ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Μίξη Τραγουδιών με χρήση Νευρωνικών Δικτύων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Εμμανουήλ Κ. Βασιλόπουλος

Επιβλέπων : Γεώργιος Στάμου

Αναπληρωτής Καθηγητής

.Αθήνα, Ιούλιος 2020

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Μίξη Τραγουδιών με χρήση Νευρωνικών Δικτύων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Εμμανουήλ Κ. Βασιλόπουλος

Επιβλέπων : Γιώργος Στάμου

Αναπληρωτής Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 3η Σεπτεμβρίου 2020.

.....	.....	.....
Γιώργος Στάμου	Στέφανος Κόλλιας	Ανδρέας-Γεώργιος Σταφυλοπάτης

Αθήνα, Ιούλιος 2020

.....

Εμμανουήλ Κ. Βασιλόπουλος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Ο σκοπός της εργασίας είναι η σχεδίαση και υλοποίηση ενός συστήματος που αυτοματοποιεί την μίξη μουσικής. Πλατφόρμες όπως το Spotify και Apple Music παρέχουν λίστες τραγουδιών ανά είδος, ημερομηνία κυκλοφορίας κλπ. Δεν παρέχουν τη δυνατότητα, όμως της ενοποίησης της μουσικής και της αναπαραγωγής των τραγουδιών χωρίς παύσεις ή με κάποια συνοχή. Το ίδιο ισχύει για τα κοινώς χρησιμοποιούμενα συστήματα αναπαραγωγής μουσικής. Το σύστημα που σχεδιάσθηκε λαμβάνει στην είσοδο του μία play-list και στην έξοδο του παράγει τα κατάλληλα σήματα που ελέγχουν ένα dj λογισμικό. Η μέθοδος που ακολουθήσαμε χρησιμοποιεί τον εντοπισμό σημείων στο χρόνο ενός τραγουδιού χρησιμοποιώντας Τεχνητά Νευρωνικά Δίκτυα. Η έρευνα της εργασίας βασίστηκε σε αυτό το πρόβλημα, με τα καλύτερα αποτελέσματα να παράγονται από Συνελικτικά Νευρωνικά Δίκτυα ή αλλιώς Convolutional Neural Networks (CNN) σε συνδυασμό με την κατάλληλη μορφή έκφρασης των δεδομένων. Τα προβλεπόμενα σημεία του Τ.Ν.Δ. χρησιμοποιούνται από ένα δεύτερο σύστημα που αυτοματοποιεί μία βασική τεχνική των δισκοθετών (deejay) για τη μετάβαση από το ένα τραγούδι στο επόμενο. Το τελικό συμπέρασμα, όσον αφορά τον εντοπισμό σημείων στο χρόνο, είναι η σημαντικότητα της μορφής της εισόδου και κυρίως της εξόδου του συστήματος κατά τη διάρκεια της εκπαίδευσης, με την αρχιτεκτονική να παίζει ρόλο στη βελτιστοποίηση του συστήματος.

<u>Λέξεις κλειδιά</u>

dj λογισμικό, CNN, μίξη, Τ.Ν.Δ., λίστες τραγουδιών, εκπαίδευση

Abstract

The purpose of this thesis is the design and implementation of a system that automates the process of mixing music. Widely used services such as Spotify and Apply Music provide playlists grouped by genre, date of release etc. They do not provide the ability of unifying the music and the reproduction of the songs with harmony or continuity. The same applies for the commonly used media players. The system designed receives in its input a play-list and it produces in its output the proper signals the control a dj software. The method we followed uses the detection of points in time of a song using Artificial Neural Networks. The research of the thesis is based on this problem, with the best results given by a Convolutional Neural Network, in combination with the proper data representation. The predicted points given by the A.N.N. are used by a second system that automates the basic technic that deejays use to transition from the current song playing to the next. The final conclusion, regarding the detection of points in time, is the significance of the form of the input and more importantly of the output, during the training phase, with the architecture contributing to the optimization of the system.

Λέξεις κλειδιά

dj software, CNN, mixing, A.N.N., playlist, training

Ευχαριστίες

Θα ήθελα να ευχαριστήσω την μητέρα μου Λένα, τον αδερφό μου Ηλία, τον πατέρα μου Κώστα και τους τρεις φίλους μου Στέλιο, Ιπποκράτη και Παντελή που πάντα με στηρίζουν και με βοηθούν στην πρόοδο μου. Επίσης, θα ήθελα να ευχαριστήσω τον καθηγητή μου κ.Γιώργο Στάμου, ο οποίος σκέφτηκε την ιδέα της εργασίας μου και με επηρέασε στις απόψεις μου ως ηλεκτρολόγος μηχανικός μέσω των μαθημάτων του. Τέλος, να ευχαριστήσω τους συναδέλφους μου Έντι και Μπάμπη που συνέβαλαν στην υλοποίηση του συστήματος. Αυτή εργασία αφιερώνεται στους λάτρεις της μουσικής.

Πίνακας Περιεχομένων

Σελίδα

Κεφάλαιο 1. Εισαγωγή 9
Κεφάλαιο 2. Ήχος και Μετασχηματισμοί 14
Κεφάλαιο 3. Σημεία Ενδιαφέροντος 30
Κεφάλαιο 4. MIDI 31
Κεφάλαιο 5. Μοντέλα Μηχανικής Μάθησης 46
Κεφάλαιο 6. Σύστημα και Αρχιτεκτονική 56
Κεφάλαιο 7. Δημιουργία Data-Set 68
Κεφάλαιο 8. Αύξηση Δεδομένων 78
Κεφάλαιο 9. Φόρτωμα Data-Set 84
Κεφάλαιο 10. Αποτελέσματα Πειραμάτων 88
Κεφάλαιο 11. Πρόβλεψη 100

Πίνακας Σχημάτων

Σχήμα 1.1 – Κυματομορφές Τραγουδιών και Σημεία στο χρόνο
Σχήμα 2.2.1. - Δύο spectrograms με διαφορετική ανάλυση
Σχήμα 2.2.2 - Δύο spectrograms με aliasing και χωρίς
Σχήμα 2.2.3 - Spectrogram
Σχήμα 2.2.4 - Mel-Spectrogram
Σχήμα 2.2.5 - MFCC
Σχήμα 2.2.6 – Constant-Q Transform
Σχήματα 2.2.7 έως 2.2.11 – Τέσσερις μετασχηματισμοί ίδιου τραγουδιού
Σχήματα 3.2.1 και 3.2.2 - Δύο διαφορετικές επιλογές για το Cue Point στο ίδιο τραγούδι.
Σχήματα 3.2.3 και 3.2.4 - Δύο διαφορετικές επιλογές για το Stop-Intro Point στο ίδιο τραγούδι
Σχήμα 3.2.5 – Mix-Point
Σχήμα 3.2.6 – Stop-Mix Point
Σχήμα 3.2.7 - Rekordbox interface
Σχήμα 3.2.8 – Rekordbox XML δομή
Σχήμα 3.2.9 – Virtual DJ Interface
Σχήμα 3.2.10 – Virtual DJ XML
Σχήμα 4.1.1. Ένα απλό σύστημα MIDI
Σχήμα 4.1.2. Προσαρμοσμένο σύστημα MIDI
Σχήμα 7.2.1. Ολόκληρος μετασχηματισμός (άνω) και 4 συνεχόμενοι υπο-μετασχηματισμοί (κάτω)
Σχήμα 7.2.2 – Ολόκληρος μετασχηματισμός και όλοι οι επιλεγμένοι υπο-μετασχηματισμοί
Σχήμα 7.2.3. Κατανομή 500 κλάσεων.. Data-set με μέγεθος παραθύρου 10 ms
Σχήμα 10.2.1 – Πείραμα 1
Σχήμα 10.2.2 – Πείραμα 2
Σχήμα 10.2.3 – Πείραμα 3
Σχήμα 10.2.4 – Πείραμα 4
Σχήμα 10.2.5 – Πείραμα 5
Σχήμα 10.2.6 – Πείραμα 6
Σχήμα 10.3.1 – Πείραμα 7
Σχήμα 10.3.2 – Πείραμα 8
Σχήμα 11.1.1 – Σύστημα Πρόβλεψης (Data Pipeline)
Σχήμα 11.2.1 – Διάνυσμα Πρόβλεψης
Σχήμα 11.2.2 – Πρόβλεψη Σημείων Ενδιαφέροντος Τραγουδιού

ΚΕΦΑΛΑΙΟ 1
ΕΙΣΑΓΩΓΗ

1.1 Περίπτωση Χρήσης

Έστω ότι έχει διοργανωθεί ένα πάρτυ γενεθλίων με πολλούς καλεσμένους. Το 50% της επιτυχίας του πάρτυ είναι η μουσική και η διάθεση που δημιουργεί. Έστω ο εορταζόμενος/η ότι δημιουργεί μία play-list που αποτελείται από 300 τραγούδια με συγκεκριμένη σειρά. Η πρώτη σκέψη για την αναπαραγωγή της μουσικής είναι να πατηθεί το “play” και να παίξουν όλα τα τραγούδια της play-list, με ένα τραγούδι να ξεκινάει όταν το προηγούμενο φτάσει στο τέλος του.

Το πρώτο πρόβλημα που εμφανίζεται είναι η μέση διάρκεια των τραγουδιών, η οποία υπολογίζεται στα τρία λεπτά. Σ’ένα πάρτυ όμως όλοι θέλουν να περάσουν ωραία καθ’όλη τη διάρκεια. Με αυτή την

προσέγγιση ακόμα και αν όλα τα τραγούδια είναι τα super-hits των τελευταίων 5 δεκαετιών, το πάρτυ θα καταλήξει βαρετό. Ένα τραγούδι συναρπάζει αρχικά όταν οι ακροατές δε γνωρίζουν ότι θα παίξει. Μετά το πρώτο λεπτό αυτός ο ενθουσιασμός αρχίζει και σβήνει. Εκείνη είναι η σωστή στιγμή να αλλάξει τραγούδι. Πρέπει όμως να περάσουν δύο λεπτά κατά μέσο όρο για να γίνει αυτό. Για παράδειγμα, έστω ότι το επόμενο κομμάτι που θα παίξει είναι το πασίγνωστο Macarena. Όλοι παγκοσμίως γνωρίζουν την χορογραφία του τραγουδιού η οποία έχει 16 κινήσεις (4 bars = 16 beats). Οι καλεσμένοι θα επαναλάβουν τον χορό 3 με 4 φόρες. Έπειτα χάνεται η "μαγεία" του και σταματάει να είναι διασκεδαστικός.

Μια λύση θα ήταν η δημιουργία ενός αυτομάτου mixer. Με αυτόν τον τρόπο κανείς δε θα χρειάζεται να ασχοληθεί με την αναπαραγωγή της μουσικής. Η μονή προϋπόθεση είναι η προετοιμασία της play-list. Ακόμα και αυτό το έργο μπορεί να αυτοματοποιηθεί αλλά δεν είναι ο στόχος αυτής της διπλωματικής. Η κυρία εστίαση βρίσκεται στη μετάβαση από ένα τραγούδι σε ένα άλλο.

Αυτή η εφαρμογή μπορεί να χρησιμοποιηθεί σε καφετερίες, bars ή ακόμα και clubs όταν βελτιστοποιηθεί και είναι έτοιμη για παράγωγη. Επίσης, μπορεί να χρησιμοποιηθεί και ως επέκταση σε streaming apps μουσικής όπως το Spotify, Tidal κλπ, παρέχοντας στους χρήστες ατελείωτα mixes.

1.2 Έμπνευση

Πως είναι δυνατό να σχεδιαστεί ένα τέτοιο σύστημα που να εξυπηρετεί αυτό το σκοπό; Βασικά, είναι απαραίτητο να υλοποιηθεί ένα πρόγραμμα-δισκοθέτη (disc jockey) και επομένως θα μελετηθεί αυτό το επάγγελμα. Όπως αναφέρεται, δίνεται μια play-list με συγκεκριμένο αριθμό τραγουδιών. Το πρόβλημα είναι *πως* θα γίνει η μίξη των τραγουδιών και πιο σημαντικά *πότε*.

Στόχος είναι η μετάβαση από το ένα κομμάτι στο επόμενο ομαλά και με <<σωστό τρόπο>>. Ο λόγος που δηλώνεται ως <<σωστός τρόπος>> είναι η υποκειμενικότητα του θέματος. Ωστόσο κρύβεται από πίσω μία λογική. Τι συμβαίνει λοιπόν, πριν, κατά τη διάρκεια και μετά τη μίξη δύο τραγουδιών. Η παρακάτω σειρά ενεργειών περιγράφει αυτό ακριβώς:

1) Ξεκινάει η αναπαραγωγή του πρώτου τραγουδιού και ακούγεται στην έξοδο των ηχείων.
2) Το δεύτερο τραγούδι προετοιμάζεται να μιξαριστεί, το οποίο σημαίνει ότι απαιτείται η εύρεση ενός cue point. Όταν πατηθεί το κουμπί play το δεύτερο τραγούδι ξεκινάει την αναπαραγωγή του.
3) Έπειτα πρέπει να βρεθεί το σημείο start-mix point του πρώτου τραγουδιού που παίζει ήδη. Πρόκειται για ένα μελλοντικό σημείο στο χρόνο. Όταν το πρώτο τραγούδι φτάσει σ'αυτό το σημείο το δεύτερο τραγούδι πρέπει να εκκινήσει.
4) Τότε το δεύτερο τραγούδι εκτίθεται στην έξοδο των ηχείων και τα δυο τραγούδια παίζουν το ένα πάνω στο άλλο ταυτόχρονα (μιξάρονται).
5) Κατά τη διάρκεια αυτής της διαδικασία το επόμενο σημείο που πρέπει να ληφθεί υπόψιν είναι το stop-mix point. Είναι ένα σημείο μετά από το start-mix point και δηλώνει το πέρας της μίξης των δυο τραγουδιών. Όταν το πρώτο τραγούδι φτάσει σ'αυτό το

σημείο τότε σταματάει την αναπαραγωγή του και συνεχίζει το δεύτερο μόνο του.

6) Επαναλαμβάνεται αυτή η διαδικασία μέχρι να τελειώσει η play-list.

Πίσω από αυτόν τον αλγόριθμο κρύβονται υπο-βήματα έτσι ώστε τα δύο τραγούδια να ακούγονται ως ένα. Το πιο σημαντικό είναι το beat-matching. Πρόκειται για μία διαδικασία ευθυγράμμισης των beats δύο διαφορετικών τραγουδιών που μπορεί να επιτευχθεί επεξεργάζοντας και δύο τραγούδια με τέτοιο τρόπο, έτσι ώστε να έχουν τα ίδια BPM ή Beats ανά λεπτό. Το beat-matching είναι διαφορετικό πρόβλημα που δε θα ερευνηθεί σ'αυτή την εργασία αλλά για τις ανάγκες της θα χρησιμοποιηθεί το κουμπί sync. Αυτό το κουμπί παρέχει την δυνατότητα ευθυγράμμισης των beats δύο τραγουδιών τραγουδιών.

