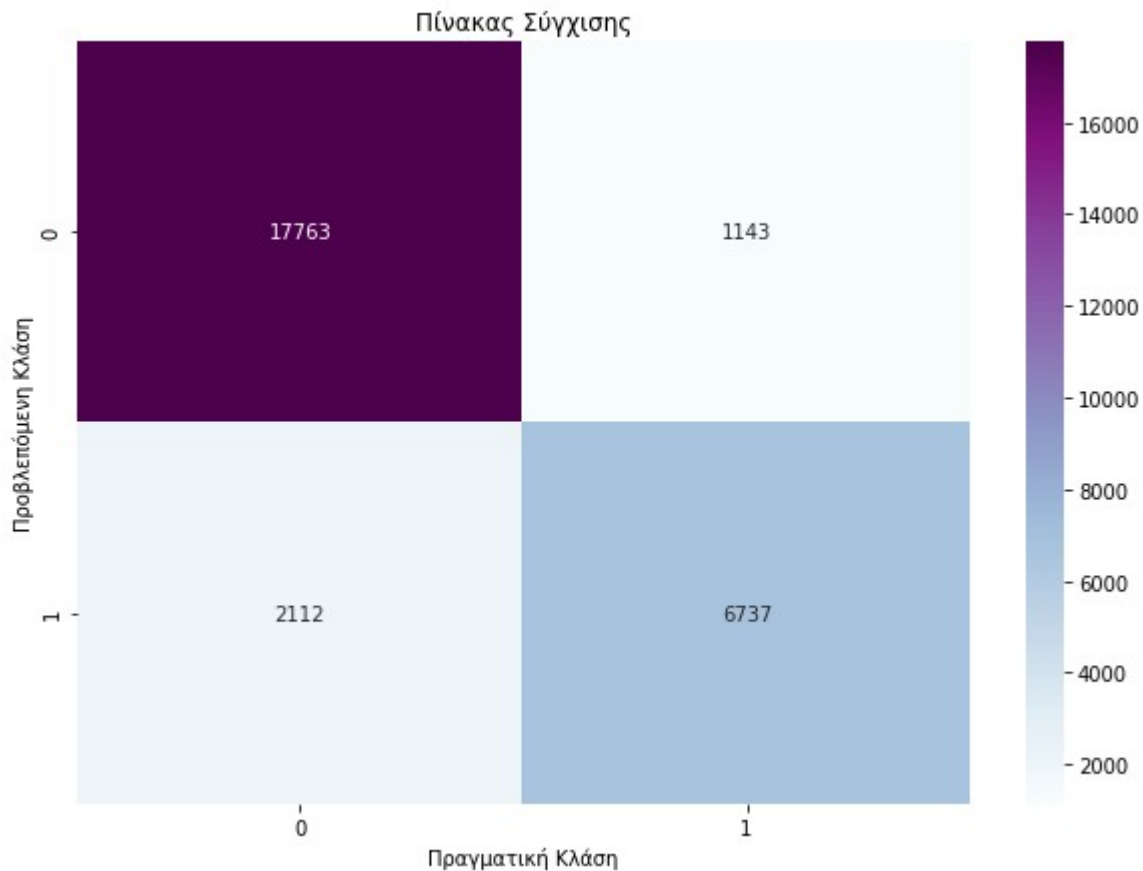
ΕΙΚΟΝΑ ΕΙΣΟΔΟΥ

Εικόνα 5.31 : 2^η αρχιτεκτονική μοντέλου

	precision	recall	f1-score	support
0	0.93954	0.89374	0.91607	19875
1	0.76133	0.85495	0.80543	7880
accuracy			0.88272	27755
macro avg	0.85044	0.87434	0.86075	27755
weighted avg	0.88895	0.88272	0.88466	27755

