

Ιατρική Οργανολογία-Ευφυή Συστήματα Ταξινόμησης Ασθενειών



Περιεχόμενα

1.Εισαγωγή	3
2.Ιατρική οργανολογία	3
Δοσιμετρία	3
MRI	3
CT	4
Ultrasound	5
NMR	5
Camera γ	7
AI	8
3.Ευφυή Συστήματα Ταξινόμησης Ασθενειών	8
Γενικές πληροφορίες-Τι είναι τα ευφυή συστήματα ταξινόμησης;	8
Νευρωνικά δίκτυα (Neural Networks)	9
Συνελικτικά νευρωνικά δίκτυα (Convolutional Neural Networks)	10
Τεχνικές Βελτίωσης	15
Συνάρτηση ενεργοποίησης (Activation function)	15
Σύνολο δεδομένων επικύρωσης (Validation data set)	15
Επανακλιμάκωση (Rescale)	15
Επαύξηση δεδομένων (Data augmentation)	15
Dropout	16
BatchNormalization	16
Πρώιμη διακοπή (Early stopping)	17
Checkpoints	17
Χρονοδιαγράμματα ρυθμού εκμάθησης (Learning rate schedulers)	17
4. Χρήση Προ-εκπαιδευμένων Νευρωνικών Δικτύων	18
5.Τεχνικές Ψηφιακής επεξεργασίας εικόνων	19
6.Υλοποίηση σε διαθέσιμα dataset	20
Ταξινόμηση εικόνων μαγνητικής τομογραφίας (MRI) για διάφορα είδη καρκίνου στον εγκέφαλο	20
Διάγνωση Κορονοϊού από εικόνες αξονικής τομογραφίας (CT) του πνεύμονα	27
Δοκιμή μοντέλων από το ένα dataset στο άλλο.	30
Συγκεντρωτικά αποτελέσματα σε πίνακες	31
7.Επίλογος	31
8.Βιβλιογραφία	32

1.Εισαγωγή

Στην παρούσα εργασία μελετώνται ορισμένες τεχνολογίες οργάνων, που χρησιμοποιούνται στην ιατρική για την διάγνωση ασθενειών. Επίσης, γίνεται αναφορά στα ευφυή συστήματα ταξινόμησης και στη χρήση τους στην ιατρική για τη διάγνωση ασθενειών. Τέλος, δοκιμάστηκε η τεχνολογία των συνελικτικών νευρωνικών δικτύων για τη διάγνωση ασθενειών σε δύο διαθέσιμα dataset που αφορούν ιατρικές εικόνες.

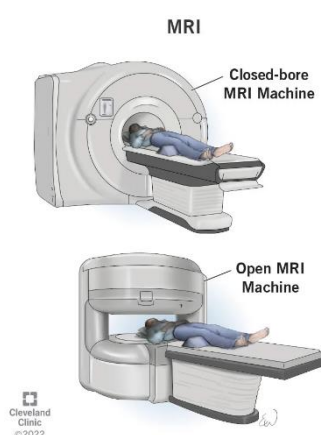
2.Ιατρική οργανολογία

Δοσιμετρία

Η «δοσιμετρία» αναφέρεται στην επιστήμη με την οποία προσδιορίζεται η δόση της ακτινοβολίας για τη θεραπεία ή την ανίχνευση ασθενειών, συμπεριλαμβανομένου του καρκίνου. Συγκεκριμένα, είναι η ποσότητα ενέργειας ακτινοβολίας που εναποτίθεται στον ιστό διαιρεμένη με τη μάζα του ιστού. Όσο υψηλότερη είναι η απορροφούμενη δόση στους όγκους, τόσο περισσότερα κύτταρα θα σκοτωθούν από την ακτινοβολία και τόσο μεγαλύτερη είναι η πιθανότητα ίασης. Ταυτόχρονα, κάποια ακτινοβολία φυσιολογικών οργάνων είναι αναπόφευκτη. Επομένως, ο ρόλος της δοσιμετρίας ακτινοβολίας στη στοχευμένη θεραπεία με ραδιονουκλείδια είναι να προσδιορίζει ειδικά, για κάθε ασθενή, τη χορηγούμενη ποσότητα του ραδιοφαρμάκου που θα θεραπεύσει αποτελεσματικότερα τη νόσο του ασθενούς, αποφεύγοντας τις απορροφημένες δόσεις που βλάπτουν τους φυσιολογικούς ιστούς. Η εξατομικευμένη δοσιμετρία ακτινοβολίας είναι κρίσιμη για τον σχεδιασμό της πιο αποτελεσματικής και ασφαλέστερης στοχευμένης θεραπείας με ραδιονουκλείδια για κάθε ασθενή.

MRI

Η μαγνητική τομογραφία (MRI) είναι μια μη επεμβατική τεχνολογία απεικόνισης που παράγει τρισδιάστατες λεπτομερείς ανατομικές εικόνες.



Οι μαγνητικοί τομογράφοι χρησιμοποιούν ισχυρούς μαγνήτες που παράγουν ένα ισχυρό μαγνητικό πεδίο που αναγκάζει τα πρωτόνια στο σώμα να ευθυγραμμιστούν με αυτό το πεδίο. Όταν στη συνέχεια ένα ρεύμα ραδιοσυχνότητας διοχετεύεται στον ασθενή, τα πρωτόνια διεγείρονται και περιστρέφονται εκτός ισορροπίας, τεντώνοντας την έλξη του μαγνητικού πεδίου. Όταν το πεδίο ραδιοσυχνότητας είναι απενεργοποιημένο, οι αισθητήρες MRI είναι σε θέση να ανιχνεύσουν την ενέργεια που απελευθερώνεται καθώς τα πρωτόνια ευθυγραμμίζονται εκ νέου με το μαγνητικό πεδίο. Ο χρόνος που χρειάζεται για να ευθυγραμμιστούν εκ νέου τα πρωτόνια με το μαγνητικό πεδίο, καθώς και η ποσότητα της ενέργειας που απελευθερώνεται, αλλάζει ανάλογα με το περιβάλλον και τη χημική φύση των μορίων. Οι

γιατροί είναι σε θέση να πουν τη διαφορά μεταξύ διαφόρων τύπων ιστών με βάση αυτές τις μαγνητικές ιδιότητες.

Οι σαρωτές μαγνητικής τομογραφίας είναι ιδιαίτερα κατάλληλοι για την απεικόνιση των μη οστικών μερών ή των μαλακών ιστών του σώματος. Διαφέρουν από την αξονική τομογραφία (CT), στο ότι δεν χρησιμοποιούν την επιβλαβή ιονίζουσα ακτινοβολία των ακτίνων Χ. Ο εγκέφαλος, ο νωτιαίος μυελός και τα νεύρα, καθώς και οι μύες, οι σύνδεσμοι και οι τένοντες φαίνονται πολύ πιο καθαρά με τη μαγνητική τομογραφία παρά με τις κανονικές ακτινογραφίες και αξονική τομογραφία.

MRI στον εγκέφαλο

Η μαγνητική τομογραφία μπορεί να διαφοροποιήσει τη λευκή και τη φαιά ουσία του εγκεφάλου και μπορεί επίσης να χρησιμοποιηθεί για τη διάγνωση ανευρυσμάτων και όγκων. Επειδή δεν χρησιμοποιεί ακτινοβολία, είναι συνήθως η μέθοδος που επιλέγεται όταν απαιτείται συχνή απεικόνιση για διάγνωση ή θεραπεία, ειδικά στον εγκέφαλο

CT

Η αξονική τομογραφία (CT scan) είναι μια διαγνωστική διαδικασία απεικόνισης που χρησιμοποιεί έναν συνδυασμό ακτίνων Χ και τεχνολογίας υπολογιστή για την παραγωγή εικόνων του εσωτερικού του σώματος. Δείχνει λεπτομερείς εικόνες οποιουδήποτε μέρους του σώματος, συμπεριλαμβανομένων των οστών, των μυών, του λίπους, των οργάνων και των αιμοφόρων αγγείων.



Οι αξονικές τομογραφίες είναι πιο λεπτομερείς από τις τυπικές ακτινογραφίες. Στις τυπικές ακτινογραφίες, μια δέσμη ενέργειας στοχεύει στο μέρος του σώματος που μελετάται. Μια πλάκα πίσω από το μέρος του σώματος καταγράφει τις διακυμάνσεις της δέσμης ενέργειας αφού περάσει από το δέρμα, τα οστά, τους μυς και άλλους ιστούς. Ενώ πολλές πληροφορίες μπορούν να ληφθούν από μια τακτική ακτινογραφία, πολλές λεπτομέρειες σχετικά με τα εσωτερικά όργανα και άλλες δομές δεν είναι διαθέσιμες.

Στην αξονική τομογραφία, η ακτίνα Χ κινείται σε κύκλο γύρω από το σώμα. Αυτό επιτρέπει πολλές διαφορετικές απόψεις του ίδιου οργάνου ή δομής και παρέχει πολύ περισσότερες λεπτομέρειες. Οι πληροφορίες ακτίνων Χ αποστέλλονται σε έναν υπολογιστή που ερμηνεύει τα δεδομένα ακτίνων Χ και τα εμφανίζει σε δισδιάστατη μορφή σε μια οθόνη. Η νεότερη τεχνολογία και το λογισμικό υπολογιστών καθιστούν δυνατές τις τρισδιάστατες εικόνες.

Οι αξονικές τομογραφίες μπορεί να πραγματοποιηθούν για να βοηθήσουν στη διάγνωση όγκων, στη διερεύνηση της εσωτερικής αιμορραγίας ή στον έλεγχο για άλλους εσωτερικούς τραυματισμούς ή βλάβες. Η αξονική τομογραφία μπορεί επίσης να χρησιμοποιηθεί για βιοψία ιστού ή υγρού.

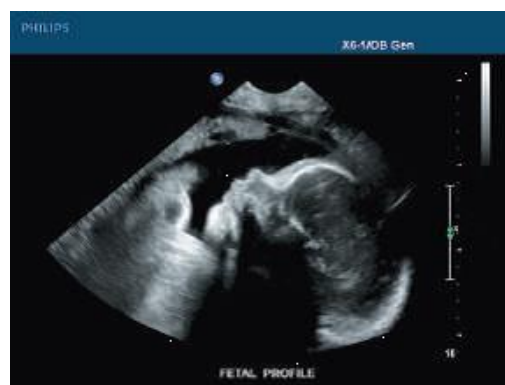
CT scan στους πνεύμονες

Η αξονική τομογραφία μπορεί να χρησιμοποιηθεί για την ανίχνευση τόσο οξείων όσο και χρόνιων αλλαγών στο πνευμονικό παρέγχυμα, τον ιστό των πνευμόνων. Είναι ιδιαίτερα σημαντικό εδώ επειδή οι κανονικές διαστάσεις ακτίνες Χ δεν δείχνουν τέτοια ελαττώματα. Χρησιμοποιείται μια ποικιλία τεχνικών, ανάλογα με την υποψία ανωμαλίας. Για την αξιολόγηση χρόνιων διάμεσων διεργασιών όπως το εμφύσημα και η ίνωση, χρησιμοποιούνται λεπτές τομές με ανακατασκευές υψηλής χωρικής συχνότητας. Συχνά πραγματοποιούνται σαρώσεις τόσο κατά την εισπνοή όσο και κατά την εκπνοή. Αυτή η ειδική τεχνική ονομάζεται CT υψηλής ανάλυσης που παράγει δειγματοληψία του πνεύμονα και όχι συνεχείς εικόνες.

Ultrasound

Το διαγνωστικό υπερηχογράφημα είναι μια μη επεμβατική διαγνωστική τεχνική που χρησιμοποιείται για την απεικόνιση μέσα στο σώμα.

Οι περισσότεροι διαγνωστικοί ανιχνευτές υπερήχων τοποθετούνται στο δέρμα. Ωστόσο, για να βελτιστοποιηθεί η ποιότητα της εικόνας, μπορούν να τοποθετηθούν ανιχνευτές στο εσωτερικό του σώματος μέσω του γαστρεντερικού σωλήνα, του κόλπου ή των αιμοφόρων αγγείων. Επιπλέον, ο υπέρηχος χρησιμοποιείται μερικές φορές κατά τη διάρκεια της χειρουργικής επέμβασης με την τοποθέτηση ενός αποστειρωμένου καθετήρα στην περιοχή που χειρουργείται.



Λειτουργεί με την παραγωγή κυμάτων υπερήχων από έναν μορφοτροπέα, ο οποίος μπορεί να εκπέμπει υπερηχητικά κύματα, καθώς και να ανιχνεύσει τις ηχώ υπερήχων που ανακλώνται πίσω. Όταν αυτές οι ηχώ χτυπούν τον μορφοτροπέα, παράγουν ηλεκτρικά σήματα που αποστέλλονται στον σαρωτή υπερήχων. Χρησιμοποιώντας την ταχύτητα του ήχου και το χρόνο επιστροφής κάθε ηχούς, ο σαρωτής υπολογίζει την απόσταση από τον μορφοτροπέα έως το όριο του ιστού. Αυτές οι αποστάσεις χρησιμοποιούνται στη συνέχεια για τη δημιουργία διαστάσεων εικόνων ιστών και οργάνων.

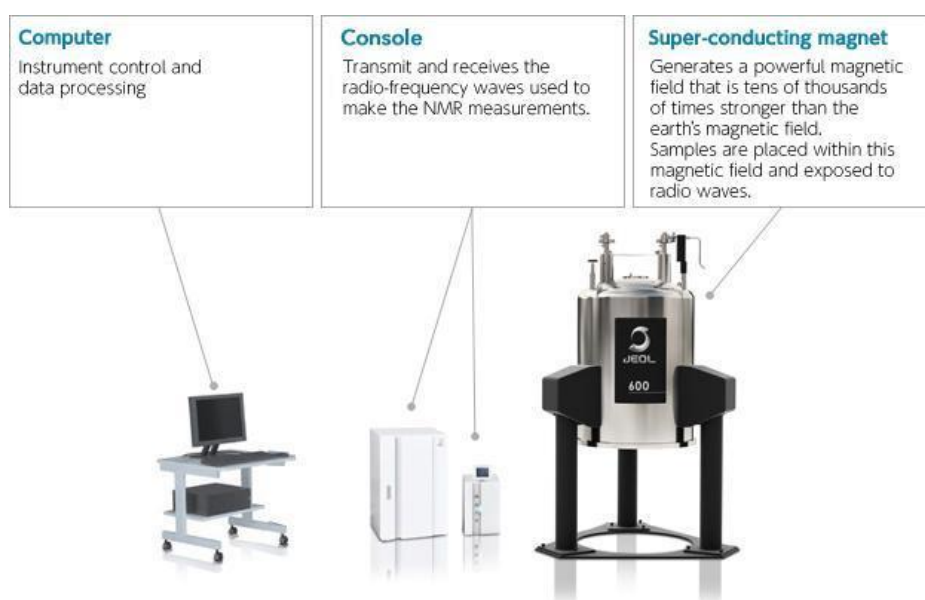
Μία από τις πιο κοινές χρήσεις του υπερήχου είναι κατά τη διάρκεια της εγκυμοσύνης, αλλά υπάρχουν πολλές άλλες χρήσεις, όπως η απεικόνιση της καρδιάς, των αιμοφόρων αγγείων, των ματιών, του θυρεοειδούς, του εγκεφάλου, του μαστού, των κοιλιακών οργάνων, του δέρματος, και μύες. Ωστόσο, δεν είναι καλό για την απεικόνιση οστών ή ιστών που περιέχουν αέρα, όπως οι πνεύμονες. Οι εικόνες υπερήχων εμφανίζονται σε 2D, 3D ή 4D (που είναι 3D σε κίνηση).