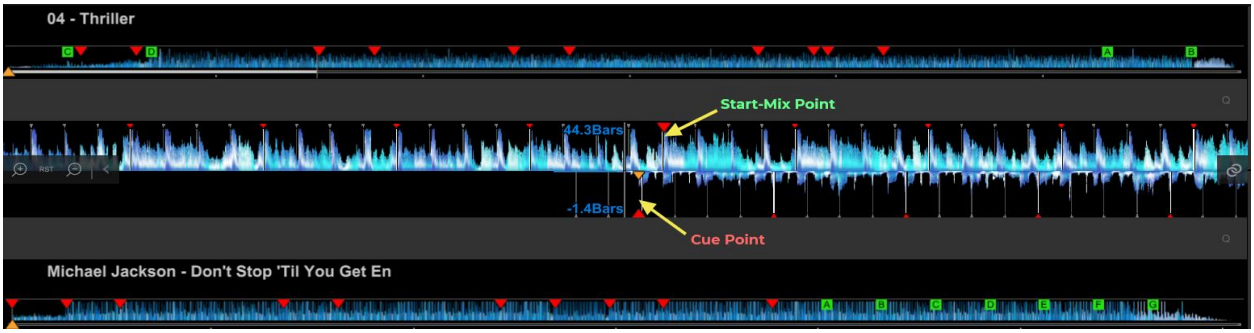
1.3 Υλοποίηση

Καταλήγουμε ότι χρειάζονται τρία σημεία στο χρόνο για να υπάρχει μια ολοκληρωμένη μετάβαση από ένα τραγούδι σ'ένα άλλο. Εν τέλει όμως θα υπολογισθεί ένα ακόμα:

1) Cue Point
2) Stop-Intro Point
3) Start-Mix Point
4) Stop-Mix Point

Stop-Intro Point: Η πλειονότητα των τραγουδιών έχει εισαγωγή. Συνήθως είναι μια ενότητα του τραγουδιού λιγότερο έντονη από τις υπόλοιπες. Δηλαδή, παίζουν λιγότερα όργανα, υπάρχουν λίγα φωνητικά ή καθόλου κλπ. Το Stop-Intro Point δηλώνει το τέλος αυτής της ενότητας. Αργότερα θα φανεί η χρησιμότητα αυτού του σημείου.

Αυτή η τετράδα σημείων δεν είναι μοναδική. Ένας δισκοθέτης μπορεί να επιλέξει διαφορετική τετράδα για το ίδιο ζευγάρι τραγουδιών. Επομένως, τα Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks) θα γίνουν το εργαλείο υπολογισμού της τετράδας σημείων, λόγω της πιθανοτικής τους φύσης. Απλώς, το σύστημα που θα σχεδιασθεί σ'αυτή την πτυχιακή εργασία παίρνει ως είσοδο ένα τραγούδι και στην έξοδο του δίνει 4 σημεία στο χρόνο. Τα σημεία της τετράδας θα ονομαστούν σημεία ενδιαφέροντος.



Σχήμα 1.1 – Κυματομορφές Τραγουδιών και Σημεία στο χρόνο

ΚΕΦΑΛΑΙΟ 2

ΉΧΟΣ ΚΑΙ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΙ

2.1 Εισαγωγή

Όλοι οι μουσικοί συνθέτουν και εκτελούν μουσική με τη βοήθεια μουσικών οργάνων με το πιο δημοφιλές να είναι οι ανθρώπινες φωνητικές χορδές. Αλλά τι είναι η μουσική; [1] Η μουσική είναι μία μορφή τέχνης και πολιτισμική δραστηριότητα, της οποίας το μέσο είναι ο ήχος. Και τι είναι ο ήχος; [2] Ο ήχος μπορεί να διαδίδεται με κάποιο μέσο όπως ο αέρας, το νερό και τα στερεά, όπως τα διαμήκη κύματα. Τα ηχητικά κύματα δημιουργούνται από μία ηχητική πηγή, όπως το δονούμενο διάφραγμα ενός ηχείου stereo. Η ηχητική πηγή δημιουργεί στο περιβάλλων μέσο. Όσο η πηγή συνεχίζει τις δονήσεις στο μέσο, οι δονήσεις διαδίδονται μακριά από την πηγή με την ταχύτητα του ήχου δημιουργώντας έτσι το ηχητικό κύμα. Ένας ήχος ή αλλιώς ένα ηχητικό κύμα μπορεί να ερμηνευτεί ως ένα συνεχές/αναλογικό σήμα.

Η μουσική που ηχογραφείται σε ένα στούντιο αποθηκευόταν σε βινύλια. Στην εποχή των υπολογιστών όμως, δημιουργήθηκε η ανάγκη για ψηφιακά σήματα και έτσι εισήχθη το Pulse-Code Modulation (PCM). [3] Είναι μια μέθοδος που χρησιμοποιείται για την ψηφιακή αναπαράσταση ενός δειγματοληπτειμένου αναλογικού σήματος. Είναι η καθιερωμένη μορφή του ψηφιακού ήχου στους υπολογιστές, compact discs (CDs), ψηφιακή τηλεφωνία και άλλων ψηφιακών εφαρμογών ήχου. Σ'ένα PCM stream, το αναλογικό σήμα δειγματοληπτείται κανονικά σε ομοιόμορφα διαστήματα, και κάθε δείγμα στρογγυλοποιείται (quantized) στη κοντινότερη τιμή εντός ενός εύρους ψηφιακών βημάτων. Πρόκειται για μια μορφή χωρίς απώλειες πληροφορίας (lossless), με τη δυνατότητα μετατροπής του ψηφιακού σήματος σε αναλογικού.

Άλλες ασυμπίεστες μορφές αρχείων ήχου είναι οι γνώστες: WAV, AIFF κλπ. Υπάρχουν και μορφές που χρησιμοποιούν συμπίεση δεδομένων με απώλειες για την κωδικοποίηση δεδομένων ήχου όπως το MP3. Όμως ένα σύστημα όπως αυτό της εργασίας δε μπορεί να επεξεργαστεί αμέσως τέτοια αρχεία ήχου και πρέπει να τα αποκωδικοποιήσει πρώτα.

Η παρατήρηση του ήχου αποδίδει κάποιες ποσότητες εξαρτημένες του χρόνου. Αυτές οι ποσότητες, οι οποίες υποθέτονται μετρήσιμες, θα λέγονται σήματα. Αντιστοιχίζονται στα μαθηματικά με την έννοια της συνάρτησης και έτσι τα σήματα μοντελοποιούνται ως συναρτήσεις. Οι υπολογιστές διαβάζουν τον ήχο ως μια συνάρτηση διακριτού χρόνου που μεταφράζεται σε ένα πινάκα. Οι μονοδιάστατοι πινάκες A[t] χρησιμοποιούνται για την αναπαραγωγή μονοφωνικών ήχων (mono). Οι διδιάστατοι πίνακες A[k, t] περιγράφουν στερεοφωνικούς ήχους με το k να αντιπροσωπεύει το αριστερό ή το δεξί κανάλι. Κάθε τιμή του t αντιστοιχίζεται σ'ένα σημείο στο χρόνο.

Ένα αρχείο ήχου MP3 υψηλής ποιότητας με bit-rate των 320 kbps ενός ήχου διάρκειας 3 λεπτών έχει μέγεθος 7.2 MB. Το αντίστοιχο WAV αρχείο με ρυθμό δειγματοληψίας 44100 Hz και bit-rate των 16 bits έχει μέγεθος 31.752 MB. Επομένως, για ένα τραγούδι είναι προφανές ότι υπάρχουν εκατομμύρια σημεία για επεξεργασία που αυξάνουν την πολυπλοκότητα του προβλήματος.

Το μέγεθος τους αρχείου WAV υπολογίζεται ως εξής:

$$f_{size} = duration * sr * br * n_{channels}$$

$$f_{bytes} = \frac{f_{size}}{8}$$

με: a) sr → sample-rate, b) br → bit-rate

Τα αρχεία ήχου είναι διακριτά σήματα το οποίο σημαίνει ότι μπορούμε να εφαρμόσουμε Ψηφιακή Επεξεργασία Σήματος (ΨΕΣ) και γνώστες μεθόδους του Music Information Retrieval (MIR). Η ανάλυση συχνά απαιτεί ένα βαθμό σύνοψης και για τον ήχο αυτό γίνεται εφικτό με εξαγωγή χαρακτηριστικών (feature extraction), ειδικά όταν το περιεχόμενο του ήχου αναλύεται και μηχανική μάθηση πρόκειται να εφαρμοστεί. Ο σκοπός είναι η μείωση της ποσότητας των δεδομένων σε ένα διαχειρίσιμο σύνολο τιμών έτσι ώστε η μάθηση να εκτελεστεί σ’ένα αποδοτικό χρονικό πλαίσιο. Στο κεφάλαιο 4 αιτιολογείται η ανάγκη των μετασχηματισμών που χρησιμοποιούνται στην Ψ.Ε.Σ. και M.I.R., όπως: Spectrogram, Mel-Spectrogram, MFCC, Constant-Q κλπ.

2.2 Μετασχηματισμοί

Οι περισσότεροι μετασχηματισμοί που αναφέρονται θα μελετηθούν και θα συγκριθούν για το ποιος δίνει τα καλύτερα αποτελέσματα.

1. Short-Time Fourier Transform (STFT)

[4] Ο όρος μετασχηματισμός Fourier αναφέρεται στην αναπαράσταση στο πεδίο της συχνότητας και στη μαθηματική λειτουργία που συνδέει αυτή την αναπαράσταση σε μία συνάρτηση του χρόνου. Ο μετασχηματισμός Fourier μίας συνάρτησης του χρόνου είναι ο ίδιος μία συνάρτηση της συχνότητας με μιγαδικές τιμές. Το πλάτος (magnitude) αντιπροσωπεύει την ποσότητα της συχνότητας που υπάρχει στην αρχική συχνότητα. Το όρισμα (arg ή argument) είναι το offset φάσης της βασικής ημιτονοειδούς σ'αυτή τη συχνότητα.

[4] Ο βραχέος χρόνου μετασχηματισμός Fourier είναι ένας μετασχηματισμός σχετικός με τον Fourier και χρησιμοποιείται για τον καθορισμό της ημιτονοειδούς συχνότητας και περιεχόμενου φάσης τοπικών τμημάτων ενός σήματος καθώς αλλάζει με την πάροδο του χρόνου. Πρακτικά, η διαδικασία υπολογισμού του STFTs είναι η διαίρεση ενός μεγάλου σε διάρκεια σήματος σε μικρότερα τμήματα ίσου μήκους και έπειτα ο υπολογισμός του μετασχηματισμού Fourier ξεχωριστά σε καθ'ένα απ'αυτά τα τμήματα. Έτσι εμφανίζεται το φάσμα Fourier σε κάθε τμήμα. Το αποτέλεσμα είναι λοιπόν πολλοί μετασχηματισμοί Fourier ο ένας μετά τον άλλον. Όποτε αυτά τα φάσματα μπορούν να απεικονισθούν ως συνάρτηση του χρόνου γνωστή και ως spectrogram.

Στην περίπτωση διακριτού χρόνου, τα δεδομένα που θα μετασχηματισθούν μπορούν να διαιρεθούν σε επικαλυπτόμενα τμήματα. Κάθε τμήμα μετασχηματίζεται (Μ/Σ Fourier) και το μιγαδικό

αποτέλεσμα προστίθεται σ'ένα πινάκα, ο οποίος καταγράφει το πλάτος και τη φάση κάθε σημείου στο χρόνο και τη συχνότητα. Αυτό μπορεί να εκφραστεί ως:

$$\mathbf{STFT}\{x[n]\}(m,\omega) \equiv X(m,\omega) = \sum_{n=-\infty}^{+\infty} x[n]w[n-m]e^{-j\omega n}$$

οπού $x[n]$ το σήμα και $w[n]$ το παράθυρο. Σ'αυτή την περίπτωση, το m είναι διακριτό και το ω είναι συνεχές, αλλά στις περισσότερες εφαρμογές ο STFT εφαρμόζεται σε υπολογιστή χρησιμοποιώντας Fast Fourier Transform, όποτε και οι δύο μεταβλητές είναι διακριτές.

Ο STFT ενός κομματιού υπολογίζεται με τη βοήθεια της βιβλιοθήκης "scipy" της γλωσσάς προγραμματισμού Python. Πιο συγκεκριμένα θα χρησιμοποιηθεί η συνάρτηση signal.stft που απαιτεί τις παραμέτρους:

- x : Χρονική σειρά (σήμα)
- f_s : Συχνότητα δειγματοληψίας του x
- $window$: Επιθυμητό παράθυρο που θα χρησιμοποιηθεί.
- $nperseg$: Μήκος κάθε τμήματος
- $noverlap$: Αριθμός επικαλυπτόμενων σημείων μεταξύ τμημάτων
- $nfft$: Μήκος του FFT

Κατά τη διάρκεια των πρώτων πειραμάτων το $noverlap$ είχε τιμή μηδέν, με παράθυρο "Hamming" διάρκειας 10 ms. Το μήκος του παράθυρου υπολογίζεται με τη βοήθεια του ρυθμού δειγματοληψίας. Για παράδειγμα, αν έχει τιμή 44100 Hz ανά δευτερόλεπτο, σημαίνει ότι 44100 δείγματα εκπροσωπούν ήχο διάρκειας 1 δευτερόλεπτου ή 1000 ms. Επομένως ο αντίστοιχος αριθμός δειγμάτων για 10 ms υπολογίζεται:

$$1\,ms \rightarrow 44.1\,samples <=> 10\,ms \rightarrow 441\,samples$$

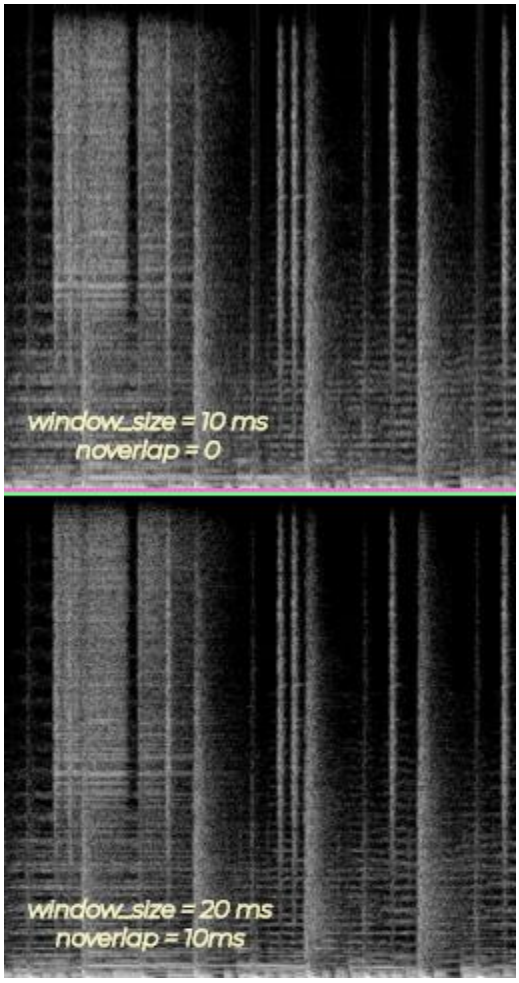
Δηλαδή, $window_{size} = 441$. Το μήκος κάθε τμήματος, δηλαδή το $nperseg$, είναι ίσο με το μήκος του παράθυρου. Ωστόσο, αυτή η

παραμετροποίηση οδήγησε σ’ένα μετασχηματισμό χαμηλό σε ανάλυση (resolution), ο οποίος είχε περιθώρια βελτίωσης.