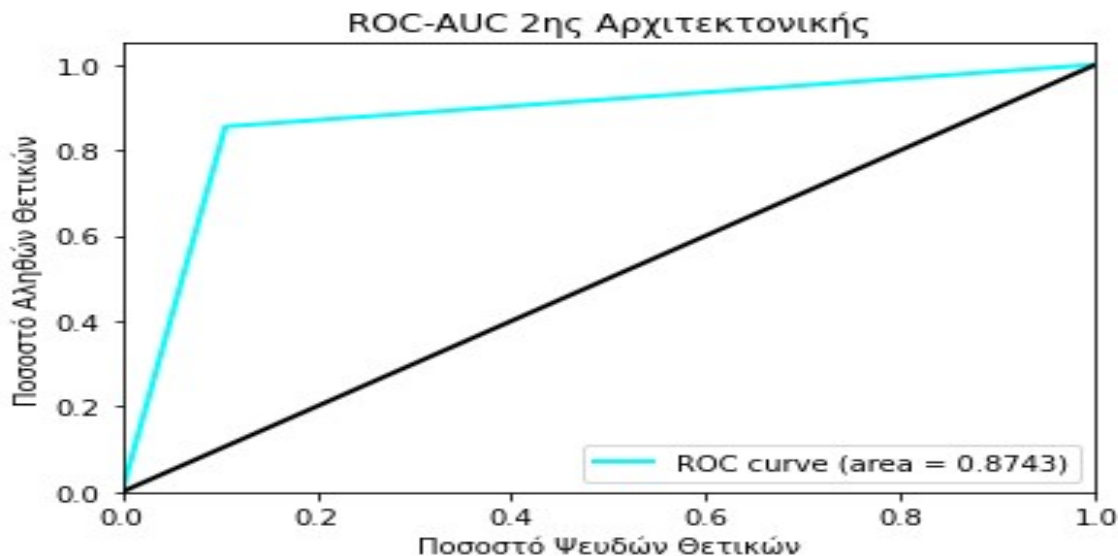
Εικόνα 5.32 : Classification Report για το 2^ο μοντέλο

Παραπάνω φαίνεται αρχικά η αρχιτεκτονική του μοντέλου που σχεδιάσαμε και τα επιμέρους επίπεδά του, ενώ στην Εικόνα 5.6 φαίνεται ότι σε σύγκριση με την 1^η αρχιτεκτονική έχουμε καλύτερο Recall όσον αφορά την κλάση 1, ωστόσο το Precision είναι χειρότερο ενώ στην κλάση 2 είναι λίγο καλύτερο το Precision ενώ το Recall όχι. Ακόμη το συνολικό macro F1-Score είναι χειρότερο από την 1^η αρχιτεκτονική όπως είναι επόμενο, αφού το 1^ο μοντέλο υπερτερεί στα Precision/Recall.



Εικόνα 5.33 : Πίνακας Σύγκρισης (Confusion Matrix) 2^{ης} αρχιτεκτονικής

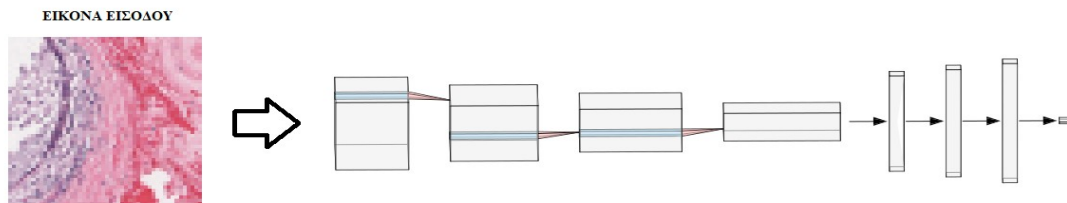
Όπως καταλαβαίνουμε και με την χρήση των μετρικών, έτσι και με τον πίνακα σύγκρισης μπορούμε να εξάγουμε το συμπέρασμα ότι ναι μεν προβλέψαμε περισσότερα σωστά με καρκίνο, ωστόσο προβλέψαμε αρκετά παραπάνω σαν False Positive (περίπου 500), δηλαδή ότι έχουν καρκίνο ενώ δεν είχαν το οποίο είναι πολύ σημαντικό και πρέπει να αποφεύγεται. Βέβαια, μπορούμε να πούμε με σιγουρία ότι πιο καίριας σημασίας είναι το γεγονός να έχουμε όσο λιγότερα False Negative μπορούμε γιατί το να πούμε σε έναν ασθενή ότι δεν έχει καρκινό και πραγματικά να έχει είναι ζήτημα ζωής και θανάτου, στο οποίο 2^ο μοντέλου έχουμε λιγότερα (1143 έναντι 1447 του 1^{ου} μοντέλου).



Εικόνα 5.34 : Καμπύλη ROC-AUC 2^{ης} αρχιτεκτονικής

Τέλος, από την καμπύλη ROC-AUC βλέπουμε ότι έχουμε αρκετά καλύτερο διαχωρισμό μεταξύ των δύο κλάσεων, συγκριτικά με το 1^ο μοντέλο μιας και η τιμή του ROC του είναι ίση με 86.62% ενώ παραπάνω έχουμε ίσο με 87.43%.