NMR

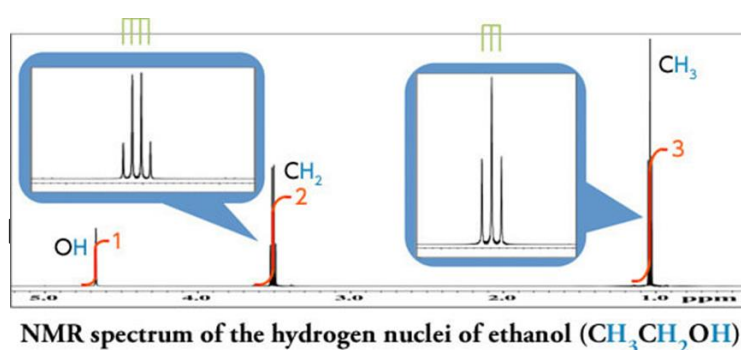
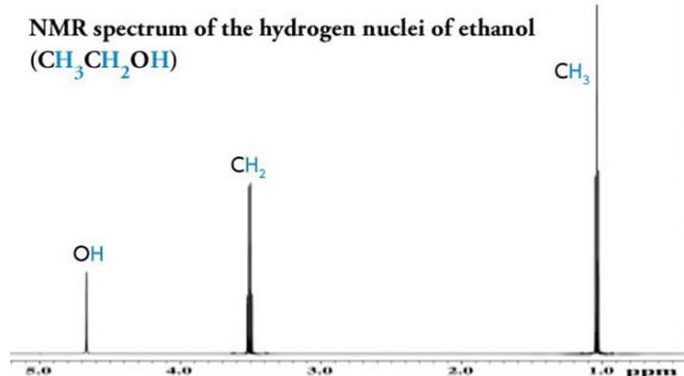
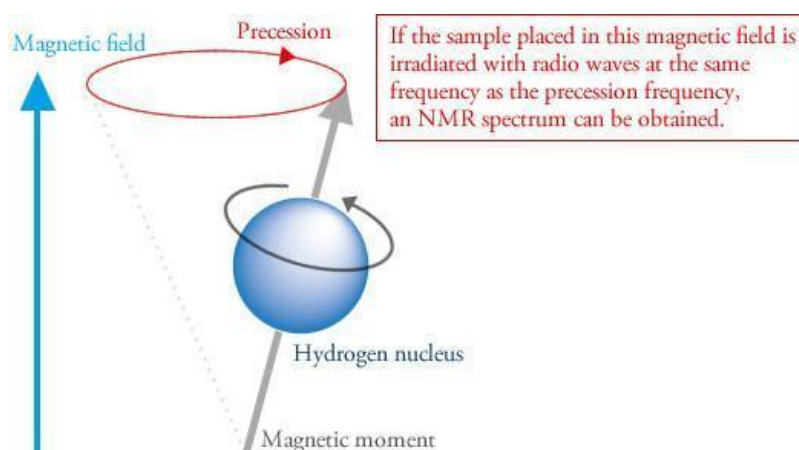
Ο πυρηνικός μαγνητικός συντονισμός (NMR) είναι μια από τις πιο ευρέως χρησιμοποιούμενες μεθόδους για την αναγνώριση χημικών ενώσεων, ιδιαίτερα οργανικών. Βασίζεται στην αρχή του πυρηνικού spin. Όταν ο πυρήνας ενός ατόμου τοποθετείται σε ένα

ισχυρό μαγνητικό πεδίο, το μαγνητικό του πεδίο ευθυγραμμίζεται με την κατεύθυνση του πεδίου. Όταν εφαρμόζεται ενέργεια, με τη μορφή ραδιοκυμάτων, ορισμένοι από τους πυρήνες περιστρέφονται ενάντια στο πεδίο. Το φασματομέτρο μετρά την ενέργεια που απελευθερώνεται όταν οι πυρήνες επιστρέφουν στην περιστροφή με το πεδίο. Από αυτές τις πληροφορίες μπορεί να ληφθεί πληθώρα δομικών δεδομένων.

Όταν ένας πυρήνας που διαθέτει μαγνητική ροπή (όπως ένας πυρήνας υδρογόνου ^1H ή ο



πυρήνας άνθρακα ^{13}C) τοποθετηθεί σε ένα ισχυρό μαγνητικό πεδίο, θα αρχίσει να κινείται, σαν μια περιστρεφόμενη κορυφή.



NMR spectrum of the hydrogen nuclei of ethanol ($\text{CH}_3\text{CH}_2\text{OH}$)

Φασματοσκοπία NMR και ιατρική διάγνωση

Την τελευταία δεκαετία, η Φασματοσκοπία Πυρηνικού Μαγνητικού Συντονισμού έπαιξε θεμελιώδη ρόλο στην πρόοδο της διαγνωστικής ιατρικής. Οι μελέτες NMR προσφέρουν μια σειρά πλεονεκτημάτων σε σχέση με άλλες τεχνικές, καθιστώντας την το μέσο επιλογής για τους ερευνητές στον τομέα της διαγνωστικής ιατρικής. Παρακάτω, παραθέτουμε τρία από τα πιο ελκυστικά οφέλη της Φασματοσκοπίας Πυρηνικού Μαγνητικού Συντονισμού κατά την ανάλυση ιατρικών δειγμάτων.

- Ευκολία στην προετοιμασία του δείγματος
- Βιολογική ασφάλεια
- Μη επεμβατικότητα και μη καταστροφικότητα

Αναγνώριση ανθρώπινων διαταραχών

Η φασματοσκοπία NMR είναι ιδιαίτερα χρήσιμη για την ανάλυση και τον εντοπισμό ανθρώπινων διαταραχών, ιδιαίτερα όταν χρησιμοποιείται παράλληλα με τη μεταβολομική ανάλυση. Αυτό συμβαίνει επειδή ο κυτταρικός μεταβολισμός προσφέρει σημαντικές ενδείξεις όταν πρόκειται για τη διάγνωση ενός ευρέος φάσματος καταστάσεων, όπως καρδιαγγειακές παθήσεις, ο καρκίνος, ακόμη και νευροψυχιατρικές διαταραχές όπως η σχιζοφρένεια, οι διαταραχές πανικού, η μείζονα κατάθλιψη, οι διπολικές διαταραχές και ο αυτισμός.

Camera γ

Η κάμερα γάμμα, γνωστή και ως κάμερα σπινθηρισμού, είναι μια συσκευή διάγνωσης και παρακολούθησης που είναι ζωτικής σημασίας για την πρακτική της πυρηνικής ιατρικής. Ραδιοφαρμακευτικές ουσίες (ιχνηθέτες) χορηγούνται στον ασθενή και καθώς ταξιδεύουν στο σώμα του ασθενή εκπέμπεται ακτινοβολία που μπορεί να συλληφθεί και να παρακολουθηθεί από έναν κρύσταλλο στην κάμερα σπινθηρίζει (αναβοσβήνει) ως απόκριση στην έκθεση σε ακτίνες γάμμα. Ο κρύσταλλος της κάμερας είναι τοποθετημένος μπροστά από μια σειρά αισθητήρων φωτός έτσι ώστε οι λάμπεις φωτός κατά την έκθεση να μετατρέπονται σε ηλεκτρικό σήμα. Αυτός είναι ο λόγος για τον οποίο η κάμερα είναι επίσης γνωστή ως κάμερα σπινθηρισμού. Αυτή η τεχνολογία βοηθά στη μέτρηση της λειτουργίας και της υγείας διαφόρων δομών και οργάνων του σώματος, καθώς οι ειδικοί μπορούν να αναγνωρίσουν καταστάσεις που βασίζονται στη συσσώρευση ή τον αποκλεισμό του ιχνηθέτη από ορισμένες περιοχές.



Οι σαρώσεις που λαμβάνονται με αυτόν τον τρόπο βοηθούν τους ειδικούς να διαγνώσουν και να εντοπίσουν διάφορους καρκίνους, συγγενείς ανωμαλίες και πολλές άλλες καταστάσεις, επιτρέποντάς τους να σχεδιάσουν μια αποτελεσματική πορεία δράσης. Διορθωτική δράση για την τροποποίηση και την αλλαγή των σχεδίων θεραπείας είναι επίσης δυνατή, καθώς οι ειδικοί είναι σε θέση να μετρήσουν την ανταπόκριση των ασθενών στις συνεχιζόμενες θεραπείες.

AI

Πλέον εκτός από τα παραπάνω, χρησιμοποιείται και η τεχνητή νοημοσύνη για τη βελτίωση της ιατρικής περίθαλψης και της βιοϊατρικής έρευνας με κάποιους από τους τρόπους που παρατίθενται στη συνέχεια:

- Ακτινολογία: Η ικανότητα της τεχνητής νοημοσύνης να ερμηνεύει τα αποτελέσματα της απεικόνισης μπορεί να βοηθήσει στον εντοπισμό μιας μικρής αλλαγής σε μια εικόνα που ένας κλινικός ιατρός μπορεί να χάσει κατά λάθος.
- Απεικόνιση: Ένα παράδειγμα είναι η χρήση της τεχνητής νοημοσύνης για την αξιολόγηση του πώς θα φαίνεται ένα άτομο μετά από χειρουργική επέμβαση προσώπου.
- Τηλε-υγεία: Οι φορητές συσκευές επιτρέπουν τη συνεχή παρακολούθηση του ασθενούς και την ανίχνευση φυσιολογικών αλλαγών που μπορεί να παρέχουν έγκαιρα προειδοποιητικά σημάδια ενός συμβάντος, όπως μια κρίση άσθματος.
- Κλινική περίθαλψη: Μεγάλη εστίαση της τεχνητής νοημοσύνης στον τομέα της υγειονομικής περίθαλψης είναι στα συστήματα υποστήριξης κλινικών αποφάσεων, τα οποία χρησιμοποιούν παρατηρήσεις υγείας και γνώση περιπτώσεων για να βοηθήσουν στη λήψη αποφάσεων θεραπείας.

3. Ευφυή Συστήματα Ταξινόμησης Ασθενειών

Γενικές πληροφορίες-Τι είναι τα ευφυή συστήματα ταξινόμησης;

Τα ευφυή συστήματα ταξινόμησης είναι συστήματα μηχανικής μάθησης που χρησιμοποιούνται για την αυτόματη ταξινόμηση δεδομένων σε διάφορες κατηγορίες (κλάσεις).

Τα συστήματα αυτά χρησιμοποιούν αλγόριθμους μηχανικής μάθησης (machine learning), όπως είναι για παράδειγμα ο αλγόριθμος Κ πλησιέστερων γειτόνων (KNN, K nearest neighbors) και οι μηχανές διανυσμάτων υποστήριξης (SVM, support vector machines). Μία υποκατηγορία της μηχανικής μάθησης είναι η βαθιά μάθηση (deep learning) στην οποία ανήκουν τα νευρωνικά δίκτυα (neural networks).

Τα ευφυή συστήματα ταξινόμησης χρησιμοποιούνται σε μια πληθώρα εφαρμογών. Στην ιατρική τα ευφυή συστήματα ταξινόμησης ασθενειών μπορούν να βοηθήσουν στη διάγνωση ασθενειών που είναι δύσκολες ή περίπλοκες να αναγνωριστούν από τον άνθρωπο, καθώς και στην αξιολόγηση και παρακολούθηση της υγείας του ασθενούς. Επιπλέον, μπορούν να χρησιμοποιηθούν για την πρόβλεψη πιθανών ασθενειών και την πρόληψη της εξάπλωσής τους. Αυτά τα συστήματα χρησιμοποιούν δεδομένα από ιατρικούς φακέλους, ιατρικές εγγραφές, εργαστηριακά αποτελέσματα και άλλες πηγές, ώστε να επιτύχουν ακρίβεια στη διάγνωση των ασθενειών. Μπορούν επίσης να χρησιμοποιηθούν και για τη σύγκριση δεδομένων από διαφορετικούς ασθενείς.

Η χρήση των ευφυών συστημάτων ταξινόμησης ασθενειών μπορεί να βοηθήσει στη βελτίωση της διάγνωσης, της θεραπείας και της πρόληψης ασθενειών. Επίσης, συμβάλλει στην εξοικονόμηση χρόνου και κόστους για τους ιατρούς και τους ασθενείς και τη βελτίωση της υγειονομικής περίθαλψης.

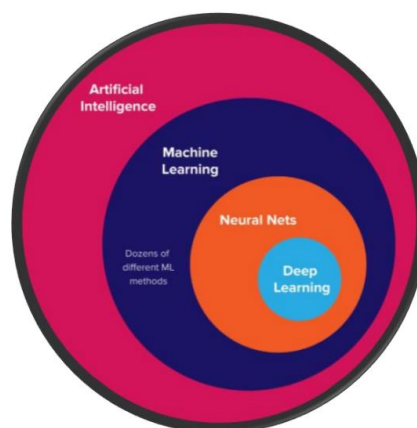
Ωστόσο, υπάρχουν και ορισμένες προκλήσεις που σχετίζονται με τη χρήση ευφυών συστημάτων ταξινόμησης ασθενειών, όπως η προστασία της ιδιωτικότητας των ασθενών.

Επιπλέον, η ευαισθησία των αλγορίθμων, μπορεί να οδηγήσει σε λάθη και παραλείψεις και ως αποτέλεσμα σε λανθασμένες διαγνώσεις και θεραπείες.

Παρόλα αυτά, η τεχνητή νοημοσύνη και η μηχανική μάθηση συνεχίζουν να βελτιώνονται και να εξελίσσονται και αναμένεται ότι τα ευφυή συστήματα ταξινόμησης ασθενειών θα είναι ακόμα πιο ακριβή και αποτελεσματικά στο μέλλον.