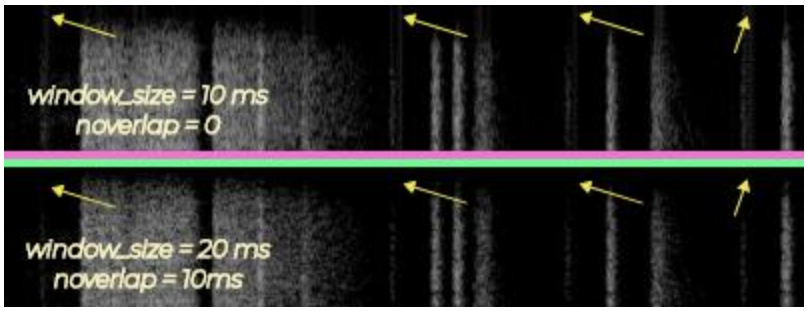
Μια καλύτερη ρύθμιση είναι:

$$window_{size} = 20\,ms \,\&\, noverlap = 50\%$$

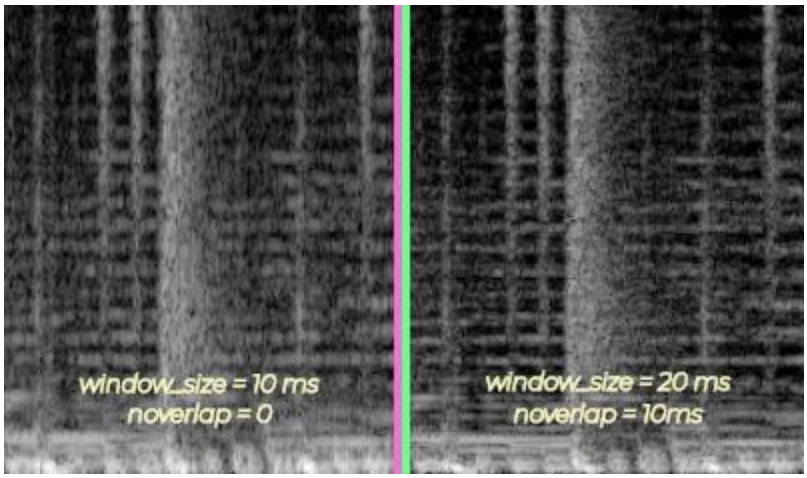
Ως αποτέλεσμα, το spectrogram έχει καλύτερη ανάλυση και λιγότερο aliasing και θα επιλεχθεί ως προκαθορισμένη ρύθμιση. Οι εικόνες 1, 2 και 3 παρουσιάζουν αυτές τις διαφορές.



Σχήμα 2.2.1. Ίδιο μέρος από spectrograms με διαφορετική παραμετροποίηση (10 - 15 seconds of Leah Labelle’s Sexify). a) $window_{size} = 10\,ms \,\&\, noverlap = 0$ *b)* $window_{size} = 20\,ms \,\&\, noverlap = 10\,ms$



Σχήμα 2.2.2. Τα μεγεθυμένα τμήματα του spectrogram δείχνουν την παρουσία aliasing με μηδενική επικάλυψη.
a) Άνω υπο-spectrogram, $window_{size} = 10\ ms$ *&* $noverlap = 0$.
b) Κάτω υπο-spectrogram, $window_{size} = 20\ ms$ *&* $noverlap = 10\ ms$



Σχήμα 2.2.3. Τα μεγεθυμένα τμήματα των spectrograms δείχνουν την διαφορά ανάλυσης που δίνουν οι δυο παραμετροποιήσεις.
a) Αριστερό υπο-spectrogram $window_{size} = 10\ ms$ *&* $noverlap = 0$.
b) Δεξί υπο-spectrogram, created with a $window_{size} = 20\ ms$ *&* $noverlap = 10\ ms$

Οι μόνες μεταβλητές των πειραμάτων είναι τα x, fs και $nfft$. Όσο υψηλότερη είναι η τιμή του $nfft$ τόσο υψηλότερη είναι η ανάλυση του STFT και συνεπώς τόσο αυξάνεται η χωρική πολυπλοκότητα.

2. Spectrogram

[5] Ένα spectrogram είναι η οπτική αναπαράσταση της ισχύος ενός σήματος στο χρόνο σε διαφορετικές συχνότητες που είναι παρούσες σε δεδομένη κυματομορφή. Αναπαρίσταται από ένα διδιάστατο γράφημα του οποίου ο χρόνος δίνεται στον οριζόντιο άξονα και η συχνότητα στον κάθετο. Το πλάτος (amplitude) των συχνοτικών συστατικών σε μία συγκεκριμένη χρονική στιγμή υποδεικνύεται από την ένταση ή το χρώμα του σημείου αυτού στο γράφημα. Χαμηλά πλάτη αναπαρίστανται από σκούρα μπλε χρώματα και υψηλά από πιο έντονα κίτρινα χρώματα. Υπολογίζεται από το σήμα εφαρμόζοντας FFT, το οποίο σχηματίζει μια αναπαράσταση χρόνου-συχνότητας. Για την ανακάλυψη συχνοτήτων σε κάθε χρονικό διάστημα, το σήμα διαιρείται σε τμήματα και στο καθένα εφαρμόζεται FFT.

Το τετράγωνο του πλάτους του STFT παράγει το spectrogram που θα χρησιμοποιηθεί από το σύστημα. Τα spectrograms χρησιμοποιούνται εκτενώς στο χώρο της μουσικής, των ραντάρ, για αναγνώριση φωνής κλπ. Ο λόγος που δε χρησιμοποιείται αμέσως ο STFT είναι λόγω της μιγαδικής του φύσεως και οι τιμές του δεν είναι διαχειρίσιμες.

Επίσης, οι τιμές ενός spectrogram είναι κοντά στο μηδέν. Έτσι, αφού υπολογιστεί το πλάτος του STFT, θα περάσει από μια λογαριθμική συναρτήσει και θα πολλαπλασιαστεί με το 10 για να μετριέται σε decibel.

$$spectrogram(t, f) = 10log(|STFT|^2) = 20log(|STFT|)$$

- t: χρόνος
- f: συχνότητα

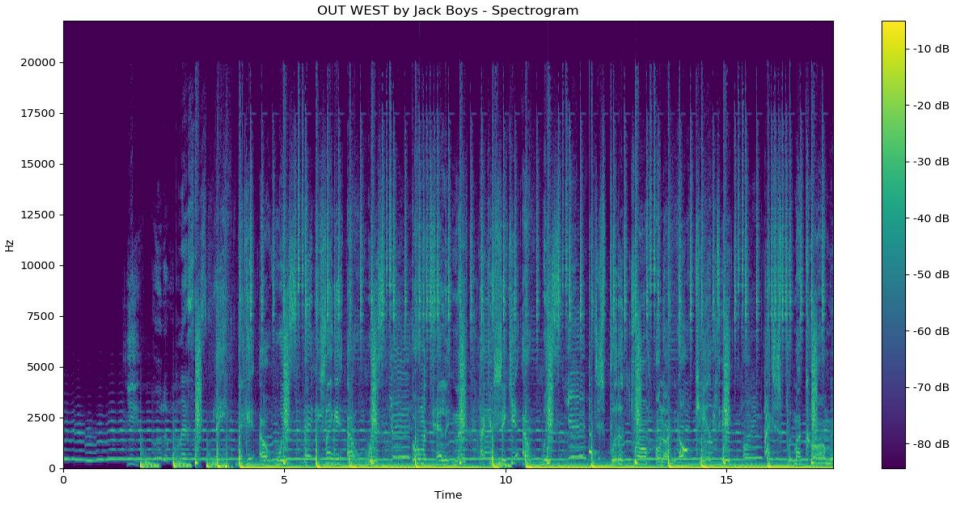
Ο οριζόντιος άξονας αναπαριστά τον χρόνο και ο κάθετος τη συχνότητα. Το εύρος τιμών στον κάθετο άξονα ξεκινάει από το 0 και φτάνει έως το ρυθμό δειγματοληψίας του αρχικού ήχου και το εύρος τιμών του οριζοντίου άξονα εξαρτάται από τη διάρκεια του αρχείου ήχου.

"t" και "f" δέχονται διακριτές τιμές μόνο. Εξαρτώντας του πειράματος:

- $t = 0 \rightarrow [0, 9]\ ms\ \ or\ [0, 19]\ ms$
- $t = 1 \rightarrow [10, 19]\ ms\ or\ [20, 39]\ ms$
- ..
- t= $n \rightarrow [n * 10, (n * 10 + 9)]\ ms\ or\ [n * 20, (n * 20 + 19)]\ ms$

Εξαρτώντας της συχνότητας δειγματοληψίας "f_s" και του "$nfft$":

- $f = 0 \rightarrow [0, f_s/nfft)$ Hz
- $f = 1 \rightarrow [f_s/nfft, 2 * f_s/nfft)$ Hz
- ..
- $f = (nfft - 1) \rightarrow [(nfft - 1) * f_s/nfft, f_s)$ Hz



Σχήμα 2.2.3 - Spectrogram

3. Mel-Spectrogram

Αφού υπολογιστεί το spectrogram ενός σήματος γίνεται να υπολογιστεί και το mel-spectrogram. Είναι το spectrogram του οποίου οι συχνότητες αναπαρίστανται στην κλίμακα mel. [6] Η κλίμακα mel είναι μια αντιληπτική κλίμακα των pitches η οποία κρίθηκε από ακροατές ότι είναι ίσα σε απόσταση το ένα από το άλλο. Το σημείο αναφοράς ανάμεσα σ'αυτή τη κλίμακα και στην κανονική μέτρηση συχνότητας ορίζεται αναθέτοντας ένα αντιληπτικό pitch των 1000 mels σε ένα τόνο 1000 Hz, 40dB πάνω από το κατώφλι του ακροατή. Περίπου 500 Hz πιο πάνω, αυξανόμενα μεγάλα διαστήματα κρίνονται από ακροατές να αναπαράγουν ίσες αυξήσεις pitch. Ως αποτέλεσμα, 4 οκτάβες στην κλίμακα hertz πάνω από τα 500 Hz κρίνονται να περιλαμβάνουν περίπου δυο οκτάβες στην κλίμακα mel. Το όνομα mel προέρχεται από τη λέξη melody έτσι ώστε να υποδείξει ότι η κλίμακα βασίζεται σε συγκρίσεις pitch .

Δεν υπάρχει μοναδικός ορισμός της κλίμακας mel. Μια δημοφιλής συνταγή δίνεται από τον O'Shaughnessy:

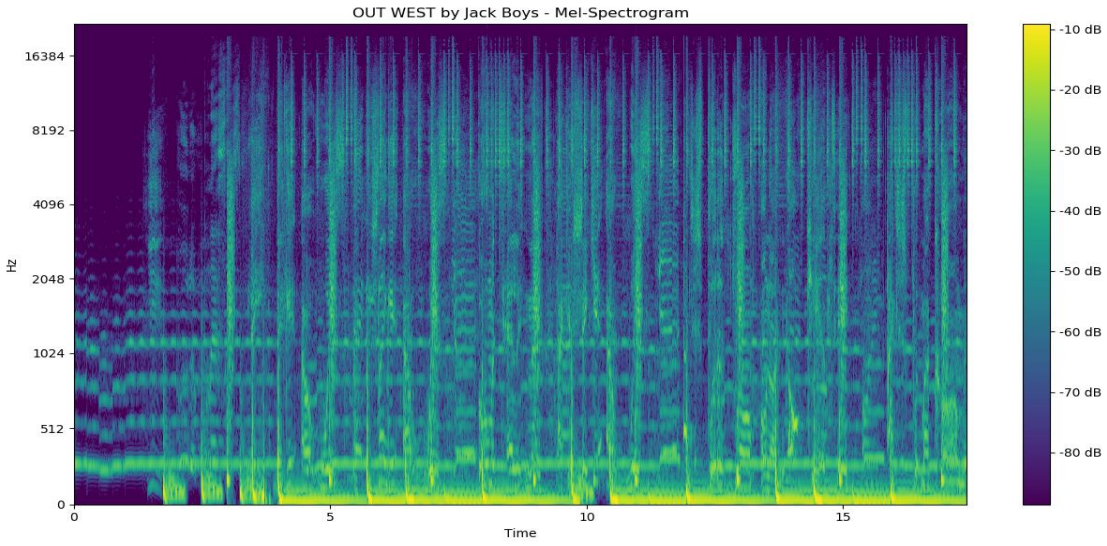
$$\mathrm{m} = 2595\log_{10}(1 + \frac{f}{700})$$

με m να είναι το pitch σε mels.

[7] Έτσι ένας τόνος με pitch (500 mels) το μισό ενός pitch ενός τόνου των 1000 Hz έχει συχνότητα περίπου 390 Hz, ενώ ένας τόνος με pitch το διπλάσιο ενός pitch ενός τόνου των 1000 Hz έχει συχνότητα περίπου 3429 Hz. Το ηχητικό σύστημα, λοιπόν, εφαρμόζει ανάλυση συχνότητας που μπορεί να εξομοιωθεί με ένα σύνολο από ζωνοπερατά φίλτρα των οποίων τα εύρη ζώνης αυξάνονται καθώς το κέντρο συχνότητας αυξάνονται.

Ένα από τα χαρακτηριστικά του pitch είναι ότι οι ανθρώπινοι ακροατές είναι εξαιρετικά ευαίσθητοι στις αλλαγές συχνότητας και μπορούν να ξεχωρίσουν δύο τόνους που διαφέρουν κατά 3 Hz ή περισσότερο αν οι τόνοι βρίσκονται κάτω από 500 Hz. Αν οι δύο τόνοι βρίσκονται πάνω από τα 500 Hz, οι άνθρωποι μπορούν να διακρίνουν ότι δύο τόνοι διαφέρουν αν διαχωρίζονται από $0.003F_0$, όπου F_0 είναι η συχνότητα του χαμηλότερου τόνου.

Τα mel-spectrograms του σύνολου δεδομένων θα υπολογισθούν με τη βοήθεια του libROSA. Το LibROSA είναι ένα πακέτο της Python για ανάλυση ήχου και μουσικής. Παρέχει τα κατάλληλα εργαλεία για τη δημιουργία συστημάτων M.I.R.

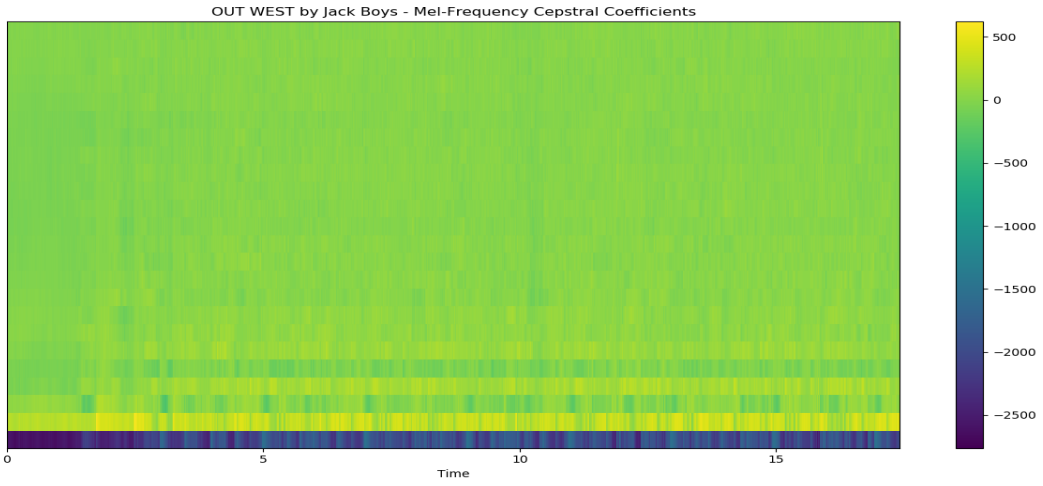


Σχήμα 2.2.4 – Mel-Spectrogram

4. Mel-Frequency Cepstral Coefficients

Mel-frequency cepstral coefficients (MFCCs) είναι συντελεστές που συνολικά αποτελούν ένα Mel-Frequency Cepstrum. *Αντλούνται από ένα τύπο cepstral αναπαράστασης του αρχείου ήχου (ένα μη-γραμμικό "spectrum-of-a-spectrum"). Η διάφορα ανάμεσα στο cepstrum και στο mel-frequency cepstrum είναι ότι στο MFC, οι ζώνες συχνοτήτων είναι σε ίσες αποστάσεις στην κλίμακα mel, που προσεγγίζει την απόκριση του ανθρωπίνου ακουστικού συστήματος πιο κοντά από τις ζώνες συχνοτήτων των οποίων οι αποστάσεις είναι γραμμικά ίσες. Αυτή διαστροφή της συχνότητας επιτρέπει την καλύτερη αναπαράσταση του ήχου. Τα MFCCs συνήθως αντλούνται αφού υπολογισθεί το mel-spectrogram, ακολουθώντας τα παρακάτω βήματα:

1. Υπολογισμός των λογαρίθμων των δυνάμεων σε κάθε συχνότητα-mel.
2. Υπολογισμός του διακριτού μετασχηματισμού συνημιτόνου της λίστας των mel λογαριθμισμένων δυνάμεων, σα να ήταν σήμα.
3. Τα MFCCs είναι το μέτρο του φάσματος που δίνεται ως αποτέλεσμα.