3. Τέλος, το 3^ο και καλύτερο μοντέλο που αναπτύχθηκε και είχε την καλύτερη μετρική F1-Score, είχε την παρακάτω δομή. Αρχικά είχαμε τα συνελεκτικά δίκτυα όπως και στα υπόλοιπα με την εφαρμογή των 32,64,128 και τέλος 256 φίλτρων πάνω στην αρχική εικόνα ακολουθούμενα από max pooling layer ενώ στο τελευταίο επίπεδο δεν είχαμε καθόλου. Έπειτα, στον κατηγοριοποιητή είχαμε 4 επίπεδα, ξεκινώντας με 256,512,1024 νευρώνες με συνάρτηση ενεργοποίησης την ReLU ενώ στο τέλος έναν σιγμοειδή για την τελική πρόβλεψη. Η κύρια διαφορά με τα προηγούμενα μοντέλα είναι ότι είχαμε εξίσωση βελτιστοποίησης τον SGD και όχι τον adam όπως στα 2 προηγούμενα μοντέλα. Αυτό οδήγησε σε καλύτερα αποτελέσματα, ωστόσο σε περισσότερο χρόνο μιας και πήρε 39 εποχές για να βρεθεί η συγκεκριμένη παραμετροποίηση. Παρακάτω θα παρουσιαστούν το μοντέλο, οι μετρικές και ο πίνακας σύγκρισης του συγκεκριμένου μοντέλου.