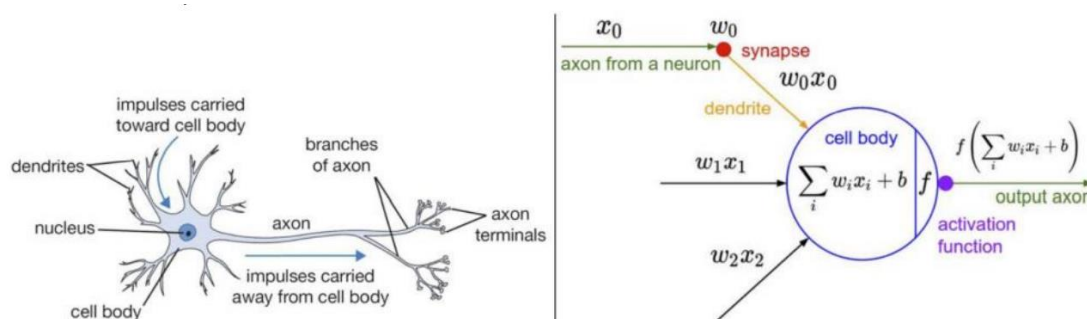
Νευρωνικά δίκτυα (Neural Networks)

Τα τεχνητά νευρωνικά δίκτυα (artificial neural networks) είναι αλγόριθμοι εμπνευσμένοι από τους βιολογικούς νευρώνες μέσα στο ανθρώπινο σώμα. Αποτελούν υποκατηγορία της μηχανικής μάθησης (machine learning) και εμπεριέχουν τα βαθιά νευρωνικά δίκτυα (deep neural networks) ή βαθιά μάθηση (deep learning). Όλα τα παραπάνω είναι υποκατηγορίες ενός εκτεταμένου πεδίου που ονομάζουμε τεχνητή νοημοσύνη (artificial intelligence).



Το πρόβλημα της ταξινόμησης ανήκει στην κατηγορία επιβλεπόμενης μηχανικής μάθησης (supervised machine learning), επειδή οι κλάσεις των εικόνων είναι γνωστές, έχουμε δηλαδή τις ετικέτες (labels).

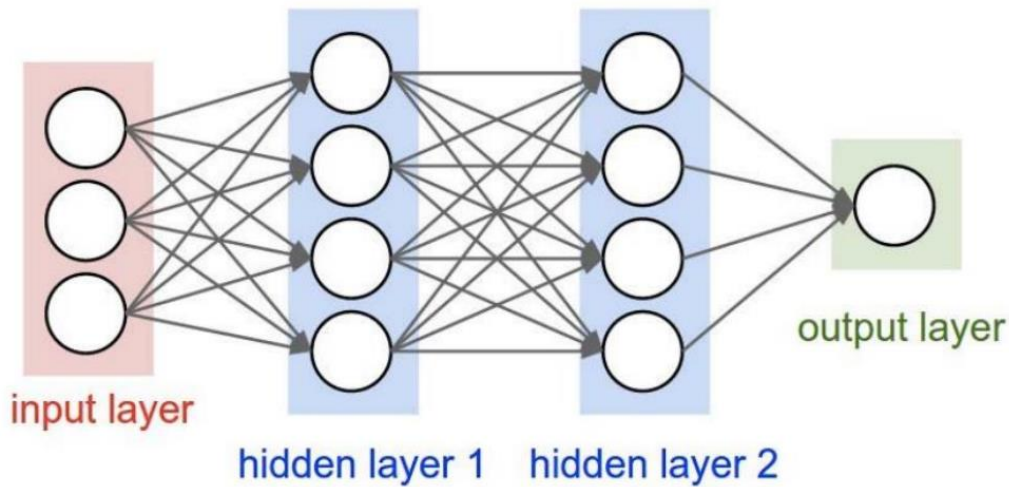
Ο τεχνητός νευρώνας είναι εμπνευσμένος από το βιολογικό νευρώνα του ανθρώπινου εγκεφάλου. Ο συσχετισμός αποτυπώνεται στην παρακάτω εικόνα και στον παρακάτω πίνακα.



Biological Neural Network	Artificial Neural Network
Dendrites	Inputs
Cell nucleus	Nodes
Synapse	Weights
Axon	Output

Ένα τεχνητό νευρωνικό δίκτυο αποτελείται από ένα επίπεδο εισόδου (input layer), ένα επίπεδο εξόδου (output layer) και κάποια κρυμμένα επίπεδα (hidden layers).

Στα προβλήματα ταξινόμησης, το επίπεδο εξόδου έχει τόσους νευρώνες, όσες και οι

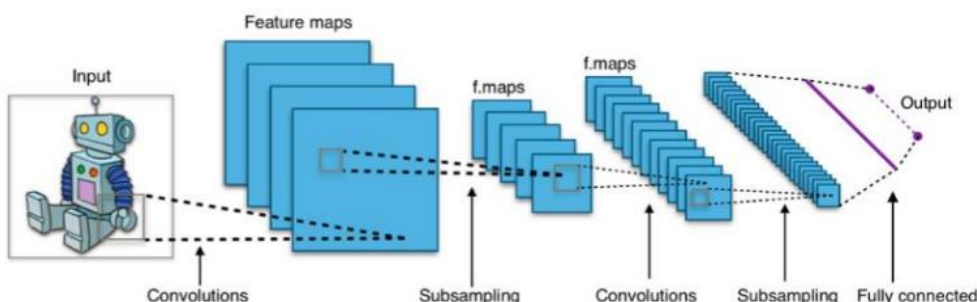
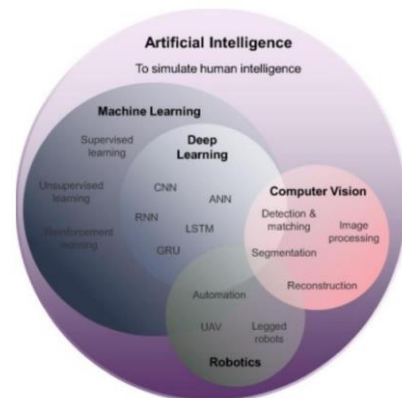


κλάσεις που θέλουμε να ταξινομήσουμε

Συνελικτικά νευρωνικά δίκτυα (Convolutional Neural Networks)

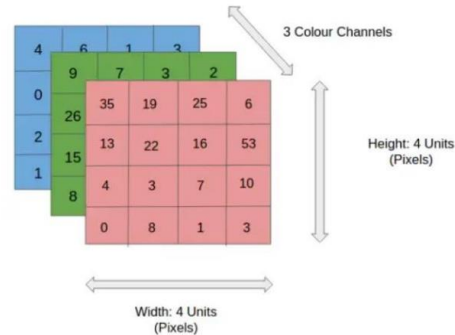
Τα νευρωνικά δίκτυα που χρησιμοποιούνται για την ταξινόμηση εικόνων, ονομάζονται συνελικτικά νευρωνικά δίκτυα (convolutional neural networks). Είναι στην ουσία υποκατηγορία των νευρωνικών δικτύων και χρησιμοποιούνται κατά κόρον στην όραση υπολογιστών, σε μια πληθώρα εφαρμογών.

Ένα CNN μπορεί να λάβει μια εικόνα στην είσοδο, να αποδώσει σημασία σε διάφορα αντικείμενα της εικόνας και να διαφοροποιήσει το ένα από το άλλο, ώστε να κατανοήσει το περιεχόμενό της. Είναι ιδιαίτερα χρήσιμα στην εύρεση μοτίβων σε οπτικές εικόνες για την αναγνώριση αντικειμένων και κλάσεων. Τα CNN χρησιμοποιούν σχετικά λίγη προ-επεξεργασία σε σύγκριση με άλλους αλγόριθμους ταξινόμησης εικόνων. Το δίκτυο μαθαίνει να βελτιστοποιεί τα φίλτρα (kernels) μέσω της αυτοματοποιημένης εκμάθησης, ενώ στους παραδοσιακούς αλγόριθμους αυτά τα φίλτρα είναι κατασκευασμένα με το χέρι. Σε ένα CNN τα κρυμμένα επίπεδα (layers) εκτελούν λειτουργίες με σκοπό την εκμάθηση χαρακτηριστικών. Τα πιο κοινά επίπεδα είναι η συνέλιξη (convolution) και η υποδειγματοληψία (pooling/subsampling).

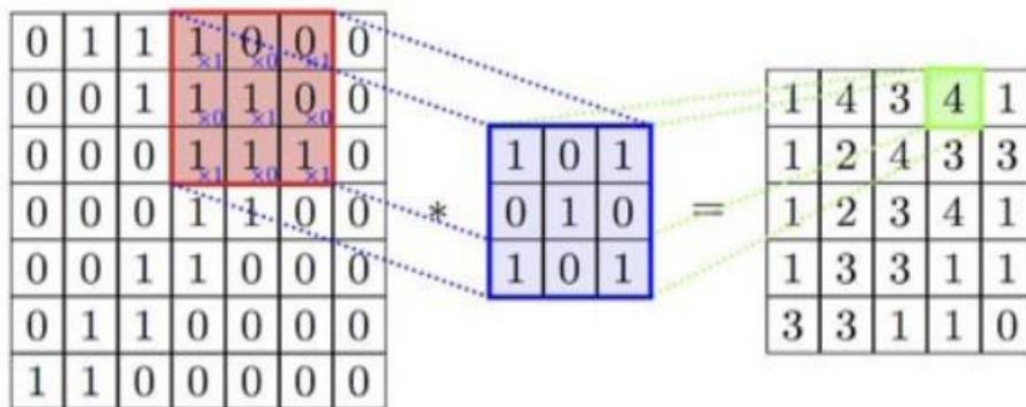


Ένα συνελκτικό επίπεδο (conv2D), κυρίως χρησιμοποιείται για την εξαγωγή χαρακτηριστικών από την εικόνα εισόδου. Τα πρώτα στρώματα είναι υπεύθυνα για την καταγραφή χαρακτηριστικών χαμηλού επιπέδου. Με πρόσθετα επίπεδα συνέλιξης, η αρχιτεκτονική προσπαθεί να εξάγει χαρακτηριστικά υψηλού επιπέδου. Τα φίλτρα Conv2D εκτείνονται μέσω των τριών καναλιών της εικόνας (κόκκινο-πράσινο-μπλε, RGB). Τα φίλτρα είναι διαφορετικά για κάθε κανάλι.

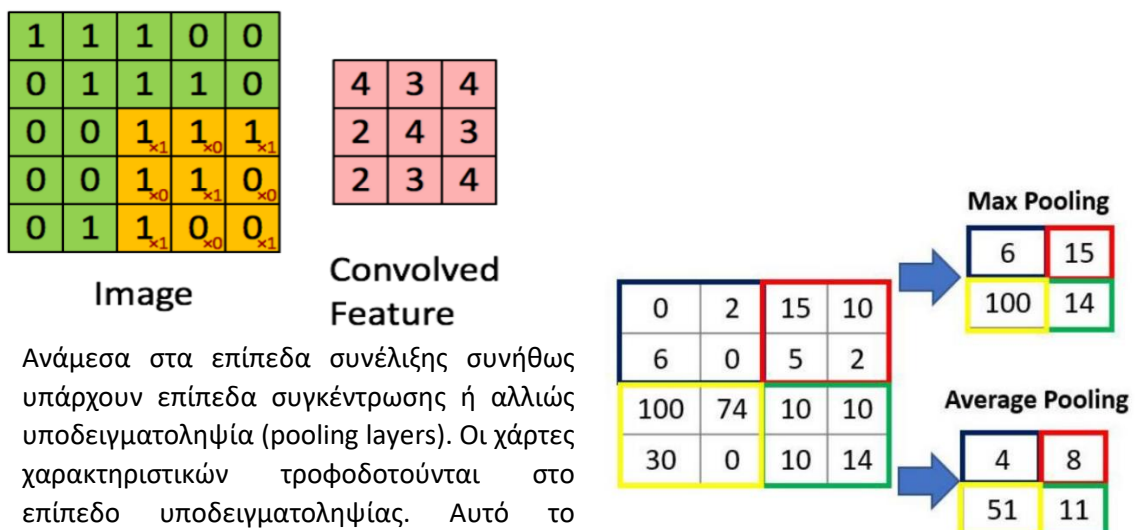
Input Image



Στην παρακάτω εικόνα η κόκκινη περιοχή περιγράφει την περιοχή του τρέχοντος φίλτρου που υπολογίζεται και ονομάζεται πεδίο υποδοχής (receptive field). Οι αριθμοί στο φίλτρο ονομάζονται βάρη ή παράμετροι. Το φίλτρο ολισθαίνει σε ολόκληρη την εικόνα και υπολογίζει τους πολλαπλασιασμούς στοιχείο προς στοιχείο και γεμίζει τον πίνακα εξόδου, ο οποίος ονομάζεται χάρτης ενεργοποίησης ή χάρτης χαρακτηριστικών (feature map).



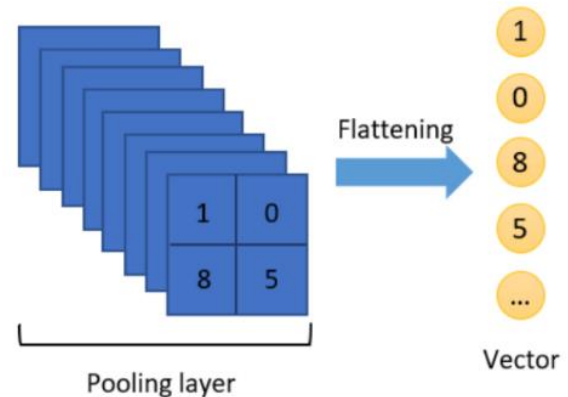
Στην παρακάτω εικόνα μπορούμε να δούμε άλλο ένα παράδειγμα συνέλιξης.



