ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ ΚΑΙ ΕΦΑΡΜΟΓΕΣ ΤΟΥΣ ΣΤΗΝ ΑΝΑΓΝΩΡΙΣΗ ΤΟΥ ΚΛΕΙΔΙΟΥ ΕΝΟΣ ΜΟΥΣΙΚΟΥ ΚΟΜΜΑΤΙΟΥ

Διπλωματική Εργασία

Καπαρουνάκης Γεώργιος

Ζεάκης Μιχαήλ

Επιβλέπων: Ευμορφόπουλος Νέστορας

Φεβρουάριος 2022

UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

ARTIFICIAL NEURAL NETWORKS AND THEIR APPLICATIONS IN RECOGNIZING THE KEY OF A PIECE OF MUSIC

Diploma Thesis

KaparounakisGeorgios

Zeakis Michael

Supervisor: Evmorfopoulos Nestoras

February 2022

Εγκρίνεται από την Επιτροπή Εξέτασης:

Επιβλέπων **Ευμορφόπουλος Νέστωρ**
Αναπληρωτής Καθηγητής, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος **Σταμούλης Γεώργιος**
Καθηγητής, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος **Ποταμιάνος Γεράσιμος**
Αναπληρωτής Καθηγητής, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

ΕΥΧΑΡΙΣΤΙΕΣ

Θέλουμε να ευχαριστήσουμε θερμά τις οικογένειες και τους φίλους μας για την στήριξη, την αγάπη και την πολύτιμη βοήθεια τους, οι οποίοι είναι πάντα εκεί για εμάς, στις σπουδές μας, στα όνειρα μας και στη ζωή μας. Θα θέλαμε επίσης να ευχαριστήσουμε τον επιβλέποντα καθηγητή μας Ευμορφόπουλο Νέστορα για τη συνέπεια του και την άριστη συνεργασία μας, κατά την περάτωση της διπλωματικής μας εργασίας.

ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΑΚΑΔΗΜΑΪΚΗΣ ΔΕΟΝΤΟΛΟΓΙΑΣ ΚΑΙ ΠΝΕΥΜΑΤΙΚΩΝ ΔΙΚΑΙΩΜΑΤΩΝ

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

Οι Δηλούντες
Καπαρουνάκης Γεώργιος
Ζεάκης Μιχαήλ

ΠΕΡΙΛΗΨΗ

Η Μηχανική Μάθηση και τα Νευρωνικά Δίκτυα αποτελούν έναν επιστημονικό τομέα βασισμένο στον τρόπο λειτουργίας του ανθρώπου και συγκεκριμένα των νευρώνων του εγκεφάλου του. Τα τελευταία προσομοιώνουν την ανθρώπινη εμπειρία μέσω της διαδικασίας της εκπαίδευσης. Η εκμάθηση τους διεκπεραιώνεται χρησιμοποιώντας μεγάλα σύνολα δεδομένων προκειμένου να επιτευχθεί η επίλυση προβλημάτων και η εκτέλεση προβλέψεων. Η παρούσα πτυχιακή πραγματεύεται την δημιουργία ενός μοντέλου νευρωνικών δικτύων με σκοπό την πρόβλεψη του κλειδιού ενός μουσικού κομματιού. Για την πραγματοποίηση του μοντέλου κρίθηκε απαραίτητη η μελέτη της μουσικής θεωρίας ώστε να προσδιοριστούν όροι όπως μουσική κλίμακα, τονικότητα και τέλος μουσικό κλειδί. Επιπλέον, τα συνελικτικά νευρωνικά δίκτυα ήταν η κατηγορία των νευρωνικών που αναπτύχθηκαν. Τα τελευταία παρέχουν την δυνατότητα αποτελεσματικότερης εκπαίδευσης σε εικόνες, και συγκεκριμένα τα φασματογραφήματα, που αποτέλεσαν και τα σύνολα δεδομένων που χρησιμοποιήθηκαν. Το μοντέλο αποτελείται από δύο συνελικτικά δίκτυα, ένα για την πρόβλεψη τονικότητας και ένα για την πρόβλεψη της κλίμακας που απαρτίζουν το μουσικό κλειδί. Το framework που χρησιμοποιήθηκε για την ανάπτυξη του μοντέλου είναι το PyTorch. Σε αυτό δημιουργήθηκαν οι κώδικες (python) για την επεξεργασία των δεδομένων, την εκπαίδευση και την αξιολόγηση των δικτύων. Κατά την διαδικασία της εκπαίδευσης και αξιολόγησης του μοντέλου, πραγματοποιήθηκαν πολλοί πειραματισμοί προκειμένου να προσδιοριστούν οι βέλτιστες τιμές των υπέρ-παραμέτρων που δίνουν το καλύτερο δυνατό ποσοστό επιτυχίας προβλέψεων. Τέλος αναπτύχθηκε πρόγραμμα που χρησιμοποιεί το εκπαιδευμένο μοντέλο ώστε να κάνει πρόβλεψη του μουσικού κλειδιού οποιουδήποτε κομματιού επιθυμούμε διαθέτοντας το αντίστοιχο αρχείο mp3.

ABSTRACT

Machine Learning and Neural Networks are a scientific field based on the way a person works and more specifically the neurons of their brain. The latter simulate human experience through the process of education. Their learning is performed using large data sets to achieve problem solving and prediction execution. This dissertation deals with the creation of a model of neural networks aiming to predict the key of a piece of music. In order to create the model, it was deemed necessary to research music theory to identify terms such as musical scale, tonality and finally a musical key. In addition, the convolutional neural networks were the architecture of the networks developed. The latter provide the most effective training in images, and in particular the spectrograms, which were the data sets used. The model consists of two convolutional networks, one for predicting tonality (tonic) and one for predicting the scale that constitute the musical key. The framework used to develop the model is PyTorch. With it, the codes (python) were created for data processing, training and evaluation of networks. During the training and evaluation process of the model, many experiments were performed in order to determine the optimal values of the hyperparameters that give the best possible prediction success rate. Finally, a program was developed that uses the trained model to predict the music key of any track we want by having the corresponding mp3 file.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Η μηχανική μάθηση στοχεύει στην ανάπτυξη υπολογιστικών συστημάτων τα οποία μιμούνται τον άνθρωπο. Συγκεκριμένα, η δημιουργία προγραμμάτων και αλγορίθμων που εκπαιδεύουν τον υπολογιστή πώς να μάθει μέσω κάποιων δεδομένων βασίζεται στον εμπειρικό τρόπο εκμάθησης του ανθρώπου. Τα τελευταία χρόνια οι βελτιωμένοι υπολογιστικοί πόροι και η εύκολη πρόσβαση σε ένα τεράστιο όγκο δεδομένων έχουν καθοριστικό ρόλο στην πρόοδο και ανάπτυξη του τομέα της μηχανικής μάθησης [1].

Η έννοια των νευρωνικών δικτύων πηγάζει από τον κλάδο της βιολογίας και συγκεκριμένα των νευρώνων του ανθρώπινου εγκεφάλου. Τα τεχνητά νευρωνικά δίκτυα, που αποτελούν μέρος της τεχνολογίας της μηχανικής μάθησης, απαρτίζονται από τεχνητούς νευρώνες και μέσω υπολογιστικών διαδικασιών προσπαθούν να επιλύσουν προβλήματα. Χρησιμοποιούνται όλο και περισσότερο σε διαφορετικούς επιστημονικούς τομείς και αποτελούν εξαιρετικά εργαλεία πρόβλεψης, ταξινόμησης και ελέγχου [2],[3].

Η παρούσα πτυχιακή εργασία προσπαθεί να συνεισφέρει στο τομέα των νευρωνικών δικτύων αναπτύσσοντας ένα μοντέλο αναγνώρισης του κλειδιού ενός μουσικού κομματιού. Αυτό αποσκοπεί στην σύνδεση ενός καλλιτεχνικού κλάδου όπως αυτός της μουσικής με τον επιστημονικό και προγραμματιστικό κλάδο της μηχανικής μάθησης. Ακόμη, έγινε ανάπτυξη ενός προγράμματος για την εύρεση του μουσικού κλειδιού με τη χρήση των νευρωνικών δικτύων που εκπαιδεύτηκαν.

Ακολουθεί η διάρθρωση των υπολοίπων κεφαλαίων με βάση το περιεχόμενο τους:

Στο κεφάλαιο 2 προσδιορίζεται η μουσική θεωρία που κρίθηκε απαραίτητη. Συγκεκριμένα, αναλύεται το κλειδί ενός μουσικού κομματιού το οποίο αποτελείται από την τονικότητα και την κλίμακα.

Το 3ο κεφάλαιο αναφέρεται στην θεωρία των νευρωνικών δικτύων αναλύοντας την διαδικασία εκμάθησης και τα διαφορετικά είδη αρχιτεκτονικής των νευρωνικών. Τέλος παρουσιάζονται τα χαρακτηριστικά των συνελικτικών νευρωνικών δικτύων.

Στο 4ο κεφάλαιο γίνεται εισαγωγή στην μηχανική μάθηση και προσδιορίζεται το framework (PyTorch) που χρησιμοποιείται για την υλοποίηση του software.

Στα κεφάλαια 5 και 6 εστιάζεται η ανάπτυξη και η υλοποίηση του λογισμικού των νευρωνικών δικτύων. Το 5ο κεφάλαιο αναφέρεται στην υλοποίηση του λογισμικού για την δημιουργία, την επεξεργασία και την επαύξηση των δεδομένων εισαγωγής των νευρωνικών δικτύων ενώ στο 6ο κεφάλαιο αναλύεται ο κώδικας της εκπαίδευσης όπως επίσης και η δομή των νευρωνικών δικτύων. Επιπλέον παρουσιάζονται τα πειράματα με τα οποία πραγματοποιήθηκε ο ορισμός των υπέρ-παραμέτρων των δικτύων.

Το κεφάλαιο 7 αφορά τα συμπεράσματα της διπλωματικής ως προς τα αποτελέσματα των προβλέψεων των νευρωνικών δικτύων και τις δυσκολίες που αντιμετωπίστηκαν. Επίσης αναφέρονται πιθανές μελλοντικές επεκτάσεις σε επίπεδο εφαρμογής και βελτιστοποιήσεις στην εκπαίδευση των δικτύων.

Τέλος, το παράρτημα Α αφορά την ανάπτυξη ενός προγράμματος με το οποίο ο χρήστης έχει την δυνατότητα να εκτελέσει προβλέψεις σε μουσικά κομμάτια με την χρήση των εκπαιδευμένων δικτύων.

ΚΕΦΑΛΑΙΟ 2

ΜΟΥΣΙΚΗ ΘΕΩΡΙΑ

2.1 Εισαγωγή στη μουσική θεωρία

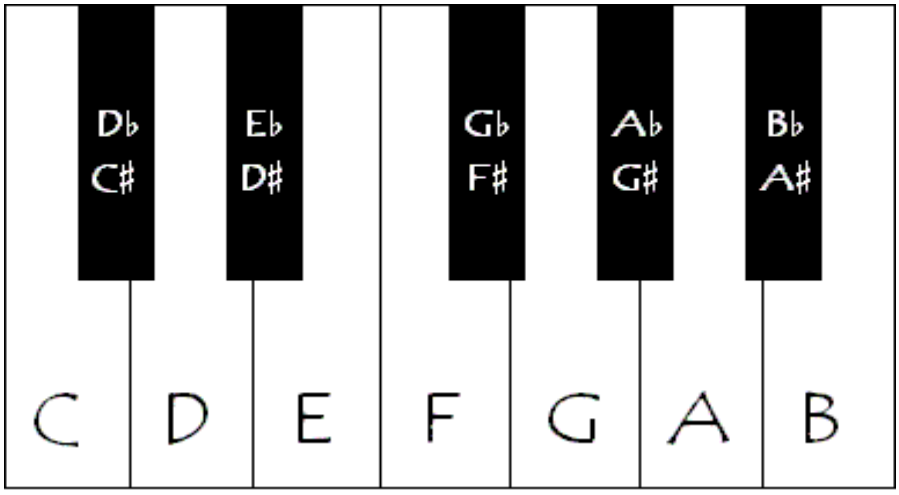
Η θεωρία της μουσικής είναι η πρακτική που χρησιμοποιείται από τους μουσικούς προκειμένου να κατανοήσουν και να επικοινωνήσουν την γλώσσα της μουσικής. Εμβαθύνει στα θεμέλια της και παρέχει ένα σύστημα ερμηνείας μουσικών συνθέσεων. Ο σκοπός της είναι να εξηγήσει για ποιο λόγο και με ποιόν τρόπο μια μελωδία (ή ένα τραγούδι) ακούγεται "σωστή" ή "λάθος". Τα στοιχεία που σχηματίζουν αρμονία, μελωδία, ρυθμό, αλλά και τα συνθετικά στοιχεία ενός τραγουδιού όπως νότες, χορδές, τονικότητα, κλίμακες, κλειδιά καθορίζονται από την βασική μουσική θεωρία. Τα κρίσιμα μουσικά στοιχεία που πραγματεύεται η παρούσα διπλωματική είναι οι κλίμακες και το κλειδί [4].

2.2 Μουσικό κλειδί, τονικότητα και κλίμακες

Κλειδί ή τονικότητα στη μουσική ορίζεται ως ένα σύνολο από νότες ή τόνους που οργανώνονται γύρω από ένα τονικό κέντρο. Αυτή η σειρά από διαφορετικούς μουσικούς φθόγγους ή νότες θεμελιώνει την μελωδία ενός μουσικού κομματιού και ονομάζεται κλίμακα. Οι πιο βασικές κλίμακες χρησιμοποιούν οκτώ νότες (εκ των οποίων η όγδοη είναι η αρχική που επαναλαμβάνεται) μέσα από μία οκτάβα. Μια οκτάβα είναι η απόσταση μεταξύ μιας νότας και της επόμενης με το ίδιο όνομα (δηλαδή ανάμεσα στη πρώτη και την όγδοη), της οποίας η συχνότητα είναι διπλασιασμένη. Αποτελείται από δώδεκα νότες, τις εφτά βασικές (τα λευκά πλήκτρα σε ένα πιάνο) και 5 επιπλέον flat ή sharp (τα μαύρα πλήκτρα) [4].

Οι βασικές νότες συμβολίζονται με γράμματα του αγγλικού αλφάβητου με τον εξής τρόπο: C D E F G A B και αντίστοιχα στα ελληνικά με: ΝΤΟ ΡΕ ΜΙ ΦΑ ΣΟΛ ΛΑ ΣΙ. Μια sharp νότα ακούγεται μισό τόνο ή μισό βήμα ψηλότερα από την προηγούμενη βασική νότα και συνοδεύεται από το σύμβολο #. Μια flat νότα ακούγεται μισό τόνο χαμηλότερα από την επόμενη βασική νότα και το σύμβολό της είναι το πεζό γράμμα του αγγλικού αλφάβητου b [4]. Για παράδειγμα εφόσον η G νότα είναι η πέμπτη νότα μιας οκτάβας, η G# είναι το επόμενο μαύρο πλήκτρο και ακούγεται μισό τόνο ψηλότερα. Παρατηρείται

λοιπόν, ότι ένα μαύρο πλήκτρο μίας οκτάβας μπορεί να έχει δύο ονομασίες: την sharp (#) της προηγούμενης νότας ή την flat (b) της επόμενης. Μπορούμε να παρατηρήσουμε όλες τις ονομασίες από τις νότες που ανήκουν σε μια οκτάβα ενός πιάνου στην εικόνα 2.1.

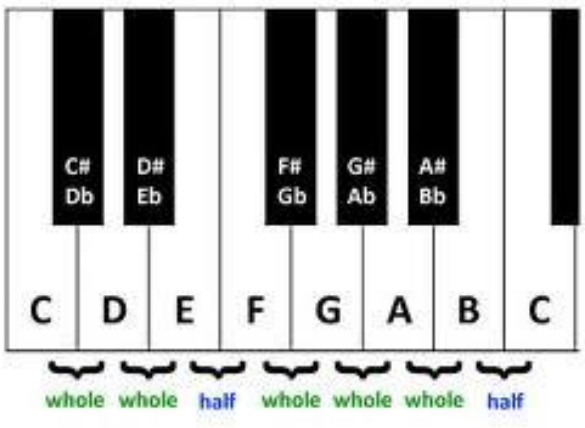


2.1 Η οκτάβα σε ένα πιάνο

Η απόσταση μεταξύ δύο νοτών στην μουσική ονομάζεται διάστημα. Τα δύο βασικά είδη διαστημάτων είναι το half step (μισός τόνος ή ημιτόνιο) και whole step. Το half step εκφράζει την απόσταση μεταξύ δύο συνεχόμενων νοτών ενώ το whole step είναι δύο ημιτόνια. Για παράδειγμα :

- Whole step: C – D, F# - G#, E – F#
- Half step: C – C#, F# - G, E - F

Στο μεγαλύτερο μέρος της μη έθνικ, ευρωπαϊκής μουσικής βιομηχανίας χρησιμοποιούνται κατά κύριο λόγο δύο ομάδες κλιμάκων, οι Μείζονες (major) και οι Ελάσσονες (minor). Οι Μείζονες κλίμακες συνήθως χαρακτηρίζονται ως “χαρούμενες” και το μοτίβο που ακολουθούν είναι : – Whole – Whole – Half – Whole – Whole – Whole – Half – (όπου – έχουμε νότα που ανήκει στη κλίμακα). Οι Ελάσσονες αντίστοιχα χαρακτηρίζονται πιο θλιβερές στο άκουσμα και ακολουθούν το μοτίβο: – Whole – Half – Whole – Whole – Half – Whole – Whole. Παραδείγματος χάριν για το κλειδί C major ξεκινώντας από την C και ακολουθώντας το μοτίβο W W H W W W H παίρνουμε τις νότες C D E F G A B όπως φαίνεται και στο σχήμα 2.2 [4].



2.2 νότες της C major

Μια κλίμακα χαρακτηρίζεται από την βασική νότα και από το είδος της (minor ή major) [4]. Έχοντας 12 νότες σε μια οκτάβα και 2 διαφορετικά είδη κλίμακας καταλήγουμε σε 24 διαφορετικά μουσικά κλειδιά τα οποία απεικονίζονται στις εικόνες 2.3 και 2.4. Συνεπώς, ένα μουσικό κλειδί αποτελεί έναν οδηγό για το ποιες νότες ακούγονται "όμορφα" μαζί και αποτελεί βασικό χαρακτηριστικό ενός κομματιού μουσικής.