Σχήμα 2.2.5 - MFCC

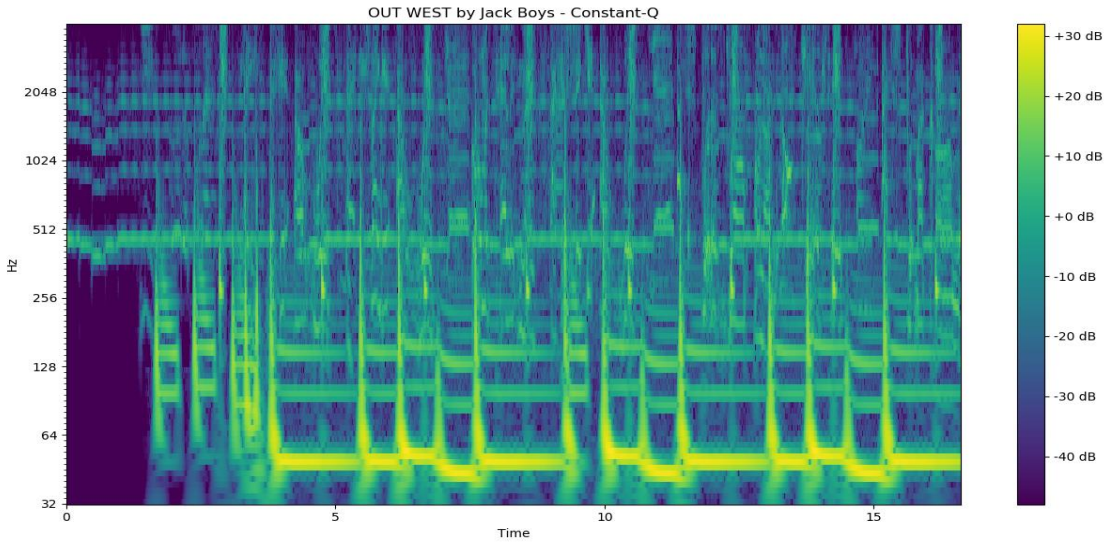
5. Constant-Q

[9] Οι συχνότητες που έχουν επιλεχθεί και αποτελούν την κλίμακα της Δυτικής μουσικής είναι γεωμετρικά χωριστές. Έτσι ο διακριτός μετασχηματισμός Fourier (DFT), παρόλο που είναι εξαιρετικά αποδοτικός στην υλοποίηση του FFT, παράγει συστατικά που δεν απεικονίζονται αποδοτικά σε μουσικές συχνότητες. Αυτό συμβαίνει διότι οι συχνότητες που υπολογίζονται με τον DFT διαχωρίζονται από μια σταθερή διάφορα και με μια σταθερή ανάλυση (resolution). Ένας παρόμοιος υπολογισμός με τον DFT με σταθερή αναλογία κεντρικής συχνότητας-αναλυσης είναι ο constant-Q μετασχηματισμός. Είναι ισοδύναμος με ένα φίλτρο 1/24-οκτάβας. Έτσι υπάρχουν δυο συχνοτικά συστατικά για κάθε μουσική νότα έτσι ώστε δυο γειτονικές νότες στη μουσική κλίμακα που παίζονται ταυτόχρονα να μπορούν να διαχωριστούν παντού στο εύρος μουσικών συχνοτήτων. Αυτός ο μετασχηματισμός έχει σχεδιασθεί για να αποκτηθεί ένα σταθερό μοτίβο στο πεδίο συχνότητας για ήχους με αρμόνικες συχνότητες. Συγκρίνεται με τον DFT που δίνει συχνότητες χωρισμένες σε ίσες αποστάσεις. Επίσης εκτός από πλεονεκτήματα ανάλυσης (resolution), η αναπαράσταση με σταθερό μοτίβο παραχωρεί το πλεονέκτημα ότι η ταυτοποίηση νότας, η αναγνώριση μουσικού οργάνου και ο διαχωρισμός σήματος μπορούν να γίνουν κομψά και ευθέως από έναν αλγόριθμο αναγνώρισης μοτίβων.

Για τον υπολογισμό του μετασχηματισμού Constant-Q ενός δείγματος ήχου θα χρησιμοποιηθεί το πακέτο της Python libROSA. Η υλοποίηση του ακολουθεί τον αλγόριθμο του “Constant-Q transform toolbox for music processing” από τους Christian Schorkhuber και Anssi Klapuri. Ο αριθμός των δειγμάτων μεταξύ διαδοχικών στηλών CQT εξαρτάται από τον επιθυμητό αριθμό bins. Αν ο αριθμός των bins n_{bins} είναι:

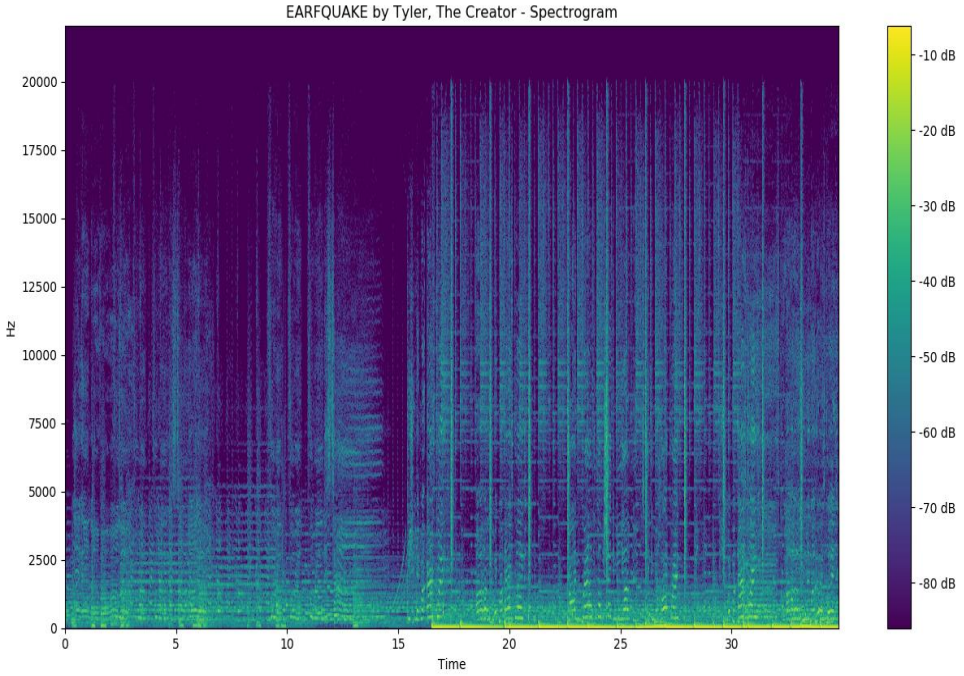
$$n_{bins} = 12 * n_{octaves}$$

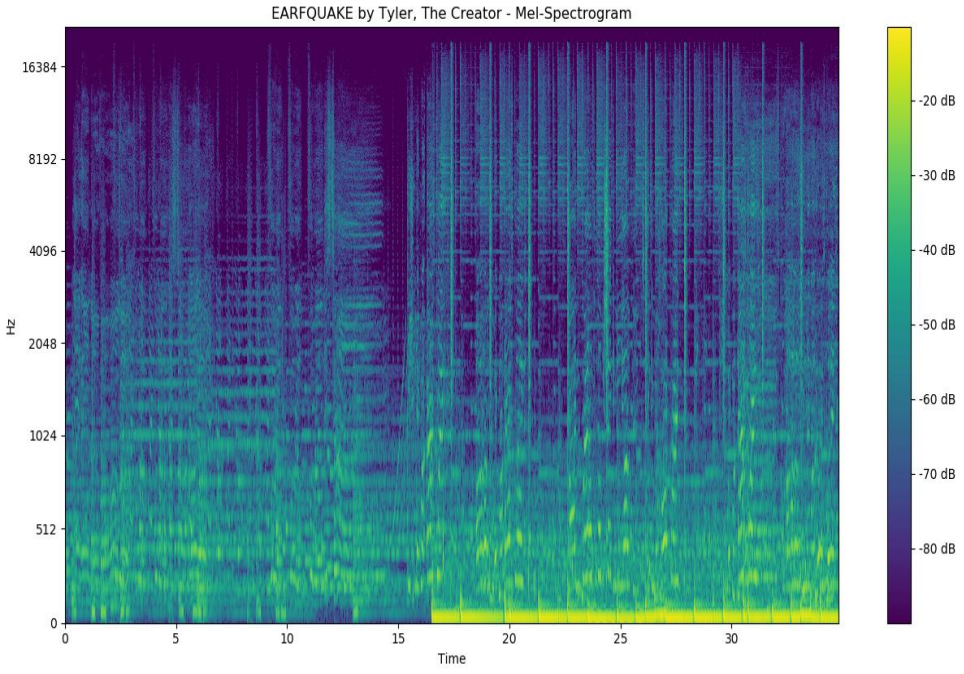
τότε ο αριθμός δειγμάτων ανάμεσα σε διαδοχικές στήλες CQT πρέπει να είναι πολλαπλάσιο του $2^{n_{octaves}}$. Επομένως, το $window_{size}$ περιορίζεται να έχει τιμές πολλαπλάσιες κάποιας δύναμης του 2.

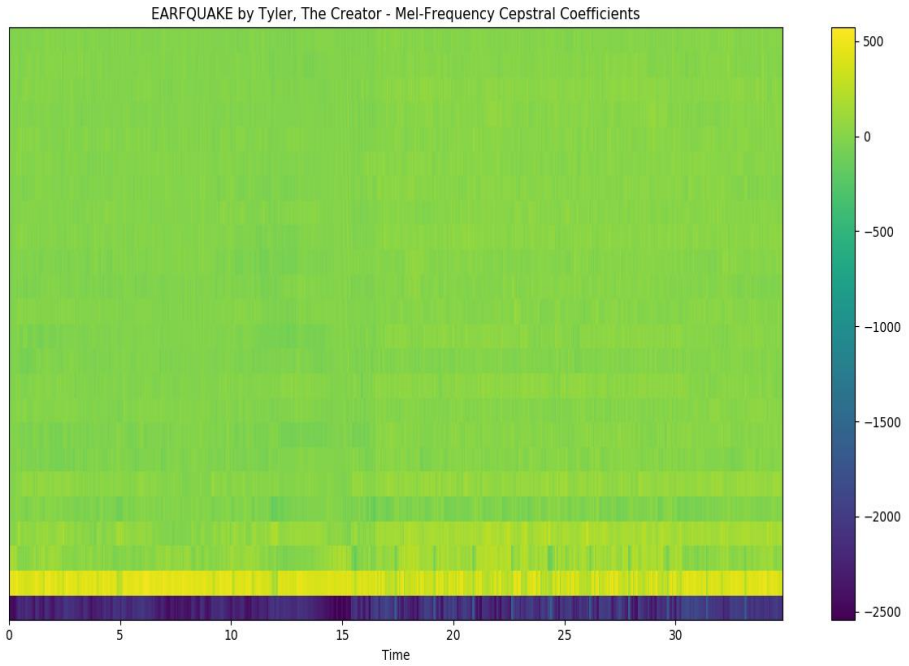


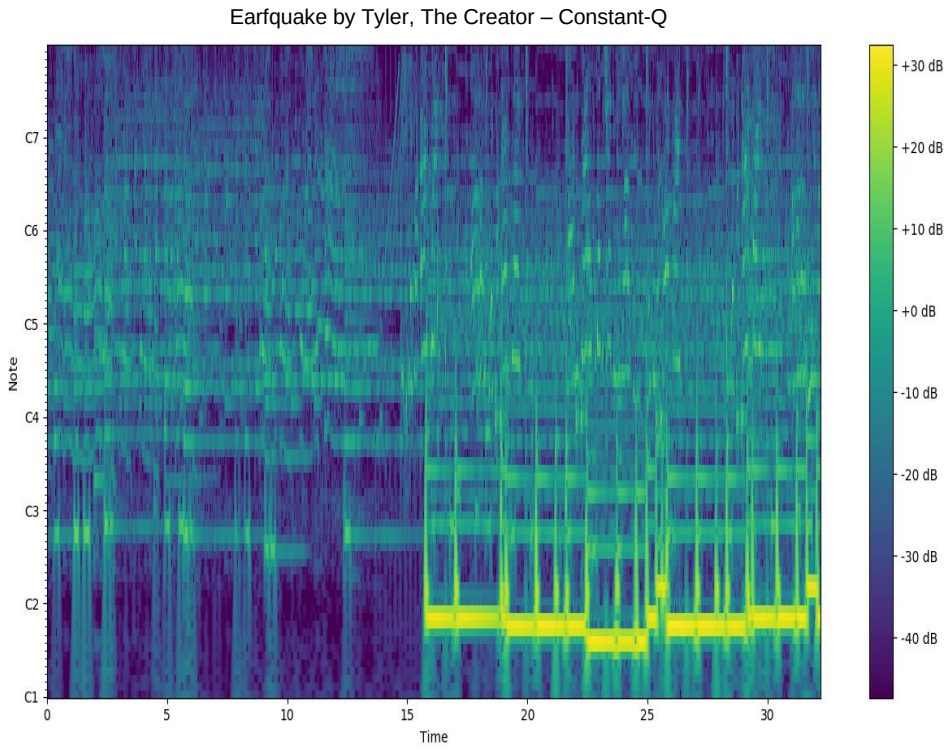
Σχήμα 2.2.6 – Constant-Q Transform

ΣΥΓΚΡΙΣΗ ΜΕΤΑΣΧΗΜΑΤΙΣΜΩΝ









Σχήματα 2.2.7 έως 2.2.11 – Τέσσερις μετασχηματισμοί ίδιου τραγουδιού

ΚΕΦΑΛΑΙΟ 3

ΣΗΜΕΙΑ ΕΝΔΙΑΦΕΡΟΝΤΟΣ (DJ)

3.1 Εισαγωγή

Όπως αναφέρθηκε στο κεφάλαιο 1, τέσσερα σημεία σ'ένα τραγούδι είναι αρκετά για τη μετάβαση απ'αυτό σ'ένα άλλο.