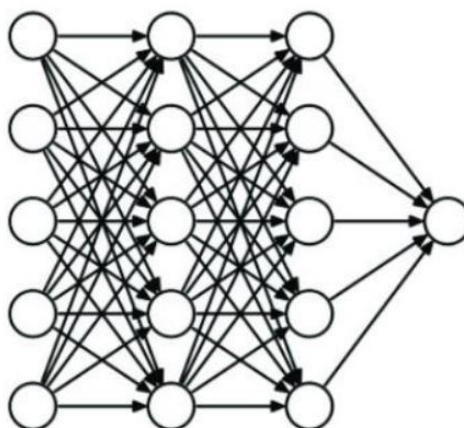
Ανάμεσα στα επίπεδα συνέλιξης συνήθως υπάρχουν επίπεδα συγκέντρωσης ή αλλιώς υποδειγματοληψία (pooling layers). Οι χάρτες χαρακτηριστικών τροφοδοτούνται στο επίπεδο υποδειγματοληψίας. Αυτό το

επίπεδο μειώνει προοδευτικά το χωρικό μέγεθος της αναπαράστασης. Μειώνει τον αριθμό των παραμέτρων και την ποσότητα υπολογισμού στο δίκτυο. Το στρώμα συγκέντρωσης ελέγχει επίσης την υπερπροσαρμογή (overfitting). Υπάρχουν δύο κύριοι τύποι επιπέδων υποδειγματοληψίας: η υποδειγματοληψία μέγιστης τιμής (max pooling) επιστρέφει τη μέγιστη τιμή από το συγκεκριμένο τμήμα και η υποδειγματοληψία μέσης τιμής (average pooling) επιστρέφει τον μέσο όρο όλων των τιμών από το συγκεκριμένο τμήμα.

Έπειτα από αλληπάλληλα επίπεδα συνέλιξης και υποδειγματοληψίας και εφόσον έχουν εξαχθεί τα χαρακτηριστικά υψηλού επιπέδου από την εικόνα εισόδου, μπορούμε να χρησιμοποιήσουμε ένα πλήρες συνδεδεμένο επίπεδο (fully connected layer/dense layer, FCL) για να συσχετίσουμε τον τελικό χάρτη χαρακτηριστικών με την κατηγορία στην οποία αντιστοιχεί η εικόνα. Ωστόσο, πριν τροφοδοτήσουμε τον τελικό χάρτη χαρακτηριστικών στο πλήρες συνδεδεμένο επίπεδο, χρειαζόμαστε ένα ενδιάμεσο επίπεδο για να μετατρέψουμε τη διάσταση των δεδομένων σε 1xη, κατάλληλη δηλαδή για ταξινόμηση με το πλήρες συνδεδεμένο επίπεδο. Αυτό το ενδιάμεσο επίπεδο ονομάζεται επίπεδο ισοπέδωσης (Flatten) και μετατρέπει τους πολυδιάστατους πίνακες σε ένα ενιαίο μεγάλο συνεχές γραμμικό διάνυσμα.

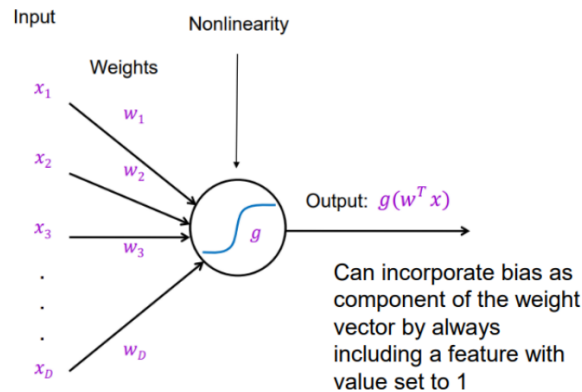


Σε ένα πλήρες συνδεδεμένο επίπεδο κάθε νευρώνας λαμβάνει είσοδο από όλους τους νευρώνες του προηγούμενου επιπέδου. Το επίπεδο έχει έναν πίνακα βάρους W (Weight matrix), ένα διάνυσμα μεροληψίας b (bias vector) και την ενεργοποίηση του προηγούμενου στρώματος a (activation function). Παρακάτω φαίνεται ένα πλήρες συνδεδεμένο νευρωνικό δίκτυο (Fully Connected/Dense Neural Network).

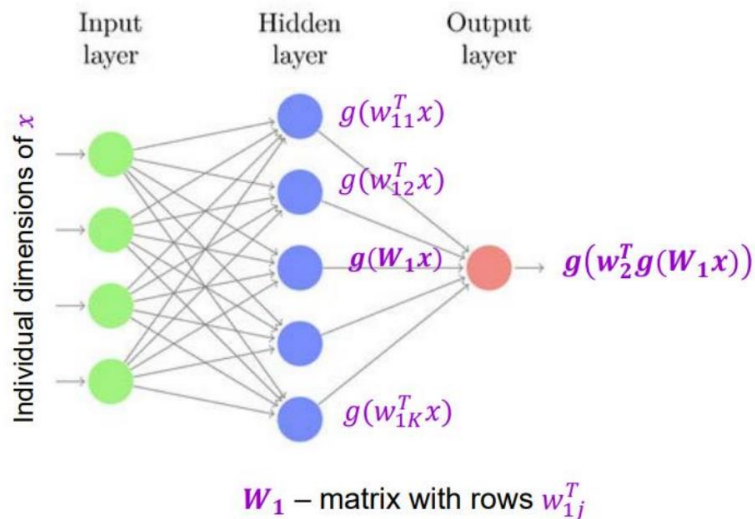


Το πλήρες συνδεδεμένο επίπεδο υλοποιεί τη λειτουργία:

$$\text{output} = \text{activation}(\text{dot}(\text{input}, \text{weight}) + \text{bias})$$



Στην παρακάτω εικόνα φαίνεται ο υπολογισμός για ένα ολόκληρο επίπεδο του δικτύου.

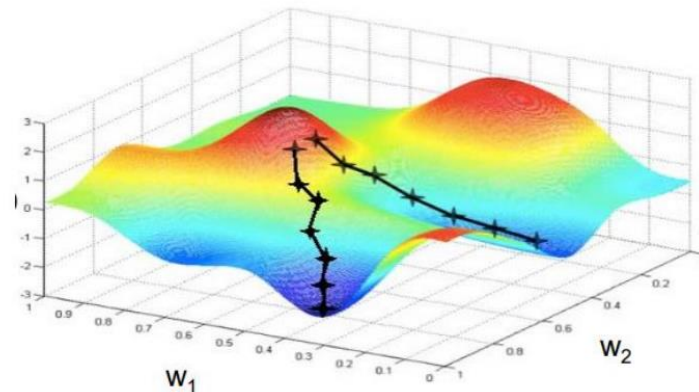


Τα βάρη και το σταθερός όρος b , δηλαδή τις παραμέτρους του δικτύου τις μαθαίνει το δίκτυο με τη μέθοδο της εκπαίδευσης (training). Αρχικά υπολογίζεται η έξοδος του δικτύου (προς τα εμπρός διάδοση, forward propagation) με τυχαίες τιμές W και b . Στη συνέχεια, εφόσον οι ετικέτες (labels) είναι γνωστές (supervised learning), συγκρίνουμε την έξοδο που υπολογίσαμε με την πραγματική έξοδο του δικτύου και υπολογίζουμε το σφάλμα (error), με τη χρήση μιας συνάρτησης κόστους (cost function). Τέλος, γίνεται διόρθωση των W και b με την τεχνική της οπισθοδιάδοσης (backpropagation) με στόχο την ελαχιστοποίηση του κόστους, δηλαδή του σφάλματος. Η επικαιροποίηση των W και b γίνεται συνήθως με τον αλγόριθμο κατάβαση/κάθοδος κλίσης (gradient descent).

Συνάρτηση κόστους:
$$E(\mathbf{w}) = \sum_i l(\mathbf{x}_i, y_i; \mathbf{w})$$

Επικαιροποίηση βαρών:
$$\mathbf{w} \leftarrow \mathbf{w} - \alpha \frac{\partial E}{\partial \mathbf{w}}$$

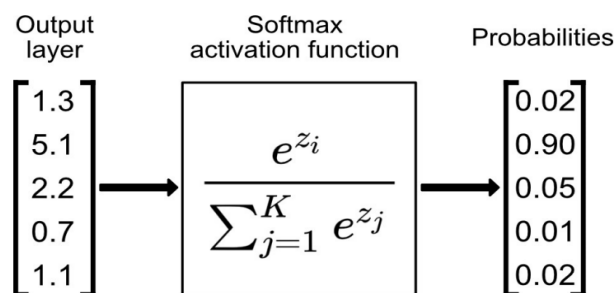
Στην παρακάτω εικόνα φαίνεται ένα παράδειγμα του αλγορίθμου gradient descent για την επικαιροποίηση δύο παραμέτρων W_1 , W_2 .



Όπως αναφέραμε παραπάνω η χρήση πλήρως συνδεδεμένου επιπέδου γίνεται για τον υπολογισμό της κατηγορίας στην οποία ανήκει η εικόνα εισόδου, δηλαδή για την ταξινόμηση της εικόνας. Για να επιτύχουμε την ταξινόμηση, όταν έχουμε πολλές κλάσεις, χρησιμοποιούμε σαν συνάρτηση ενεργοποίησης του πλήρους συνδεδεμένου επιπέδου, τη softmax. Με τη συνάρτηση αυτή για n κλάσεις, λαμβάνουμε n πιθανότητες. Η κλάση με την υψηλότερη πιθανότητα, αντιστοιχεί στην κλάση που έχουμε προβλέψει ότι ανήκει η εικόνα εισόδου. Αν ο ταξινομητής μας είναι ο βέλτιστος, τότε η πιθανότητα μιας κλάσης θα είναι κοντά στο 1 και οι υπόλοιπες πιθανότητες θα είναι κοντά στο 0.

$$\text{softmax}(f_1, \dots, f_c) = \left(\frac{\exp(f_1)}{\sum_j \exp(f_j)}, \dots, \frac{\exp(f_c)}{\sum_j \exp(f_j)} \right)$$

Παρακάτω φαίνεται η μετατροπή των τιμών εξόδου του δικτύου σε πιθανότητες, μέσω της συνάρτησης ενεργοποίησης. Έχουμε 5 κλάσεις και η εικόνα εισόδου με βάση την μεγαλύτερη πιθανότητα, ανήκει στην κλάση 2 (πιθανότητα 0.90).



Σαν συνάρτηση κόστους χρησιμοποιείται η cross-entropy.

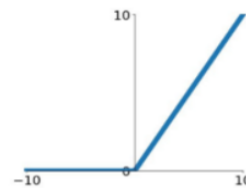
$$\begin{aligned}
 & \text{Indicator variable} \quad \text{Prob of class } j \\
 & - \sum_{j=1}^M y_j \log(p(y_j)) \\
 & \quad \quad \quad \text{Sum over classes} \\
 & \text{Sum over trials} \\
 & - \sum_{i=1}^N y_i \log(p(y_i)) + (1 - y_i) \log(1 - p(y_i)) \\
 & \quad \quad \quad \text{Label} \quad \text{Prob of positive class} \quad \text{Label} \quad \text{Prob of positive class}
 \end{aligned}$$

Τεχνικές Βελτίωσης

Συνάρτηση ενεργοποίησης (Activation function)

Η συνάρτηση ενεργοποίησης που χρησιμοποιείται στα κρυμμένα στρώματα (hidden layers) είναι η Rectified Linear Unit (ReLU).

ReLU
 $\max(0, x)$



Σύνολο δεδομένων επικύρωσης (Validation data set)

Το σύνολο δεδομένων συνήθως αποτελείται από το σύνολο δεδομένων εκπαίδευσης και δοκιμής. Θα ήταν χρήσιμο να χωρίσουμε το σύνολο εκπαίδευσης και να πάρουμε ένα μικρό κομμάτι του για να δημιουργήσουμε το σύνολο δεδομένων επικύρωσης.

Το σύνολο επικύρωσης χρησιμοποιείται για την παροχή μιας αμερόληπτης αξιολόγησης ενός μοντέλου κατά την ρύθμιση των υπερπαραμέτρων του μοντέλου. Το μοντέλο βλέπει περιστασιακά αυτά τα δεδομένα, αλλά δεν μαθαίνει από αυτά. Χρησιμοποιούμε τα αποτελέσματα του συνόλου επικύρωσης για να ενημερώσουμε τις υπερπαραμέτρους του δικτύου.

Επανακλιμάκωση (Rescale)

Η μέγιστη τιμή που μπορεί να πάρει ένα εικονοστοιχείο μιας εικόνας είναι η τιμή 255. Οπότε μπορούμε να διαιρέσουμε κάθε εικονοστοιχείο, κάθε εικόνας εισόδου με το 255, ώστε κάθε τιμή εισόδου να κυμαίνεται στο εύρος 0 έως 1. Αν οι τιμές εισόδου κυμαίνονται στο ίδιο εύρος, έχει αποδειχθεί ότι βελτιώνεται η απόδοση του δικτύου. Με το όρισμα rescale της κλάσης ImageDataGenerator πολλαπλασιάζουμε τα δεδομένα με την τιμή που παρέχεται, δηλαδή με 1/255, άρα στην ουσία διαιρούμε με την τιμή 255.

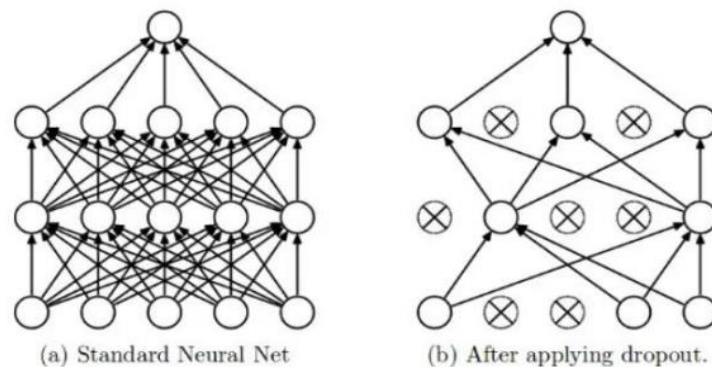
Επαύξηση δεδομένων (Data augmentation)

Η επαύξηση δεδομένων είναι ένα σύνολο τεχνικών για την τεχνητή αύξηση της ποσότητας δεδομένων με τη δημιουργία νέων δεδομένων από τα υπάρχοντα. Αυτό περιλαμβάνει την πραγματοποίηση μικρών αλλαγών στα δεδομένα εκπαίδευσης ή τη χρήση μοντέλων βαθιάς μάθησης για τη δημιουργία νέων σημείων δεδομένων. Εμείς θα εστιάσουμε στην

πραγματοποίηση κάποιων μετασχηματισμών για την επαύξηση του συνόλου δεδομένων μας. Με την εφαρμογή κατάλληλου είδους και ποσότητας μετασχηματισμών στις εικόνες είναι δυνατόν να αυξηθεί η απόδοση του δικτύου.