Minor κλίμακες	
Κλειδί	Νότες
C minor	C D Eb F G Ab Bb
G minor	G A Bb C D Eb F
D minor	D E F G A Bb C
A minor	A B C D E F G
E minor	E F# G A B C D
B minor	B C# D E F# G A
F minor	F G Ab Bb C Db Eb
F# minor	F# G# A B C# D E
Db minor	Db Eb Fb Gb Ab Bbb Cb
Ab minor	Ab Bb Cb Db Eb Fb Gb
Eb minor	Eb F Gb Ab Bb Cb Db
Bb minor	Bb C Db Eb F Gb Ab

2.3 Minor Κλίμακες / Κλειδιά

Major Κλίμακες	
Κλειδί	Νότες
C Major	C D E F G A B
G Major	G A B C D E F#
D Major	D E F# G A B C#
A Major	A B C# D E F# G#
E Major	E F# G# A B C# D#
B Major	B C# D# E F# G# A#
F Major	F G A Bb C D E
Bb Major	Bb C D Eb F G A
Eb Major	Eb F G Ab Bb C D
Ab Major	Ab Bb C Db Eb F G
Db Major	Db Eb F Gb Ab Bb C
Gb Major	Gb Ab Bb Cb Db Eb F

2.4 Major Κλίμακες / Κλειδιά

ΚΕΦΑΛΑΙΟ 3

ΑΝΑΓΝΩΡΙΣΗ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ ΜΟΥΣΙΚΗΣ ΜΕ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

3.1 Music Information Retrieval (MIR) and Music Composition

Ο κλάδος της μηχανικής μάθησης που ασχολείται και χρησιμοποιείται στην μουσική βιομηχανία και την μουσική γενικότερα διακρίνεται σε δύο κυρίαρχους τομείς. Αυτοί είναι η σύνθεση μουσικών κομματιών (music composition) και η αναγνώριση χαρακτηριστικών μουσικής (Music Information Retrieval). Τα δύο υπό πεδία αυτά ενώ δεν καλύπτουν όλες τις εφαρμογές που μπορεί να έχει η μηχανική μάθηση και τα νευρωνικά δίκτυα στην μουσική, κατηγοριοποιούν σε μεγάλο βαθμό τις τεχνολογίες που επικρατούν. Πιο συγκεκριμένα το MIR απαιτείται σχεδόν σε κάθε εφαρμογή και τεχνολογία που σχετίζεται με τον κλάδο [5].

Το υπό πεδίο της μουσικής σύνθεσης είναι πιο εύκολο να προσδιοριστεί από αυτό της αναγνώρισης χαρακτηριστικών, και αφορά την χρήση μηχανικής και βαθιάς μάθησης για την παραγωγή νέων ήχων και ολόκληρων μουσικών κομματιών. Η αναγνώριση χαρακτηριστικών μπορεί θεωρητικά να αποτελεί διαφορετικό κλάδο από την μουσική σύνθεση, αλλά στην πραγματικότητα παίζει καθοριστικό ρόλο στην σύνθεση μουσικών κομματιών. Συγκεκριμένα, για να μπορέσει ένα νευρωνικό να μάθει να συνθέτει, θα πρέπει πρώτα να έχει εκπαιδευτεί στον διαχωρισμό διαφόρων μουσικών χαρακτηριστικών όπως ρυθμό, κλίμακες, νότες που αποτελούν κομμάτι του MIR. Πέρα από τη σύνθεση ο τομέας του music composition συμπεριλαμβάνει επιπλέον την απομόνωση ήχων, την μετατροπή του σε άλλον ή τον εμπλουτισμό του [5].

Το MIR είναι ένα πεδίο που συνδυάζει επιστημονικούς κλάδους όπως της στατιστικής, των υπολογιστών, της μουσικολογίας και του κλάδου επεξεργασίας ψηφιακού σήματος. Παρά το γεγονός ότι υπάρχουν πληροφορίες σε ένα μουσικό κομμάτι όπως το τέμπο και η ένταση, που είναι εύκολα κατανοητά από τον ακροατή, υπάρχουν και πληροφορίες που μπορεί να γίνουν αρκετά περίπλοκες. Για παράδειγμα, έννοιες όπως η

διάθεση, το συναίσθημα και η έμπνευση ενός τραγουδιού αποτελούν χαρακτηριστικά που αφορούν τον τρόπο που άνθρωπος αντιλαμβάνεται το εκάστοτε μουσικό κομμάτι, με την μοντελοποίηση τους να αποτελεί πολύ δυσκολότερη διαδικασία. Σε αυτές τις περιπτώσεις πραγματοποιείται και ο συνδυασμός τον κλάδων όπως αυτού της στατιστικής προκειμένου να προσδιοριστεί καλύτερα το μοντέλο. Γενικότερα το MIR αποτελεί πεδίο μείζονος σημασίας για οποιαδήποτε εφαρμογή μηχανικής μάθησης που βασίζεται στη μουσική, παρά το γεγονός ότι η κύρια χρήση του αφορά συστήματα συστάσεων (recommendation systems) στη μουσική βιομηχανία. [5].

Η παρούσα πτυχιακή αποτελεί κομμάτι του κλάδου αναγνώρισης χαρακτηριστικών μουσικής και συγκεκριμένα του μουσικού κλειδιού ενός τραγουδιού. Αποτελείται από δύο νευρωνικά δίκτυα που εκπαιδεύτηκαν σε dataset τα οποία απαρτίζονταν από μουσικά κομμάτια ηλεκτρονικής μουσικής (EDM) και κατάφεραν να φτάσουν σε ποσοστό επιτυχημένων προβλέψεων 59,9% (59,9% για την τονικότητα και 99,7% για την κλίμακα οπότε θεωρούμε σχεδόν αμελητέα την πιθανότητα λάθους της κλίμακας). Γίνεται αντιληπτό ότι το μοντέλο μας για να παρουσιάσει αντίστοιχα ποσοστά επιτυχίας και σε άλλα είδη μουσικής θα πρέπει να έχει εκπαιδευτεί και σε αυτά.

3.2 Υπάρχουσες υλοποιήσεις εύρεσης μουσικού κλειδιού με νευρωνικά δίκτυα

Σε αυτό το υπό κεφάλαιο θα γίνει ανάπτυξη και αναφορά σε προ υπάρχουσες έρευνες και υλοποιήσεις που αφορούν την εύρεση του μουσικού κλειδιού ενός κομματιού και θα συγκριθούν με το μοντέλο που αναπτύχθηκε στην παρούσα διπλωματική εργασία.

3.2.1 Μουσικός ρυθμός και εκτίμηση κλειδιού με συνελικτικά νευρωνικά δίκτυα και φίλτρα κατεύθυνσης

Στο συγκεκριμένο άρθρο ερευνάτε σε ποιο βαθμό μπορούν να χρησιμοποιηθούν τα συνελικτικά νευρωνικά δίκτυα, προκειμένου να βρεθεί το μουσικό κλειδί και ο ρυθμός ενός μουσικού κομματιού εκμεταλλευόμενοι την πληροφορία που περιέχουν τα φασματογραφήματα (spectrograms). Για τις δύο εργασίες χρησιμοποιούνται οι ίδιες αρχιτεκτονικές και γίνεται προσπάθεια απόδειξης ότι οι αρχιτεκτονικές με ευθυγραμμισμένο άξονα μπορούν να έχουν παρόμοιες επιδόσεις με δίκτυα όπως τα VGG νευρωνικά δίκτυα. Στη συνέχεια γίνονται αναφορές στο είδος των φίλτρων που

επιλέγονται για MIR εφαρμογές, και καταλήγουν στην ανάγκη έρευνας για το πως και γιατί τα κατευθυντικά και τετράγωνα φίλτρα συμβάλλουν στα αποτελέσματα που επιτυγχάνονται από συστήματα ταξινόμησης για εργασίες MIR [6].

Η εκτίμηση κλειδιού επιχειρεί να προβλέψει το σωστό κλειδί για ένα δεδομένο μουσικό κομμάτι, και ουσιαστικά αποτελεί ένα πρόβλημα ταξινόμησης 24 διαφορετικών κατηγοριών,12 διαφορετικές νότες μιας οκτάβας και 2 βασικές κλίμακες (minor, major). Στην έρευνα παρουσιάζεται πως ακολουθήθηκε παρόμοια προσέγγιση με τις αρχιτεκτονικές τύπου VGG με τετράγωνα φίλτρα. Πιο συγκεκριμένα, η είσοδος των νευρωνικών αποτελείται από φασματογραφήματα διαστάσεων 168x60, στα οποία έχει υπάρξει συχνοτική μετατόπιση [6]. Η λειτουργία αυτή αποτελεί ακριβώς την ίδια μέθοδο που χρησιμοποιήθηκε στην παρούσα πτυχιακή κατά τη προσπάθεια επαύξησης των δεδομένων. Ωστόσο στην έρευνα η μετατόπιση πραγματοποιήθηκε τυχαία μεταξύ των ημιτονίων (-4,-3...,6,7) προκειμένου να αντιμετωπιστεί η ανισορροπία των κλάσεων και όχι για να επιτευχθεί επαύξηση δεδομένων.

Αναλυτικότερα, στο συγκεκριμένο άρθρο χρησιμοποιούνται δύο πολύ διαφορετικά είδη αρχιτεκτονικών έτσι ώστε να επιτευχθεί η απόκτηση πληροφοριών σχετικά με την επιρροή των φίλτρων στην εκτίμηση του κλειδιού και του ρυθμού μουσικής. Αυτές είναι μια ρηχή αλλά εξειδικευμένη αρχιτεκτονική και μία πολύ βαθιά. Η αρχιτεκτονική που αποτελεί την εξειδικευμένη αλλά ρηχή προσέγγιση αποτελείται από δύο ενότητες. Μια ενότητα εξαγωγής χαρακτηριστικών (ShallowMod) και ακολουθείται από μία ενότητα ταξινόμησης (ClassMod). Για την εκτίμηση του ρυθμού χρησιμοποιούνται χρονικά φίλτρα με συγκέντρωση κατά μήκος του άξονα συχνότητας και για την εκτίμηση του κλειδιού φασματικά φίλτρα με συγκέντρωση κατά μήκος του άξονα του χρόνου. Και οι δύο αρχιτεκτονικές έχουν ονομαστεί από τις κατευθύνσεις φίλτρων τους, ShallowTemp και ShallowSpec, αντίστοιχα [6].

Η πολύ βαθιά αρχιτεκτονική μοιάζει πολύ περισσότερο με αυτή που αναπτύχθηκε στην διπλωματική, και ουσιαστικά αποτελείται από 6 συνελικτικά επίπεδα φίλτρων 5x5 ακολουθούμενα από φίλτρα 3x3. Όπως και στην προηγούμενη αρχιτεκτονική μετά το dropout επίπεδο έχουμε την ίδια ενότητα ταξινόμησης ClassMod. Η αρχική παραλλαγή ονομάζεται DeepSquare και κατηγοριοποιείται όπως στα ρηχά δίκτυα, σε κατευθυντικές

παραλλαγές DeepTemp και DeepSpec. Πιο αναλυτικά οι αρχιτεκτονικές των δικτύων φαίνονται στο σχήμα 3.1 [6].

(a) `ShallowMod`

Layer	Temp	Spec	Square
Input			
Conv, ReLU	$k, 1 \times 3$	$k, 3 \times 1$	n.a.
Dropout	p_{D}	p_{D}	n.a.
AvgPool	$F_{\mathrm{T}} \times 1$	$1 \times T_{\mathrm{K}}$	n.a.
Conv, ReLU	$64k, 1 \times T_{\mathrm{T}}$	$64k, F_{\mathrm{K}} \times 1$	n.a.
Dropout	p_{D}	p_{D}	n.a.

(b) `DeepMod`

Layer	Temp	Spec	Square
Input			
Conv, ReLU BatchNorm	$2^{\ell}k, 1 \times 5$	$2^{\ell}k, 5 \times 1$	$2^{\ell}k, 5 \times 5$
Conv, ReLU BatchNorm	$2^{\ell}k, 1 \times 3$	$2^{\ell}k, 3 \times 1$	$2^{\ell}k, 3 \times 3$
MaxPool	2×2	2×2	2×2
Dropout	p_{D}	p_{D}	p_{D}

(c) `ClassMod`

Layer	Temp	Spec	Square
Input			
Conv, ReLU GlobalAvgPool Softmax	$N_{\mathrm{T}}, 1 \times 1$	$N_{\mathrm{K}}, 1 \times 1$	n.a.

Table 2: Layer definitions for the three modules `ShallowMod`, `ClassMod`, and `DeepMod`, describing number of filters (e.g., k or $64k$) and their respective shapes (e.g., 1×3 or 5×5).

3.1 Αρχιτεκτονικές νευρωνικών δικτύων που χρησιμοποιήθηκαν [6]

Τέλος, στον πίνακα 3.2 παρουσιάζονται οι ακρίβειες των μοντέλων που χρησιμοποιήθηκαν από το συγκεκριμένο άρθρο. Αξίζει να παρατηρήσουμε την ακρίβεια στο dataset GS(Giansteps MTG key dataset) το οποίο αποτελεί ένα από τα δύο dataset που χρησιμοποιήθηκε και στην παρούσα διπλωματική. Παρατηρείται ότι στο συγκεκριμένο σύνολο δεδομένων για την πρόβλεψη του κλειδιού, έχει επιτευχθεί ακρίβεια 50,8 (ShallowSpec-ρηχή αλλά εξειδικευμένη αρχιτεκτονική) και 58.5 (DeepSquare-βαθιά αρχιτεκτονική). Παρατηρείται λοιπόν ότι οι ακρίβειες είναι αρκετά κοντά με αυτήν της διπλωματικής μας, με την τελευταία να είναι κατά 1,4% πιο ακριβείς. Επιπλέον, η έρευνα καταφέρνει να αναδείξει ότι οι ρηχές αρχιτεκτονικές CNN εμπνευσμένες από την επεξεργασία σήματος που χρησιμοποιούν φίλτρα κατεύθυνσης μπορούν να χρησιμοποιηθούν με επιτυχία τόσο για την ανίχνευση ρυθμού όσο και για τον εντοπισμό του κλειδιού [6].

Architecture	GS	GT	LMD	BR
ShallowTemp	86.5 1.5	60.3 2.7	94.0 1.0	87.9 2.3
DeepTemp	**88.7** 0.6	63.1 0.6	94.5 0.7	88.2 2.4
ShallowSpec	4.5 1.9	11.5 1.3	9.4 2.1	16.7 5.7
DeepSpec	49.6 2.5	40.2 1.4	73.0 2.4	59.6 9.1
DeepSquare	88.1 1.3	**64.7** 2.1	**96.2** 0.4	**92.4** 1.7
Literature	82.5 [3]	78.3 [29]	—	92.0 [3]

(a) Tempo

GS	GT	LMD
1.7 0.4	4.9 0.7	11.0 3.7
46.8 4.3	38.4 2.4	60.7 0.4
50.8 3.8	43.8 1.4	67.1 0.9
55.4 2.7	44.8 2.0	**71.3** 0.2
58.5 3.9	**49.9** 2.0	68.9 2.5
67.9 [2]	~45 [28]	—

(b) Key

3.2 Πίνακες ακρίβειας για tempo και key predition [6]

3.2.2 Εκτίμηση μουσικού κλειδιού από άκρο σε άκρο με χρήση συνελικτικού νευρωνικού δικτύου

Η συγκεκριμένη έρευνα που πραγματοποιήθηκες από το Johannes Kepler University αποτέλεσε έμπνευση στην ανάπτυξη της παρούσας πτυχιακής εργασίας. Αφορά την χρήση συνελικτικών νευρωνικών δικτύων για την πρόβλεψη του κλειδιού ενός μουσικού κομματιού [7]. Η αρχιτεκτονική του νευρωνικού που αναπτύχθηκε αποτελεί ένα πολυταξικό νευρωνικό δίκτυο 24 κλάσεων. Οι κλάσεις αποτελούν όλα τα επιθυμητά κλειδιά που προκύπτουν από τις 12 νότες μιας οκτάβας σε συνδυασμό με τις 2 βασικές κλίμακες. Το νευρωνικό δίκτυο δέχεται σαν είσοδο ένα φασματογράφημα, το οποίο στη συνέχεια περνά από τα 5 συνελικτικά επίπεδα του φίλτρων 5x5. Ακολουθεί ένα επίπεδο υπό δειγματοληψίας μέσης τιμής, ένα linear επίπεδο με έξοδο 48 νευρώνες και ένα softmax επίπεδο το οποίο καταλήγει στις 24 κλάσεις πρόβλεψης. Τα dataset που έχουν χρησιμοποιηθεί είναι τα δύο που έχουν χρησιμοποιηθεί στην παρούσα πτυχιακή μαζί με το McGill Billboard Dataset [8]. Επιπρόσθετα χρησιμοποιήθηκε αλγόριθμος επαύξησης δεδομένων που είχε ακριβώς την ίδια λογική με το Pitch Shift [7].

Μια αξιοσημείωτη διαφορά της διπλωματικής μας με την παραπάνω έρευνα, εκτός από διαφορές σε αρχιτεκτονική και σύνολα εκπαίδευσης, αποτελεί η προσπάθεια μας να απλουστεύσουμε το πρόβλημα σε δύο μικρότερα. Πιο συγκεκριμένα, έγινε προσπάθεια να καταλήξουμε από τις 24 διαφορετικές κλάσεις στο συνδυασμό δύο προβλέψεων, μία πρόβλεψη για την τονικότητα (12 επιλογές) και μία για την κλίμακα (2 επιλογές). Αυτή η προσέγγιση έφερε αμέσως τεράστια βελτίωση στην ακρίβεια του μοντέλου που αναπτύσσαμε παρά το γεγονός ότι βρισκόμασταν ακόμα σε αρχικό στάδιο πειραματισμών. Πριν χωρίσουμε το μοντέλο σε δύο τμήματα, η ακρίβεια μας κυμαινόταν στο 25%, και μετά την αλλαγή καταλήξαμε σε ένα ποσοστό 43%.