- Cue Point: σημείο στο χρόνο από το οποίο το επόμενο τραγούδι θα εκκινήσει την αναπαραγωγή του, όταν πατηθεί το κουμπί "play".
- Stop-Intro Point: σημείο που υποδεικνύει το μέγιστο χρονικό σημείο σύμφωνα με το οποίο το επόμενο τραγούδι μπορεί να μιξαριστεί με το τραγούδι που παίζει ήδη. Αν το "Stop-Mix Point" του τρέχοντος τραγουδιού δεν έχει ξεπεραστεί και το επόμενο φτάσει στο "Stop-Intro Point" τότε το επόμενο πρέπει να ξεκινήσει από το "Cue Point".

- Mix Point: σημείο στο χρόνο του τρέχοντος κομματιού. Μόλις ξεπερασθεί αυτό το σημείο, πρέπει να πατηθεί “play” για το επόμενο κομμάτι.
- Stop-Mix Point: σημείο στο χρόνο του τρέχοντος κομματιού. Όταν ξεπερασθεί αυτό το σημείο το τρέχων τραγούδι πρέπει να διακόψει την αναπαραγωγή του και το επόμενο κομμάτι να συνεχίζει μόνο του στην έξοδο των ηχείων.

Σημείωση:

$$t_{cuepoint} < t_{stopintropoint} < t_{mixpoint} < t_{stopmixpoint}$$

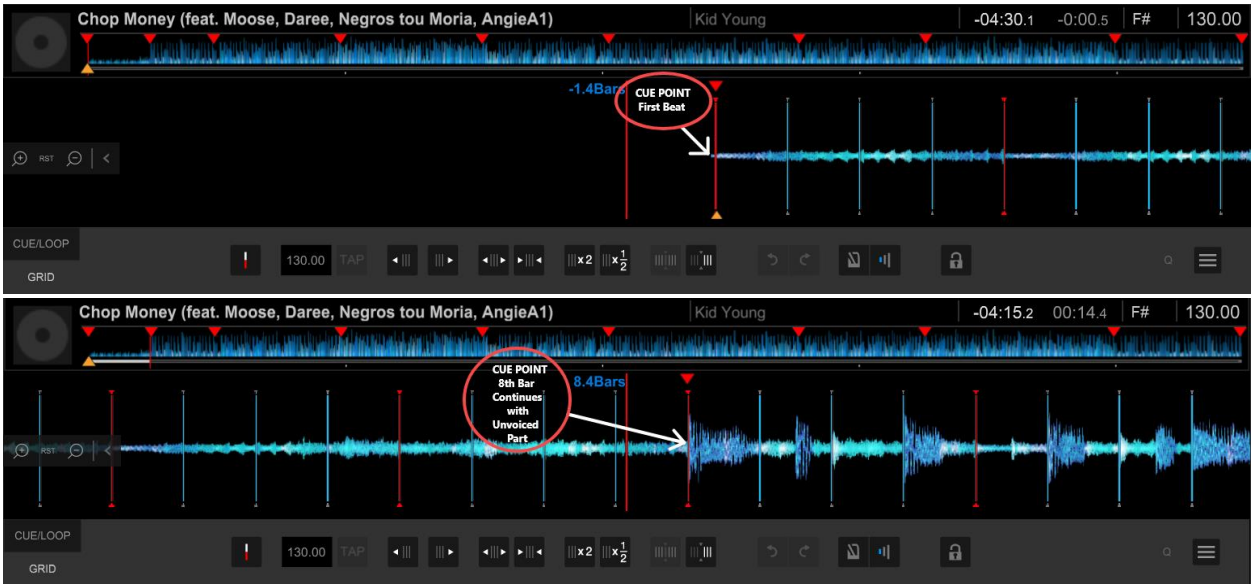
Τα σημεία ενδιαφέροντος θα χωριστούν σε δύο τύπους. Ο λόγος προκύπτει από το γεγονός ότι είναι απαραίτητο να υπάρχει μεγάλο σύνολο δεδομένων (data-set) που θα χρησιμοποιηθεί για την εκπαίδευση ενός συστήματος μηχανικής μάθησης κατάλληλα και για να υπάρχει μεγαλύτερη ακρίβεια. Ο αλγόριθμος υπολογισμού των σημείων ενδιαφέροντος περιγράφεται στο Κεφάλαιο 6 μαζί με την περιγραφή της αρχιτεκτονικής.

Σημείωση: Η αγγλική ονομασία των σημείων ενδιαφέροντος είναι Points Of Interest. Παρακάτω χρησιμοποιείται ο όρος P.O.I. που αναφέρεται στα σημεία.

3.2 Τοποθεσίες Σημείων Ενδιαφέροντος

Πως επιλέγονται αυτά τα σημεία; Πρώτα, πρέπει να εντοπίζονται εντός ενός χρονικού διαστήματος 40ms, με το μέσο του διαστήματος να είναι η ακριβής τοποθεσία του σημείου ενδιαφέροντος. Οπότε μια απόκλιση των 20ms από το πραγματικό σημείο είναι αποδεκτή από ένα πραγματικό deejay. Ωστόσο, καθώς το τελικό σύστημα θα χρησιμοποιήσει μία αυτοματοποιημένη μέθοδο για το beat-matching το αποδεκτό σφάλμα μπορεί να είναι μεγαλύτερο και πιο συγκεκριμένα το σφάλμα μπορεί να είναι μικρότερο του μισού της διάρκειας ενός beat.

Cue Point: βρίσκεται κυρίως στην αρχή ενός κομματιού. Συνήθως πριν απ'αυτό το σημείο υπάρχει λίγος ή καθόλου ήχος, εξαρτώντας του κομματιού. Μπορεί να είναι το πρώτο beat του κομματιού χωρίς να είναι απαραίτητο. Ένα άλλο μπορεί να βρίσκεται 16 beat μετά το πρώτο. Εξαρτάται κυρίως από τις προτιμήσεις του deejay και τον τρόπο μετάβασης από το ένα τραγούδι στο άλλο.



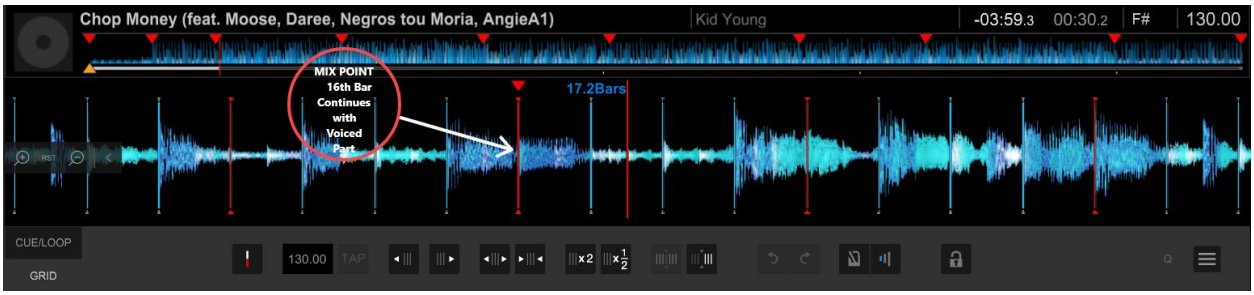
Σχήματα 3.2.1 και 3.2.2 - Δύο διαφορετικές επιλογές για το Cue Point στο ίδιο τραγούδι.

Το μέρος μετά το “Cue Point” της εικόνας-2 είναι άφωνο. Αυτό το καθιστά κατάλληλο “Cue Point” διότι υπάρχει χώρος για να μιξαριστεί με το επόμενο κομμάτι. Ένα άλλο “Cue Point” μπορεί να βρίσκεται ακριβώς πριν το έμφωνο μέρος του τραγουδιού. Κάνοντας αυτή την επιλογή, όμως, η μετάβαση θα πρέπει να γίνει με διαφορετικό τρόπο, πιο απότομο. Ο λόγος είναι ότι η μίξη έμφωνων ήχων σπάνια είναι αρμονική. Ένα απλό παράδειγμα είναι να βάλουμε δύο μουσικούς να τραγουδήσουν ταυτόχρονα διαφορετικά τραγούδια. Θα δημιουργηθεί σύγχυση στον ακροατή.

Stop-Intro Point: Εντοπίζεται 4, 8, 16 ή περισσότερα beats μετά το “Cue Point”. Η εισαγωγή ενός τραγουδιού μπορεί να ξεκινήσει χωρίς τύμπανα (drums). Σ’αυτή την περίπτωση ένα “Stop-Intro Point” μπορεί να εντοπιστεί τη χρονική στιγμή που ξεκινάνε τα drums. Επίσης, μετά το πέρας της εισαγωγής ξεκινάει ένα κουπλέ ή ένα ρεφρέν. Το σημείο διαχωρισμού των δυο ενοτήτων του τραγουδιού είναι το “Stop-Intro Point”.

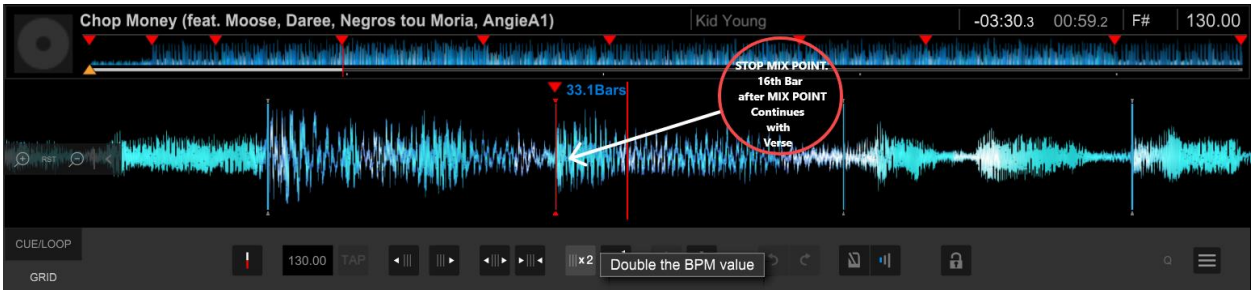
Σχήματα 3.2.3 και 3.2.4 - Δύο διαφορετικές επιλογές για το Stop-Intro Point στο ίδιο τραγούδι

Mix Point: Αυτό το σημείο εντοπίζεται στο πρώτο beat του ρεφρέν ή στο πρώτο beat της γέφυρας ή κάποιου άφωνου τμήματος του κομματιού.



Σχήμα 3.2.5 – Mix-Point

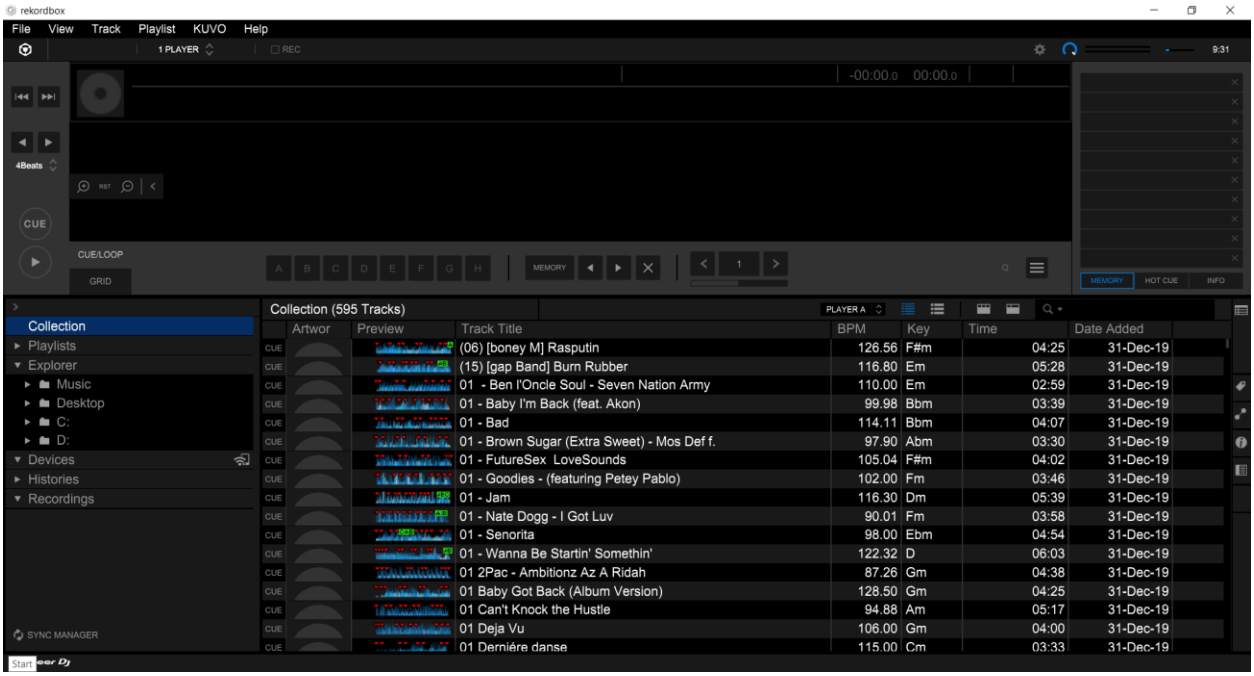
Stop Mix Point: Βρίσκεται 16 ή 32 μπάρες μετά το "Mix Point". Το σημείο αυτό δηλώνει το τέλος της αντίστοιχης ενότητας.



Σχήμα 3.2.6 – Stop-Mix Point

3.2.1 Σημεία Τύπου-1 (On Rekordbox P.O.I.)

Το Rekordbox είναι ένα λογισμικό που βοηθά τους δισκοθέτες να προετοιμάσουν το σετ τους πριν από μία εκδήλωση. Παρέχει μία φιλική-προς-το-χρήστη διεπαφή με βοηθητικές συντομεύσεις πληκτρολογίου και λειτουργίες ποντικιού που το καθιστά ιδανικό για τη δημιουργία του συνόλου δεδομένων.



Σχήμα 3.2.7 - Rekordbox interface

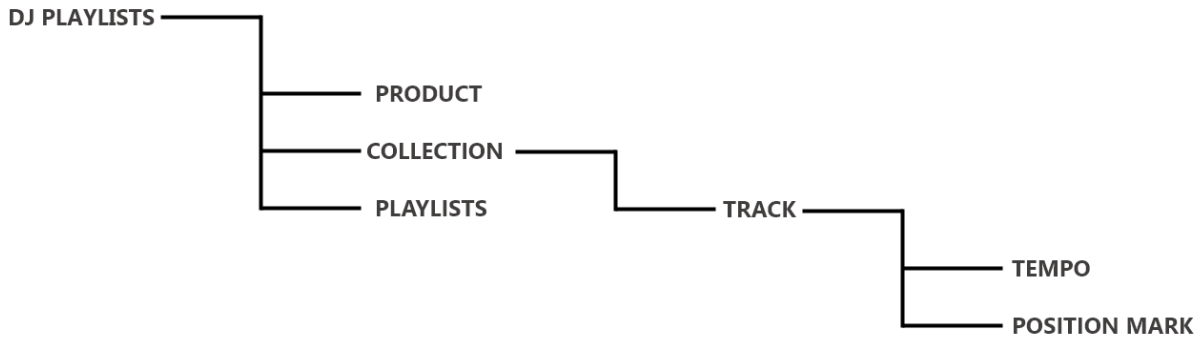
Η βασική ενέργεια του χρήστη είναι η εισαγωγή μουσικής σε οποιαδήποτε μορφή αρχείου, όπως MP3. Όταν γίνει αυτό το Rekordbox ανοίγει, αναλύει τα αρχεία μουσικής και διαβάζει τα meta-data τους. Μετά την ανάλυση, το Rekordbox αποθηκεύει στη βάση δεδομένων του καθιερωμένες πληροφορίες:

1. Όνομα αρχείου
2. Όνομα τραγουδιού
3. Όνομα καλλιτέχνη
4. Όνομα άλμπουμ
5. Κλειδί

6. BPM, beat τοποθετήσεις κλπ.

Τότε ο χρήστης μπορεί να αποθηκεύσει στη βάση δεδομένων σημεία στο χρόνο και με το πάτημα ενός κουμπιού μπορεί να μετακινηθεί σ'αυτά χωρίς να χρειάζεται να το εντοπίσει από την αρχή με το αυτί. Ο μέγιστος αριθμός σημείων ανά τραγούδι είναι 20. Οι δυνατότητες που προσφέρει είναι οι κατάλληλες για τη δημιουργία του data-set.