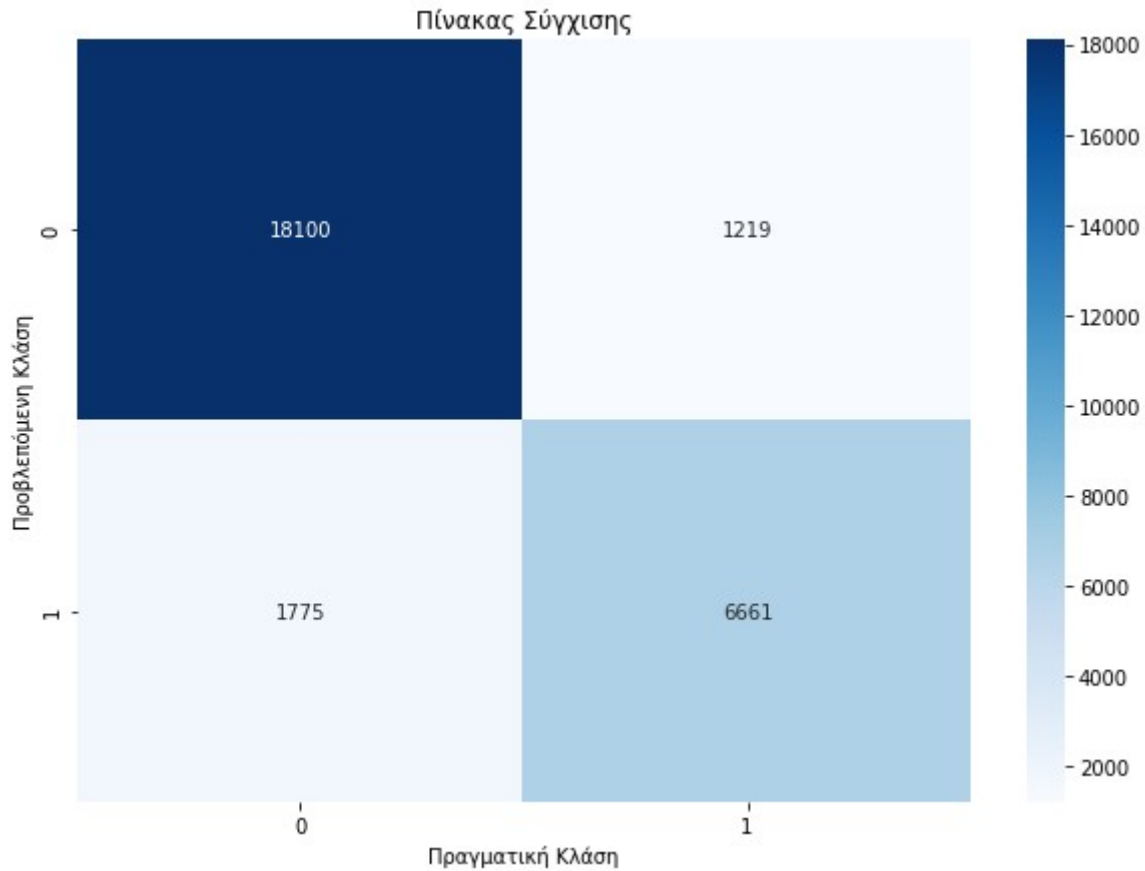


Εικόνα 5.35 : 3^η αρχιτεκτονική μοντέλου

	precision	recall	f1-score	support
0	0.93690	0.91069	0.92361	19875
1	0.78959	0.84530	0.81650	7880
accuracy			0.89213	27755
macro avg	0.86325	0.87800	0.87005	27755
weighted avg	0.89508	0.89213	0.89320	27755

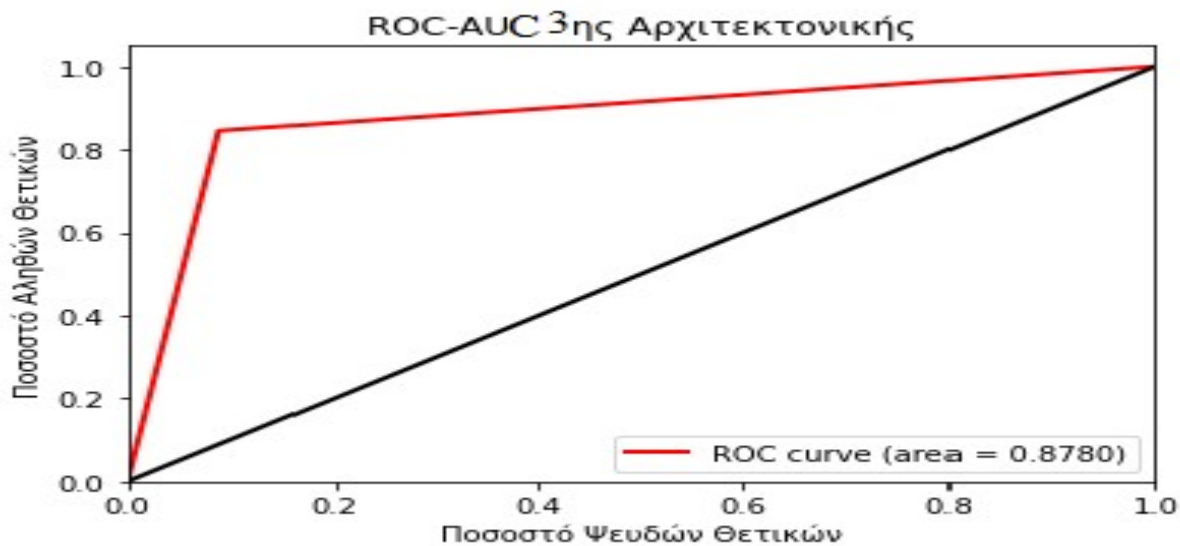
Εικόνα 5.36 : Classification Report για το 3^ο μοντέλο

Παρατηρώντας και τις μετρικές των άλλων μοντέλων βλέπουμε ότι υπερτερεί σε όλα για λίγο, ενώ η κύρια διαφορά είναι ότι στην κλάση 1 που μας ενδιαφέρει και περισσότερο, έχουμε τα καλύτερα αποτελέσματα όσον αφορά το precision/recall και συνεπώς, το f1-score. Τέλος παρουσιάζεται ο πίνακας σύγκρισης του μοντέλου αυτού στα δεδομένα ελέγχου.



Εικόνα 5.37 : Πίνακας Σύγχισης (Confusion Matrix) 3^{ης} αρχιτεκτονικής

Βλέποντας τον πίνακα σύγχισης του μοντέλου στα δεδομένα ελέγχου, παρατηρούμε ότι σε σύγκριση με τα υπόλοιπα 2 μοντέλα, έχουμε λιγότερα λάθη και στα παραδείγματα που είπαμε ότι είχαν καρκίνο ενώ πραγματικά δεν είχαν, ενώ και στα παραδείγματα που δεν είχαν και είπαμε ότι είχαν. Το 1^ο μοντέλο που παρουσιάσαμε μας δίνει πίνακα σύγχισης που βγάζει παραπάνω σωστά ως μη καρκινικά, ωστόσο στα καρκινικά βγάζει λιγότερα, ενώ στο 2^ο μοντέλο ισχύει το αντίθετο, δηλαδή έχουμε περισσότερα σωστά καρκινικά ενώ έχουμε λιγότερα μη-καρκινικά.