Επιπρόσθετα, σημαντική διαφορά αποτελεί η χρήση μιας στρατηγικής αξιολόγησης από το συγκεκριμένο άρθρο. Αυτή στην πραγματικότητα αποτελεί έναν αλγόριθμο ο οποίος καθορίζει πόσο λάθος θεωρείται κάποια πρόβλεψη. Η αξιολόγηση της εύρεσης του κλειδιού ενός μουσικού κομματιού αποτελεί μια διαδικασία περίπλοκη συγκριτικά με άλλα προβλήματα ταξινόμησης 24 κλάσεων. Αυτό συμβαίνει διότι στη μουσική υπάρχουν παραδείγματα όπως το κλειδί του A-minor, που ονομάζεται "σχετική ελάσσονα" με το κλειδί του C-major, καθώς μοιράζονται όλες τις κατηγορίες τόνου και διαφέρουν μόνο ως προς το τονικό. Με αυτό τον τρόπο έχει δημιουργηθεί και ο πίνακας ακρίβειας της έρευνας, όπως φαίνεται στον πίνακα 3.3, όπου έχουμε τις ακρίβειες Correct και Weighted με τις δεύτερες να αποτελούν τις ακρίβειες μετά την χρήση του παραπάνω αλγορίθμου. Εδώ μπορεί να παρατηρηθεί από τη στήλη Correct, που αποτελεί και το μέτρο σύγκρισης με την παρούσα διπλωματική, ότι οι ακρίβειες της έρευνας για την εκπαίδευση σε ίδια δεδομένα (GS, GS mtg), παρουσιάζουν καλύτερη ακρίβεια από την πτυχιακή μας αλλά όχι σε πολύ μεγάλο βαθμό (+ 1-7%). Πιο αναλυτικά η ακρίβειες του μοντέλου που αναπτύχθηκαν φαίνονται στο σχήμα 3.3

Test Set	Method	Train Set	Weighted	Correct	Fifth	Relative	Parallel	Other
GS	CK^1	GS^{MTG}	**74.3**	**67.9**	6.8	7.1	4.3	**13.9**
	CK^2	BB^{TV}	57.3	47.0	6.5	12.6	16.6	17.4
	CK^3	GS^{MTG}, BB^{TV}	69.2	61.9	6.8	8.6	6.3	16.4
	EDM^A		65.6	57.8	7.3	6.6	10.8	17.6
	EDM^M		70.1	63.7	8.6	2.7	6.5	18.5
	EDM^T		44.6	33.6	8.8	15.4	9.9	32.3
	QM		50.4	39.6	11.9	13.2	4.3	31.0
BB^{TE}	CK^1	GS^{MTG}	72.8	62.5	7.6	13.2	12.5	**4.2**
	CK^2	BB^{TV}	**83.9**	**77.1**	9.0	4.9	4.2	4.9
	CK^3	GS^{MTG}, BB^{TV}	79.7	70.8	9.7	9.0	6.3	**4.2**
	EDM^A		78.7	70.8	11.8	2.8	5.6	9.0
	EDM^M		28.9	14.6	2.1	16.0	42.4	25.0
	EDM^T		75.4	66.7	12.5	6.3	2.8	11.8
	QM		60.9	52.1	11.8	4.2	8.3	23.6

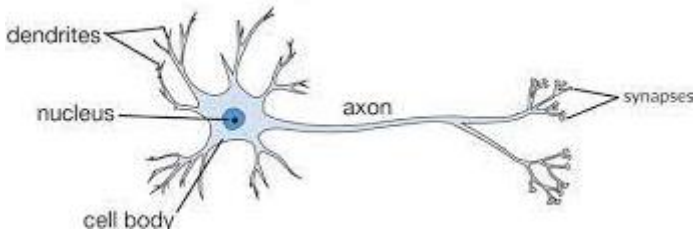
3.3 Πίνακας ακρίβειας διάφορων διαμορφώσεων εκπαίδευσης του μοντέλου [7]

ΚΕΦΑΛΑΙΟ 4

ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

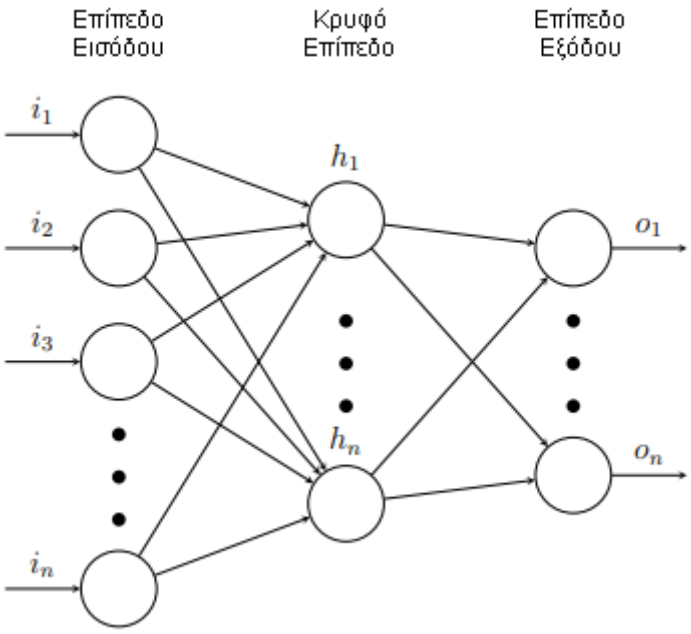
4.1 Τι είναι ένα νευρωνικό δίκτυο;

Ο όρος νευρωνικά δίκτυα αναφέρεται στην προσπάθεια να προσεγγίσουμε τον τρόπο λειτουργίας του ανθρώπινου εγκεφάλου. Ο εγκέφαλος είναι ένας πολύπλοκος, μη-γραμμικός και παράλληλος υπολογιστής με την ικανότητα να πραγματοποιεί υπολογισμούς με τεράστιες ταχύτητες. Αυτό οφείλεται στον τρόπο οργάνωσης των νευρώνων που διαθέτει. Πιο συγκεκριμένα, κατά τη γέννηση του, κατασκευάζει τους δικούς του κανόνες "εμπειρία" η οποία μεγαλώνει με τα χρόνια. Κατά τα δύο πρώτα χρόνια της ζωής του ανθρώπου, ο εγκέφαλος του βρίσκεται στην μέγιστη ανάπτυξη του, δημιουργώντας 1 εκατομμύριο συνάψεις το δευτερόλεπτο. Οι συνάψεις είναι βασικές δομικές και λειτουργικές μονάδες που μεσολαβούν στην ενδοεπικοινωνία των νευρώνων και συνδέουν τους άξονες με τους δενδρίτες [9],[10] .



4.1 Βιολογικός Νευρώνας

Τα νευρωνικά δίκτυα αποτελούνται από νευρώνες (ή αλλιώς επίπεδα): Εισόδου, Κρυφούς και Εξόδου όπως αναπαρίστανται στο σχήμα 4.2.



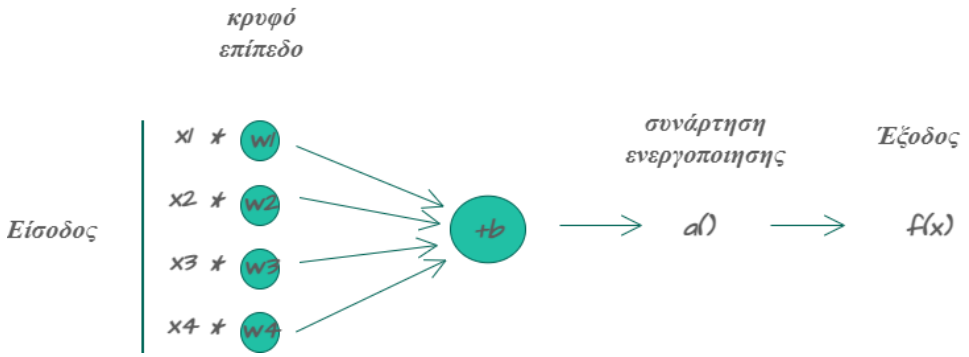
4.2 Τα διάφορα επίπεδα ενός νευρωνικού δικτύου

Κάθε νευρωνικό δίκτυο το οποίο διαθέτει τουλάχιστον ένα κρυφό επίπεδο ονομάζεται βαθύ. Επίσης, κάθε νευρώνας συνδέεται με όλους του προηγούμενους και επόμενους από αυτόν νευρώνες με συνάψεις, οι οποίες αναπαρίστανται με έναν δεκαδικό αριθμό, και ονομάζονται βάρη (W). Τα βάρη συμβάλλουν στην ενεργοποίηση κάθε επόμενου νευρώνα με τη χρήση μιας μη γραμμικής συνάρτησης ενεργοποίησης (a). Έτσι αν έχουμε σαν είσοδο ένα διάνυσμα x τότε η έξοδος f(x) ορίζεται ως εξής:

$$f(x) = a\,(Wx + b)$$

Όπου b (κατώφλι) είναι μια προαιρετική σταθερά που προστίθεται στην είσοδο της συνάρτησης ενεργοποίησης προκειμένου να μετατοπιστεί αριστερά ή δεξιά κατά b για να ταιριάζει καλύτερα στα δεδομένα προς επεξεργασία. Στην περίπτωση που θέσουμε το b = 0 τότε η συνάρτηση ενεργοποίησης γίνεται:

$$f(x) = a(Wx)$$ [10]



4.3 Μαθηματικό μοντέλο συνάρτησης ενεργοποίησης νευρώνων με $b \neq 0$

4.2 Διαδικασία εκμάθησης

Μια από τις σημαντικότερες ιδιότητες των νευρωνικών δικτύων είναι η ικανότητα τους να μαθαίνουν από το περιβάλλον τους και να βελτιώνουν τις επιδόσεις τους με την πάροδο του χρόνου. Επαναληπτικά "ελέγχονται" και ανανεώνονται οι τιμές των συναπτικών βαρών W και των κατωφλίων b προκειμένου να ολοκληρωθεί η διαδικασία της εκμάθησης. Σύμφωνα με τους Mendel και McLaren (1970) ο ορισμός της εκμάθησης ενός νευρωνικού δικτύου αναλύεται στα εξής βήματα [9]:

1. Το νευρωνικό δίκτυο ενεργοποιείται και αντιδρά στο περιβάλλον
2. Ως αποτέλεσμα γίνονται αλλαγές στις τιμές του νευρωνικού
3. Το δίκτυο δίνει διαφορετικές απαντήσεις λόγω των αλλαγών που υπέστη

Μάθηση λοιπόν, είναι η διαδικασία κατά την οποία το νευρωνικό δίκτυο προσαρμόζεται στο περιβάλλον στο οποίο βρίσκεται μέσω της επανειλημμένης ανανέωσης των παραμέτρων του. Η εκπαίδευση έχει ως κύριο στόχο τον υπολογισμό ενός διανύσματος εξόδου με την μεγαλύτερη δυνατή ακρίβεια έτσι ώστε να ανταποκρίνεται στην είσοδο του περιβάλλοντος. Η επιτυχής πραγματοποίηση της διαδικασίας αυτής ονομάζεται γενίκευση, και συμβαίνει όταν είτε για εισόδους στις οποίες έχει εκπαιδευτεί είτε για άγνωστες, το νευρωνικό δίκτυο εκτιμά σωστά διανύσματα εξόδου [9].

Κατά τη διαδικασία εκμάθησης του δικτύου τα δεδομένα εισόδου διασπώνται σε δύο σύνολα. Αρχικά στο **σύνολο εκπαίδευσης**, το οποίο τροφοδοτείται στο νευρωνικό δίκτυο για την διαδικασία της εκμάθησης επαναληπτικά. Η διαδικασία τερματίζει είτε μετά από έναν καθορισμένο αριθμό επαναλήψεων, είτε όταν επιτευχθεί κάποιος σκοπός, όπως για παράδειγμα όταν το σφάλμα μειωθεί κάτω από μια προκαθορισμένη τιμή. Στο **σύνολο δοκιμής**, το οποίο χρησιμοποιεί στοιχεία στα οποία το δίκτυο δεν έχει εκπαιδευτεί και δίνει τη δυνατότητα να ελεγχθεί η αποτελεσματικότητά του και η ικανότητα γενίκευσής του. Αυτά τα δύο ξένα μεταξύ τους σύνολα δεδομένων εισόδου επιτρέπουν να πραγματοποιηθεί συνολική αξιολόγηση της διαδικασίας εκπαίδευσης του νευρωνικού δικτύου [11].

Ανάλογα το πρόβλημα προς επίλυση που τίθεται στη διαδικασία της μάθησης μπορεί να χωριστεί σε δυο βασικές κατηγορίες:

1. Supervised learning

Το βασικό χαρακτηριστικό της επιβλεπόμενης μάθησης (supervised learning) είναι η χρήση ενός "δασκάλου". Πιο συγκεκριμένα, ο δάσκαλος γνωρίζει το περιβάλλον του

δικτύου και παρέχει δεδομένα σε ζεύγη τιμών εισόδου και εξόδου. Έτσι η εκπαίδευση πραγματοποιείται με στόχο το αποτέλεσμα του δικτύου να ταυτίζεται με την έξοδο του παραδείγματος, που αντιστοιχεί στην είσοδο που δόθηκε [9].

2. Unsupervised learning

Στη μάθηση χωρίς επίβλεψη (unsupervised learning) δεν υπάρχει εξωτερικός δάσκαλος για να επιβλέπει τη διαδικασία μάθησης και δεν υπάρχουν παραδείγματα στα οποία αντιστοιχεί μια επιθυμητή τιμή εξόδου, ώστε να εκπαιδευτεί σε αυτά το δίκτυο. Σε αυτή την περίπτωση το νευρωνικό έχει τη δυνατότητα να κωδικοποιεί μόνο του τα χαρακτηριστικά της εισόδου και οι έξοδοί του δεν έχουν μια προκαθορισμένη σημασία που τους έχει δοθεί από κάποιον εξωτερικό παράγοντα, όπως τον άνθρωπο [9].

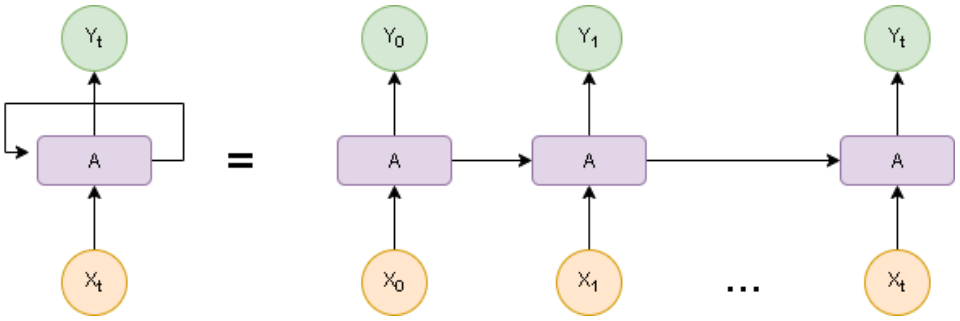
4.3 Είδη αρχιτεκτονικής νευρωνικών δικτύων

Υπάρχουν δύο βασικές αρχιτεκτονικές νευρωνικών δικτύων. Ένα από αυτά είναι τα πλήρως συνδεδεμένα νευρωνικά δίκτυα πρόσθιας τροφοδότησης. Στα τελευταία, μεταξύ δύο συνεχόμενων επιπέδων, η έξοδος του πρώτου προωθείται στην είσοδο του δεύτερου. Ουσιαστικά όλες οι τιμές εξόδου ενός επιπέδου δίνονται ως είσοδος σε όλους τους νευρώνες του επόμενου επιπέδου μέχρι να καταλήξουμε στο επίπεδο εξόδου [10].

Η δεύτερη βασική αρχιτεκτονική, είναι τα νευρωνικά δίκτυα των οποίων η είσοδος ενός επιπέδου είναι:

- Το επόμενο διάνυσμα εισόδου
- Το διάνυσμα εξόδου του ίδιου επιπέδου

Πιο συγκεκριμένα στα αναδρομικά νευρωνικά δίκτυα, αν θεωρήσουμε ότι παίρνουμε σαν είσοδο ένα σύνολο διανυσμάτων $\mathbf{x_0}, \mathbf{x_1}, \ldots, \mathbf{x_n}$ με αντίστοιχες εξόδους $\mathbf{y_0}, \mathbf{y_1}, \ldots, \mathbf{y_n}$ τότε μια έξοδος του νευρωνικού $\mathbf{y_t}$ θα αντιστοιχεί σε είσοδο όλων των διανυσμάτων $\mathbf{x_0}, \mathbf{x_1}, \ldots, \mathbf{x_t}$ και των προηγούμενων εξόδων $\mathbf{y_0}, \mathbf{y_1}, \ldots, \mathbf{y_{t-1}}$.Στο σχήμα 4.4 μπορούμε να δούμε σχηματικά τη λειτουργία των αναδρομικών νευρωνικών δικτύων.



4.4 Παράδειγμα αναδρομικού νευρωνικού δικτύου

4.4 Συνελικτικά Νευρωνικά Δίκτυα -ΣΝΔ (Convolutionalneuralnetowrk - CNN)

4.4.1 Ορισμός

Τα συνελικτικά δίκτυα αποτελούν κατηγορία των πολυεπίπεδων δικτύων πρόσθιας τροφοδότησης αλλά διαφέρουν από τις αρχιτεκτονικές των πλήρως συνδεδεμένων νευρωνικών δικτύων. Έχουν τη δυνατότητα να δέχονται ολόκληρες εικόνες σαν είσοδο με αποτέλεσμα την εμφάνιση πολύ καλών επιδόσεων σε προβλήματα ταξινόμησης και αναγνώρισης εικόνας. Τα συνήθη πλήρως συνδεδεμένα δίκτυα διασπούν την τρισδιάστατη μορφή των εικόνων και τη μετατρέπουν σε ένα διάνυσμα μιας διάστασης. Αυτή η διαδικασία όμως δεν είναι υπολογιστικά αποδοτική και αυξάνει την πολυπλοκότητα σε μεγάλο βαθμό. Η μεγάλη πρόοδος των συνελικτικών δικτύων είναι ότι έχουν την δυνατότητα να διατηρήσουν στην είσοδο την τρισδιάστατη φύση των έγχρωμων εικόνων με αποτέλεσμα την αποδοτικότερη ανάλυση τους. Εν συντομία, τα ΣΝΔ στα πρώτα τους επίπεδα μοντελοποιούν την πληροφορία σε μικρά τμήματα (περιοχές σάρωσης) και στη συνέχεια τη συνενώνουν για να δημιουργήσουν μεγαλύτερα [10],[11].

Η αρχιτεκτονική των ΣΝΔ είναι σχεδιασμένη για να αναγνωρίζει διάφορα είδη παραμορφώσεων σε δισδιάστατα σχήματα. Οι μορφές περιορισμών που περιλαμβάνουν τα συνελικτικά δίκτυα είναι η **διεξαγωγή χαρακτηριστικών**, όπου κάθε νευρώνας λαμβάνει είσοδο από ένα τοπικό δεκτικό πεδίο με αποτέλεσμα να εξάγει τοπικά χαρακτηριστικά. Έπειτα από την εξαγωγή του χαρακτηριστικού από τα δεδομένα, η ακριβής του θέση γίνεται λιγότερη σημαντική εφόσον διατηρείται σχετική διάταξη μεταξύ των χαρακτηριστικών. Επιπλέον μορφή περιορισμού είναι η **αντιστοίχιση χαρακτηριστικών** στην οποία κάθε επίπεδο του δικτύου αποτελείται από πολλούς χάρτες χαρακτηριστικών (feature maps) [11]. Σε κάθε τέτοιο χάρτη όλοι οι νευρώνες μοιράζονται τα ίδια συναπτικά βάρη με αποτέλεσμα τα εξής πλεονεκτήματα:

- Επιτυγχάνεται η μη ευαισθησία του δικτύου ως προς τη μετατόπιση, αφού στον χάρτη χαρακτηριστικών χρησιμοποιούμε συνέλιξη με έναν πυρήνα (kernel) μικρού μεγέθους [11].
- Μείωση του αριθμού των ελεύθερων παραμέτρων.

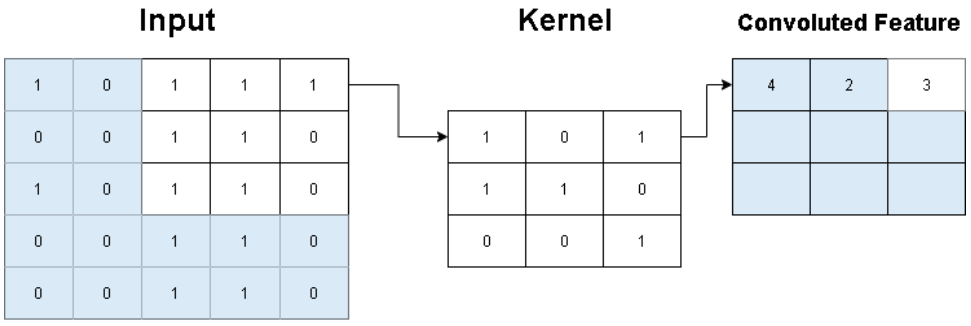
Τέλος, κατά την **υποδειγματοληψία** κάθε συνελικτικό επίπεδο ακολουθείται από ένα υπολογιστικό επίπεδο το οποίο μειώνει την ανάλυση του χάρτη χαρακτηριστικών. Η λειτουργία αυτή έχει σαν αποτέλεσμα την μείωση της ευαισθησίας του χάρτη χαρακτηριστικών εξόδου σε παραμορφώσεις και μετατοπίσεις [11].