Dropout

Το Dropout είναι μια μέθοδος, η οποία συμβάλει στην αποφυγή της υπερπροσαρμογής (overfitting). Κατά τη διάρκεια της εκπαίδευσης, ορισμένοι νευρώνες απορρίπτονται τυχαία με κάποια πιθανότητα, σε κάθε εποχή. Όλες οι συνδέσεις προς τα εμπρός και προς τα πίσω με έναν κόμβο που έχει απορριφθεί αφαιρούνται προσωρινά, δημιουργώντας έτσι μια νέα αρχιτεκτονική δικτύου, διαφορετική από του αρχικού. Οι κόμβοι απορρίπτονται με πιθανότητα απόρριψης p .



Στο πρόβλημα υπερπροσαρμογής, το μοντέλο μαθαίνει τον στατιστικό θόρυβο. Για την ακρίβεια, το κύριο κίνητρο της εκπαίδευσης είναι η μείωση της συνάρτησης απώλειας, δεδομένων όλων των μονάδων (νευρώνες). Όμως αν το μοντέλο προσαρμοστεί πάρα πολύ καλά στα δεδομένα εκπαίδευσης, μπορεί να αποτυγχάνει να γενικευτεί στο σύνολο δεδομένων δοκιμής. Εάν χρησιμοποιήσουμε το dropout, διασφαλίζεται ότι το μοντέλο γενικεύεται και συνεπώς μειώνει το πρόβλημα υπερπροσαρμογής.

BatchNormalization

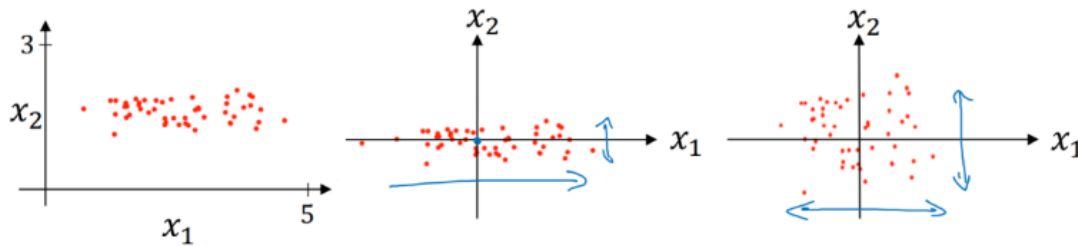
Το batchnormalization είναι μια μέθοδος κανονικοποίησης που συμβάλει στην αποφυγή της υπερπροσαρμογής και στο πρόβλημα των exploding gradients (όταν οι παράγωγοι γίνονται πολύ μεγάλες). Οι παράμετροι εκπαίδευσης είναι η μέση τιμή (mean a , c) και η τυπική απόκλιση (std b , d), των κατανομών εισόδου και προβολής. Αρχικά από τις τιμές εισόδου, αφαιρείται το mean και διαιρείται το αποτέλεσμα με το std, για να φέρουμε τα δεδομένα στο ίδιο εύρος τιμών.

$$\text{Normalization Operation: } y_1 = \frac{x-a}{b}$$

Στη συνέχεια, για να προβάσουμε τα δεδομένα, πολλαπλασιάζουμε με το std και προσθέτουμε το mean.

$$\text{Projection Operation } y = y_1 * d + c$$

Αυτή η νέα προβολή έχει αποδειχτεί ότι συμβάλει στην αύξηση της απόδοσης του δικτύου. Ένα παράδειγμα φαίνεται στην παρακάτω εικόνα.



Πρώιμη διακοπή (Early stopping)

Κατά την εκπαίδευση ενός δικτύου, υπάρχει ένα σημείο κατά τη διάρκεια της εκπαίδευσης που το μοντέλο θα σταματήσει να γενικεύει και θα αρχίσει να μαθαίνει τον στατιστικό θόρυβο στο σύνολο δεδομένων εκπαίδευσης. Αυτή η υπερπροσαρμογή (overfitting) του συνόλου δεδομένων εκπαίδευσης θα οδηγήσει σε σφάλματα κατά τη γενίκευση, καθιστώντας το μοντέλο λιγότερο χρήσιμο για την πραγματοποίηση προβλέψεων σε νέα δεδομένα. Η πρόκληση είναι να εκπαιδεύσουμε το δίκτυο για αρκετές εποχές, αλλά όχι τόσο πολύ, ώστε να υπερπροσαρμοστεί στα δεδομένα εκπαίδευσης. Μία εποχή είναι ένα πλήρες πέρασμα των δεδομένων από το δίκτυο.

Μια προσέγγιση είναι να εκπαιδεύσουμε το μοντέλο μία φορά για μεγάλο αριθμό εποχών. Κατά τη διάρκεια της εκπαίδευσης, το μοντέλο αξιολογείται σε ένα σύνολο δεδομένων επικύρωσης, μετά από κάθε εποχή. Εάν η απόδοση του μοντέλου στο σύνολο δεδομένων επικύρωσης αρχίσει να υποβαθμίζεται (π.χ. η απώλεια αρχίζει να αυξάνεται ή η ακρίβεια αρχίζει να μειώνεται), τότε η διαδικασία εκπαίδευσης διακόπτεται. Στη συνέχεια χρησιμοποιείται το μοντέλο από τη στιγμή που διακόπτεται η εκπαίδευση και είναι γνωστό ότι επιτυγχάνει τη γενίκευση. Αυτή η διαδικασία ονομάζεται πρώιμη διακοπή (early stopping).

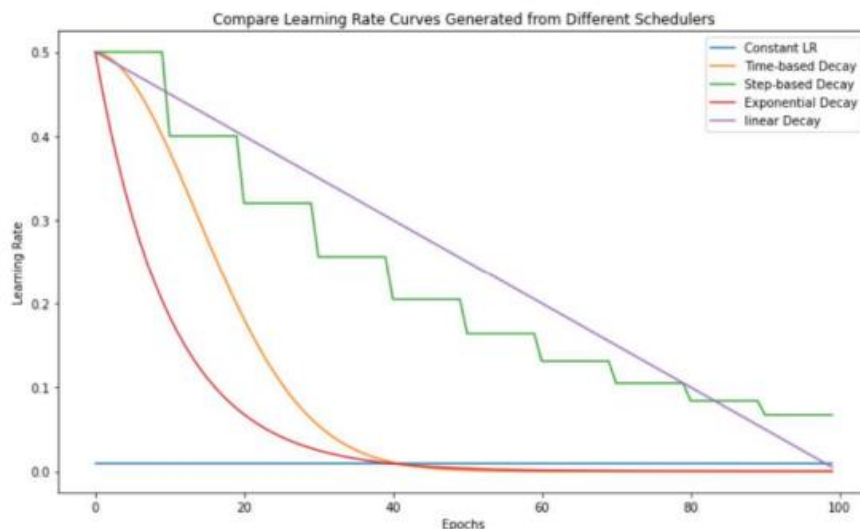
Checkpoints

Ένας τρόπος να αποθηκεύσουμε τα βάρη του μοντέλου μας, όταν έχουμε ρυθμίσει κατάλληλα τις υπερπαραμέτρους είναι με τη χρήση σημείων ελέγχου (checkpoints). Στο τέλος κάθε εποχής, αν το σφάλμα επικύρωσης είναι μικρότερο από πριν, αποθηκεύεται ένα checkpoint στο φάκελο checkpoint_filepath. Μόλις ολοκληρωθεί η εκπαίδευση, έχουμε κρατήσει το μοντέλο με την καλύτερη απόδοση. Οπότε, μπορούμε χωρίς να επαναλάβουμε ολόκληρη τη διαδικασία της εκπαίδευσης να φορτώσουμε το μοντέλο από το φάκελο και να το χρησιμοποιήσουμε για προβλέψεις.

Χρονοδιαγράμματα ρυθμού εκμάθησης (Learning rate schedulers)

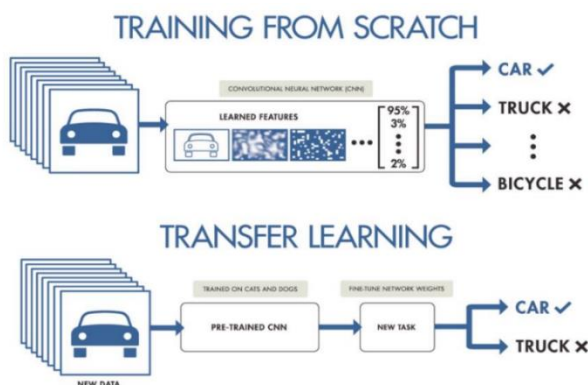
Κατά την εκπαίδευση, είναι συχνά χρήσιμο να μειώνεται ο ρυθμός εκμάθησης καθώς προχωρά η εκπαίδευση. Αυτό μπορεί να γίνει χρησιμοποιώντας προκαθορισμένα χρονοδιαγράμματα ρυθμού εκμάθησης (learning rate schedules) ή προσαρμοστικές μεθόδους ρυθμού εκμάθησης (adaptive learning rate methods). Τα χρονοδιαγράμματα ρυθμού εκμάθησης επιδιώκουν να προσαρμόσουν τον ρυθμό εκμάθησης (learning rate) Στην αρχή της εκπαίδευσης, ο ρυθμός εκμάθησης έχει οριστεί να είναι μεγάλος, προκειμένου να φτάσει σε ένα σύνολο βαρών που δίνουν σχετικά καλή απόδοση. Με την πάροδο του χρόνου, αυτά τα βάρη προσαρμόζονται ώστε να επιτυγχάνουν υψηλότερη ακρίβεια αξιοποιώντας ένα μικρό ποσοστό εκμάθησης. Παρακάτω φαίνονται ορισμένοι ρυθμοί εκμάθησης με

χρονοδιαγράμματα, κατά τη διάρκεια της εκπαίδευσης. Μια πολύ γνωστή κλάση που χρησιμοποιείται για να επιβάλλει στο ρυθμό εκμάθησης ένα χρονοδιάγραμμα είναι η ReduceLROnPlateau. Ανήκει στην κατηγορία step-based decay. Αυτό σημαίνει ότι ο ρυθμός εκμάθησης είναι σταθερός για ένα συγκεκριμένο αριθμό εποχών και στη συνέχεια μειώνεται σε κάποια άλλη τιμή για έναν ακόμα αριθμό εποχών. Στο διάγραμμα φαίνεται σαν μία σκάλα, δηλαδή κάθε φορά που μειώνεται ο ρυθμός εκμάθησης δημιουργείται ένα σκαλοπάτι.



4. Χρήση Προ-εκπαιδευμένων Νευρωνικών Δικτύων

Το Keras περιέχει προεκπαιδευμένα μοντέλα ταξινόμησης εικόνων τα οποία εκπαιδεύονται σε δεδομένα του Imagenet. Το Imagenet είναι μια μεγάλη συλλογή εικόνων που περιέχει 1000 κατηγορίες. Αυτά τα προεκπαιδευμένα μοντέλα είναι σε θέση να ταξινομήσουν οποιαδήποτε εικόνα εμπίπτει σε αυτές τις 1000 κατηγορίες. Το προεκπαιδευμένο δίκτυο περιέχει συνελκτικά επίπεδα που ακολουθούνται από πλήρως συνδεδεμένα επίπεδα. Τα επίπεδα συνέλιξης εξάγουν χαρακτηριστικά από την εικόνα και τα πλήρως συνδεδεμένα επίπεδα ταξινομούν την εικόνα χρησιμοποιώντας τα χαρακτηριστικά που έχουν εξαχθεί. Όταν εκπαιδεύουμε ένα CNN σε δεδομένα εικόνων, φαίνεται ότι τα ανώτερα στρώματα του δικτύου μαθαίνουν να εξάγουν γενικά χαρακτηριστικά από εικόνες όπως άκρες, κατανομή



χρωμάτων κ.λπ. Καθώς συνεχίζουμε να πηγαίνουμε βαθιά στο δίκτυο, τα επίπεδα τείνουν να εξάγουν περισσότερα εξειδικευμένα χαρακτηριστικά. Οπότε, μπορούμε να χρησιμοποιήσουμε αυτά τα προεκπαιδευμένα μοντέλα που ήδη γνωρίζουν πώς να εξάγουν χαρακτηριστικά και να αποφύγουμε την εκπαίδευση από την αρχή. Αυτή η έννοια είναι γνωστή ως μεταφερόμενη μάθηση (transfer learning).

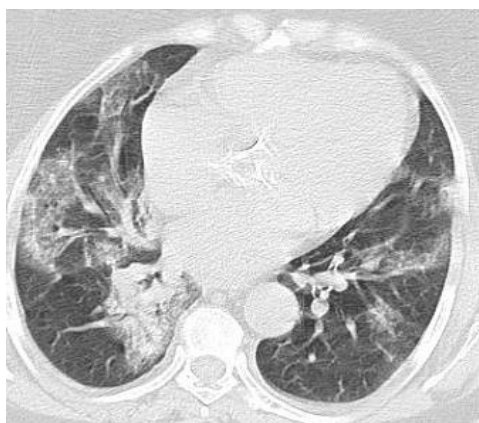
Πριν εκπαιδεύσουμε το δίκτυο, μπορούμε να παγώσουμε (freeze) μερικά από τα επίπεδά του. Με τον όρο freeze εννοούμε ότι τα βάρη του δεν ενημερώνονται κατά τη διαδικασία της εκπαίδευσης. Συχνά είναι χρήσιμο να παγώνουμε τα αρχικά επίπεδα που συμβάλουν στην εξαγωγή γενικών χαρακτηριστικών και να εκπαιδεύουμε μόνο τα τελευταία επίπεδα που συμβάλουν στην εξαγωγή ειδικών χαρακτηριστικών και στη διαδικασία ταξινόμησης.

5.Τεχνικές Ψηφιακής επεξεργασίας εικόνων

Εκτός από τα παραπάνω, μπορούν να χρησιμοποιηθούν και μέθοδοι όρασης υπολογιστών στις εικόνες από τα datasets για τον εντοπισμό σημαντικών χαρακτηριστικών, που με βάση αυτά μπορεί να πραγματοποιηθεί ταξινόμηση των ασθενειών.

Πιο συγκεκριμένα, έγινε χρήση μιας από αυτές τις τεχνικές για το σύνολο των δεδομένων από την αξονική τομογραφία στους πνεύμονες από υγιείς ανθρώπους και από αυτούς που πάσχουν από κορονοϊό. Η τεχνική αυτή ονομάζεται image thresholding και σύμφωνα με αυτή ορίζεται ένα όριο (threshold), με σκοπό κάθε pixel που έχει τιμή κάτω από το όριο αυτό να παίρνει την τιμή 0 (μαύρο χρώμα), διαφορετικά να παίρνει την τιμή 255 (άσπρο). Έτσι, η εικόνα μετατρέπεται σε δυαδική.