Το Rekordbox εξάγει τη βάση δεδομένων του σε αρχείο XML το οποίο βοηθά πολύ στην προεπεξεργασία. Τα fields του αρχείου οργανώνονται ως εξής:



Σχήμα 3.2.8 – Rekordbox XML δομή

Τα fields που θα χρησιμοποιηθούν είναι τα "TRACK" και "POSITION MARK".

"TRACK" Attributes:

a. Name
b. Artist
c. Genre
d. Size
e. TotalTime
f. Year
g. AverageBPM

h. BitRate
i. SampleRate
j. PlayCount
k. Location
l. Tonality

“POSITION MARK” Attributes:

a. Name
b. Type
c. Start
d. Num

Το πρώτο P.O.I. είναι το “Cue Point”. Έπειτα εμφανίζονται τα “Stop-Intro Point”, “Mix Point” και “Stop-Mix Point”. Το ζευγάρι “Cue Point”, “Stop-Intro Point” εμφανίζεται μία μόνο φορά σε αντίθεση με το άλλο ζευγάρι σημείων που εμφανίζεται πολλαπλές φορές.

$$t_{cue} < t\_stop_{intro} < t_{mix1} < t\_stop_{mix1} < t_{mix2} < t\_stop_{mix2} < \dots, (\#)$$

Αυτό είναι ο πρώτος τύπος σημείων ενδιαφέροντος. Το πλεονέκτημα αυτής της διαδικασίας είναι ότι δεν είναι απαραίτητο να γίνει σημείωση των σημείων με ταμπέλες (labels).

Τα κομμάτια που σημειώθηκαν είναι συνολικά 1000 περίπου. Με την αύξηση δεδομένων (data augmentation) φτάνουν τα 7000 σημειωμένα κομμάτια. Περισσότερες λεπτομέρειες αναγράφονται στο κεφάλαιο 8.

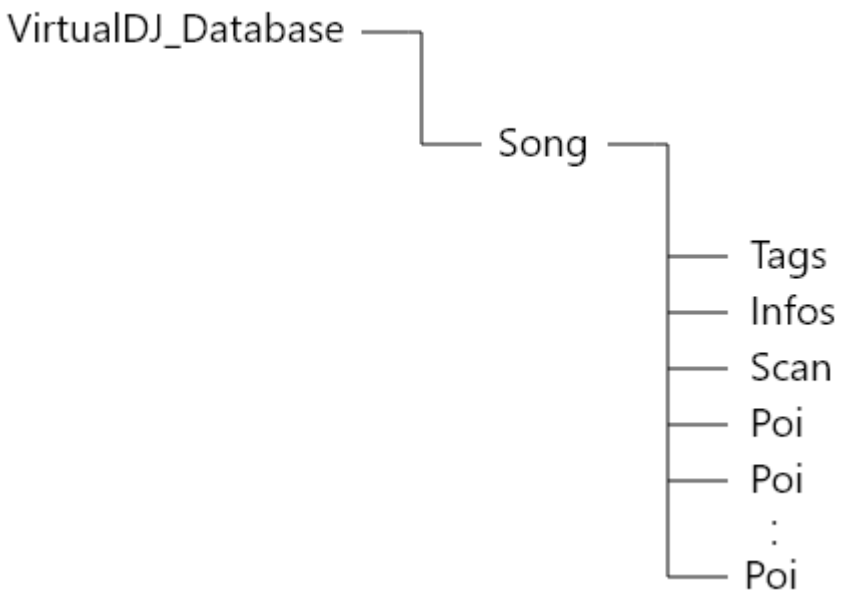
3.2.2 Σημεία Τύπου-2 (On Virtual DJ P.O.I.)

Το Virtual DJ προσφέρει ολόκληρο τον εξοπλισμό που χρειάζεται ένας deejay σε λογισμικό. Μπορεί να χρησιμοποιηθεί μονάχα με ένα ποντίκι και ένα πληκτρολόγιο. Επίσης παρέχει τη δυνατότητα ελέγχου κάθε κουμπιού, knob ή fader με ένα ελεγκτή deejay (deejay controller) μέσω MIDI. Λειτουργεί παρόμοια με το Rekordbox, όσον αφορά την ανάλυση των τραγουδιών και την πληροφορία που αποθηκεύεται στη βάση του.

Σχήμα 3.2.9 – Virtual DJ Interface

Ο κύριος λόγος που θα χρησιμοποιηθεί και το Virtual DJ είναι η απεριόριστη παροχή σημείωσης P.O.I. σε αντίθεση με το Rekordbox. Επίσης θα σημειωθούν κι άλλα σημεία εκτός από τα 4 βασικά που έχουν αναφερθεί, διότι θα χρησιμοποιηθούν σε διαφορετικό πείραμα.

Το Virtual DJ επίσης εξάγει τη βάση του σε XML:



Σχήμα 3.2.10 – Virtual DJ XML

Ακολουθώντας τη δομή του XML, θα χρησιμοποιηθούν εκείνα τα "Poi" των οποίων τα "attributes" περιέχουν:

1) Name
2) Pos (Τοποθεσία σε δευτερόλεπτα με ακρίβεια 5 δεκαδικών ψηφίων)
3) Type="cue"

Κάποια σημεία απ'αυτά θα ανήκουν στις 4 βασικές ομάδες σημείων ενδιαφέροντος. Ωστόσο, θα σημειωθούν και ενδιάμεσες χρονικές στιγμές. Συνήθως εντοπίζονται ανά 2-beats ξεκινώντας από ένα βασικής ομάδας.

Από αυτό το data-set θα γίνει μία αντιστοίχηση ενός τμήματος του κομματιού με ένα P.O.I., δηλαδή θα γίνει εκπαίδευση ενός νευρωνικού δικτύου το οποίο θα έχει είσοδο αυτό το τμήμα του κομματιού και έξοδο το P.O.I. που του αντιστοιχίζεται.

3.3 Μετατροπή Δεδομένων

Και στις δύο περιπτώσεις τα XML αρχεία θα μετατραπούν σε Comma-Separated-Values (CSV) αρχεία συμπεριλαμβάνοντας:

1) Όνομα αρχείου ήχου
2) Λίστα P.O.I. του αρχείου

Για παράδειγμα:

$(03)\ [boney\ M]\ Daddy\ Cool.wav, \|[0.378, 11.705, 57.9, 73.337, 111.917, 127.378]\|$

ΚΕΦΑΛΑΙΟ 4

MIDI

4.1 Εισαγωγή

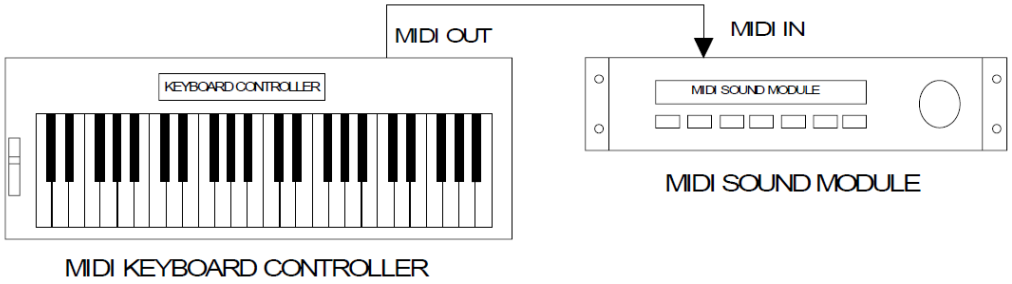
[10] Το M.I.D.I. (Musical Instrument Digital Interface) είναι ένα καθιερωμένο πρωτόκολλο τεχνολογίας που συνδέει προϊόντα από πολλές εταιρείες συμπεριλαμβάνοντας ψηφιακά όργανα μουσικής, υπολογιστές, tablets και smartphones. Χρησιμοποιείται καθημερινά σε παγκόσμιο επίπεδο από μουσικούς, deejays, παραγωγούς, εκπαιδευτές, καλλιτέχνες κλπ. για τη δημιουργία, αναπαραγωγή και εκμάθηση μουσικής και καλλιτεχνικών έργων.

[11] Το πρωτόκολλο MIDI παρέχει καθιερωμένα και αποδοτικά μέσα που μεταφέρουν μουσική πληροφορία ως ηλεκτρονικά δεδομένα. Η πληροφορία MIDI μεταδίδεται με χρήση των MIDI-μηνυμάτων, που μπορούν να θεωρηθούν ως εντολές που καθοδηγούν μία συμβατή συσκευή πως να παίξει ένα κομμάτι μουσικής. Η συσκευή που λαμβάνει MIDI δεδομένα πρέπει να δημιουργεί τους πραγματικούς ήχους. Το

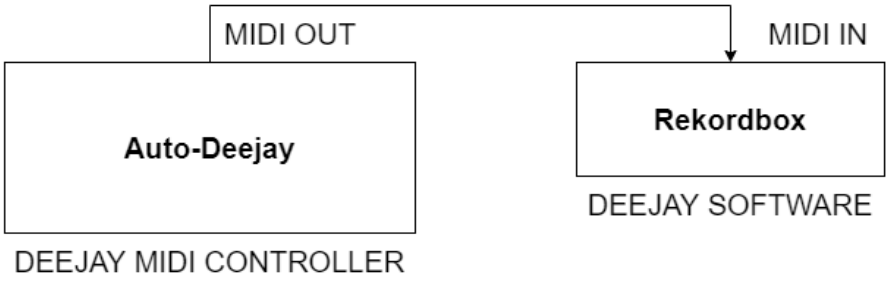
MIDI 1.0 Detailed Specification παρέχει μία ολοκληρωμένη περιγραφή του πρωτοκόλλου MIDI.

Το MIDI-stream δεδομένων είναι ένα ασύγχρονο μονοκατευθυντικό bit stream στα 31.25kbits ανά δευτερόλεπτο. Η διεπαφή MIDI σ'ένα όργανο MIDI θα περιέχει γενικά τρεις διαφορετικούς συνδέσμους, με την ένδειξη IN,OUT και THRU. Το MIDI stream προέρχεται από έναν ελεγκτή MIDI, όπως ένα μουσικό όργανο-πληκτρολόγιο, ή από έναν διαδοχέα MIDI. Ένας ελεγκτής MIDI είναι μία συσκευή που παίζεται όπως ένα όργανο και μεταφράζει την εκτέλεση σε MIDI stream δεδομένων σε πραγματικό χρόνο (καθώς παίζεται). Ένας διαδοχέας MIDI είναι μία συσκευή, η οποία επιτρέπει σε ακολουθίες MIDI δεδομένων να αποθηκεύονται, επεξεργάζονται, συνδυάζονται και να ξαναπαίζονται. Τα MIDI δεδομένα εξόδου ενός ελεγκτή ή ενός διαδοχέα μεταδίδονται μέσω των συνδέσμων MIDI OUT των συσκευών.

Ο παραλήπτης του MIDI stream δεδομένων είναι κοινώς μία MIDI γεννήτρια ήχου ή μία μονάδα ήχου, ή οποία λαμβάνει MIDI μηνύματα στον σύνδεσμο MIDI IN, και ανταποκρίνεται σ'αυτά τα μηνύματα παίζοντας ήχους. Η εικόνα 1 δείχνει ένα απλό σύστημα MIDI, που περιέχει MIDI ελεγκτή πληκτρολόγιο και μία MIDI μονάδα ήχου. Να σημειωθεί ότι πολλά MIDI όργανα πληκτρολόγια περιέχουν και τον ελεγκτή και τη μονάδα ήχου στην ίδια συσκευή. Σ'αυτές υπάρχει εσωτερική σύνδεση ανάμεσα σ'αυτά τα δύο που ενεργοποιείται ή απενεργοποιείται αναλόγως τη θέση του διακόπτη που παρέχεται.



Σχήμα 4.1.1. Ένα απλό σύστημα MIDI



Σχήμα 4.1.2. Προσαρμοσμένο σύστημα MIDI

4.2 ΜΗΝΥΜΑΤΑ MIDI

Ένα μήνυμα MIDI αποτελείται από μία ακολουθία των 8-bits ή αλλιώς status byte που γενικά ακολουθείται από ένα ή δύο bytes δεδομένων. Υπάρχουν διαφορετικοί τύποι μηνυμάτων MIDI. Στο υψηλότερο επίπεδο, τα μηνύματα διαχωρίζονται σε <<μηνύματα καναλιού>> και <<μηνύματα συστήματος>>. Τα μηνύματα καναλιού εφαρμόζονται σε συγκεκριμένο κανάλι, και ο αριθμός καναλιού περιέχεται στο status byte. Αντιθέτως στα μηνύματα καναλιού δεν καθορίζεται τέτοιος αριθμός στο status byte.

Τα μηνύματα καναλιού διαχωρίζονται σε <<μηνύματα φωνής καναλιού>> και σε <<μηνύματα Mode καναλιού>>. Τα πρώτα μεταφέρουν μουσικά δεδομένα εκτέλεσης και αυτά περιλαμβάνουν την πλειοψηφία της συνολικής κίνησης μηνυμάτων σ'ένα τυπικό stream δεδομένων MIDI. Τα μηνύματα Mode επηρεάζουν τον τρόπο ο παραλήπτης-όργανο θα ανταποκριθεί στα μηνύματα φωνής.

Τα μηνύματα φωνής χρησιμοποιούνται για την αποστολή πληροφορίας εκτέλεσης. Τα μηνύματα σ'αυτή την κατηγορία είναι τα Note On, Note Off, Polyphonic Key Pressure, Channel Pressure, Pitch Bend Change, Program Change και Control Change.

4.3 ΧΡΗΣΗ MIDI ΜΗΝΥΜΑΤΩΝ

Σ'αυτή την εργασία τα MIDI μηνύματα θα αντικαταστήσουν τα χέρια του deejay. Η εκτέλεση ενός deejay set απαιτεί το πάτημα κουμπιών (εργαλείο τύπου-1) και την μετακίνηση της θέση ολισθητών ευθύγραμμης (εργαλείο τύπου-2) και στροφίκης κίνησης (εργαλείο τύπου-3). Αυτά τα εργαλεία μπορούν να ελεγχθούν από μηνύματα MIDI.

Τα εργαλεία τύπου-1 έχουν δύο πιθανές καταστάσεις . Πατημένο κουμπί και μη-πατημένο κουμπί. Τα κατάλληλα μηνύματα φωνής γι'αυτά είναι επομένως τα Note On και Note Off. Τα εργαλεία τύπου-2 έχουν κατάσταση μέσα σ'ένα εύρος συνεχών τιμών κατάστασης όπως και τα εργαλεία τύπου-3. Γι'αυτά, τα κατάλληλα μηνύματα φωνής είναι τα Control Change.