4.4.2 Είδη επιπέδων των Συνελικτικών Νευρωνικών Δικτύων

Ένα ΣΝΔ μπορεί να έχει τρεις τύπους νευρωνικών επιπέδων τα οποία λειτουργούν διαφορετικά κατά την εκτέλεση τους. Ειδικότερα αυτά είναι το συνελικτικό επίπεδο, το επίπεδο υποδειγματοληψίας ή μέγιστης συγκέντρωσης και το πλήρως συνδεδεμένο επίπεδο [12].

Ένα convolutional layer βασίζεται στην πράξη της συνέλιξης και είναι το βασικό χαρακτηριστικό των συνελικτικών δικτύων. Αποτελείται από φίλτρα το καθένα από τα οποία έχει σταθερές τιμές και διαστάσεις αναλόγως την είσοδο που παίρνουν, δηλαδή για δισδιάστατη είσοδο τα φίλτρα είναι τετραγωνικά, για τρισδιάστατη κυβικά κ.ο.κ. Τα φίλτρα διαπερνούν σε βάθος ολόκληρη την είσοδο και ο σκοπός τους είναι η αναγνώριση και η διεξαγωγή χαρακτηριστικών. Στην ουσία ψάχνουν να βρουν "μοτίβα" στα δεδομένα εισόδου, δηλαδή σε ένα μοντέλο αναγνώρισης εικόνας το νευρωνικό με τα συνελικτικά επίπεδα προσπαθεί να διακρίνει γραμμές, καμπύλες, γωνίες και άλλες ιδιαιτερότητες της εκάστοτε εισόδου [12].

Στη συνέχεια έχουμε τα επίπεδα υποδειγματοληψίας (pooling layer) τα οποία συνήθως τοποθετούνται ακριβώς μετά από συνελικτικά επίπεδα. Σκοπός ενός τέτοιου επιπέδου είναι να μειώσει το μέγεθος της εξόδου ενός συνελικτικού επιπέδου ώστε να εστιάσει το δίκτυο στην εύρεση χαρακτηριστικών και να επιτευχθεί γενίκευση. Επομένως παίρνοντας σαν είσοδο έναν πίνακα χαρακτηριστικών (έξοδος ενός συνελικτικού επιπέδου) δίνουν ως έξοδο μια μοναδική τιμή που είναι είτε η μέγιστη τιμή του πίνακα (max-pooling layers) είτε η μέση τιμή του (average-pooling layers) [12].

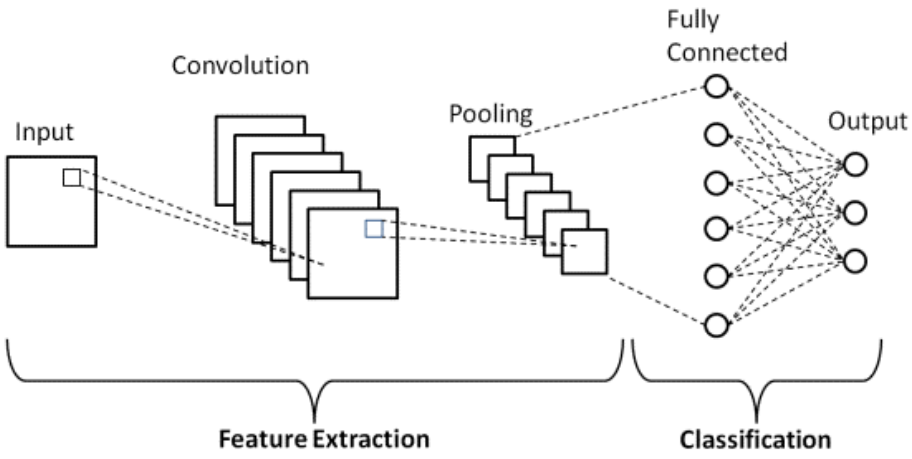


4.5 Σχηματική αναπαράσταση της εξαγωγής χαρακτηριστικών

Τέλος έχουμε τα πλήρως συνδεδεμένα επίπεδα τα οποία συνήθως συγκεντρώνονται στο τέλος ενός συνελικτικού νευρωνικού δικτύου και σε αντίθεση με τα συνελικτικά επίπεδα παίρνουν σαν είσοδο όλες τις τιμές εξόδου του προηγούμενου επιπέδου. Τα

επίπεδα αυτά βρίσκονται στο τέλος διότι η έξοδός τους φανερώνει το πόσο συχνά εμφανίζεται ένα “μοτίβο” που πήραν ως είσοδο από το τελευταίο συνελικτικό επίπεδο [12].

Για παράδειγμα στη παρούσα πτυχιακή έχει σχεδιαστεί ένα δίκτυο το οποίο παίρνει σαν είσοδο ένα φασματογράφημα διαστάσεων 105 x 600, με rgb τιμές σε κάθε pixel. Συνεπώς το σύνολο των δεδομένων που δέχεται σαν είσοδο το πρώτο επίπεδο του νευρωνικού είναι 105 x 600 x 3. Αυτό με τη σειρά του παράγει σαν έξοδο δεδομένα που έχουν τρισδιάστατη μορφή και τροφοδοτούνται στα επόμενα επίπεδα. Ο όγκος των δεδομένων σταδιακά μικραίνει λόγω των επιπέδων υποδειγματοληψίας και τελικά παράγεται μια πρόβλεψη αποτελούμενη από 12 κλάσεις.



4.6 Γραφική αναπαράσταση ενός ΣΝΔ

ΚΕΦΑΛΑΙΟ 5

ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ ΚΑΙ PYTORCH

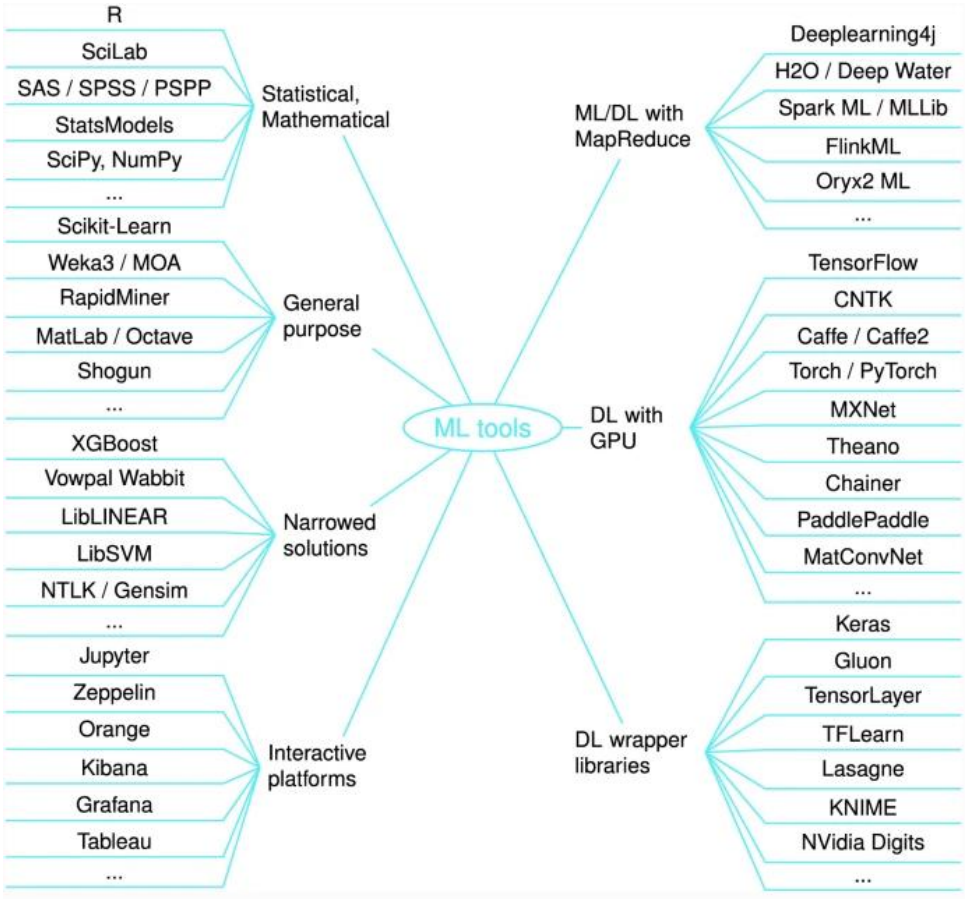
5.1 Εισαγωγή στη Μηχανική Μάθηση

Ένα από τα βασικότερα χαρακτηριστικά του ανθρώπου και της νοημοσύνης του είναι η ικανότητα του να μαθαίνει. Παρά το γεγονός ότι η διαδικασία της μάθησης δεν έχει καταφέρει να γίνει πλήρως κατανοητή, υπάρχει η δυνατότητα ανάπτυξης αρκετά πολύπλοκων υπολογιστικών συστημάτων που προσεγγίζουν τον τρόπο που μαθαίνει ο άνθρωπος. Σύμφωνα με τον Tom M.Mitchell(1997) η Μηχανική μάθηση ορίζεται ως: «Ένα πρόγραμμα υπολογιστή λέγεται ότι μαθαίνει από εμπειρία Ε ως προς μια κλάση εργασιών Τ και ένα μέτρο επίδοσης Ρ, αν η επίδοσή του σε εργασίες της κλάσης Τ, όπως αποτιμάται από το μέτρο Ρ, βελτιώνεται με την εμπειρία Ε». Σύμφωνα με τον ορισμό αυτό η μηχανική μάθηση έχει σκοπό τη δημιουργία υπολογιστικών μεθόδων που θα χρησιμοποιούν την εμπειρία προκειμένου να βελτιώσουν τις επιδόσεις και να κάνουν επιτυχημένες προβλέψεις. Ως εμπειρία θεωρούμε διαθέσιμες πληροφορίες που έχουμε από το παρελθόν και συνήθως έχουν την μορφή των συνόλων εκπαίδευσης, ή δεδομένα που έχουν παραχθεί από αλληλεπίδραση με το περιβάλλον. Σε κάθε περίπτωση, η επιτυχία της πρόβλεψης καθορίζεται κατά κύριο λόγο από την ποιότητα και το μέγεθος της πληροφορίας που διαθέτουμε.

5.2 Frameworks και Pytorch

Τα νευρωνικά δίκτυα αποτελούν ένα σημαντικό κομμάτι του κλάδου της μηχανικής μάθησης. Ο αριθμός των αλγορίθμων αλλά και των διαφορετικών υλοποιήσεων που χρησιμοποιούνται για την πραγματοποίηση της μηχανικής μάθησης και κατ' επέκταση την εκπαίδευση των νευρωνικών δικτύων είναι τεράστιος. Για την επίτευξη των παραπάνω διαδικασιών απαιτείται η ανάπτυξη και χρήση προγραμματιστικών περιβαλλόντων ικανών να αναλύουν περίπλοκα δεδομένα και λειτουργίες. Μερικά παραδείγματα είναι: οι πλατφόρμες ανάλυσης, τα συστήματα προβλέψεων, η επεξεργασία κειμένου, εικόνας και

ήχου. Τα προγραμματιστικά περιβάλλοντα αυτά ονομάζονται machine learning frameworks και είναι software "βιβλιοθήκες" που χρησιμοποιούνται στην δική μας περίπτωση για την εύκολη και γρήγορη ανάπτυξη ενός νευρωνικού δικτύου. Παραδείγματα τέτοιων κατηγοριοποιημένων frameworks βλέπουμε στο σχήμα 5.1 [13].



5.1 Machine learning tools [14]

Στην παρούσα πτυχιακή για την ανάπτυξη των νευρωνικών δικτύων χρησιμοποιήθηκε το pytorch. Το pytorch είναι το πιο διαδεδομένο open-source framework γραμμένο σε python, υποστηρίζει GPU-accelerated εκμάθηση και είναι βασισμένο στο torch. Συγκεκριμένα το torch αποτελεί ένα επιστημονικό υπολογιστικό framework που παρέχει αλγορίθμους μηχανικής μάθησης. Ο πυρήνας του έχει στηριχθεί στη βιβλιοθήκη tensor, η οποία υποστηρίζει λειτουργίες CPU (openMP) και GPU (CUDA). Επιπρόσθετα, είναι μια δομή δεδομένων που επιτρέπει την αναπαράσταση αριθμών σε πολλές διαστάσεις και στη δική μας περίπτωση αποτελεί την είσοδο του νευρωνικού δικτύου. Κάποια από τα βασικά χαρακτηριστικά και λειτουργίες που παρέχει το pytorch είναι [15]:

- Η μηχανή αυτόματης διαφοροποίησης autograd για την εκμάθηση των νευρωνικών δικτύων
- Αυτόματη διαφοροποίηση για NumPy και SciPy
- Ευέλικτο προγραμματιστικό περιβάλλον για ανάπτυξη σε python
- Υποστηρίζει το Open Neural Network Exchange (ONNX) format

Το pytorch, πέρα από τις βασικές λειτουργίες που χρειάζεται ένα νευρωνικό δίκτυο για να λειτουργήσει, παρέχει και επιπλέον βιβλιοθήκες επεξεργασίας δεδομένων. Είναι σημαντικό να αναφερθούν τα πακέτα που χρειάστηκαν για την υλοποίηση των μοντέλων [15]:

- Torch για την εισαγωγή των παραμέτρων και χρήσιμων συναρτήσεων του νευρωνικού δικτύου όπως του optimizer και του loss function
- Torchaudio για την επεξεργασία αρχείων ήχου (mp3, WAV) και την μετατροπή τους σε spectrogram
- Numpy για τις μαθηματικές πράξεις μεταξύ των tensors και την διαχείρισή τους

Το κάθε πακέτο παρέχει έτοιμες δομές και συναρτήσεις που βοηθούν σημαντικά στην υλοποίηση οποιουδήποτε προβλήματος μηχανικής μάθησης και πιο συγκεκριμένα μοντέλα νευρωνικών δικτύων.

ΚΕΦΑΛΑΙΟ 6

DATASETS

6.1 Εισαγωγή

Όπως έχει προαναφερθεί τα νευρωνικά δίκτυα μαθαίνουν μέσω μιας διαδικασίας που ονομάζεται επιβλεπόμενη μάθηση. Αυτή, προϋποθέτει την ύπαρξη ή ανάπτυξη δεδομένων τέτοιων ώστε το νευρωνικό δίκτυο να μπορεί να συλλέξει χαρακτηριστικά τα οποία μέσω επαναληπτικών διαδικασιών θα του μαθαίνουν να προβλέπει το μουσικό κλειδί ενός κομματιού. Συμπερασματικά υπάρχει η ανάγκη κάποιων σετ από δεδομένα (datasets) που θα παρέχουν αρχεία με μουσικά κομμάτια και τα αντίστοιχα μουσικά κλειδιά τους.

6.2 Datasets που χρησιμοποιήθηκαν

Για την εκπαίδευση των νευρωνικών δικτύων έχουν χρησιμοποιηθεί δύο datasets. Αυτά είναι το GiantSteps Key Dataset [16] και το GiantSteps MTG Key dataset [17]. Τα περιεχόμενα του πρώτου dataset απαρτίζονται από:

- 604 audio files τραγουδιών σε μορφή mp3
- Αρχείο σημειώσεων κάθε μουσικού κομματιού που αφορούν
    - Το είδος
    - Και το κλειδί τους

Το GiantSteps MTG Key dataset αντίστοιχα αποτελείται από:

- 1486 μουσικά κομμάτια σε μορφή mp3
- Annotation file που αφορά το μουσικό τους κλειδί.

Τα audio files αποτελούνται από μουσικά κομμάτια διάρκειας 120 δευτερολέπτων και αφορούν κυρίως ηλεκτρονική μουσική (EDM). Και τα δύο datasets χρησιμοποιούνται στις διαδικασίες εκμάθησης και αξιολόγησης των νευρωνικών δικτύων.

6.3 Προ επεξεργασία Δεδομένων

Τα νευρωνικά δίκτυα όπως έχει αναφερθεί δέχονται σαν είσοδο “αριθμούς”. Πιο συγκεκριμένα τα συνελικτικά νευρωνικά δίκτυα επεξεργάζονται εικόνες, επομένως είναι απαραίτητο κατά την είσοδό τους τα pixels να μετατραπούν σε numpy arrays (πολυδιάστατοι πίνακες από αριθμούς). Αυτό καθιστά αναγκαία την χρήση εικόνων που έχουν τη δυνατότητα να περιγράψουν σημαντικά χαρακτηριστικά ενός μουσικού κομματιού. Το spectrogram (φασματογράφημα) επειδή αναπαριστά τις διακυμάνσεις των συχνοτήτων, και κατ’ επέκταση όλες τις νότες και τις συγχορδίες ενός τραγουδιού, συναρτήσει του χρόνου αποτελεί την είσοδο των νευρωνικών δικτύων που αναπτύχθηκαν. Τα βήματα της προ επεξεργασίας είναι τα εξής:

1. Μετατροπή των mp3 αρχείων σε wav

2. Δημιουργία ενός spectrogram για κάθε wav αρχείο

6.3.1 Μετατροπή soundfiles σε wav format

Μετά την λήψη των mp3 αρχείων από τα δύο datasets αναπτύσσεται ένα script σε python για την μετατροπή τους σε wav. Το αρχείο αυτό χρησιμοποιεί μια βιβλιοθήκη που ονομάζεται pydub και παρέχει δυνατότητες επεξεργασίας ήχου, προκειμένου να πραγματοποιηθεί η μετατροπή των αρχείων στην επιθυμητή μορφή. Η μετατροπή αυτή είναι απαραίτητη γιατί η βιβλιοθήκη librosa η οποία χρησιμοποιείται παρακάτω μπορεί να δημιουργήσει φασματογραφήματα μόνο από αρχεία τέτοιου τύπου.

```python
import os
import pydub
from pydub import AudioSegment

pydub.AudioSegment.converter = r"C:\ffmpeg\bin\ffmpeg.exe"

path = "C:/users/Michalis Zeakis/Desktop/university/ptyxiaki/datasets/our_dataset"
#path = "C:/Users/Michalis Zeakis/Desktop/university/ptyxiaki/datasets/test_dataset_mp3"

#Change working directory
os.chdir(path)

audio_files = os.listdir()

# we dont need the number of files in the folder, just iterate over them directly using:
for file in audio_files:

    #spliting the file into the name and the extension
    name, ext = os.path.splitext(file)
    mp3_sound = AudioSegment.from_mp3(file)
    #rename them using the old name + ".wav"
    mp3_sound.export("C:/Users/Michalis Zeakis/Desktop/university/ptyxiaki/datasets/test_dataset_wav/{0}.wav".format(name), format="wav")

print("Done !!")
```