Ο λόγος που επιλέχθηκε να εφαρμοστεί αυτή η τεχνική είναι επειδή παρατηρήθηκε ότι οι πνεύμονες των ανθρώπων που έχουν κορονοϊό, εμφανίζουν λόγω μεγάλης συγκέντρωσης υγρού, περισσότερες ανοιχτόχρωμες περιοχές στους πνεύμονες σε σχέση με τους υγιείς. Άρα, θα πρέπει ο αριθμός των μαύρων pixels στα υγιή άτομα να είναι μεγαλύτερος σε σχέση με τους φορείς του ιού.



Ασθενής με COVID



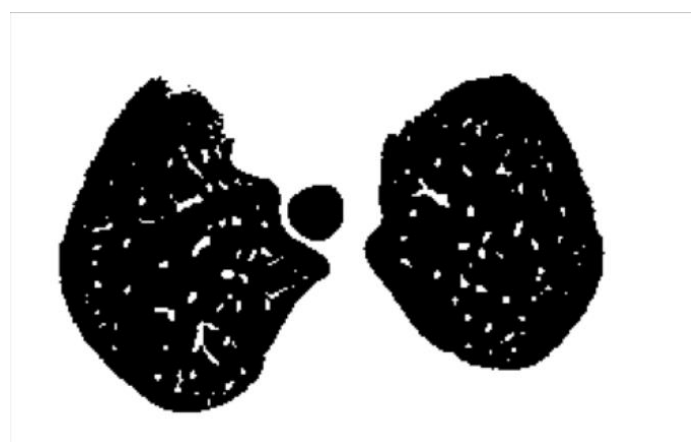
Υγιής

Μετά την εφαρμογή του threshold = 120 για δύο εικόνες προκύπτουν τα παρακάτω:



Ασθενής με COVID

black pixels covid = 56301



Υγιής

black pixels no covid = 64104

Άρα όντως στο συγκεκριμένο παράδειγμα επιβεβαιώνεται η υπόθεση.

Η υπόλοιπη υλοποίηση του κώδικα αποτελεί την εύρεση της μέσης τιμής των μαύρων pixels για το 70% των εικόνων του Covid και αντίστοιχα του No-Covid. Οι υπόλοιπες εικόνες χρησιμοποιήθηκαν για τη δοκιμή και τον έλεγχο της ακρίβειας της μεθόδου. Η ταξινόμηση έγινε ανάλογα με το σε ποια από τις δύο μέσες τιμές ήταν πλησιέστερα ο αριθμός των μαύρων pixels της κάθε εικόνας από το σύνολο των δοκιμών.

Το αποτέλεσμα της ακρίβειας είναι **65.5%**, νούμερο σχετικά υψηλό για την απλότητα της μεθόδου.

Παρακάτω θα φανεί υλοποίηση ταξινόμησης με συνελκτικό νευρωνικό δίκτυο, όπου τα αποτελέσματα είναι πολύ καλύτερα.

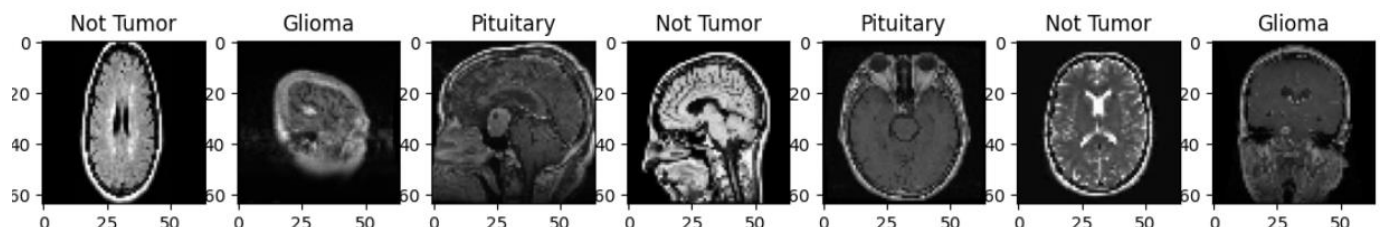
6.Υλοποίηση σε διαθέσιμα dataset

Στη συνέχεια για να επιτύχουμε καλύτερη πρόβλεψη υλοποιήσαμε μεθόδους συνελκτικών νευρωνικών δικτύων σε δύο διαθέσιμα dataset.

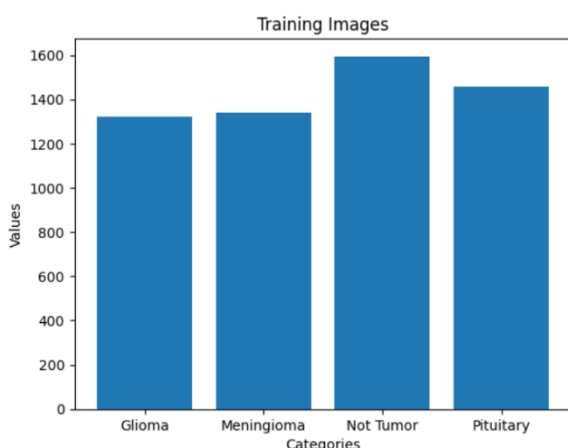
Ταξινόμηση εικόνων μαγνητικής τομογραφίας (MRI) για διάφορα είδη καρκίνου στον εγκέφαλο

Χρησιμοποιήθηκε μια διαθέσιμη βάση δεδομένων που περιέχει 7023 εικόνες, από μαγνητική τομογραφία (MRI) που απεικονίζουν τον ανθρώπινο εγκέφαλο. Οι εικόνες ταξινομούνται σε 4 κατηγορίες: γλοίωμα (glioma) – μηνιγγίωμα (meningioma) - χωρίς όγκο (no tumor) και υπόφυση (pituitary).

Παρακάτω φαίνονται ενδεικτικά κάποιες εικόνες με τις ετικέτες τους.



Στο παρακάτω διάγραμμα μπορούμε να δούμε πως μοιράζονται οι εικόνες σε κάθε κατηγορία. Ιδανικά θα θέλαμε να έχουμε ίδιο αριθμό εικόνων για κάθε κλάση.



Για να φορτώσουμε τις εικόνες και να κάνουμε rescale χρησιμοποιήθηκαν data generators με τη βοήθεια του keras της βιβλιοθήκης tensorflow. Το batch size επιλέξαμε να είναι 32 και εκπαιδεύσαμε για 50 εποχές. Οι εικόνες φορτώνονται σε μέγεθος 64x64x3. Το 3 σημαίνει ότι τις φορτώνουμε έγχρωμες (RGB, 3 channels). Το dataset είναι ήδη χωρισμένο σε εικόνες εκπαίδευσης και εικόνες δοκιμής. Χωρίσαμε τις εικόνες εκπαίδευσης περαιτέρω, σε 20% εικόνες δοκιμής και 80% εικόνες εκπαίδευσης.

Επίσης, χρησιμοποιήθηκαν οι callback συναρτήσεις early stopping, checkpoints και reduce_lr_scheduler για τους λόγους που αναφέραμε και πιο πάνω στο θεωρητικό κομμάτι. Με βάση το validation loss, αν δεν υπάρξει βελτίωση για 15 εποχές η εκπαίδευση σταματάει και αποκαθίστανται τα βάρη από την καλύτερη εποχή. Ο ρυθμός μάθησης ανά 12 εποχές που δεν υπάρχει βελτίωση μειώνεται στο μισό.

Το μοντέλο περιλαμβάνει **τρία επίπεδα Συνέλιξης- BatchNormalization- MaxPooling2D** που **επαναλαμβάνονται τρεις φορές**. Σε κάθε επίπεδο συνέλιξης ο αριθμός των φίλτρων διπλασιάζεται (από 64, σε 128 και 256). Το kernel που χρησιμοποιήθηκε είναι 7x7. Η συνάρτηση ενεργοποίησης είναι η relu. Στη συνέχεια ακολουθεί **ένα επίπεδο flatten, ένα επίπεδο Dropout (50%) και πέντε επίπεδα Dense (256 νευρώνες) - Dropout (20%) - Dense (128 νευρώνες) - Dropout (20%) - Dense (64 νευρώνες)**. Το **τελικό επίπεδο** αποτελείται από 4 νευρώνες (όσες και οι κλάσεις) και συνάρτηση ενεργοποίησης τη softmax (για κατηγοριοποίηση).

Τέλος, το μοντέλο γίνεται compile, χρησιμοποιείται ο βελτιστοποιητής Adam και για συνάρτηση κόστους η CategoricalCrossentropy.

Παρακάτω φαίνεται ενδεικτικά ο κώδικας με όσα αναφέραμε.

```
rescale = 1./255

train_image_generator = ImageDataGenerator(rescale=rescale,
                                           validation split=0.2)

test_image_generator = ImageDataGenerator(rescale=rescale)

train_data_gen = train_image_generator.flow_from_directory(
    directory=train_path,
    target_size=(IMG_HEIGHT, IMG_WIDTH),
    batch_size=batch_size,
    class_mode='categorical',
    subset='training', seed=6)

val_data_gen = train_image_generator.flow_from_directory(
    directory=train_path,
    target_size=(IMG_HEIGHT, IMG_WIDTH),
    batch_size=batch_size,
    class_mode='categorical',
    subset='validation', seed=6)

test_data_gen = test_image_generator.flow_from_directory(
    directory=test_path,
    target_size=(IMG_HEIGHT, IMG_WIDTH),
    batch_size=batch_size,
    class_mode='categorical',
    shuffle=False)
```

```

from keras.callbacks import EarlyStopping
from keras.callbacks import ModelCheckpoint

# Early stopping
early_stopping = EarlyStopping(monitor='val_loss', mode='min', restore_best_weights=True, patience=15, verbose=1)

# Checkpoints
checkpoint_filepath = '/content/drive/MyDrive/classification_of_diseases/brain_tumor_model.hdf5'

model_checkpoint = ModelCheckpoint(filepath=checkpoint_filepath, monitor='val_loss', mode='min', save_best_only=True, verbose=1)

# Scheduler
reduce_lr = tf.keras.callbacks.ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=12, min_lr=1e-6)

model = tf.keras.Sequential([
    tf.keras.layers.Conv2D(64, (7, 7), activation='relu', padding='same', input_shape=(IMG_HEIGHT, IMG_WIDTH, IMG_COLOR)),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPooling2D((2, 2)),

    tf.keras.layers.Conv2D(128, (7, 7), activation='relu', padding='same'),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPooling2D((2, 2)),

    tf.keras.layers.Conv2D(256, (7, 7), activation='relu', padding='same'),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPooling2D((2, 2)),

    tf.keras.layers.Flatten(),
    tf.keras.layers.Dropout(0.5),

    tf.keras.layers.Dense(256, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(num_of_classes, activation='softmax')
])

model.compile(optimizer=tf.keras.optimizers.Adam(),
              loss=tf.keras.losses.CategoricalCrossentropy(),
              metrics=['accuracy'])

```

Σύνολο έχουμε 17 επίπεδα. Ένα input layer, ένα output layer και 15 hidden layers.

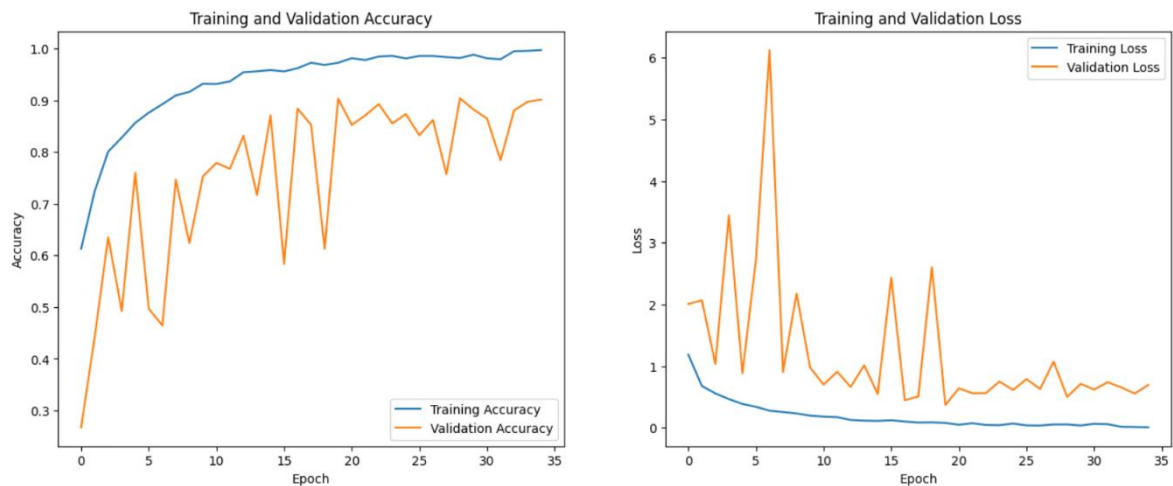
Προχωράμε στην εκπαίδευση του δικτύου η οποία στις 35 εποχές σταματάει και τα βάρη αποκαθίστανται από την εποχή 20.

```
history = model.fit(x=train_data_gen,
                    validation_data=val_data_gen,
                    epochs=epochs, batch_size=batch_size, shuffle=True, callbacks=[early stopping, model checkpoint, reduce_lr], verbose=1)
```

```
Epoch 19: val_loss did not improve from 0.44846
143/143 [=====] - 15s 105ms/step - loss: 0.0892 - accuracy: 0.9687 - val_loss: 2.6031 - val_accuracy: 0.6126 - lr: 0.0010
Epoch 20/50
143/143 [=====] - ETA: 0s - loss: 0.0797 - accuracy: 0.9731
Epoch 20: val_loss improved from 0.44846 to 0.37163, saving model to /content/drive/MyDrive/classification_of diseases/brain_tumor_model.hdf5
143/143 [=====] - 16s 114ms/step - loss: 0.0797 - accuracy: 0.9731 - val_loss: 0.3716 - val_accuracy: 0.9036 - lr: 0.0010
Epoch 21/50
143/143 [=====] - ETA: 0s - loss: 0.0511 - accuracy: 0.9818
Epoch 21: val_loss did not improve from 0.37163
143/143 [=====] - 17s 121ms/step - loss: 0.0511 - accuracy: 0.9818 - val_loss: 0.6416 - val_accuracy: 0.8528 - lr: 0.0010
```

```
Epoch 35: val_loss did not improve from 0.37163
143/143 [=====] - 15s 106ms/step - loss: 0.0091 - accuracy: 0.9976 - val_loss: 0.6963 - val_accuracy: 0.9018 - lr: 5.0000e-04
Epoch 35: early stopping
```

Τα διαγράμματα για την ακρίβεια και την απώλεια φαίνονται παρακάτω.