ΚΕΦΑΛΑΙΟ 5

ΜΟΝΤΕΛΑ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ

5.1 Convolutional Neural Networks

[12] Ένα Convolutional Neural Network (CNN) ή συνελικτικό νευρωνικό δίκτυο είναι μία πολύ γνωστή αρχιτεκτονική εμπνευσμένη από τη φυσικό οπτικό μηχανισμό των ζωντανών οργανισμών. Υπάρχουν διάφορες παραλλαγές των CNN αρχιτεκτονικών στη βιβλιογραφία. Ωστόσο, τα βασικά εξαρτήματα, τα οποία είναι στρώματα convolutional, pooling και fully-connected, είναι κοινά για όλες. Το convolutional στρώμα στοχεύει στη μάθηση χαρακτηριστικών που αντιπροσωπεύουν την είσοδο. Ένα convolutional στρώμα αποτελείται από πολλούς convolutional πυρήνες (kernels) που χρησιμοποιούνται για τον υπολογισμό διάφορων απεικονίσεων χαρακτηριστικών. Πιο συγκεκριμένα, κάθε νευρώνας μίας απεικόνισης χαρακτηριστικών συνδέεται με μία περιοχή γειτονικών νευρώνων στο προηγούμενο στρώμα. Η νέα απεικόνιση χαρακτηριστικών μπορεί να ληφθεί πρώτα συνελίσοντας (convolution) την είσοδο με ένα πυρήνα και έπειτα εφαρμόζοντας μία μη-γραμμική συνάρτηση ενεργοποίησης σε κάθε στοιχείο των

χαρακτηριστικών. Για να παράγει κάθε απεικόνιση χαρακτηριστικών, ο πυρήνας μοιράζεται από όλες τις χωρικές περιοχές της εισόδου. Οι ολοκληρωμένες απεικονίσεις λαμβάνονται χρησιμοποιώντας διάφορους πυρήνες. Η συνάρτηση ενεργοποίησης εισαγάγει μη-γραμμικότητες στο CNN, οι οποίες είναι επιθυμητές για δίκτυο πολλών στρωμάτων για τον εντοπισμό μη-γραμμικοτήτων. Τυπικές συναρτήσεις ενεργοποίησης είναι οι sigmoid, tanh και παραλλαγές της ReLU (Rectified Linear Unit). Το pooling-στρώμα στοχεύει στη αναλλοίωση μεταβλητότητας (shift-invariance) μειώνοντας την ανάλυση των απεικονίσεων. Τοποθετείται συνήθως ανάμεσα σε δύο convolution blocks. Κάθε convolution block αποτελείται από μία σειρά convolution-στρωμάτων με συναρτήσεις ενεργοποίησης. Κάθε απεικόνιση χαρακτηριστικών ενός pooling-στρώματος συνδέεται με την είσοδο ενός convolution block ή άλλου είδους νευρωνικού δικτύου. Τυπικές pooling-διαδικασίες αποτελούν οι average pooling και max pooling.

Συστατικά των C.N.N.s - [13]

Convolutional Στρώμα

Το convolutional-στρώμα είναι το βασικό συστατικό ενός CNN. Οι παράμετροι αποτελούνται από ένα σύνολο <<μαθητευόμενων>> φίλτρων (ή πυρήνων), που έχουν ένα μικρό δεκτικό πεδίο (receptive field), που όμως επεκτείνονται σε όλο το βάθος του όγκου της εισόδου (input volume). Κατά το forward pass, κάθε φίλτρο συνελίσεται σε όλο το πλάτος και ύψος της εισόδου, υπολογίζοντας το εσωτερικό γινόμενο ανάμεσα στο φίλτρο και την είσοδο, παράγοντας έτσι μία διδιάστατη απεικόνιση αυτού του φίλτρου. Ως αποτέλεσμα, το δίκτυο μαθαίνει φίλτρα που ενεργοποιούνται όταν εντοπίζει συγκεκριμένο τύπο χαρακτηριστικών σε κάποια θέση στο χώρο της εισόδου.

Βάζοντας τις απεικονίσεις των ενεργοποιήσεων σε όλα τα φίλτρα σε κάθε διάσταση, σχηματίζεται η ολοκληρωμένη έξοδος του convolution-στρώματος. Κάθε στοιχείο της εξόδου μπορεί έτσι να μεταφραστεί και ως η έξοδος ενός νευρώνα που μοιάζει με μία μικρή περιοχή της εισόδου που μοιράζεται τις παραμέτρους του με άλλους νευρώνες στην ίδια ενεργοποίηση.

Όταν αντιμετωπίζονται είσοδοι υψηλών διαστάσεων όπως οι εικόνες, δεν είναι πρακτική η σύνδεση κάθε νευρώνα με όλους του υπόλοιπους (fully-connected) του προηγούμενου στρώματος διότι μία τέτοια αρχιτεκτονική δικτύου δεν λαμβάνει υπόψη τη χωρική δομή των δεδομένων. Τα convolutional δίκτυα εκμεταλλεύονται την τοπική συσχέτιση αναγκάζοντας ένα αραιό τοπικό μοτίβο συνδεσιμότητας ανάμεσα σε νευρώνες γειτονικών στρωμάτων: κάθε νευρώνας συνδέεται μόνο με μία μικρή περιοχή της εισόδου.

Η έκταση της συνδεσιμότητας είναι υπερπαράμετρος που λέγεται δεκτικό πεδίο του νευρώνα. Οι συνδέσεις είναι τοπικές στο χώρο (κατά μήκος του πλάτους και του ύψους), αλλά εκτείνονται κατά μήκος ολόκληρους του βάθους της εισόδου. Μία τέτοια αρχιτεκτονική εξασφαλίζει ότι τα μαθημένα φίλτρα παράγουν την καλύτερη απόκρισης σ’ένα χωρικά τοπικό μοτίβο εισόδου (spatially local input pattern).

Pooling Στρώμα

Μία άλλη σημαντική έννοια των CNNs είναι το pooling. Πρόκειται για μία μορφή μη-γραμμικής υποδειγματοληψίας. Υπάρχουν δίαφορες μη-γραμμικές συναρτήσεις που υλοποιούν το pooling, από οποίες η max-pooling είναι η πιο κοινή. Χωρίζει την εικόνα εισόδου σε ένα σύνολο μή-γραμμικών επικαλυπτόμενων ορθογώνιων και, για κάθε τέτοια υπο-περιοχή, βγάζεις ως έξοδο τη μέγιστη τιμή.

Διαισθητικά, η ακριβής τοποθεσία ενός χαρακτηριστικού είναι λιγότερο σημαντική από τη σχετική της θέση με άλλα χαρακτηριστικά. Αυτή είναι η ιδέα πίσω από το pooling σε ένα convolutional νευρωνικό δίκτυο. Το pooling-στρώμα εξυπηρετεί στο να μειώνει προοδευτικά το χωρικό μέγεθος των αναπαραστάσεων. Ο λόγος είναι η μείωση του αριθμού των παραμέτρων, της μνήμης και της χρονικής πολυπλοκότητας στο δίκτυο, και με σκοπό τον έλεγχο του overfitting (υπερεκπαίδευση). Είναι κοινή η περιοδική εισαγωγή ενός pooling-στρώματος ανάμεσα σε διαδοχικά convolutional-στρώματα σε μία αρχιτεκτονική CNN. [*] Η διαδικασία pooling παρέχει μία άλλη μορφή του αναλλοίωτου μετάφρασης.

ReLU Στρώμα

Η ReLU ή αλλιώς Rectified Linear Unit εφαρμόζει τη συνάρτηση ενεργοποίησης $f(x) = max(0, x)$. Αφαιρεί αποτελεσματικά τις αρνητικές τιμές από μία απεικόνιση ενεργοποίησης ορίζοντας τους την τιμή μηδέν. Αυξάνει τις μη-γραμμικές ιδιότητες της συνάρτησης απόφασης και του ολικού δικτύου χωρίς να επηρεάζει τα δεκτικά πεδία ενός convolutional-στρώματος.

Κι άλλες συναρτήσεις χρησιμοποιούνται για την αύξηση της μη-γραμμικότητας, όπως η υπερβολική εφαπτομένη (tanh) και η sigmoid. Η ReLU συχνά προτιμάται από άλλες συναρτήσεις διότι επιτρέπει στο νευρωνικό δίκτυο να εκπαιδευτεί αρκετές φορές πιο γρήγορα χωρίς να εισαγάγει σημαντικά σφάλματα στην ακρίβεια γενίκευσης.

LeakyReLU Στρώμα

Η LeakyReLU σε αντίθεση με την απλή ReLU είναι μία συνάρτηση που κρατάει μικρό κομμάτι αρνητικών τιμών $f(x) = max(a * x, x)$, με το a να είναι αντιπροσωπεύει μικρή θετική κλίση. Ενδεικτικές τιμές είναι: 0.01, 0.03 κλπ.

Fully Connected Στρώμα

Τελικά, μετά από αρκετά convolutional και pooling στρώματα, η λογική υψηλού επιπέδου στο νευρωνικό δίκτυο αναπτύσσεται μέσω των fully connected στρωμάτων. Νευρώνες σ'ένα τέτοιο στρώμα έχουν συνδέσεις με όλες τις ενεργοποιήσεις του προηγούμενου δικτύου, όπως φαίνεται σε κανονικά τεχνητά νευρωνικά δίκτυα. Οι ενεργοποιήσεις τους μπορούν έτσι να υπολογιστούν σαν ένα παράλληλο μετασχηματισμό, με πολλαπλασιασμό πινάκων που ακολουθείται από μία πόλωση (bias)

Loss Στρώμα ή Στρώμα Απώλειας

Το στρώμα απώλειας καθορίζει πως η εκπαίδευση <<τιμωρεί>> την απόκλιση ανάμεσα στην προβλεπόμενη και την πραγματική έξοδο του συστήματος. Συνήθως είναι το τελευταίο στρώμα ενός νευρωνικού δικτύου. Μπορούν να χρησιμοποιηθούν διάφορες συναρτήσεις απώλειας ανάλογα το έργο προς υλοποίηση.

Η απώλεια categorical cross-entropy σε συνδυασμό με softmax ως τελευταία συνάρτηση ενεργοποίησης χρησιμοποιείται για την πρόβλεψη μίας κλάσης ανάμεσα σε k αμοιβαίως αποκλειστικές κλάσεις. Η απώλεια binary cross-entropy sigmoid χρησιμοποιείται σε συνδυασμό με sigmoid για την πρόβλεψη k ανεξάρτητων τιμών πιθανότητας στο διάστημα $[0, 1]$. Η ευκλείδεια απώλεια Mean Squared Error (MSE) ή Μέσο Τετραγωνικό Σφάλμα χρησιμοποιείται με τη γραμμική συνάρτηση ενεργοποίησης και έχει σκοπό την πρόβλεψη μίας τιμής στο συνεχές διάστημα $(-\infty, +\infty)$

5.2 Recurrent Neural Networks

[14] Ένα Recurrent Neural Network (RNN) είναι νευρωνικό δίκτυο που εξομοιώνει ένα δυναμικό σύστημα διακριτού χρόνου που έχει μία είσοδο, μία έξοδο και μία κρυμμένη κατάσταση (hidden state). Είναι μία από τις κύριες λύσεις σε προβλήματα αλληλουχίας. Τα RNN ονομάζονται recurrent διότι εκτελούν το ίδιο έργο για κάθε στοιχείο της αλληλουχίας, με την έξοδο να εξαρτάται από τους προηγούμενους υπολογισμούς. Έχουν μνήμη που συλλαμβάνει πληροφορία που άφορα τι έχει υπολογιστεί έως εδώ και επιτρέπουν την λειτουργιά με αλληλουχία διανυσμάτων. Η εκπαίδευση είναι διαφορετική σε σχέση με τα Feed-Forward νευρωνικά δίκτυα. Η πιο κοινή μέθοδος εκπαίδευσης RNN είναι Backpropagation Through Time (BPTT) και Real-Time Recurrent Learning (RTRL), με την πρώτη να είναι η πιο κοινή. Η κυρία διάφορα στις δυο μεθόδους είναι ο τρόπος που υπολογίζονται οι αλλαγές των βαρών. Η αρχική διατύπωση των Long Short Term Memory (LSTM) RNNs χρησιμοποιούσε συνδυασμό BPTT και RTRL. [15] Με τις συμβατικές μεθόδους BPTT ή RTRL τα σήματα σφάλματος που <<ρέουν πίσω στο χρόνο>> τείνουν να σκάνε ή να εξαφανίζονται. Η χρονική εξέλιξη του backpropagated σφάλματος έχει εκθετική εξάρτηση με το μέγεθος των βαρών. Η περίπτωση που σκάει μπορεί να οδηγήσει σε ταλαντευόμενα βάρη, ενώ η άλλη περίπτωση οπού υπάρχει vanishing learning το αποτέλεσμα είναι η μικρή ή καθόλου πρόοδος. Για την αντιμετώπιση αυτού του προβλήματος δημιουργήθηκαν τα LSTM RNNs και οι απλοτερη τους εκδοσή Gated Recurrent Unit (GRU) RNNs.

[16] Μία κοινή LSTM μονάδα αποτελείται από ένα πυρήνα (cell), μια πύλη εισόδου (input gate), μια πύλη εξόδου (output gate) και μια πύλη forget (forget gate). Ο πυρήνας θυμάται τις τιμές καθώς περνάνε αυθαίρετα χρονικά διάστημα και οι τρεις πύλες ρυθμίζουν τη ροή πληροφορίας προς τα μέσα και προς τα έξω του πυρήνα.

[17] Το GRU είναι σαν το LSTM αλλά με λιγότερες παραμέτρους καθώς δεν περιλαμβάνει πύλη εξόδου. Η απόδοση του GRU σε συγκεκριμένα έργα όπως μοντελοποίηση πολυφωνικής μουσικής και μοντελοποίηση σημάτων φωνής αποδείχτηκε ότι είναι παρόμοια με αυτής των LSTM.

Συστατικά των R.N.N.s

Fully Recurrent

Βασικά RNNs είναι δίκτυα νευροειδών κόμβων οργανομένων σε συνεχή στρώματα. Κάθε κόμβος σε δεδομένο στρώμα συνδέεται με μία μονόπλευρη κατευθυνόμενη σύνδεση προς κάθε κόμβο στο επόμενο στρώμα. Κάθε κόμβος-νευρώνας έχει μία χρονικά-μεταβλητή ενεργοποίηση. Οι κόμβοι είναι είτε κόμβοι εισόδου (δέχονται δεδομένα έξω από το δίκτυο), κόμβοι εξόδου (δίνοντας αποτελέσματα) ή κρυμμένοι κόμβοι (που τροποποιούν δεδομένα καθ'οδόν από την είσοδο προς την έξοδο).

Για επιτηρούμενη μάθηση σε ρυθμίσεις διακριτού χρόνου, σειρές από διανύσματα με πραγματικές τιμές φτάνουν στους κόμβους εισόδου, με ένα διάνυσμα κάθε φορά. Σε οποιοδήποτε χρονικό βήμα, κάθε μονάδα που δεν είναι είσοδος υπολογίζει την τρέχουσα ενεργοποίηση ως μια μη-γραμμική συνάρτηση του επιβαρυμένου αθροίσματος (weighted sum) των ενεργοποιήσεων όλων των μονάδων που συνδέονται με αυτή.