6.1 Κώδικας μετατροπής mp3 σε wav

6.3.2 Δημιουργία Spectrogram

Για την δημιουργία του φασματογραφήματος χρειάστηκε η βιβλιοθήκη librosa. Αυτή αποτελεί ένα πακέτο python το οποίο δίνει τη δυνατότητα ανάλυσης και επεξεργασίας ήχου. Συγκεκριμένα από τη βιβλιοθήκη αυτή, χρησιμοποιήθηκε η συνάρτηση melspectrogram. Αυτή παίρνει σαν είσοδο το wav αρχείο ήχου, το sample rate και τις επιθυμητές διαστάσεις του φασματογραφήματος (105x600) και επιστρέφει ένα numpy array που εκφράζει το αντίστοιχο spectrogram. Τα ΣΝΔ πραγματοποιούν πράξεις με τις τιμές των pixels των εικόνων που δέχονται σαν είσοδο. Προκειμένου να απλοποιηθούν οι πράξεις αυτές μετατρέπουμε τις τιμές ενός pixel, οι οποίες κυμαίνονται από 0 έως 255, σε μια κλίμακα από 0 έως 1. Στη συνέχεια γίνεται flip του φασματογραφήματος προκειμένου να μετακινήσουμε τις χαμηλές συχνότητες στο κάτω μέρος της εικόνας. Τέλος κάνουμε όλα τα pixel της εικόνας πιο μαύρα. Οι αλλαγές αυτές συμβάλλουν κυρίως στην καλύτερη οπτικοποίηση των φασματογραφημάτων για να είναι ευκολότερα κατανοητά στον άνθρωπο.

```python
import librosa
import numpy as np
import skimage.io
import os
import numpy

def scale_minmax(X, min=0.0, max=1.0):
    #scale the values between 0 - 1
    X_std = (X - X.min()) / (X.max() - X.min())
    X_scaled = X_std * (max - min) + min
    return X_scaled

def spectrogram_image(y, sr, out, hop_length, n_mels):
    # use log-melspectrogram
    mels = librosa.feature.melspectrogram(y=y, sr=sr, n_mels=n_mels,
                                          n_fft=hop_length*2, hop_length=hop_length)
    mels = numpy.log(mels + 1e-9) # add small number to avoid log(0)

    # min-max scale to fit inside 8-bit range
    img = scale_minmax(mels, 0, 255).astype(np.uint8)
    img = numpy.flip(img, axis=0) # put low frequencies at the bottom in image
    img = 255-img # invert. make black==more energy

    # save as PNG
    skimage.io.imsave(out, img)

if __name__ == '__main__':
    # settings
    hop_length = 4096 # number of samples per time-step in spectrogram
    n_mels = 105 # number of bins in spectrogram. Height of image
    time_steps = 599 # number of time-steps. Width of image

    # extract a fixed length window
    start_sample = 0 # starting at beginning
    length_samples = time_steps*hop_length


    # load audio. Using example from librosa
    path = "C:/Users/Michalis Zeakis/Desktop/university/ptyxiaki/datasets/test_dataset_wav"

    #path = "C:/test"
    os.chdir(path)
    audio_files = os.listdir()
    for file in audio_files:
        name, ext = os.path.splitext(file)
        y, sr = librosa.load(file, sr=44100)
        out = 'C:/outs/testSpects/{0}.png'.format(name)
        window = y[start_sample:start_sample+length_samples]
        # convert to PNG
        spectrogram_image(window, sr=sr, out=out, hop_length=hop_length, n_mels=n_mels)



    print('wrote file', out)
```

6.2 Κώδικας δημιουργίας spectrogram

6.4 Training & Testing datasets

Μέχρι στιγμής για κάθε αρχείο ήχου έχει δημιουργηθεί το αντίστοιχο spectrogram και το annotation file για το μουσικό κλειδί του. Είναι απαραίτητο λοιπόν να σχεδιαστεί μια δομή δεδομένων που θα περιέχει όλη την πληροφορία συγκεντρωμένη, δηλαδή μια αντιστοιχία των spectrogram με το μουσικό τους κλειδί.

```
[9]: print (key_training_data[0])

     [array([[[0.47843137, 0.47843137, 0.47843137],
              [0.47843137, 0.47843137, 0.47843137],
              [0.51764706, 0.51764706, 0.51764706],
              ...,
              [0.84705882, 0.84705882, 0.84705882],
              [0.84705882, 0.84705882, 0.84705882],
              [0.83921569, 0.83921569, 0.83921569]],

             [[0.41568627, 0.41568627, 0.41568627],
              [0.42352941, 0.42352941, 0.42352941],
              [0.46666667, 0.46666667, 0.46666667],
              ...,
              [0.97254902, 0.97254902, 0.97254902],
              [0.90980392, 0.90980392, 0.90980392],
              [0.87058824, 0.87058824, 0.87058824]],

             [[0.42745098, 0.42745098, 0.42745098],
              [0.42352941, 0.42352941, 0.42352941],
              [0.45882353, 0.45882353, 0.45882353],
              ...,
              [0.76862745, 0.76862745, 0.76862745],
              [0.82745098, 0.82745098, 0.82745098],
              [0.85882353, 0.85882353, 0.85882353]],

             ...,

             [[0.11372549, 0.11372549, 0.11372549],
              [0.09019608, 0.09019608, 0.09019608],
              [0.06666667, 0.06666667, 0.06666667],
              ...,
              [0.08235294, 0.08235294, 0.08235294],
              [0.16470588, 0.16470588, 0.16470588],
              [0.14117647, 0.14117647, 0.14117647]],

             [[0.09019608, 0.09019608, 0.09019608],
              [0.10196078, 0.10196078, 0.10196078],
              [0.06666667, 0.06666667, 0.06666667],
              ...,
              [0.02352941, 0.02352941, 0.02352941],
              [0.03137255, 0.03137255, 0.03137255],
              [0.05882353, 0.05882353, 0.05882353]],

             [[0.1372549 , 0.1372549 , 0.1372549 ],
              [0.18431373, 0.18431373, 0.18431373],
              [0.29803922, 0.29803922, 0.29803922],
              ...,
              [0.20392157, 0.20392157, 0.20392157],
              [0.12941176, 0.12941176, 0.12941176],
              [0.10588235, 0.10588235, 0.10588235]]]), tensor([0])]
```

6.3 Απεικόνιση του dataset σαν δομή του PyTorch

Κατά τον σχηματισμό αυτής της δομής στη πραγματικότητα ορίζουμε έναν αριθμό για κάθε πιθανή πρόβλεψη. Πιο συγκεκριμένα στην περίπτωση του νευρωνικού δικτύου πρόβλεψης κλίμακας θέτουμε την τιμή 0 για την minor κλίμακα και το 1 για την major. Αντίστοιχα για το νευρωνικό δίκτυο πρόβλεψης τονικότητας έχουμε τιμές από 0 έως 11 για τους τόνους C, C#, D, D#, E, F, F#, G, G#, A, A#, B. Επιπλέον κάθε spectrogram αναπαρίσταται στη μορφή ενός numpy array προκειμένου να είναι εφικτή η τροφοδότηση τους στα δίκτυα. Τέλος διαχωρίζεται το dataset σε υποσύνολα εκπαίδευσης και δοκιμής,

παίρνοντας το 90% του για την διαδικασία της εκπαίδευσης και το υπόλοιπο 10% για την αξιολόγηση.

6.5 Data augmentation

Data augmentation είναι οι τεχνικές που χρησιμοποιούνται για την αύξηση του αριθμού των δεδομένων που είναι διαθέσιμα σε ένα dataset. Στην περίπτωση μας, το dataset αποτελείται από 2088 μουσικά κομμάτια, αριθμός ο οποίος είναι αρκετά μικρός για την επίτευξη γενικευμένης μάθησης από τα νευρωνικά μας δίκτυα. Οι μέθοδοι που χρησιμοποιήθηκαν για την αύξηση των δεδομένων είναι:

1. Pitch shift
2. Square Crop
3. Center Crop

6.5.1 Pitch shift

Η μέθοδος pitch shift αφορά την μετατόπιση ολόκληρου του μουσικού κομματιού κατά ένα ημιτόνιο μετά, αλλάζοντας όλες τις νότες του με τον ίδιο τρόπο. Έτσι αν εφαρμόσουμε pitch shift σε ένα wav αρχείο με κλειδί C major θα λάβουμε σαν έξοδο ένα κομμάτι μουσικής σε C# major. Αυτή η διαδικασία πραγματοποιείται επαναληπτικά καλύπτοντας και τις 11 νότες μιας οκτάβας και επιτυγχάνεται με την χρήση της συνάρτησης effects.pitch_shift() της βιβλιοθήκης librosa. Επομένως για κάθε μουσικό κομμάτι δημιουργούνται 11 καινούρια spectrograms έχοντας συνολικά 11*2088 + 2088(τα αρχικά) = 25056 αρχεία για το dataset.

6.5.2 Square crop

Η μέθοδος square crop αφορά επεξεργασία εικόνας και όχι ήχου σε αντίθεση με την τεχνική pitch shift. Συγκεκριμένα για κάθε φασματογράφημα που έχει δημιουργηθεί και έχει διαστάσεις 105x600, χωρίζεται σε 5 με ίσες διαστάσεις (105x105). Διασπάται δηλαδή στα εξής μέρη:

1. [0:105, 0:105]
2. [0:105, 105:210]
3. [0:105, 210:315]
4. [0:105, 315:420]
5. [0:105, 420:525]

Με αυτό τον τρόπο τα δεδομένα πενταπλασιάζονται, αλλά έχουν μικρότερες διαστάσεις με αποτέλεσμα να αλλάζουν και αυτές στην είσοδο των νευρωνικών δικτύων. Συγκεκριμένα: 5*2088 = 10440, και αν έχει πραγματοποιηθεί pitch shift πρώτα : 25056*5 = 125280.

6.5.3 Center Crop

Η μέθοδος center crop αφορά και αυτή επεξεργασία εικόνας διαφορετικής όμως από εκείνη του square crop. Σε αυτή την περίπτωση κάθε ένα spectrogram χωρίζεται με τον ίδιο τρόπο σε δύο, δηλαδή στις διαστάσεις [0:105, 0:299] και [0:105, 300:600]. Αφού έχουν παραχθεί τα δύο καινούργια φασματογραφήματα ακολουθεί η διαδικασία "τεντώματος" στις διαστάσεις 105x600, έχοντας έτσι τριπλασιάσει τα δεδομένα μας.

6.6 Εφαρμογή τεχνικών data augmentation

Οι τρεις τεχνικές που εξετάσαμε παρουσίαζαν σημαντική αύξηση των δεδομένων με τον συνδυασμό pitch shift με square crop να δίνει τα περισσότερα δεδομένα για τον σχηματισμό των dataset. Καθώς όμως ακολούθησαν πειραματισμοί με αυτά παρατηρήθηκε ότι μόνο η μέθοδος pitch shift βελτίωσε σημαντικά την αποδοτικότητα και την ακρίβεια των νευρωνικών δικτύων. Συγκεκριμένα η μέθοδος αυτή κάνει συχνοτικές μετατροπές δίνοντας ως αποτέλεσμα τελείως καινούρια αρχεία ενώ οι τεχνικές αποκοπής (crop) προσφέρουν ουσιαστικά το ίδιο αρχείο ήχου πολλές φορές.

6.7 Κώδικας δημιουργίας dataset

Αρχικά, για την δημιουργία της δομής που θα αποτελεί το dataset με όλα τα φασματογραφήματα και την αντιστοίχιση με τη τονικότητα του κλειδιού τους ή την κλίμακα του, χρησιμοποιείται το annotation file. Το αρχείο αυτό περιέχει όλα τα ονόματα

των φασματογραφημάτων και των κλειδιών τους. Στη συνέχεια, καθώς διατρέχονται σειριακά τα περιεχόμενα του "φορτώνονται" τα spectrogram, τα οποία μετατρέπονται σε numpy array και αποθηκεύονται σε μια λίστα ακολουθούμενα από τα annotations που χρειάζονται (κλίμακα ή τονικότητα ανάλογα το νευρωνικό). Μετά το τέλος της επαναληπτικής διαδικασίας αυτής τα δεδομένα της λίστας "ανακατεύονται" και αποθηκεύονται.

```python
import os
import cv2
import numpy as np
from tqdm import tqdm
import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim

import warnings
warnings.filterwarnings("ignore", category=np.VisibleDeprecationWarning)

key_training_data = []

class Labeling():

    def make_training_data():


        keys = ['C','C#','Db','D','D#','Eb','E','F','F#','Gb','G','G#','Ab','A','A#','Bb','B']
        keyIndex = [0, 1, 1, 2, 3, 3, 4, 5, 6, 6, 7, 8, 8, 9, 10, 10, 11]
        keys_counters= [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
        path = "C:/outs/spects4096"
        #path = "C:/outs/square4096"
        #path = "C:/outs/our_spects"
        os.chdir(path)
        spectrograms = os.listdir()
        key_count = [0] * 24
        annotations = open("C:/outs/shiftedAnnotations4096.txt", "r")
        #annotations = open("C:/outs/square4096.txt", "r")
        while True:
            line = annotations.readline()
            if not line:
                break;


            file, key, chord = line.split(' ', 2)
            file = file.strip()
            key = key.strip()
            chord = chord.strip()
```

6.4 Κώδικας δημιουργίας dataset τονικότητας

```
            file = file + ".LOFI.png"
            spect_path = os.path.join(path, file)
            spect_path = cv2.imread(spect_path)

            if spect_path is None :
                continue;


            for j,i in enumerate(keys):

                temp = keyIndex[j]
                if keys_counters[temp] > 999 :
                    continue;

                if key == i:
                    key_training_data.append([np.array(spect_path)/255, torch.LongTensor([keyIndex[j]])])
                    keys_counters[temp] += 1



        np.random.shuffle(key_training_data)
        print("saving data...\n")
        np.save("1000-test_training_data.npy", key_training_data)

        print(keys_counters)

if __name__ == '__main__':
    Labeling.make_training_data()


```

6.4 Κώδικας δημιουργίας dataset κλίμακας

ΚΕΦΑΛΑΙΟ 7

ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ

7.1 Εισαγωγή

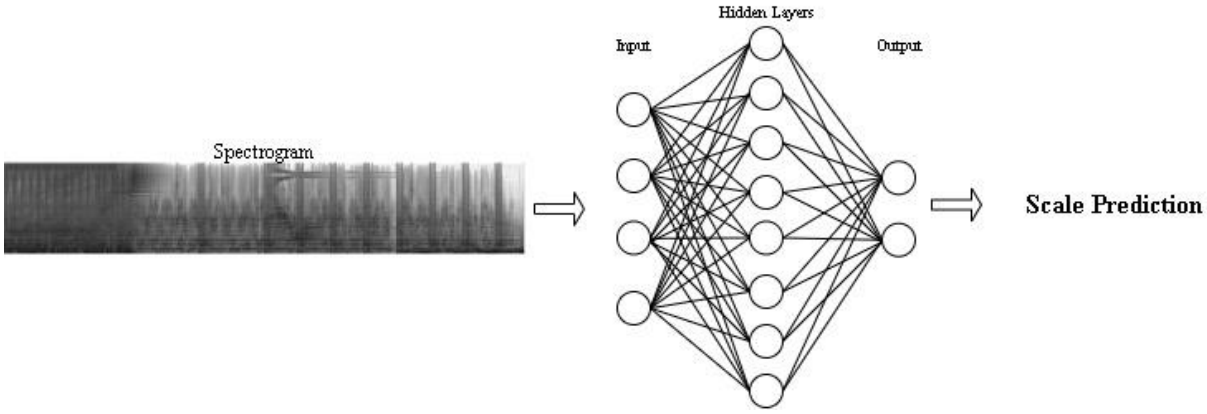
Η εκπαίδευση αποτελεί ιδιότητα πρωτεύουσας σημασίας στα νευρωνικά δίκτυα, και όπως έχουμε αναφέρει η βελτίωση της απόδοσης επιτυγχάνεται με την πάροδο του χρόνου σύμφωνα με κάποιο προκαθορισμένο μέτρο όπως είναι η βελτιστοποίηση των βαρών και των κατωφλιών του κάθε νευρώνα. Εξίσου σημαντική διαδικασία για την σωστή εκμάθηση των νευρωνικών αποτελεί η αξιολόγηση, κατά την οποία πραγματοποιείται έλεγχος και σύγκριση των προβλέψεων ενός νευρωνικού με αυτές που έχουμε προκαθορίσει ότι περιμένουμε σαν έξοδο για συγκεκριμένη είσοδο [18]. Η εκπαίδευση του νευρωνικού δικτύου αποτελείται από δύο φάσεις, την εμπρόσθια (forward) και την οπίσθια (backward). Κατά την forward φάση η είσοδος του περνά από όλα τα επίπεδα του νευρωνικού στα οποία και αποθηκεύονται όλα τα απαραίτητα δεδομένα προκειμένου στη backward φάση να γίνει η ανανέωση των βαρών ανάλογα με το βαθμό λάθους της πρόβλεψης. Για να πετύχουμε καλύτερη και πιο γενικευμένη εκμάθηση του δικτύου "ψάχνουμε" τις καλύτερες τιμές για τις μεταβαλλόμενες παραμέτρους (W, b) αφού πρώτα ορίσουμε τις υπέρ-παραμέτρους του [19]. Οι τελευταίες χωρίζονται σε 2 κατηγορίες:

1. Σε αυτές που καθορίζουν την δομή του δικτύου (π.χ. αριθμός κρυφών επιπέδων, αριθμός νευρώνων κάθε επιπέδου κτλ.)
2. Σε αυτές που καθορίζουν τον τρόπο με τον οποίο θα εκτελεστεί η εκπαίδευση

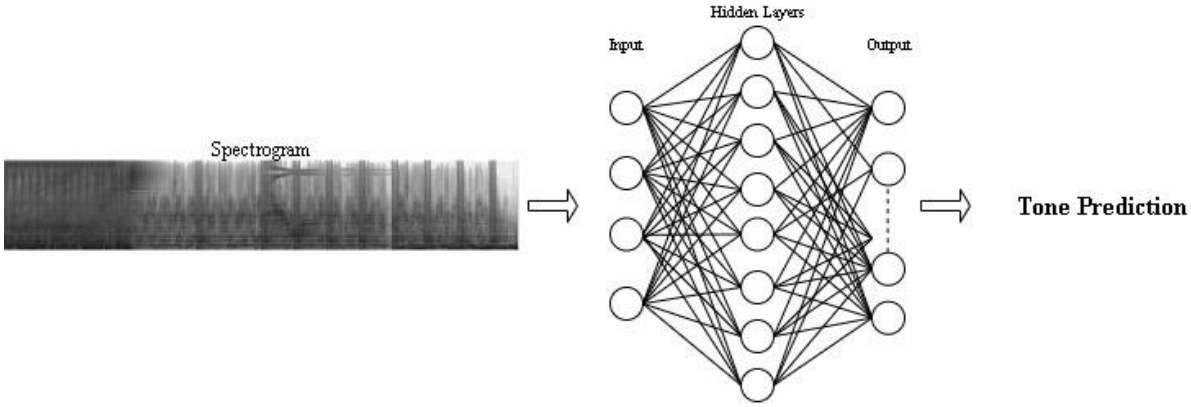
7.2 Μοντέλο και δομή νευρωνικών δικτύων

Το μοντέλο μας αποτελείται από δύο συνελικτικά νευρωνικά δίκτυα για την αναγνώριση του μουσικού κλειδιού ενός κομματιού μουσικής, το ένα προβλέπει την τονικότητα (C, C#, D, D#, E, F, F#, G, G#, A, A#, B) του κλειδιού και το δεύτερο την κλίμακα (minor, major). Το πρώτο αφορά ένα πολυταξικό νευρωνικό δίκτυο πρόβλεψης 12 κλάσεων όπως φαίνεται στο σχήμα 3.8. Αυτό σημαίνει πως το επίπεδο εξόδου του θα

έχει 12 νευρώνες και συνεπώς 12 διαφορετικές τιμές πρόβλεψης. Το δεύτερο νευρωνικό δίκτυο είναι δυαδικής πρόβλεψης, δηλαδή θα έχει 2 νευρώνες εξόδου και τιμές πρόβλεψης, σχήμα 3.7. Έτσι δίνοντας ως είσοδο το ίδιο φασματογράφημα και στα δύο νευρωνικά παίρνουμε συνολικά τις 24 κλάσεις που αποτελούν όλα τα διαφορετικά κλειδιά που επιθυμούμε. Το μοντέλο αναπτύχθηκε σύμφωνα με την επιβλεπόμενη μάθηση (supervised learning) γνωρίζοντας το αποτέλεσμα των δεδομένων εισαγωγής. Η επιλογή των χαρακτηριστικών και παραμέτρων για τα παραπάνω νευρωνικά δίκτυα είναι αποτέλεσμα δοκιμών και πειραματισμού [20].