Φαίνεται ότι ενώ η απώλεια στο training set μειώνεται σταθερά στο validation set κάνει πολλά σκαμπανεβάσματα. Έχουμε πολύ μεγάλη διακύμανση, συνολικά όμως υπάρχει καθοδική πορεία.

```
[ ] train_loss, train_acc = model.evaluate(train_data_gen, verbose=2)
print("Train accuracy: " + "{:.2f}".format(train_acc*100) + "%")

143/143 - 10s - loss: 0.0409 - accuracy: 0.9884 - 10s/epoch - 72ms/step
Train accuracy: 98.84%
```

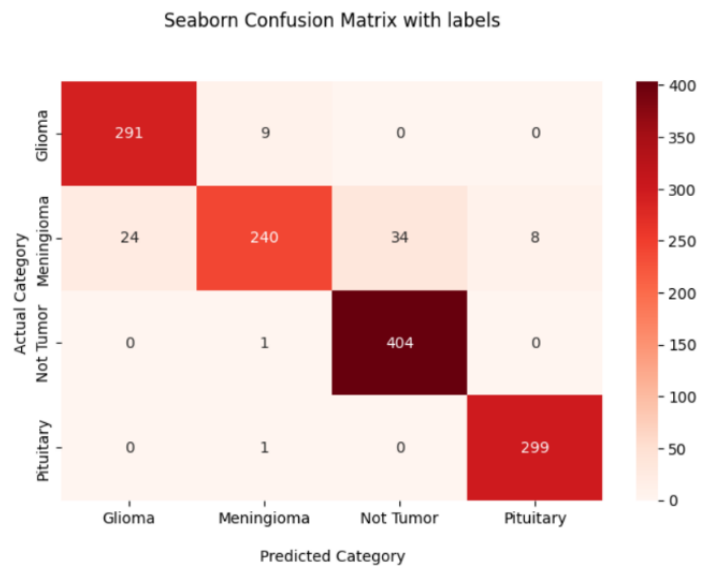
```
[ ] test_loss, test_acc = model.evaluate(test_data_gen, verbose=2)
print("Test accuracy: " + "{:.2f}".format(test_acc*100) + "%")

41/41 - 4s - loss: 0.1869 - accuracy: 0.9413 - 4s/epoch - 90ms/step
Test accuracy: 94.13%
```

Η ακρίβεια στο σύνολο εκπαίδευσης είναι πάρα πολύ καλή, **98.84%**, και στο σύνολο δοκιμής είναι **94.13%**. Η απόκλιση οφείλεται σε υπερπροσαρμογή (overfitting), την οποία δεν καταφέραμε να εξαλείψουμε εντελώς με τις μεθόδους batchnormalization και dropout.

Παρακάτω φαίνονται κάποιες άλλες μετρικές για την αξιολόγηση του μοντέλου, όπως είναι ο πίνακας σύγχυσης (confusion matrix) και το classification report.

Από τον πίνακα σύγχυσης μπορούμε να δούμε ποιες κλάσεις κάναμε λάθος στην πρόβλεψη. Στον κατακόρυφο άξονα έχουμε τις πραγματικές κλάσεις που ανήκει η κάθε εικόνα και στον οριζόντιο άξονα τις προβλέψεις μας. Θέλουμε ιδανικά όλες οι εικόνες να βρίσκονται πάνω στη διαγώνιο. Τα νούμερα που είναι διαφορετικά από μηδέν και δε βρίσκονται πάνω στη διαγώνιο είναι εικόνες που δεν ταξινομήθηκαν σωστά.



Το classification report χρησιμοποιεί τρεις διαφορετικές μετρικές απόδοσης: το precision (ακρίβεια), το recall (ανάκληση), και το f1-score.

Accuracy: Αποτελεί μια βασική μετρική για την αξιολόγηση ενός μοντέλου και ορίζεται ως το ποσοστό των προβλέψεων που είναι σωστές.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{\text{Σωστές Προβλέψεις}}{\text{Όλες οι Προβλέψεις}}$$

Η συγκεκριμένη μετρική ωστόσο, έχει νόημα να χρησιμοποιηθεί όταν το σύνολο δεδομένων είναι ισορροπημένο (balanced).

Precision: Εκφράζει το πηλίκο των σωστών προβλέψεων για μία κατηγορία (TP) προς το σύνολο των παραδειγμάτων τα οποία προέβλεψε το μοντέλο ότι ανήκουν στη κατηγορία αυτή (TP+FP).

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall/Sensitivity: Εκφράζει το πηλίκο των σωστών προβλέψεων για μία κατηγορία (TP) προς το σύνολο των παραδειγμάτων που ανήκουν στη κατηγορία αυτή (TP+FN).

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1-score: Αποτελεί τον αρμονικό μέσο όρο του Precision και του Recall και δίνει συνολικότερη εικόνα για την απόδοση του μοντέλου.

$$F_1 = \frac{2}{\text{Precision}^{-1} + \text{Sensitivity}^{-1}}$$

Παρακάτω φαίνεται το classification report για το μοντέλο μας. Εφόσον, το dataset είναι balanced μπορούμε να χρησιμοποιήσουμε το accuracy σαν μετρική αξιολόγησης του μοντέλου.

```
[ ] from sklearn.metrics import classification_report
report = classification_report(y_test, predicted_classes)
print(report)
```

	precision	recall	f1-score	support
0	0.92	0.97	0.95	300
1	0.96	0.78	0.86	306
2	0.92	1.00	0.96	405
3	0.97	1.00	0.99	300
accuracy			0.94	1311
macro avg	0.94	0.94	0.94	1311
weighted avg	0.94	0.94	0.94	1311

Επιπλέον, για να επιτύχουμε κάποιο καλύτερο αποτέλεσμα, δοκιμάσαμε να χρησιμοποιήσουμε ένα προεκπαιδευμένο μοντέλο, το ResNet50.

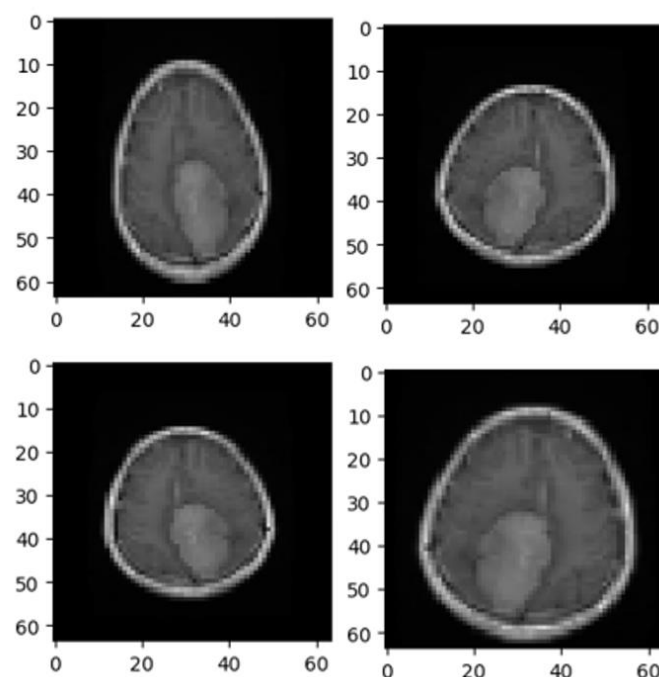
Εδώ πλέον εκτός από το rescale των εικόνων, χρησιμοποιήσαμε και κάποια data augmentations για καλύτερα αποτελέσματα (ζουμ, εύρος διάτμησης, οριζόντια ανατροπή).

```
rescale = 1./255

train_image_generator = ImageDataGenerator(rescale=rescale,
                                           shear_range=0.2,
                                           zoom_range=0.2,
                                           horizontal_flip=True,
                                           validation_split=0.2)

test_image_generator = ImageDataGenerator(rescale=rescale)
```

Ένα παράδειγμα data augmentation φαίνεται παρακάτω.



Αυτή τη φορά εκπαιδεύουμε για 150 εποχές, με patience πρώιμης διακοπής 25. Με τον τρόπο αυτό η εκπαίδευση θα διαρκέσει περισσότερο και ίσως πετύχουμε καλύτερο αποτέλεσμα.

Το resnet50 το φορτώνουμε με την παρακάτω εντολή.

```
base_model = tf.keras.applications.resnet.ResNet50(include_top=False, weights="imagenet", input_shape=(IMG_HEIGHT, IMG_WIDTH, IMG_COLOR))
```

Αποτελείται από 175 επίπεδα.

```
[ ] len(base_model.layers)
175
```

Το resnet50 το χρησιμοποιούμε σαν βάση και στην κορυφή τοποθετούνται 5 επίπεδα, MaxPooling2D-Flatten-Dropout (30%)-Dense (150 νευρώνες)-Dense (4 νευρώνες).

Άρα σύνολο έχουμε $175+5=180$ επίπεδα στο δίκτυό μας.

```
model = tf.keras.Sequential([
    base_model,
    tf.keras.layers.MaxPooling2D(),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dropout(0.3),
    tf.keras.layers.Dense(150, activation='relu'),
    tf.keras.layers.Dense(num_of_classes, activation='softmax')
])

model.compile(optimizer=tf.keras.optimizers.Adam(),
              loss=tf.keras.losses.CategoricalCrossentropy(),
              metrics=['accuracy'])
```

Επιλέξαμε να το εκπαιδεύσουμε ολόκληρο, χωρίς να παγώσουμε κάποια αρχικά επίπεδα.

Στην εποχή 57 σταματάει η εκπαίδευση.

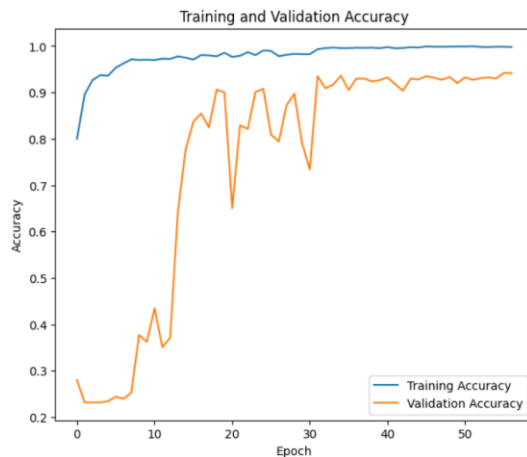
```
[ ] train_loss, train_acc = model.evaluate(train_data_gen, verbose=2)
print("Train accuracy: " + "{:.2f}".format(train_acc*100) + "%")

72/72 - 15s - loss: 0.0230 - accuracy: 0.9919 - 15s/epoch - 215ms/step
Train accuracy: 99.19%
```

```
[ ] test_loss, test_acc = model.evaluate(test_data_gen, verbose=2)
print("Test accuracy: " + "{:.2f}".format(test_acc*100) + "%")

21/21 - 3s - loss: 0.1260 - accuracy: 0.9611 - 3s/epoch - 144ms/step
Test accuracy: 96.11%
```

Τα αποτελέσματα βελτιώθηκαν αρκετά σε σχέση με πριν. Η ακρίβεια στο σύνολο εκπαίδευσης είναι πάρα πολύ καλή, **99.19%**, και στο σύνολο δοκιμής είναι **96.11%**.



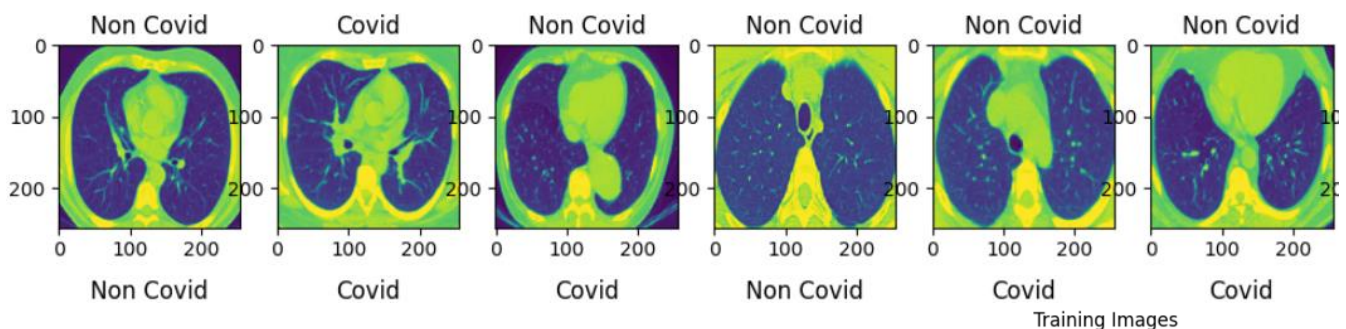
Η διακύμανση μειώθηκε σε σχέση με το προηγούμενο μοντέλο (μη προεκπαιδευμένο).

Διάγνωση Κορονοϊού από εικόνες αξονικής τομογραφίας (CT) του πνεύμονα

Χρησιμοποιήθηκε ένα dataset που περιέχει δεδομένα αξονικής τομογραφίας (σάρωσης CT) SARS-CoV-2, με 1252 αξονικές τομογραφίες που είναι θετικές για λοίμωξη SARS-CoV-2 (COVID-19) και 1230 αξονικές τομογραφίες για ασθενείς που δεν έχουν μολυνθεί από SARS-CoV-2, 2482 αξονικές τομογραφίες συνολικά. Αυτά τα δεδομένα έχουν συλλεχθεί από πραγματικούς ασθενείς σε νοσοκομεία από το Σάο Πάολο της Βραζιλίας.

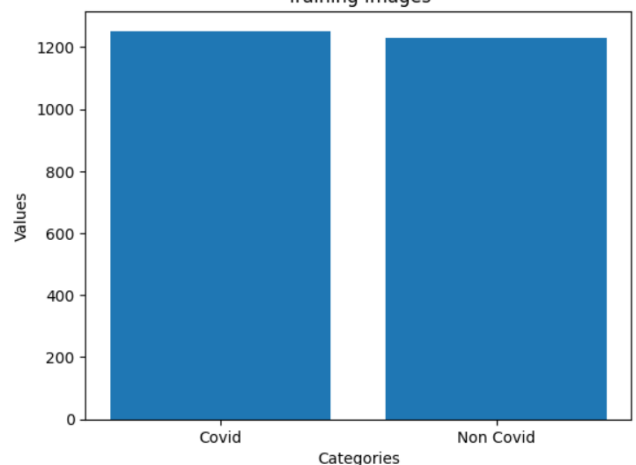
Το ίδιο dataset χρησιμοποιήθηκε και στην ενότητα 5, Τεχνικές Ψηφιακής επεξεργασίας εικόνων. Σε αυτό το σημείο, έγινε προσπάθεια βελτίωσης των προβλέψεων με τη χρήση συνελκτικών νευρωνικών δικτύων.