Εικόνα 5.38 : Καμπύλη ROC-AUC 3^{ης} αρχιτεκτονικής

Τέλος, παρατηρούμε ότι από τα 3 μοντέλα τελικά, αυτό είναι που έχει και το καλύτερο ROC-AUC σκορ και μπορεί να «ξεχωρίσει» καλύτερα παραδείγματα των δύο κλάσεων.

6 Συμπεράσματα και μελλοντικά σχέδια

Τα νευρωνικά δίκτυα και τα συνελεκτικά νευρωνικά δίκτυα όπως γίνεται αντιληπτό από την παρούσα εργασία και πληθώρα άλλων, αποτελούν ένα πανίσχυρο εργαλείο πρόβλεψης του οποίου έγινε χρήση για την πρόβλεψη του καρκίνου του μαστού σε εικόνες, με χρήση συνελεκτικών νευρωνικών δικτύων. Το σύνολο των δεδομένων ήταν μεγάλο και μη ισοροπημένο. Η δυσκολία σε τέτοιου είδους προβλήματα είναι ότι οι καρκινικές εικόνες και οι μη, είναι σχεδόν ίδιες και είναι πολύ δύσκολο να εξάγεις χαρακτηριστικά που θα κατηγοριοποιούν σωστά όλες τις εικόνες. Η προσέγγιση της εργασίας βασίστηκε αποκλειστικά σε συνελεκτικά νευρωνικά δίκτυα και κατηγοριοποιητή νευρωνικό δίκτυο, μιας και είναι η state of the art μέθοδος που χρησιμοποιείται για κατηγοριοποίηση εικόνων. Αναπτύχθηκαν συνολικά 3 μοντέλα, εκ των οποίων το τελευταίο ήταν και το καλύτερο μιας και έδινε τις καλύτερες δυνατές μετρικές για το συγκεκριμένο πρόβλημα.

Επίσης, φάνηκε ο αλγόριθμος του Stochastic Gradient Descent να λειτουργεί καλύτερα στο συγκεκριμένο πρόβλημα μετά από μεγάλο πλήθος εποχών, ωστόσο ο Adam έβγαζε λίγο χειρότερα αποτελέσματα σε λιγότερο πλήθος εποχών. Ακόμη, τα μοντέλα που αναπτύχθηκαν αντιμετώπισαν προβλήματα υπερπροσαρμογής, ωστόσο τα αντιμετωπίσαμε με την μέθοδο του drop out και της τεχνικής Batch Normalization για να έχουμε όσο πιο γενικευμένο μοντέλο μπορούμε, ενώ συμπεράναμε ότι η τεχνική του data augmentation δεν βοηθούσε, μιας και καθυστέρουσε κατά πολύ την διαδικασία της εκπαίδευσης ενώ δεν πρόσφερε πολλά.

6.1 Μελλοντικές Επεκτάσεις

Όπως αναφέρθηκε, το σύνολο δεδομένων που είχαμε και γενικότερα εικόνες τέτοιου είδους (με ιατρικό περιεχόμενο) και συνεπώς σύνολα δεδομένων, υπάρχουν πάρα πολλά, οπότε δεν είναι δυνατό να μελετηθούν όλες οι τεχνικές.

Για την καλύτερη απόδοση άλλων μοντέλων, θα μπορούσαμε να κάνουμε πειράματα με διαφορετικούς κατηγοριοποιητές και όχι μόνο με νευρωνικά δίκτυα (π.χ. Support Vector Machine, Decision Trees κλπ), ενώ θα μπορούσε να γίνει και χρήση LSTM που είναι ένα είδος μνήμης που χρησιμοποιείται σε συνδυασμό με τα αναδρομικά νευρωνικά δίκτυα όπως αναφέρθηκε παραπάνω. Ωστόσο, δεν μπορούμε να πούμε με σιγουρία ότι θα είχαμε καλύτερα αποτελέσματα. Επιπλέον, κάτι που θα βοηθούσε σίγουρα είναι να γίνονταν μια προεπεξεργασία άλλου είδους στις εικόνες και όχι μόνο με τα συνελεκτικά νευρωνικά δίκτυα.