7.1 Μοντέλο νευρωνικού δικτύου πρόβλεψης κλίμακας

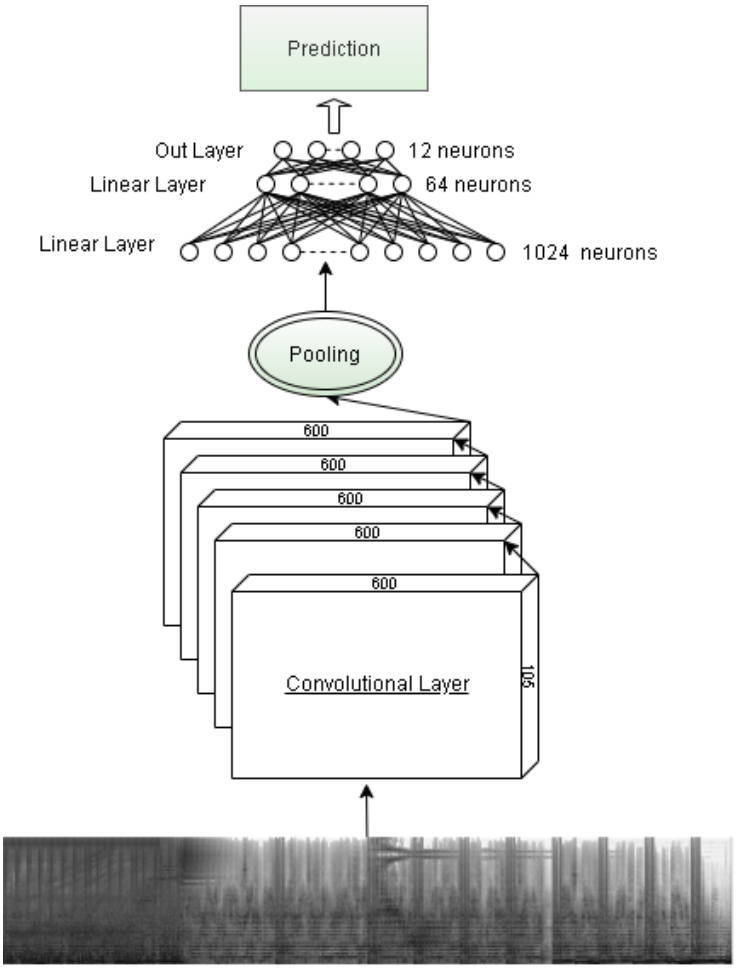


7.2 Μοντέλο νευρωνικού δικτύου πρόβλεψης τονικότητας

7.2.1 Χαρακτηριστικά νευρωνικού δικτύου αναγνώρισης τονικότητας

Η αρχιτεκτονική του πρώτου νευρωνικού δικτύου έχει ως είσοδο ένα τρισδιάστατο φασματογράφημα (spectrogram) διαστάσεων 105x600. Αποτελείται από 5 συνελικτικά επίπεδα από 8 συνελίξεις εισόδου και εξόδου το καθένα με φίλτρα διαστάσεων 5x5 για την αναγνώριση χαρακτηριστικών. Έπειτα ακολουθεί ένα επίπεδο υπό δειγματοληψίας μέσης

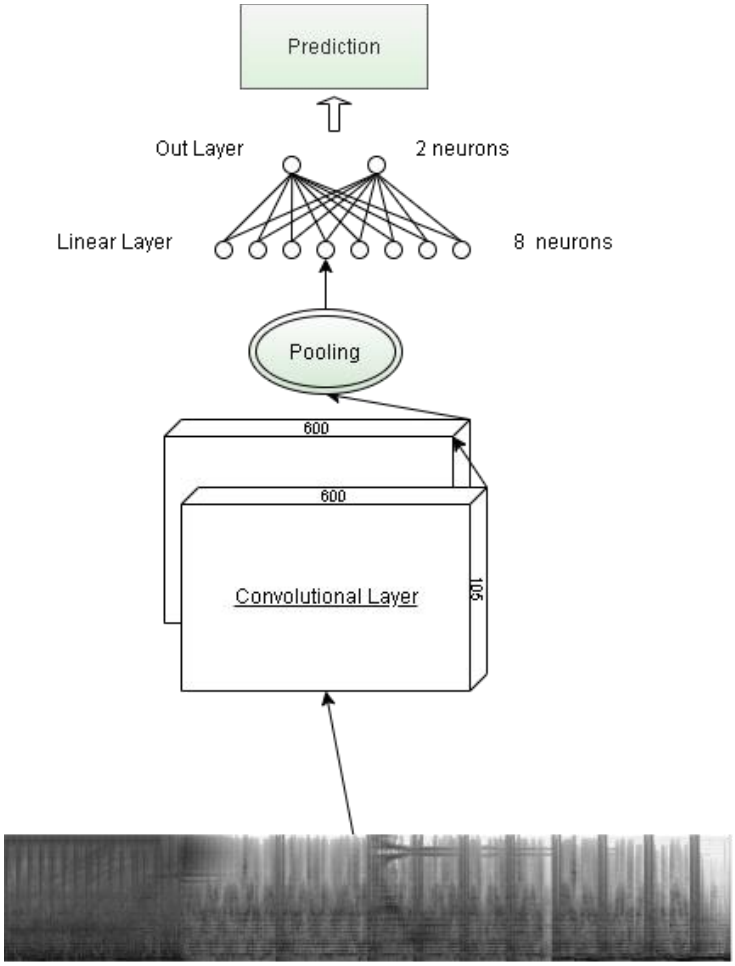
τιμής για την μείωση των χαρακτηριστικών που έχουν εξαχθεί από τα 5 συνελικτικά επίπεδα. Στη συνέχεια έχουμε 3 πλήρως συνδεδεμένα επίπεδα, με τα 2 πρώτα να έχουν 1024 και 64 νευρώνες αντίστοιχα και τέλος το επίπεδο εξόδου με 12 νευρώνες, όσες είναι και οι κλάσεις (C, C#, D, D#, E, F, F#, G, G#, A, A#, B).



7.3 Αρχιτεκτονική νευρωνικού δικτύου αναγνώρισης τονικότητας

7.2.2 Χαρακτηριστικά νευρωνικού δικτύου αναγνώρισης κλίμακας

Το νευρωνικό δίκτυο αναγνώρισης κλίμακας επειδή έχει να ξεχωρίσει μόνο δυο κλάσεις είναι λιγότερο πολύπλοκο από το προηγούμενο. Έτσι έχει σαν είσοδο ένα μονοδιάστατο αυτή τη φορά φασματογράφημα (spectrogram) διαστάσεων 105x600. Αυτό σημαίνει πως το spectrogram για κάθε pixel έχει μόνο μια τιμή που καθορίζει την αντίστοιχη greyscale τιμή του. Αποτελείται από 2 συνελικτικά επίπεδα με 16 συνελίξεις το καθένα με φίλτρα 5x5 και ακολουθείται ένα επίπεδο υπό δειγματοληψίας μέσης τιμής. Τέλος έχουμε 2 πλήρως συνδεδεμένα επίπεδα με το πρώτο να έχει 8 νευρώνες και το δεύτερο 2 που αντιστοιχούν στις κλάσεις πρόβλεψης (minor, major).



7.4 Αρχιτεκτονική νευρωνικού δικτύου αναγνώρισης κλίμακας

7.3 Αλγόριθμος εκπαίδευσης

Η δημιουργία του dataset όπως είδαμε στην ενότητα 6.6 αποτελεί θεμελιώδη ενέργεια για την πραγματοποίηση της εκπαίδευσης του νευρωνικού. Μετά την δημιουργία της δομής αυτής συνεχίζουμε την διαδικασία διαχωρισμού τους σε train και test set. Ο διαχωρισμός αυτός πραγματοποιείται επειδή θέλουμε η αξιολόγηση να εκτελείτε σε δεδομένα "άγνωστα" για το νευρωνικό, δεδομένα δηλαδή που δεν είχε συναντήσει ξανά κατά την εκπαίδευση. Αφού χωριστούν τα δεδομένα σε εκπαίδευσης και αξιολόγησης, θέλουμε να τα φορτώσουμε σε tensors και να αρχίσουμε να τα τροφοδοτούμε στο νευρωνικό. Ο τρόπος που διενεργείται η τροφοδοσία αυτή παίζει σημαντικό ρόλο στη διαδικασία της εκπαίδευσης, καθώς προτιμούμε να περνάμε τα δεδομένα μας σε μικρές

παρτίδες και ανακατεμένες για κάθε epoch. Κάθε επανάληψη κατά την οποία έχουν δοθεί όλα τα δεδομένα από μια φορά ονομάζεται μια εποχή (epoch). Ένα εργαλείο που διαθέτει το pytorch και μας δίνει τη δυνατότητα να τροφοδοτούμε με τέτοιο τρόπο τα δεδομένα μας στο νευρωνικό είναι το dataloader [21] [22].

```
CLASS torch.utils.data.DataLoader(dataset, batch_size=1, shuffle=False, sampler=None,
        batch_sampler=None, num_workers=0, collate_fn=None, pin_memory=False, drop_last=False,
        timeout=0, worker_init_fn=None, multiprocessing_context=None, generator=None, *,
        prefetch_factor=2, persistent_workers=False) [SOURCE]
```

7.5 DataLoader

Πριν το τέλος κάθε εποχής καλούμε την συνάρτηση σφάλματος η οποία συγκρίνει τα αποτελέσματα του νευρωνικού με τις "σωστές" προβλέψεις, όπως έχουν οριστεί από το annotation file. Έπειτα εκτελούμε την backward διαδικασία της εκπαίδευσης για την ανανέωση των βαρών. Στο τέλος κάθε εποχής ελέγχουμε την ακρίβεια και την συγκρίνουμε με την μεγαλύτερη ακρίβεια που μπορεί να είχε το νευρωνικό σε προηγούμενη εποχή και αποθηκεύουμε τις παραμέτρους της εποχής που είχε ως αποτέλεσμα την καλύτερη. Στα παρακάτω στιγμιότυπα κώδικα φαίνεται η διάσπαση των δεδομένων και η χρήση του dataloader (εικόνα 7.5) και ο αλγόριθμος που υλοποιεί την διαδικασία της εκπαίδευσης (εικόνα 7.6).

```
1  from torch.utils.data import DataLoader
2
3  val_size = int(len(key_training_data)*0.1)
4  training_data = key_training_data[:len(key_training_data)-val_size]
5  test_data = key_training_data[-val_size:]
6
7  batchSize = 16
8  train_dataloader = DataLoader(training_data, batch_size=batchSize , shuffle=True, pin_memory=True, drop_last=True)
9  test_dataloader = DataLoader(test_data, batch_size = batchSize, shuffle=True, drop_last=True)
10
```

7.5 Data split & dataloader

```python
#TRAINING KEY NEURAL NETWORK


k_Net.train()
num_epochs = 15
acc = 101
our_acc = 0
flag = True # flag for not changing again and again the lr when loss remains below 0.5

for epoch in range(0, num_epochs):
    print("-------------------------------------------------------------")

    for i, data in enumerate(train_dataloader, 0):
        # Get inputs
        inputs = data[0].view(-1,3,105,600)
        targets = data[1]
        inputs = inputs.float()

        inputs, targets = inputs.to(device), targets.to(device)

        optimizer.zero_grad()

        # Generate outputs
        outputs = k_Net(inputs)

        # Set total and correct
        targets = targets.view(len(data[1]))

        loss = loss_function(outputs, targets)
        loss.backward()
        optimizer.step()


    flag = lr_handler(loss, flag) # check learning rate
    acc = train_evaluation(test_dataloader, epoch) # evaluate the model every epoch

    if acc != 101:

        print("| Epoch : %2d | Loss: %.5f | currenct accuracy: %2.5f |" % (epoch, loss, acc))
        if (our_acc < acc): # if the accurace imporves save the models parameteres
            our_acc = acc
            torch.save(k_Net.state_dict(), 'C:/Users/Michalis Zeakis/Desktop/university/ptyxiaki/saved_nns/batchSize1-best-key-network-model-parameters1.pt')
        acc = 101
    else:
        print("| Epoch : %2d | Loss: %.5f | " % (epoch ,loss))
```

7.6 Αλγόριθμος εκπαίδευσης

7.4 Υπέρ-παράμετροι

Για την εκπαίδευση των νευρωνικών δικτύων είναι απαραίτητο να οριστεί ένας optimizer (βελτιστοποιητής) ο οποίος είναι υπεύθυνος για τον τρόπο με τον οποίο μεταβάλλονται τα βάρη. Η βελτιστοποίηση σαν διαδικασία παρέχει έναν τρόπο ελαχιστοποίησης της συνάρτησης λάθους (loss function) [23]. Η συνάρτηση λάθους αποτελεί αναπαράσταση της απόστασης μεταξύ της πραγματικής και της προβλεπόμενης τιμής του στόχου και παίρνει μη αρνητικές τιμές, με το μηδέν να αποτελεί την τέλεια πρόβλεψη. Στη παρούσα πτυχιακή χρησιμοποιείται η συνάρτηση Cross Entropy Loss. Η τελευταία αφορά την βελτιστοποίηση πολυταξικών μοντέλων και βασίζεται στην εντροπία (entropy). Η εντροπία μιας μεταβλητής X ορίζεται ως το επίπεδο αβεβαιότητας στο πιθανό αποτέλεσμα των μεταβλητών. Έτσι η Cross Entropy Loss συγκρίνει την προβλεπόμενη με την πραγματική απόδοση κλάσης, υπολογίζει μια απώλεια και βάση αυτής “τιμωρεί” την πιθανότητα ανάλογα το πόσο απέχει από την αναμενόμενη τιμή. Τέλος, η συγκεκριμένη ποινή είναι λογαριθμικής φύσης [24] [25].

Ο optimizer που χρησιμοποιείται είναι ο Stochastic Gradient Descent [26]. Δεδομένου ενός training dataset με n δεδομένα, συνάρτηση λάθους fi(x) όπου i ο αριθμός του

παραδείγματος εκπαίδευσης και x ο πίνακας παραμέτρων, η συνάρτηση γενίκευσης του SGD θα είναι:

$$f(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{x}).$$

Η κλίσης της συνάρτησης αυτής υπολογίζεται από τον τύπο :

$$\nabla f(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\mathbf{x}).$$

Ο Stochastic Gradient Descent βελτιώνει σημαντικά το υπολογιστικό κόστος σε κάθε επανάληψη ανεξάρτητης μεταβλητής σε O(1) από O(n) που είναι στον απλό Gradient Descent. Στο pytorch παρέχεται μια βιβλιοθήκη από βελτιστοποιητές που ονομάζεται optim. Με τη συνάρτησης που βλέπουμε και στην εικόνα 7.7 μπορεί να χρησιμοποιηθεί ο SGD αφού πρώτα οριστούν κάποιες άλλες πολύ σημαντικές υπέρ-παράμετροι [27].

```
CLASS  torch.optim.SGD(params, lr=<required parameter>, momentum=0, dampening=0, weight_decay=0,
       nesterov=False) [SOURCE]
```

7.7 Stochastic Gradient Descent στο pytorch

Πιο συγκεκριμένα μια από τις σημαντικότερες υπέρ-παραμέτρους που συναντάτε στη διαδικασία της εκπαίδευσης ενός νευρωνικού δικτύου είναι το learing rate (lr). Ο ρυθμός εκμάθησης ελέγχει πόσο θα αλλάζει το μοντέλο ως απόκριση στο εκτιμώμενο σφάλμα μετά από κάθε ενημέρωση των βαρών του. Η επιλογή του κατάλληλου learning rate αποτελεί μια αρκετά δύσκολη και ταυτόχρονα σημαντική διαδικασία. Αν το learning rate είναι πολύ μικρό η διαδικασία εκμάθησης παίρνει πάρα πολύ χρόνο (πολλά epochs), με τις αλλαγές στα βάρη να είναι μικρές και το αποτέλεσμα να μην είναι ιδανικό, ενώ αν είναι πολύ μεγάλο τότε το optimization μπορεί να συγκλίνει σε μια λύση που δεν είναι βέλτιστη. Το learning rate που χρησιμοποιείται είναι 10^{-3} και όταν το σφάλμα έπεφτε κάτω από μια προκαθορισμένη τιμή τότε μειωνόταν σε $0.5*10^{-3}$ [28].

Ο απλός SGD δεν υπολογίζει την ακριβή παράγωγο του loss function, αλλά αυτό γίνεται σε παρτίδες (batches). Αυτό σημαίνει ότι δεν κινείται πάντα προς την σωστή κατεύθυνση γιατί η παράγωγος έχει “θόρυβο”. Προκειμένου να αποφευχθεί το παραπάνω πρόβλημα και να αξιοποιηθεί η μείωση της διακύμανσης (variance reduction) αντικαθίσταται ο υπολογισμός της κλίσης με ένα “leaky average” (διαρροή μέσου όρου). Δηλαδή υπολογίζεται ένας μέσος όρος βασισμένος σε πολλές προηγούμενες διαβαθμίσεις

και ονομάζεται momentum (ορμή). Στη παρούσα πτυχιακή χρησιμοποιείται SGD με momentum, με τιμή 0.9 [27] [29].

Μια εξίσου σημαντική υπέρ-παράμετρος που πρέπει να οριστεί στον optimizer του pytorch είναι το weight decay. Το τελευταίο είναι γνωστό και σαν L2 reguralization και αποτελεί έναν από τους πιο διαδεδομένους τρόπους κανονικοποίησης παραμετρικών μοντέλων μηχανικής μάθησης. Η χρησιμότητα του είναι κατά κύριο λόγο η αντιμετώπιση του overfitting. Αυτό είναι ένα από τα πιο συνηθισμένα προβλήματα που συναντιούνται στην διαδικασία της εκπαίδευσης και ουσιαστικά εκφράζει την μεγάλη διαφορά του validation και training error. Συγκεκριμένα ο όρος αναφέρεται στις περιπτώσεις που το σφάλμα εκπαίδευσης είναι μεγαλύτερο από το σφάλμα επικύρωσης. Πρακτικά αυτό υποδηλώνει καλή εκπαίδευση στο train dataset, αλλά αδυναμία γενίκευσης της γνώσης με αποτέλεσμα τη χαμηλή ακρίβεια στο test dataset. Η τεχνική του weight decay βασίζεται στην διαίσθηση ότι η συνάρτηση f=0, δηλαδή αυτή που θα έχει όλες τις εισόδους μηδενικές, είναι η πιο "απλή". Έτσι μετράμε την πολυπλοκότητα μιας συνάρτησης ανάλογα με την απομάκρυνση της από το μηδέν. Το weight decay που χρησιμοποιείται είναι ίσο με 10^{-4} [30].