Παρακάτω φαίνονται ενδεικτικά κάποιες εικόνες με τις ετικέτες τους.



Στο διάγραμμα μπορούμε να δούμε πως μοιράζονται οι εικόνες σε κάθε κατηγορία.

Για να φορτώσουμε τις εικόνες και να κάνουμε rescale χρησιμοποιήθηκαν data generators με τη βοήθεια του keras της βιβλιοθήκης tensorflow. Το batch size επιλέξαμε να είναι 32 και εκπαιδεύσαμε για 100 εποχές. Οι εικόνες φορτώνονται σε μέγεθος 256x256x1. Το 1 σημαίνει ότι τις φορτώνουμε σαν κλίμακα του γκρι (grayscale).



Το dataset δεν είναι χωρισμένο, οπότε το χωρίζουμε σε εικόνες εκπαίδευσης 70%, εικόνες επικύρωσης 20% και εικόνες δοκιμής 10%.

Επίσης, χρησιμοποιήθηκαν οι callback συναρτήσεις early stopping και checkpoints. Με βάση το validation loss, αν δεν υπάρξει βελτίωση για 8 εποχές η εκπαίδευση σταματάει και αποκαθίστανται τα βάρη από την καλύτερη εποχή.

Το μοντέλο περιλαμβάνει **δύο επίπεδα Συνέλιξης - MaxPooling2D** που επαναλαμβάνονται **τέσσερις φορές**. Σε κάθε επίπεδο συνέλιξης ο αριθμός των φίλτρων διπλασιάζεται (από 16, 32, 64 και 128). Το kernel που χρησιμοποιήθηκε είναι 3x3. Η συνάρτηση ενεργοποίησης είναι η relu. Στη συνέχεια ακολουθούν **τρία επίπεδα Συνέλιξης - BatchNormalization - MaxPooling2D**. Έπειτα, έχουμε **ένα επίπεδο flatten**, **ένα επίπεδο Dense** (500 νευρώνες) και **ένα επίπεδο Dropout** (30%). Το **τελικό επίπεδο** αποτελείται από 2 νευρώνες (όσες και οι κλάσεις) και συνάρτηση ενεργοποίησης τη softmax (για κατηγοριοποίηση).

Οπότε έχουμε συνολικά 15 επίπεδα.

Τέλος, το μοντέλο γίνεται compile, χρησιμοποιείται ο βελτιστοποιητής Adam και για συνάρτηση κόστους η CategoricalCrossentropy.

Παρακάτω φαίνεται ενδεικτικά ο κώδικας με όσα αναφέραμε.

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Rescaling(1./255, input_shape=(256, 256, 1)),
    tf.keras.layers.Conv2D(16, 3, padding='same', activation='relu'),
    tf.keras.layers.MaxPooling2D(),
    tf.keras.layers.Conv2D(32, 3, padding='same', activation='relu'),
    tf.keras.layers.MaxPooling2D(),
    tf.keras.layers.Conv2D(64, 3, padding='same', activation='relu'),
    tf.keras.layers.MaxPooling2D(),
    tf.keras.layers.Conv2D(128, 3, padding='same', activation='relu'),
    tf.keras.layers.MaxPooling2D(),

    tf.keras.layers.Conv2D(256, 3, padding='same', activation='relu'),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPooling2D(),

    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(500, activation='relu')]

model.add(tf.keras.layers.Dropout(0.3))
model.add(tf.keras.layers.Dense(2, activation='softmax'))
```



```

callbacks = []

checkpoint_filepath = '/content/drive/MyDrive/classification of diseases/covid model.h
df5'
save_best_callback = tf.keras.callbacks.ModelCheckpoint(filepath=checkpoint_filepath,
monitor='val_loss', mode='min', save_best_only=True, verbose=1)
callbacks.append(save_best_callback)

early_stop_callback = tf.keras.callbacks.EarlyStopping(patience=8, restore_best_weight
s=True, verbose=1)
callbacks.append(early_stop_callback)

history = model.fit(train_generator, validation_data=validation_generator, steps per e
poch=20,

                    epochs=100,
                    validation_steps=15,
                    verbose=1,
                    callbacks=callbacks)

```

Προχωράμε στην εκπαίδευση του δικτύου η οποία στις 38 εποχές σταματάει και τα βάρη αποκαθίστανται από την εποχή 30.

```

Epoch 38: val_loss did not improve from 0.14766
Restoring model weights from the end of the best epoch: 30.
20/20 [=====] - 5s 246ms/step - loss: 0.0227 - accuracy: 0.9937 - val_loss: 0.3695 - val_accuracy: 0.8938
Epoch 38: early stopping

```

```

[ ] train_loss, train_acc = model.evaluate(train_generator, verbose=2)
print("Train accuracy: " + "{:.2f}".format(train_acc*100) + "%")

55/55 - 7s - loss: 0.0766 - accuracy: 0.9810 - 7s/epoch - 132ms/step
Train accuracy: 98.10%

```

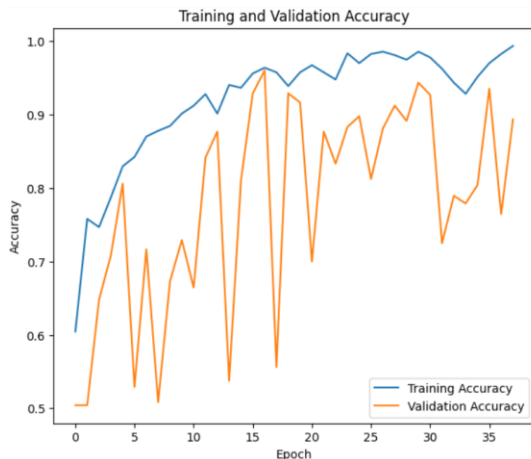
```

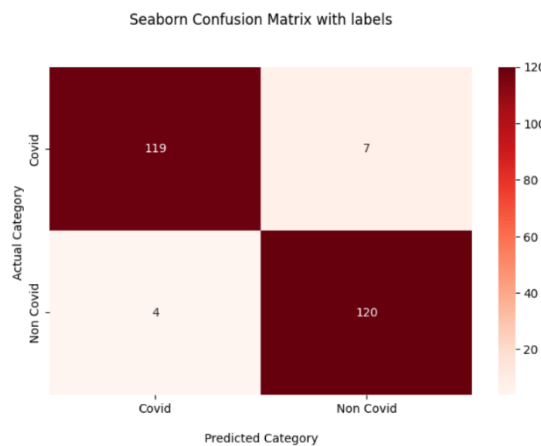
[ ] test_loss, test_acc = model.evaluate(test_generator, verbose=2)
print("Test accuracy: " + "{:.2f}".format(test_acc*100) + "%")

8/8 - 1s - loss: 0.1413 - accuracy: 0.9560 - 1s/epoch - 165ms/step
Test accuracy: 95.60%

```

Τα αποτελέσματα είναι πάρα πολύ καλά και δεν έχουν καμία σχέση με την προηγούμενη προσπάθεια που έγινε στην ενότητα 5. Η ακρίβεια στο σύνολο εκπαίδευσης είναι πολύ καλή, **98.10%**, και στο σύνολο δοκιμής είναι **95.60%**.





Από τον πίνακα σύγχυσης μπορούμε να δούμε ότι κάναμε λάθος πρόβλεψη σε μόλις 7+4=11 εικόνες.

Παρακάτω φαίνεται και το classification report.

	precision	recall	f1-score	support
0	0.97	0.94	0.96	126
1	0.94	0.97	0.96	124
accuracy			0.96	250
macro avg	0.96	0.96	0.96	250
weighted avg	0.96	0.96	0.96	250

Δοκιμή μοντέλων από το ένα dataset στο άλλο.

Το μη προεκπαιδευμένο μοντέλο που χρησιμοποιήσαμε στο brain tumor dataset, τώρα το δοκιμάσαμε στο covid dataset. Τα αποτελέσματα φαίνονται παρακάτω και είναι ικανοποιητικά, δεδομένου ότι οι παράμετροι βελτιώθηκαν για ένα άλλο dataset με διαφορετικές εικόνες.

```
[ ] train_loss, train_acc = model.evaluate(train_data_gen, verbose=2)
print("Train accuracy: " + "{:.2f}".format(train_acc*100) + "%")

143/143 - 10s - loss: 0.3456 - accuracy: 0.8766 - 10s/epoch - 71ms/step
Train accuracy: 87.66%

[ ] test_loss, test_acc = model.evaluate(test_data_gen, verbose=2)
print("Test accuracy: " + "{:.2f}".format(test_acc*100) + "%")

41/41 - 3s - loss: 0.5522 - accuracy: 0.7864 - 3s/epoch - 63ms/step
Test accuracy: 78.64%
```

Το μοντέλο που χρησιμοποιήσαμε στο covid dataset, τώρα το δοκιμάσαμε στο brain tumor dataset. Τα αποτελέσματα φαίνονται παρακάτω και είναι πολύ καλά.

```
[ ] train_loss, train_acc = model.evaluate(train_generator, verbose=2)
print("Train accuracy: " + "{:.2f}".format(train_acc*100) + "%")

55/55 - 8s - loss: 0.0977 - accuracy: 0.9516 - 8s/epoch - 148ms/step
Train accuracy: 95.16%

[ ] test_loss, test_acc = model.evaluate(test_generator, verbose=2)
print("Test accuracy: " + "{:.2f}".format(test_acc*100) + "%")

8/8 - 4s - loss: 0.2021 - accuracy: 0.8800 - 4s/epoch - 524ms/step
Test accuracy: 88.00%
```

Το μοντέλο που χρησιμοποιήσαμε στο covid dataset φαίνεται να λειτουργεί πολύ καλά και στα δυο διαφορετικά dataset που έχουμε διαθέσιμα.

Συγκεντρωτικά αποτελέσματα σε πίνακες

Στους παρακάτω πίνακες συγκεντρώσαμε τα αποτελέσματα από όλες τις δοκιμές που κάναμε. Τα δεδομένα μας είναι balanced, οπότε μπορούμε να χρησιμοποιήσουμε το accuracy σαν μετρική αξιολόγησης.

Brain Tumor dataset		
Evaluation	Train Accuracy	Test Accuracy
Non pretrained CNN model	98.84%	94.13%
Pretrained CNN model	99.19%	96.11%
Covid dataset		
Evaluation	Train Accuracy	Test Accuracy
Thresholding method	-	65.5%
Non pretrained CNN model	98.10%	95.60%

Brain tumor non pretrained CNN model used in Covid dataset	
Train Accuracy	Test Accuracy
87.66%	78.64%
Covid non pretrained CNN model used in Brain tumor dataset	
Train Accuracy	Test Accuracy
95.16%	88.00%

Οι κώδικες που δημιουργήσαμε για την εργασία, είναι γραμμένοι σε μορφή jupyter notebooks και μπορούν να τρέξουν στο online περιβάλλον Google Collab. Τα datasets βρίσκονται με links στη βιβλιογραφία, και μπορούν να φορτωθούν εύκολα στους κώδικες μέσω σύνδεσης στο google drive.

7.Επίλογος

Απ' όλα τα παραπάνω, συμπεραίνουμε ότι με τη χρήση συνελκτικών νευρωνικών δικτύων μπορούμε να επιτύχουμε πολύ μεγάλη ακρίβεια στις προβλέψεις μας. Αξίζει να σημειωθεί, ότι η επίτευξη υψηλής ακρίβειας στην ταξινόμηση ασθενειών απαιτεί σωστή προεπεξεργασία των εικόνων και καλή επιλογή της αρχιτεκτονικής του δικτύου. Επίσης, είναι σημαντικό να υπάρχει μια επαρκής ποσότητα δεδομένων εκπαίδευσης. Επομένως, η χρήση συνελκτικών νευρωνικών δικτύων αναμένεται να επεκταθεί και να χρησιμοποιηθεί ευρέως στην ταξινόμηση ασθενειών στο μέλλον, σε πολλές ιατρικές εφαρμογές.

8.Βιβλιογραφία

<https://www.wikihow.life/Run-a-Nuclear-Magnetic-Resonance-Spectrum>

<https://www.labmate-online.com/news/mass-spectrometry-and-spectroscopy/41/breaking-news/nmr-spectroscopy-in-medicine-a-complete-guide/56546>

<https://www.hopkinsmedicine.org/health/treatment-tests-and-therapies/computed-tomography-ct-scan>

https://en-m-wikipedia-org.translate.goog/wiki/CT_scan?_x_tr_sl=en&_x_tr_tl=el&_x_tr_hl=el&_x_tr_pto=sc
<https://www.nibib.nih.gov/science-education/science-topics/magnetic-resonance-imaging-mri>

<https://www.snmmi.org/AboutSNMMI/Content.aspx?ItemNumber=31086>

<https://www.medindia.net/patients/patientinfo/gamma-camera.htm>

<https://blockgeni.com/essentials-of-ai-vs-ml-vs-deep-learning/>

https://www.researchgate.net/figure/Biological-Neuron-versus-Artificial-Neural-Network_fig4_325870973

<https://www.javatpoint.com/artificial-neural-network>

<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

<https://towardsai.net/p/machine-learning/beginner-guides-to-convolutional-neural-network-from-scratch-kuzushiji-mnist-75f42c175b21>

<https://chris-said.io/2020/12/26/two-things-that-confused-me-about-cross-entropy/>

<https://towardsdatascience.com/softmax-activation-function-explained-a7e1bc3ad60>

<https://medium.com/mlearning-ai/optimizers-in-deep-learning-7bf81fed78a0>

<https://towardsdatascience.com/loss-functions-and-their-use-in-neural-networks-a470e703f1e9>

https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator

<https://www.analyticsvidhya.com/blog/2020/08/image-augmentation-on-the-fly-using-keras-imagedatagenerator/>

<https://machinelearningmastery.com/early-stopping-to-avoid-overtraining-neural-network-models/>

<https://machinelearningmastery.com/check-point-deep-learning-models-keras/>

<https://neptune.ai/blog/how-to-choose-a-learning-rate-scheduler>

https://keras.io/api/callbacks/reduce_lr_on_plateau/

<https://towardsdatascience.com/step-by-step-guide-to-using-pretrained-models-in-keras-c9097b647b29>

<https://medium.datadriveninvestor.com/introducing-transfer-learning-as-your-next-engine-to-drive-future-innovations-5e81a15bb567>

https://keras.io/getting_started/

Διαφάνειες μαθήματος “Όραση Υπολογιστών”, Δημοκρίτειο Πανεπιστήμιο Θράκης

<https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset>

<https://www.kaggle.com/datasets/plameneduardo/sarscov2-ctscan-dataset>