Τέλος, όπως αναφέρθηκε παραπάνω η εκπαίδευση και γενικότερα όλη η προσέγγιση της παρούσας εργασίας ήταν με την χρήση patches εικόνων, δηλαδή κομμάτια από μία μεγαλύτερη εικόνα. Ωστόσο, είχαμε τις συνενταγμένες του κάθε patch του ασθενή ώστε να μπορέσουμε να ξανά «συναρμολογήσουμε» την εικόνα, οπότε θα μπορούσαμε να κάνουμε αρχικά αυτό που κάναμε στην παρούσα εργασία και έπειτα έναν άλλο μετά-κατηγοριοποιητή ο οποίος θα έβγαζε μία συνολική πρόβλεψη όσον αφορά τον ασθενή βάσει όλων των patch που έχει στην διάθεσή του αν τελικά έχει καρκίνο του μαστού ή όχι.

References

- [1] "Understanding Cancer," 9 February 2015. [Online]. Available: <https://www.cancer.gov/about-cancer/understanding/what-is-cancer#related-diseases>.
- [2] "Worldwide cancer data," 2018. [Online]. Available: <https://www.wcrf.org/dietandcancer/cancer-trends/worldwide-cancer-data>.
- [3] D. Bardou, K. Zhang and S. M. Ahmad, "Classification of Breast Cancer Based on Histology Images Using Convolutional Neural Networks," *IEEE Access*, vol. 6, pp. 24680-24693, 2018.
- [4] Macenko, Marc, et al., "A method for normalizing histology slides for quantitative analysis," *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro. IEEE*, 2009.
- [5] Araújo, Teresa, et al., "Classification of breast cancer histology images using convolutional neural networks," *PloS one* 12.6, 2017.
- [6] Α. Γεωργούλη, in "Τεχνητή Νοημοσύνη" – Μια εισαγωγική προσέγγιση, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών, 2015, p. 156.
- [7] A. L. Chandra, "<https://towardsdatascience.com/>," 24 July 2018. [Online]. Available: <https://towardsdatascience.com/mcculloch-pitts-model-5fdf65ac5dd1>. [Accessed 8 April 2020].
- [8] A. Géron, "The Perceptron," in *Hands-On Machine Learning with Scikit-Learn & TensorFlow*, O'Reilly Media, 2017, p. 322.
- [9] Sazli, Murat., "A brief review of feed-forward neural networks.," *Communications, Faculty Of Science, University of Ankara*, vol. 50, pp. 11-17, 2006.
- [10] S. Haykin, Νευρωνικά Δίκτυα και Μηχανική Μάθηση, Εκδόσεις Παπασωτηρίου, 2009.
- [11] Ramachandran, P., Zoph, B. and Le, Q.V., Searching for activation functions. arXiv preprint, 2017.
- [12] A. S. Walia, "Types of Optimization Algorithms used in Neural Networks and Ways to Optimize Gradient Descent," 10 Jun 2017. [Online]. Available: <https://towardsdatascience.com/types-of-optimization-algorithms-used-in-neural-networks-and-ways-to-optimize-gradient-95ae5d39529f>. [Accessed 15 April 2020].
- [13] L. Mishra, "www.pylenin.com," 11 June 2019. [Online]. Available: <https://www.pylenin.com/blogs/10-benefits-of-switching-to-python-3/>.

- [14] M. Kamaruzzaman, "towardsdatascience.com," 4 Feb 2020. [Online]. Available: <https://towardsdatascience.com/top-10-in-demand-programming-languages-to-learn-in-2020-4462eb7d8d3e>.
- [15] G. Colaboratory, "Google Colaboratory," [Online]. Available: <https://research.google.com/colaboratory/faq.html#resource-limits>.
- [16] TensorFlow, "Tensorflow," [Online]. Available: <https://www.tensorflow.org/>.
- [17] Keras, "Keras," [Online]. Available: <https://www.keras.io>.
- [18] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," *Computing in Science Engineering*, vol. 9, no. 3, pp. 90-95, 2007.
- [19] Pedregosa, Fabian and Varoquaux, Ga{\`e}l and Gramfort, Alexandre and Michel, Vincent and Thirion, Bertrand and Grisel, Olivier and Blondel, Mathieu and Prettenhofer, Peter and Weiss, Ron and Dubourg, Vincent and others, "Scikit-learn: Machine learning in Python," *Journal of machine learning research*, vol. 12, pp. 2825-2830, 2011.
- [20] Seaborn, "Seaborn," [Online]. Available: <https://seaborn.pydata.org/>.
- [21] S. Folders, "Split Folders," [Online]. Available: <https://pypi.org/project/split-folders/>.
- [22] Tqdm, "Tqdm," [Online]. Available: <https://pypi.org/project/tqdm/>.