Τέλος, δύο λιγότερο επιδραστικές αλλά εξίσου σημαντικές υπέρ-παράμετροι είναι το batch size και οι εποχές (Epochs). Το πρώτο αφορά τον τρόπο με τον οποίο εισέρχονται τα δεδομένα στα νευρωνικά δίκτυα και το δεύτερο την διαδικασία της εκμάθησης. Πιο αναλυτικά το batch size αναφέρεται στην διαδικασία τροφοδοσίας των δεδομένων του train dataset στο νευρωνικό. Η διαδικασία αυτή πραγματοποιείται σε κομμάτια (batches) και στην συγκεκριμένη πτυχιακή χρησιμοποιείται batch sizes μεγέθους 16. Η δεύτερη υπέρ-παράμετρος, εποχές, αναφέρεται στον αριθμό των επαναλήψεων που θα πραγματοποιηθούν πριν τελειώσει η εκπαίδευση. Ειδικότερα, εκφράζει το πόσες φορές θα τροφοδοτηθούν όλα τα δεδομένα της εκπαίδευσης στο νευρωνικό. Όσο αυξάνουμε τον αριθμό των epochs το μοντέλο βελτιώνεται. Αυτό όμως συμβαίνει μέχρι ένα συγκεκριμένο αριθμό εποχών που όταν ξεπεράσει το μοντέλο αρχίζει να κάνει overfit στα training data. Σε αυτό το σημείο η ακρίβεια του νευρωνικού για άγνωστα δεδομένα σταματάει να βελτιώνεται. Ο αριθμός των εποχών που χρησιμοποιήθηκαν είναι 15. Ένα σημαντικό χαρακτηριστικό των υπέρ-παραμέτρων αυτών είναι ότι επηρεάζουν σημαντικά τους χρόνους που διαρκεί η διαδικασία της εκπαίδευσης. Για παράδειγμα, πολύ μεγάλα batch sizes, αυξάνουν την ταχύτητα της εκπαίδευσης, αλλά απαιτούν και πιο ισχυρό hardware. Αντίστοιχα και ο μεγάλος αριθμός epochs αυξάνει σημαντικά τον χρόνο εκπαίδευσης ενώ τα μικρά τον ελαττώνουν. Χαρακτηριστικά μετά την αναβάθμιση της μνήμης ram από 8

giga byte σε 32 επιτράπηκε η χρήση του batch size 16 , καθώς και η εκπαίδευση σε 100 εποχές.

Στους παρακάτω πίνακες παραθέτονται οι τιμές των υπέρ-παραμέτρων για κάθε νευρωνικό δίκτυο ξεχωριστά (πίνακες 7.8 7.9).

model parameters	2D convolutional layers	2	
	neurons of convolutional layers	16	
	filter size	5x5	
	average pool layer filter size	2x2	
	linear layers	2	
	neurons of linear layers	1st	8
		2nd	2
training parameters	learning rate	$0.5*10^{-3}$	
	weight decay	10^{-4}	
	momentum	0.9	
	batch size	8	
	epochs	20	
	optimizer	SGD	
	loss function	cross entropy	

7.8 Υπέρ-παράμετροι νευρωνικού δικτύου πρόβλεψης κλίμακας

model parameters	2D convolutional layers	5	
	neurons of convolutional layers	8	
	filter size	5x5	
	average pool layer filter size	2x2	
	linear layers	3	
	neurons of linear layers	1st	1024
		2nd	64
		3rd	12
training parameters	learning rate	10^{-3}	
	weight decay	10^{-4}	
	momentum	0.9	
	batch size	16	
	epochs	20	
	optimizer	SGD	
	loss function	cross entropy	

7.9 Υπέρ-παράμετροι νευρωνικού δικτύου πρόβλεψης τονικότητας

7.5 Πειράματα

Όπως αναφέρθηκε και νωρίτερα η επιλογή των τιμών των υπέρ-παραμέτρων και η αρχιτεκτονική των νευρωνικών δικτύων πραγματοποιήθηκε με δοκιμές και πειραματισμούς (trial and error). Ο λόγος για τον οποίο συμβαίνει αυτό είναι ότι τα νευρωνικά εκπαιδεύονται με supervised learning, διαδικασία κατά την οποία γνωρίζουμε τα αποτελέσματα των δεδομένων εισόδου "ψάχνοντας" έτσι τις καλύτερες δυνατές παραμέτρους ώστε να λάβουμε τις επιθυμητές εξόδους. Η διαδικασία που ακολουθήθηκε είναι η εξής και για τα δύο νευρωνικά δίκτυα:

1. Αρχικοποίηση παραμέτρων είτε τυχαία είτε σύμφωνα με άλλα μοντέλα [7]
2. Εκτέλεση της εκπαίδευσης με λίγα δεδομένα αρχικά
3. Εκτέλεση αξιολόγησης αποθηκεύοντας την ακρίβεια και το σφάλμα
4. Μεταβολή των παραμέτρων εκτελώντας ξανά την εκπαίδευση
5. Αποθήκευση της νέας ακρίβειας και του νέου σφάλματος μετά την αξιολόγηση
6. Επανάληψη των βημάτων 2-5 μέχρι να λάβουμε ικανοποιητικά υψηλή ακρίβεια και χαμηλό σφάλμα

Στη συνέχεια προστίθεται ένα μικρό πλήθος δεδομένων για κάθε κλάση επαναλαμβάνοντας την παραπάνω διαδικασία πειραματισμών. Τα δεδομένα που εισάχθηκαν αυξάνονται σταδιακά προκειμένου να λαμβάνονται πιο γρήγορα αποτελέσματα στις πρώτες δοκιμές, παρατηρώντας έτσι ποιες υπέρ-παράμετροι έχουν μεγαλύτερο βαθμό επιρροής στην έξοδο των νευρωνικών. Οι δοκιμές που ακολουθούν αφορούν το νευρωνικό δίκτυο πρόβλεψης τονικότητας καθώς οι 12 κλάσεις το καθιστούν δυσκολότερο στην εκμάθηση ενώ το δίκτυο πρόβλεψης κλίμακας δεν χρειάστηκε τέτοιο πλήθος πειραματισμών, δίνοντας 99,7% ακρίβεια με τις πρώτες εκπαιδεύσεις. Στους παρακάτω πίνακες ο αριθμός των epochs που αναγράφεται στην κάθε δοκιμή είναι το epoch το οποίο έδωσε την μεγαλύτερη ακρίβεια σύμφωνα με το train evaluation και όχι τον συνολικό αριθμό epochs της κάθε εκπαίδευσης.

Πιο συγκεκριμένα τροφοδοτήθηκαν 550 spectrogram ανά κλάση για τις πρώτες δοκιμές θέτοντας 3 συνελικτικά επίπεδα με φίλτρα διαστάσεων 5x5 και 2 γραμμικά με νευρώνες εισόδου και εξόδου όπως φαίνεται στον πίνακα 7.8. Σε αυτές τις δοκιμές παρατηρήθηκε πως το μεγάλο batch size (2η δοκιμή 32) μετά από εκπαίδευση με 10 epochs καταλήγει σε μικρότερη ακρίβεια (45,1 %) σε σχέση με την 1η δοκιμή η οποία έχει ακριβώς τις ίδιες τιμές στις υπέρ-παραμέτρους εκτός από το batch size (16). Επίσης παρατηρήθηκε στην 4η δοκιμή ότι το weight decay δεν επηρέασε σημαντικά την ακρίβεια παρά μόνο το σφάλμα το οποίο δεν κατάφερε να μειωθεί αρκετά. Έπειτα αφού προστέθηκε dropout επίπεδο και διπλασιάστηκε το learning rate έγινε αντιληπτό ότι η ακρίβεια αυξήθηκε ελάχιστα σε αντίθεση με το σφάλμα το οποίο έφτασε σε αρκετά χαμηλότερο επίπεδο. Καθώς οι διαφορές στην ακρίβεια και στο σφάλμα δεν ήταν ικανοποιητικές προστέθηκε ένα ακόμα συνελικτικό επίπεδο κάτι το οποίο οδήγησε σε αύξηση της ακρίβειας κατά 2,9%. Παρακάτω ακολουθεί ένας αναλυτικός πίνακας με τις πρώτες 6 δοκιμές συμπεριλαμβάνοντας όλες τις παραμέτρους που χρησιμοποιήθηκαν. Με κόκκινο χρώμα

φαίνονται οι τιμές των υπέρ-παραμέτρων οι οποίες μεταβλήθηκαν σε σχέση με την προηγούμενη δοκιμή και με πράσινο τις καλύτερες τιμές της ακρίβειας και του σφάλματος.

550 spectrogram per class (6.660 total data)										
parameters / δοκιμές	convolutional layers	linear layers	learning rate	weight decay	momentum	dropout	batch size	epochs	accuracy (%)	loss
1η	(e,8,5) (8,16,5) (16,16,5)	(s,16) (16,12)	0,0005	10^{-4}	0,9	-	16	10	48,2	1,26
2η	(e,8,5) (8,16,5) (16,16,5)	(s,16) (16,12)	0,0005	10^{-4}	0,9	-	32	10	45,1	1,7
3η	(e,8,5) (8,16,5) (16,16,5)	(s,16) (16,12)	0,0005	10^{-4}	0,9	-	8	10	48,6	1,28
4η	(e,8,5) (8,16,5) (16,16,5)	(s,16) (16,12)	0,0005	10^{-3}	0,9	-	8	12	48	2,27
5η	(e,8,5) (8,16,5) (16,8,5)	(s,16) (16,12)	0,001	10^{-4}	0,9	0,5	16	12	48,6	0,9
6η	(e,8,5) (8,16,5) (16,8,5) (8,8,5)	(s,16) (16,12)	0,001	10^{-4}	0,9	0,5	16	12	51,5	2,01

7.10 Δοκιμές με 6660 spectrogram για εκπαίδευση και αξιολόγηση (όπου e η είσοδος για το πρώτο συνελικτικό επίπεδο και s η είσοδος του πρώτου γραμμικού επιπέδου)

Στη συνέχεια τροφοδοτήθηκαν συνολικά 7800 spectrogram (650 ανά κλάση) και εκπαιδεύτηκε το νευρωνικό στις ίδιες τιμές των παραμέτρων οι οποίες έδωσαν την καλύτερη ακρίβεια στις προηγούμενες δοκιμές, δίνοντας ως αποτέλεσμα 5,5% χειρότερη ακρίβεια αλλά καλύτερο σφάλμα 0,69. Αυτό σημαίνει πως το δίκτυο εκπαιδεύτηκε και "έμαθε" πολύ καλά τα δεδομένα εκπαίδευσης αλλά ήταν "ανίκανο" να αποδώσει στα δεδομένα αξιολόγησης, έχοντας έτσι ένα overfitted δίκτυο. Ο σκοπός των πειραμάτων που ακολουθούν είναι να αποφευχθεί το overfitting όσο προστίθενται δεδομένα. Κατά την 3η δοκιμή παρατηρήθηκε πως τα περισσότερα epochs οδηγούν σε ένα overfitted μοντέλο όπως φαίνεται στην 4η δοκιμή στην οποία αφαιρέθηκε το dropout layer (μηδενικό σφάλμα ενώ μειώθηκε το accuracy). Οι επόμενες 5 δοκιμές αφορούν την δομή των συνελικτικών επιπέδων κατά τις οποίες παρατηρήθηκε ότι οι περισσότεροι νευρώνες (δοκιμή 6η) και τα περισσότερα επίπεδα (δοκιμή 9η) οδήγησαν σε καλύτερες τιμές της ακρίβειας χωρίς overfit (μη μηδενικό ή κοντά στο μηδέν σφάλμα). Ακολουθεί ο πίνακας 7.11 με τις 9 δοκιμές που πραγματοποιήθηκαν δίνοντας ως είσοδο 650 spectrogram ανά κλάση.

650 spectrogram per class (7.800 total data)										
parameters / δοκιμές	convolutional layers	linear layers	learning rate	weight decay	momentum	dropout	batch size	epochs	accuracy (%)	loss
1η	(e,8,5) (8,16,5) (16,8,5) (8,8,5)	(s,16) (16,12)	0,001	10^{-4}	0,9	0,5	16	14	46	0,69
2η	(e,8,5) (8,8,5) (8,8,5) (8,8,5)	(s,16) (16,12)	0,001	10^{-4}	0,9	0,5	16	14	49,5	0,81
3η	(e,8,5) (8,8,5) (8,8,5) (8,8,5)	(s,16) (16,12)	0,001	10^{-4}	0,9	0,5	16	22	45,2	0,6
4η	(e,8,5) (8,8,5) (8,8,5) (8,8,5)	(s,16) (16,12)	0,001	10^{-4}	0,9	-	16	14	46,6	0,003
5η	(e,8,3) (8,8,3) (8,8,3) (8,8,3)	(s,16) (16,12)	0,001	10^{-4}	0,9	0,5	16	14	50,3	1,21
6η	(e,8,5) (8,16,5) (16,8,5) (8,8,5)	(s,16) (16,12)	0,001	10^{-4}	0,9	0,5	16	12	50,1	1,3
7η	(e,8,5) (8,16,5) (16,8,5) (8,8,5)	(s,64) (64,12)	0,001	10^{-4}	0,9	0,5	16	12	48,5	0,46
8η	(e,8,5) (8,16,5) (16,8,5) (8,8,5)	(s,64) (64,12)	0,001	10^{-4}	0,9	0,5	32	16	45,05	0,8
9η	(e,8,3) (8,8,3) (8,8,3) (8,8,3) (8,8,3) (8,8,3)	(s,16) (16,12)	0,001	10^{-4}	0,9	0,5	16	12	51,04	2,03

7.11 Δοκιμές με 7800 spectrogram για εκπαίδευση και αξιολόγηση

Οι επόμενες δοκιμές πραγματοποιήθηκαν με βάση το πλήθος των δεδομένων παρατηρώντας πως είναι δυνατό να επιτευχθεί μεγαλύτερη ακρίβεια όσο αυξάνονται τα δεδομένα. Με την πρώτη κιόλας δοκιμή, στην οποία χρησιμοποιήθηκαν οι καλύτερες τιμές των παραμέτρων του προηγούμενου πίνακα, παρατηρείται μια άνοδος στην ακρίβεια κατά 2,16% μετά από 32 epochs, αποδεικνύοντας έτσι πως τα περισσότερα δεδομένα οδηγούν σε μεγαλύτερη ακρίβεια. Στις επόμενες 2 δοκιμές παρατηρήθηκε ότι οι αλλαγές στην δομή του νευρωνικού (2x2 διαστάσεις φίλτρων και 16 νευρώνες στο πρώτο συνελικτικό επίπεδο) δεν οδήγησαν σε καλύτερα αποτελέσματα για αυτό και διατηρήθηκαν οι προηγούμενες τιμές. Η προσθήκη όμως ενός επιπλέον συνελικτικού επιπέδου στην 4η

δοκιμή με τον ίδιο αριθμό νευρώνων, οδήγησε στην μείωση της ακρίβειας με διαφορά ενός epoch από την 1η δοκιμή. Συνεπώς διατηρώντας την λογική ότι τα περισσότερα δεδομένα καταλήγουν σε μεγαλύτερη ακρίβεια παρέμειναν οι τιμές των παραμέτρων ίδιες και προστέθηκαν 20 επιπλέον spectrogram ανά κλάση όπως φαίνεται στην 5η δοκιμή (ακρίβεια: 54,1% σφάλμα: 0,04). Στις επόμενες 3 δοκιμές όμως, παρατηρήθηκε ότι δεν αρκεί απλά η εισαγωγή περισσότερων δεδομένων αλλά και η απλοποίηση του νευρωνικού με την χρήση του lr_handler σε συνδυασμό με τα επιπρόσθετα δεδομένα. Παρακάτω παρατίθεται ο πίνακας 7.12 στον οποίο φαίνονται αναλυτικά οι τιμές των παραμέτρων για κάθε δοκιμή.

Δοκιμές με βάση τα δεδομένα εισαγωγής										
parameters / δοκιμές	convolutional layers	linear layers	learning rate	weight decay	momentum	dropout	batch size	epochs	accuracy (%)	loss
1η 700data/class	(e,8,3) (8,8,3) (8,8,3) (8,8,3) (8,8,3) (8,8,3)	(s,16) (16,12)	0,001	10^{-4}	0,9	0,5	16	32	53,2	0,09
2η 700data/class	(e,8,2) (8,8,2) (8,8,2) (8,8,2) (8,8,2) (8,8,2)	(s,16) (16,12)	0,001	10^{-4}	0,9	0,5	16	20	48,19	1,53
3η 700data/class	(e,16,3) (16,8,3) (8,8,3) (8,8,3) (8,8,3) (8,8,3)	(s,16) (16,12)	0,001	10^{-4}	0,9	0,5	16	30	50,4	0,3
4η 750data/class	(e,8,3) (8,8,3) (8,8,3) (8,8,3) (8,8,3) (8,8,3) (8,8,3)	(s,16) (16,12)	0,001	10^{-4}	0,9	0,5	16	33	49,27	0,38
5η 770data/class	(e,8,3) (8,8,3) (8,8,3) (8,8,3) (8,8,3) (8,8,3)	(s,16) (16,12)	0,001	10^{-4}	0,9	0,5	16	42	54,1	0,04
6η 790data/class	(e,8,3) (8,8,3) (8,8,3) (8,8,3) (8,8,3) (8,8,3)	(s,16) (16,12)	0,001	10^{-4}	0,9	0,5	16	20	52,1	0,6
7η 790data/class (χωρίς lr_handler)	(e,8,3) (8,8,3) (8,8,3) (8,8,3) (8,8,3) (8,8,3)	(s,16) (16,12)	0,001	10^{-4}	0,9	0,5	16	20	48,8	0,46
8η 800data/class	(e,8,3) (8,8,3) (8,8,3) (8,8,3) (8,8,3) (8,8,3) (8,8,3)	(s,16) (16,12)	0,001	10^{-4}	0,9	0,5	40	16	48,9	0,3

7.12 Δοκιμές με κύριο κριτήριο το πλήθος δεδομένων εισαγωγής (part 1)

Συνεχίζοντας τις δοκιμές με βάση το πλήθος δεδομένων εισαγωγής, οι οποίες όπως φαίνεται στις τρεις πρώτες παρουσίασαν σημαντική βελτίωση στην ακρίβεια και στο σφάλμα, πραγματοποιήθηκαν δοκιμές που αφορούν τα γραμμικά επίπεδα. Αρχικά δοκιμάστηκε να αυξηθούν οι νευρώνες του πρώτου επιπέδου από 16 σε 64 δίνοντας ως

αποτέλεσμα σε μόλις 5 epochs 59,9% ακρίβεια χωρίς όμως το σφάλμα να φτάνει σε ικανοποιητικό σημείο. Προσθέτοντας 1 επιπλέον γραμμικό επίπεδο (4η δοκιμή) και αυξάνοντας σταδιακά τους νευρώνες του, παρατηρείται πως η ακρίβεια παραμένει σε σχετικά σταθερές τιμές με το σφάλμα να παρουσιάζει μείωση φτάνοντας μέχρι 0,9 (6η δοκιμή). Στον πίνακα 7.13 φαίνονται αναλυτικά οι τιμές των παραμέτρων της κάθε δοκιμής, η ακρίβεια και το σφάλμα.

Δοκιμές με βάση τα δεδομένα εισαγωγής										
parameters / δοκιμές	convolutional layers	linear layers	learning rate	weight decay	momentum	dropout	batch size	epochs	accuracy (%)	loss
1η 810 data/class	(e,8,3) (8,8,3) (8,8,3) (8,8,3) (8,8,3)	(s,16) (16,12)	0,001	10^{-4}	0,9	0,5	16	20	55,7	0,25
2η 820 data/class	(e,8,3) (8,8,3) (8,8,3) (8,8,3) (8,8,3)	(s,16) (16,12)	0,001	10^{-4}	0,9	0,5	16	30	52,12	0,25
3η 1000 data/class	(e,8,3) (8,8,3) (8,8,3) (8,8,3) (8,8,3)	(s,16) (16,12)	0,001	10^{-4}	0,9	0,5	16	5	56,08	1,9
4η 1000 data/class	(e,8,3) (8,8,3) (8,8,3) (8,8,3) (8,8,3)	(s,64) (64,12)	0,001	10^{-4}	0,9	0,5	16	5	59,9	1,3
5η 1000 data/class	(e,8,3) (8,8,3) (8,8,3) (8,8,3) (8,8,3)	(s,64) (64,128) (128,12)	0,001	10^{-4}	0,9	0,5	16	5	56,7	1,17
6η 1000 data/class	(e,8,3) (8,8,3) (8,8,3) (8,8,3) (8,8,3)	(s,32) (32,64) (64,12)	0,001	10^{-4}	0,9	0,5	16	9	59,1	0,9
7η 1000 data/class	(e,8,3) (8,8,3) (8,8,3) (8,8,3) (8,8,3)	(s,1024) (1024,64) (64,12)	0,001	10^{-4}	0,9	0,5	16	10	59,2	1,19
8η 1200 data/class	(e,8,3) (8,8,3) (8,8,3) (8,8,3) (8,8,3)	(s,1024) (1024,64) (64,12)	0,001	10^{-4}	0,9	0,5	16	11	58,12	1,3

7.13 Δοκιμές με κύριο κριτήριο το πλήθος δεδομένων εισαγωγής (part 2)

ΚΕΦΑΛΑΙΟ 8

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΕΠΕΚΤΑΣΕΙΣ

8.1 Συμπεράσματα

Έπειτα από την ανάπτυξη της παρούσας εργασίας συμπεραίνουμε ότι οι μέθοδοι και οι αλγόριθμοι μηχανικής μάθησης, και συγκεκριμένα τα συνελικτικά νευρωνικά δίκτυα, αποτελούν ένα εξαιρετικό εργαλείο για την αναγνώριση της κλίμακας ενός μουσικού κομματιού. Στην περίπτωση της αναγνώρισης τονικότητας έχουμε λιγότερο ικανοποιητικά αποτελέσματα συγκριτικά με αυτά της κλίμακας. Η διαφορά στην ακρίβεια των δύο μοντέλων δεν οφείλεται μόνο στην διαφορετική τους δομή (οι τονικότητες είναι πολυταξικό δίκτυο ενώ οι κλίμακες δυαδικό). Η πρόβλεψη τονικότητας απαιτεί λεπτομερή ποσοτική ανάλυση, και δεν μπορεί να εκφραστεί σε βάθος με βαθμολογίες ακρίβειας [7]. Συγκεκριμένα, έχουμε κλάσεις που είναι σημασιολογικά κοντά η μία στην άλλη σύμφωνα με τη μουσική θεωρία. Τέτοιο παράδειγμα αποτελούν οι σχετικές κλίμακες. Μια σχετική ελάσσονα κλίμακα μιας μείζονος είναι η ελάσσονα κλίμακα του έκτου βαθμού αυτής της τονικότητας. Για παράδειγμα, η A minor (A minor scale: A, B, C, D, E, F, G) ονομάζεται σχετική ελάσσονα κλίμακα στο κλειδί της C major (C major scale: C, D, E, F, G, A, B) καθώς μοιράζονται όλες τις νότες αλλά διαφέρουν στην τονικότητα.

Ένας σημαντικός περιορισμός που παρατηρήθηκε κατά την διαδικασία ανάπτυξης αλλά και εκτίμησης των μοντέλων ήταν ότι τα σύνολα δεδομένων αφορούσαν κυρίως το μουσικό είδος της ηλεκτρονικής μουσικής. Αυτό είχε σαν αποτέλεσμα την παρατήρηση προβλέψεων μικρότερης ακρίβειας σε τραγούδια διαφορετικών ειδών από ότι στο test dataset.

8.2 Μελλοντικές επεκτάσεις και βελτιώσεις

Παρατίθενται προτάσεις για μελλοντικές βελτιώσεις και επεκτάσεις της παρούσας πτυχιακής εργασίας:

- **Ανάπτυξη νέου και εμπλουτισμένου συνόλου δεδομένων.**

  Προτείνεται ο εμπλουτισμός των διαθέσιμων dataset με νέα και πιο σύγχρονα μουσικά κομμάτια από όσο το δυνατόν περισσότερα διαφορετικά είδη μουσικής είναι εφικτό. Με αυτό τον τρόπο η ακρίβεια των νευρωνικών θα αυξηθεί. Τα μοντέλα θα έχουν τη δυνατότητα να αναγνωρίζουν γενικευμένα χαρακτηριστικά προκειμένου να κάνουν την αντίστοιχη πρόβλεψη. Επιπλέον προβλέπεται ότι η καλύτερη ποιότητα που παρατηρείται στα πιο σύγχρονα μουσικά κομμάτια θα είχε επίσης θετικό αντίκτυπο στην εκπαίδευση των νευρωνικών.

- **Επανεκπαίδευση του δικτύου**

  Μια επιπλέον πρόταση αφορά την εκπαίδευση του δικτύου σε ένα καλύτερο σύνολο δεδομένων και σε υπολογιστές με καλύτερα χαρακτηριστικά. Πολύ καλύτερο hardware επιτρέπει τον πειραματισμό διαφόρων στοιχείων της αρχιτεκτονικής χωρίς τη μεγάλη χρονική επιβάρυνση που έχει ένας "χειρότερος" υπολογιστής. Χαρακτηριστικά παραδείγματα είναι οι 32 ώρες που χρειάστηκαν για το βήμα του pitch data augmentation και οι 2 ώρες τρεξίματος για κάθε εποχή.

- **Ανάπτυξη εφαρμογής light show**

  Προτεινόμενο πλάνο επέκτασης της εργασίας αποτελεί η ανάπτυξη μια εφαρμογής "αντιστοίχισης" του μουσικού κλειδιού σε φάσμα χρωμάτων για την αναπαραγωγή light show. Η αντιστοιχία αυτή θα είχε άμεση χρησιμότητα σε χώρους διασκέδασης και συναυλιών και μπορεί να παρέχει αυτοματοποιήσεις στα φωτορυθμικά του εκάστοτε χώρου.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] M. I. Jordan and T. M. Mitchell, “Machine learning: Trends, perspectives, and prospects,” Science, vol. 349, no. 6245, pp. 255–260, 2015.

[2] D. Graupe, Principles Of Artificial Neural Networks (3rd Edition). World Scientific Publishing Company, 2013.

[3] J. A. (ed). Flores, Focus on artificial neural networks. Nova Science Publishers, Incorporated, 2011.

[4] Udemy.com. [Online]. Available: https://www.udemy.com/course/music-theory-complete/. [Accessed: 18-Dec-2021].

[5] mlearnere, “How to start implementing Machine Learning to music,” Towards Data Science, 09-Apr-2021. [Online]. Available: https://towardsdatascience.com/how-to-start-implementing-machine-learning-to-music-4bd2edccce1f. [Accessed: 13-Feb-2022].

[6] H. Schreiber and M. Müller, “Musical tempo and key estimation using Convolutional Neural Networks with directional filters,” arXiv [cs.SD], 2019.

[7] F. Korzeniowski and G. Widmer, “End-to-end musical key estimation using a convolutional neural network,” in 2017 25th European Signal Processing Conference (EUSIPCO), 2017.

[8] “The McGill billboard project,” Mcgill.ca. [Online]. Available: http://ddmal.music.mcgill.ca/research/billboard. [Accessed: 13-Feb-2022].

[9] Upatras.gr. [Online]. Available: https://eclass.upatras.gr/modules/document/file.php/CEID1041/%CE%94%CE%B9%CE%B4%CE%B1%CE%BA%CF%84%CE%B9%CE%BA%CE%AD%CF%82%20%CE%A3%CE%B7%CE%BC%CE%B5%CE%AF%CF%89%CF%83%CE%B5%CE%B9%CF%82/Chapter2.pdf. [Accessed: 18-Dec-2021].

[10] Α. Πάχος, “Ανάπτυξη Ανεξάρτητου Συστήματος Αναγνώρισης Πτώσεων με χρήση Βαθιών Νευρωνικών Δικτύων και Edge Computing,” ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ, 2021.

[11] Κ. Νικολάου, “Ανάκτηση Εικόνας Βασισμένη σε Βαθιά Συνελικτικά Νευρωνικά Δίκτυα,” Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, 2016.

[12] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in 2017 International Conference on Engineering and Technology (ICET), 2017, pp. 1–6.

[13] JovicA ,Brkic K, Bogunovic N, An overview of free software tools for general data mining. 37th international convention on information and communication technology, electronics and microelectronics (MIPRO), 2014.

[14] G. Nguyen et al., “Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey,” Artif. Intell. Rev., vol. 52, no. 1, pp. 77–124, 2019.

[15] “Torch,” Torch.ch. [Online]. Available: http://torch.ch/. [Accessed: 18-Dec-2021].

[16] giantsteps-key-dataset: This repository contains the annotations and download scripts for the audio files of the GiantSteps Key data set. This data set was published at ISMIR 2015: P. Knees et al.: “Two data sets for tempo estimation and key detection in electronic dance music annotated from user corrections.” .

[17] Giantsteps-mtg-key-dataset. .

[18] I. E. Λιβιέρης, “ΑΠΟΤΙΜΗΣΗΜΕΘΟΔΩΝΕΚΠΑΙΔΕΥΣΗΣΤΕΧΝΗΤΩΝΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝΚΑΙΕΦΑΡΜΟΓΕΣ,” Πανεπιστήμιο Πατρών, 2008.

[19] V. Zhou, “Training a Convolutional Neural Network from scratch,” Towards Data Science, 06-Jun-2019. [Online]. Available: https://towardsdatascience.com/training-a-convolutional-neural-network-from-scratch-2235c2a25754. [Accessed: 18-Dec-2021].

[20] P. Radhakrishnan, “What are Hyperparameters ? and How to tune the Hyperparameters in a Deep Neural Network?,” Towards Data Science, 09-Aug-2017. [Online]. Available: https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a. [Accessed: 18-Dec-2021].

[21] “Datasets &Data Loaders — PyTorch Tutorials 1.10.1+cu102 documentation,” Pytorch.org. [Online]. Available: https://pytorch.org/tutorials/beginner/basics/data_tutorial.html. [Accessed: 18-Dec-2021].

[22] “torch.utils.data — PyTorch 1.10.0 documentation,” Pytorch.org. [Online]. Available: https://pytorch.org/docs/stable/data.html. [Accessed: 18-Dec-2021].

[23] “3.1. Linear Regression — Dive into Deep Learning 0.17.1 documentation,” D2l.ai.[Online]. Available: https://d2l.ai/chapter_linear-networks/linear-regression.html?highlight=loss%20function. [Accessed: 18-Dec-2021].

[24] J. Brownlee, “A gentle introduction to cross-entropy for machine learning,” Machine Learning Mastery, 20-Oct-2019.[Online]. Available: https://machinelearningmastery.com/cross-entropy-for-machine-learning/. [Accessed: 18-Dec-2021].

[25] K. E. Koech, “Cross-entropy loss function,” Towards Data Science, 02-Oct-2020. [Online]. Available: https://towardsdatascience.com/cross-entropy-loss-function-f38c4ec8643e. [Accessed: 18-Dec-2021].

[26] “11.4. Stochastic gradient descent — dive into deep learning 0.17.1 documentation,” D2l.ai. [Online]. Available: https://d2l.ai/chapter_optimization/sgd.html?highlight=stochastic. [Accessed: 18-Dec-2021].

[27] “SGD — PyTorch 1.10.0 documentation,” Pytorch.org. [Online]. Available: https://pytorch.org/docs/stable/generated/torch.optim.SGD.html. [Accessed: 18-Dec-2021].

[28] “11.11. Learning rate scheduling — dive into deep learning 0.17.1 documentation,” D2l.ai. [Online]. Available: https://d2l.ai/chapter_optimization/lr-scheduler.html?highlight=learning%20rate. [Accessed: 18-Dec-2021].

[29] “11.6. Momentum — Dive into Deep Learning 0.17.1 documentation,” D2l.ai.[Online]. Available: https://d2l.ai/chapter_optimization/momentum.html?highlight=learning%20rate. [Accessed: 18-Dec-2021].

[30] “4.5. Weight Decay — Dive into Deep Learning 0.17.1 documentation,” D2l.ai. [Online]. Available: https://d2l.ai/chapter_multilayer-perceptrons/weight-decay.html?highlight=weight%20decay. [Accessed: 18-Dec-2021].

ΠΑΡΑΡΤΗΜΑΤΑ

ΠΑΡΑΡΤΗΜΑ Α

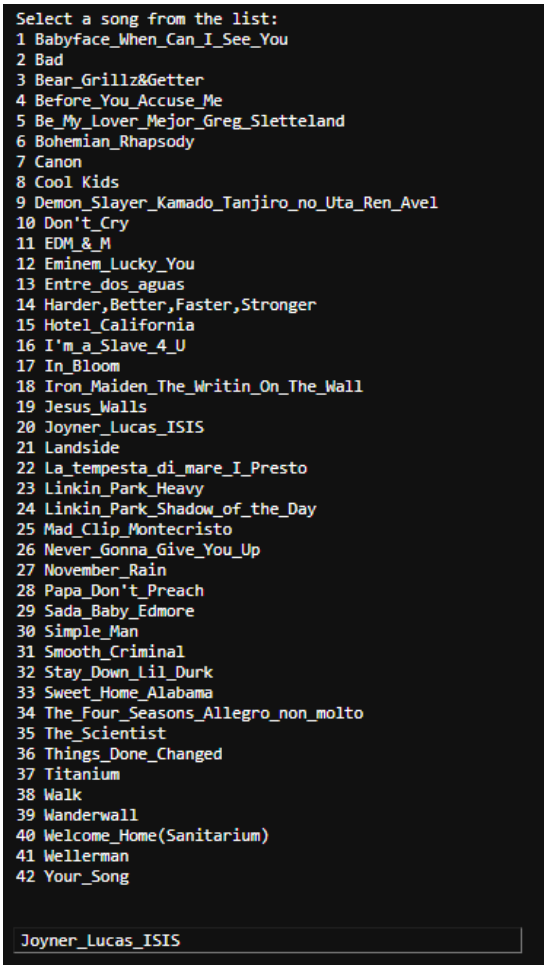
Α.1 Εισαγωγή

Ολοκληρώνοντας την εκπαίδευση, την αξιολόγηση και την αποθήκευση των καλύτερων παραμέτρων των νευρωνικών δικτύων δύναται να επιτευχθούν προβλέψεις mp3 αρχείων εκτός των δεδομένων που χρησιμοποιήθηκαν. Έτσι αναπτύχθηκε ένα πρόγραμμα σε python (keyFinder.py) το οποίο με την βοήθεια των αποθηκευμένων παραμέτρων των νευρωνικών δικτύων προβλέπει το μουσικό κλειδί οποιουδήποτε mp3 file του δοθεί. Προτού εκτελεσθεί το keyFinder.py είναι απαραίτητο να πραγματοποιηθούν τα εξής βήματα:

1. Ορισμός της δομής των 2 νευρωνικών
2. Λήψη ενός mp3 file (ή περισσότερων) και η αποθήκευση του στον φάκελο final_dataset_mp3

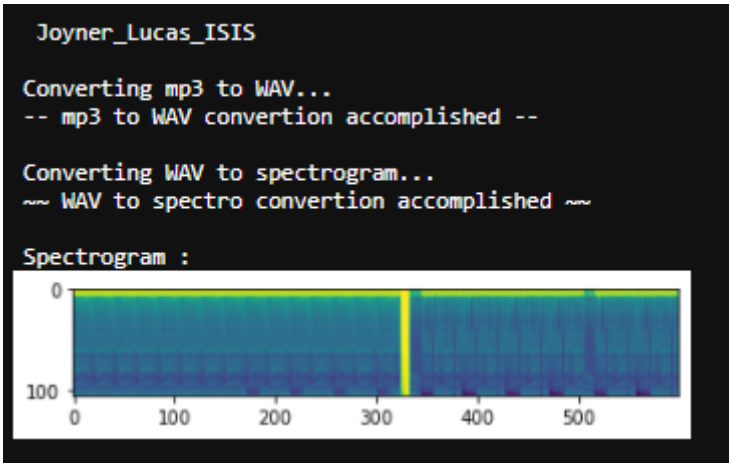
Α.2 KeyFinder.py

Κατά την εκτέλεση του keyFinder.py ο χρήστης καλείται να πληκτρολογήσει το όνομα του μουσικού τραγουδιού που επιθυμεί από την λίστα των αρχείων που βρίσκονται στον φάκελο final_dataset_mp3 όπως φαίνεται στη εικόνα Α.1 .



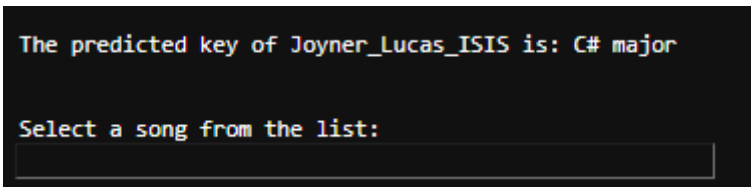
Α.1 Εισαγωγή ονόματος mp3 file

Στην συνέχεια το πρόγραμμα είναι υπεύθυνο για την μετατροπή του mp3 αρχείου στην κατάλληλη μορφή προς επεξεργασία για τα νευρωνικά δίκτυα. Πρώτα μετατρέπεται σε μορφή wav και έπειτα δημιουργείται το spectrogram ακριβώς όπως αναφέρθηκε στην ενότητα 5.3. Για κάθε επιτυχή μετατροπή εμφανίζεται ένα μήνυμα προς τον χρήστη όπως επίσης και το ίδιο το spectrogram για να ακολουθήσει η διαδικασία της πρόβλεψης (εικόνα Α.2).



Α.2 Μετατροπή του mp3 αρχείου

Αφού το spectrogram μετατραπεί σε tensor τροφοδοτείται στα δύο νευρωνικά προκείμενου να προβλεφθεί το μουσικό του κλειδί. Έπειτα εμφανίζεται η τελική "απάντηση" (εικόνα Α.3) των δικτύων και ζητείται από τον χρήστη να εισάγει άλλο αρχείο. Για να τερματίσει το πρόγραμμα ο χρήστης πρέπει να εισάγει ως όνομα ενός mp3 file το 0.



Α.3 Πρόβλεψη νευρωνικών