Κάθε σειρά παράγει ένα σφάλμα ως άθροισμα των αποκλίσεων όλων των σημάτων-στόχου (target signals) από τις αντίστοιχες ενεργοποιήσεις που υπολογίζονται από το δίκτυο. Για ένα σύνολο εκπαίδευσης με πολυάριθμες σειρές, το συνολικό σφάλμα είναι το άθροισμα των σφαλμάτων κάθε ξεχωριστής σειράς.

Long Short-Term Memory

Είναι ένα σύστημα deep learning που αποφεύγει το vanishing gradient πρόβλημα. Τα LSTM κανονικά ενισχύεται από recurrent πύλες που ονομάζονται "forget gates". Τα LSTM αποτρέπουν τα backpropagated σφάλματα από το να εξαφανίζονται ή να εκρήγνυνται. Αντί αυτού, τα σφάλματα ρέων πίσω μέσω απεριόριστων αριθμών εικονικών στρωμάτων που ξεδιπλώνονται στο χώρο. Δηλαδή, τα LSTM μπορούν να μάθουν έργα που απαιτούν μνήμη γεγονότων που συνέβησαν πριν από χιλιάδες ή εκατομμύρια διακριτά χρονικά βήματα. Μπορούν να δημιουργηθούν τοπολογίες τύπου LSTM προσαρμοσμένες στο πρόβλημα. Τα

LSTM λειτουργούν ακόμα και με μεγάλες καθυστερήσεις μεταξύ σημαντικών γεγονότων και μπορούν να χειριστούν σήματα που αναμιγνύουν στοιχεία χαμηλών και υψηλών συχνοτήτων.

5.3 Principal Component Analysis

[18] Η ιδέα πίσω από τη μέθοδο Principal Component Analysis ή Ανάλυση Κυρίων Συνιστωσών είναι να βρεθεί ένα μικρός αριθμός γραμμικών συνδυασμών συσχετισμένων παραμέτρων που περιγράφουν την μεταβλητότητα στο data-set με ένα μικρό αριθμό νέων ασυσχέτιστων παραμέτρων. Η PCA μετασχηματίζει τα δεδομένα σε ένα νέο σύστημα συντεταγμένων, όπου η μεγαλύτερη διακύμανση από οποιαδήποτε προβολή των δεδομένων εντοπίζεται κατά μήκος της πρώτης συντεταγμένης (πρώτο principal component), η δεύτερη μεγαλύτερη διακύμανση κατά μήκος της δεύτερης συντεταγμένης κ.ο.κ. Υπάρχουν τόσα principal components όσα και οι παράμετροι, αλλά τυπικά μόνο οι πρώτες χρειάζονται για την περιγραφή της συνολικής μεταβλητότητας.

Σε εφαμογές όπως συμπίεση εικόνας ή αναγνώριση προσώπου χρησιμοποιείται η μέθοδος PCA και είναι διαδεδομένη για τον προσδιορισμό μοτίβων σε δεδομένα πολλών διαστάσεων. Η προσέγγιση PCA χρησιμοποιείται για τη μείωση των διαστάσεων των δεδομένων, μέσω της συμπίεσης δεδομένων και αποκαλύπτει την πιο αποδοτική δομή χαμηλών διαστάσεων μοτίβων προσώπου. Το πλεονέκτημα της μείωσης διαστάσεων είναι ότι αφαιρεί την άχρηστη πληροφορία και συγκεκριμένα αποσυνθέτει τη δομή του προσώπου σε συστατικά που είναι ασυσχέτιστα και είναι γνωστά ως ιδιο-πρόσωπα (Eigen faces). Σε αυτή τη δημοσίευση [19], δηλώνεται ότι το μεγάλο όφελος της μεθόδου είναι ότι μπορεί να μειώσει τα δεδομένα για την αναγνώριση της οντότητας κατά 1000 φορές των υπαρχόντων δεδομένων. Μετά την εξαγωγή χαρακτηριστικών, το επόμενο βήμα είναι η ταξινόμηση.

Η εργασία χρησιμοποιεί ένα μοντέλο Incremental-PCA σε Constant-Q (CQT) υπο-μετασχηματισμούς. Για να επιτευχθεί αυτό, αρχικά ο CQT υπο-μετασχηματισμός ενός κομματιού που έχει σχήμα:

$$(n_{time}, n_{feats})$$

θα μετατραπεί σε διάνυσμα μεγέθους:

$$(n_{time} * n_{feats})$$

Το σύνολο των υπο-μετασχηματισμών μετά την μετατροπή θα δώσει ένα πίνακα σχήματος:

$$(n_{subtransforms}, n_{time} * n_{feats})$$

Για:

- $n_{feats} = 108, n_{time} = 100 => (n_{subtransforms}, 10800)$

Ο αλγόριθμος Incremental-PCA είναι αποδοτικός, λόγω της μερικής-βηματικής προσαρμογής του μοντέλου με mini-batches. Η έξοδος του μοντέλου θα είναι ένας πίνακας με σχήμα:

$$(batch_{size}, n_{components})$$

Ο αριθμός των συνιστωσών (components) επιλέγεται να έχει τιμή 20 φορές μικρότερη της τιμής $n_{time} * n_{feats}$. Δηλαδή:

- $n_{components} = 10800 / 20 = 504$

ΚΕΦΑΛΑΙΟ 6

ΣΥΣΤΗΜΑ ΚΑΙ ΑΡΧΙΤΕΚΤΟΝΙΚΗ

6.1 Εισαγωγή

Όπως αναφέρθηκε στο Κεφάλαιο 2, είναι τέτοιο το είδος των δεδομένων που έχει ως αποτέλεσμα να δημιουργούν μεγάλη χωρητική πολυπλοκότητα. Ένα τραγούδι τριών λεπτών με ρυθμό δειγματοληψίας 44100Hz ⇔ 44100 $\delta\epsilon\acute{\iota}\gamma\mu\alpha\tau\alpha \,/\, second$ που αποθηκεύεται σε μία μεταβλητή ενός προγράμματος έχει μέγεθος (7938000, 1). Αυτό είναι ένα διάνυσμα πολύ μεγάλου μέτρου ή σειρά χρόνου, δύσκολου στη διαχείριση του. Μετασχηματίζοντας το διάνυσμα σε spectrogram με ένα παράθυρο των 10ms και με ανάλυση $nfft = 512$ δείγματα τα νέα δεδομένα θα έχουν σχήμα (18000, 512) με τη πρώτη διάσταση να αντιπροσωπεύει τον χρόνο και η δεύτερη τη συχνότητα. Ως έχει το

αποτέλεσμα συνεχίζει να έχει μεγάλο μέγεθος. Η βασική ιδέα που εφαρμόζεται είναι η παρακάτω:

Έστω πίνακας $B(n_{time}, n_{freq})$ που αντιπροσωπεύει τον μετασχηματισμό ενός τραγουδιού. Το σύστημα της εργασίας θα δέχεται εισόδους ενός υποπίνακα του B, $B_{tf} = B(t, f)$, με t και f να είναι εύρη τιμών ή μίας σειράς υποπινάκων $[B_{0f}, B_{1f}, .., B_{\tau f}]$.

Για παράδειγμα ένα υπο-spectrogram του αρχικού θα έχει σχήμα $(500, 512)$, ή μία σειρά από υπο-spectrograms θα έχει σχήμα $(36, 500, 512)$.

Το πλεονέκτημα αυτής της μεθοδολογίας είναι το μοίρασμα του φορτίου των δεδομένων. Θα μπορούσε ένα τραγούδι να μεταφραστεί σε μία σειρά που αποτελείται από 36 χρονικά βήματα. Το καθένα απ'αυτά διαρκεί 5 δευτερόλεπτα. Το κάθε χρονικό βήμα μπορεί να δεχτεί μία ατομική επεξεργασία κι έπειτα μία συνολική όλων μειώνοντας έτσι κατά πολύ την πολυπλοκότητα.

6.2 ΠΕΡΙΓΡΑΦΗ ΣΥΣΤΗΜΑΤΟΣ

Ο στόχος του συστήματος είναι να παίξει μία σειρά τραγουδιών το ένα μετά το άλλο με τον καλύτερο δυνατό τρόπο ως deejay. Δεδομένης play-list ως είσοδο, η παραγόμενη έξοδος είναι ένα live-mix. Το σύστημα αποτελείται από δύο υποσυστήματα. Το πρώτο δέχεται την play-list και προβλέπει σημεία ενδιαφέροντος σε κατάλληλη μορφή αρχείου - τύπου Comma Separated Values (CSV). Το δεύτερο υποσύστημα λαμβάνει αυτό το αρχείο και παράγει παραμετροποιημένες <<κινήσεις deejay>> που μεταφράζονται σε MIDI messages, για τη δημιουργία του τελικού mix. Τα MIDI messages με τη σειρά τους στέλνονται σ'ένα deejay software τα οποία το ελέγχουν.

Το πλεονέκτημα αυτής της δομής βρίσκεται στη χρήση πολλών deejay software, που δέχονται κατάλληλα προσαρμοσμένα MIDI μηνύματα. Υπάρχει μία κοινή deejay-ορολογία και κάθε software εφαρμόζει αυτή.

Το μειονέκτημα της μεθόδου βρίσκεται στην περιορισμένη πρόσβαση των δεδομένων-αρχείων ήχου, δια μέσου του deejay software. Πρόσβαση στα δεδομένα δίνει τη δυνατότητα διαχείρισης τους με μεγαλύτερη ακρίβεια. Το παρακάτω παράδειγμα διευκρινίζει αυτό το μειονέκτημα:

Έστω τραγούδι φορτωμένο (loaded) με σκοπό την αναπαραγωγή του. Το τραγούδι κατέχει ένα δείκτη ο οποίος δείχνει το τρέχων σημείο στο χρόνο. Όταν φορτώνεται ένα τραγούδι ο δείκτης δείχνει στην αρχή του, έστω στη θέση 0. Ο deejay θα πρέπει να μετακινήσει το δείκτη αυτό σε άλλη θέση πριν πατήσει το κουμπί αναπαραγωγής τραγουδιού. Οπότε έχοντας πρόσβαση στα δεδομένα ο δείκτης μπορεί να μετακινηθεί με ακρίβεια στο σημείο ενδιαφέροντος. Ο δεύτερος τρόπος, με MIDI messages δηλαδή, δίνει μια εντολή όπως <<κράτα πατημένο το

κουμπί fast forward για τ χρονικό διάστημα>>. Ο δείκτης τότε θα μετακινηθεί κατά $\tau + \varepsilon$, με $\varepsilon \ll 1$, εισάγοντας ένα μικρό σφάλμα.

6.3 Υποσύστημα Α

Το πρώτο υποσύστημα περιλαμβάνει δύο επιμέρους υποσυστήματα σε σειρά, που ονομάζονται “Relative Locator” ή “Locator” και “Preciser”. Και τα δύο είναι νευρωνικά δίκτυα, των οποίων η συνδυαστική λειτουργία, αναδίδει Σημεία Ενδιαφέροντος χρησιμοποιώντας Recurrent Neural Networks (RNN) και Convolutional Neural Networks. Το “Relative Locator” προβλέπει σε ποιό χρονικό βήμα εντοπίζεται ένα σημείο ενδιαφέροντος. Το χρονικό βήμα διαρκεί κάποιο χρονικό διάστημα. Ουσιαστικά σκοπός του δικτύου είναι ο εντοπισμός ενός πλαισίου ενός μετασχηματισμού. Το “Preciser θα δεχτεί αυτό το πλαίσιο και θα υπολογίσει το σημείο ενδιαφέροντος εντός αυτού με ακρίβεια.

Αντί να δημιουργηθεί ένα σύστημα που προβλέπει σημεία ενδιαφέροντος άμεσα, το μεγάλο φορτίο δεδομένων μοιράζεται διαχωρίζοντας τη λύση σε υπο-έργα. Ο χρόνος είναι η κύρια μεταβλητή. Το είδος των δεδομένων απαιτεί τη χρήση RNN και CNN τα οποία προϋποθέτουν πολυπλοκότητα. Επομένων, αυτή η προσέγγιση βοηθά στην εκπαίδευση των νευρωνικών πιο απλά.

6.3.1. PRECISER

Ένα κομμάτι κατέχει ένα συγκεκριμένο αριθμό σημείων ενδιαφέροντος σχετικά με αυτή την εργασία. Στο “Virtual DJ” τα σημεία ενδιαφέροντος μαρκάρονται μαζί με πολλά άλλα για να υπάρχει μεγαλύτερο data-set. Μία ενότητα ενός τραγουδιού ξεκινάει σ’ένα σημείο $t_{section}$. Τα επιπλέον σημεία εντοπίζονται με την εξίσωση (A) έως το επόμενο σημείο ενδιαφέροντος ή την επόμενη ενότητα του τραγουδιού.

- $X_{tra} = \{t_{section} + d + e_1, t_{section} + 2 * d + e_2, .., t_{section} + k * d + e_k\}$ (A)

where:

- $d = (BPM/60) * 8 \Leftrightarrow 2\ bars$
- $-20\ ms \leq e_i \leq 20\ ms\ \forall i.$

Ο συνολικός αριθμός σημείων ενδιαφέροντος που μαρκάρονται στο “Virtual DJ” υπολογίζονται κοντά στα 28.000. Αυτά θα χρησιμοποιηθούν για την εκπαίδευση ενός CNN. Ο λόγος που χρησιμοποιείται CNN είναι οι δύο διαστάσεις ενός μετασχηματισμού. Ως αποτέλεσμα, μπορεί να θεωρηθεί ως εικόνα, και οι περισσότερες εφαρμογές των CNN γίνονται σε εικόνες.

6.3.2 RELATIVE LOCATOR version 1

Ένας μετασχηματισμός είναι μία συνάρτηση του χρόνου και της συχνότητας. Αφού ο χρόνος είναι μία μεταβλητή, το spectrogram είναι μία σειρά από χαρακτηριστικά (FFT). Η πρώτη σκέψη είναι η χρήση RNN, με κάθε χρονικό-βήμα να είναι μία στιγμή στο χρόνο του spectrogram. Ωστόσο, ένα τραγούδι τριών λεπτών με παράθυρο των 10ms έχει 18.000 χρονικά βήματα. Ένα Recurrent Cell ξεδιπλωμένο (unfolded) 18.000 φορές εισαγάγει τεράστια πολυπλοκότητα. Με κάποιο τρόπο πρέπει αυτή η σειρά να μειώσει το μήκος της.

Μερικές λύσεις σ'αυτό το πρόβλημα περιλαμβάνουν τη χρήση μεγαλύτερων παραθύρων σε διάρκεια, Truncated Backpropagation Through Time ή τη μείωση των χρονικών βημάτων. Με ένα παράθυρο των 20ms τα χρονικά βήματα πέφτουν στα 9.000 κ.ο.κ. Το μειονέκτημα της χρήσης μακρύτερου παραθύρου είναι υποχρεωτική χρήση περισσότερων δειγμάτων συχνότητας.

Η μείωση του αριθμού των χρονικών βημάτων επιτυγχάνεται με το <<κόψιμο>> ενός μετασχηματισμού σε n-υπο-μετασχηματισμούς.

$$transform = [subtransform_1 | .. | subtransform_n]$$

Αντί να θεωρηθούν οι μετασχηματισμοί ως σειρές διανυσμάτων μήκους $nfft$, θα αντιμετωπίζονται ως σειρές υπο-μετασχηματισμών. Έτσι μειώνονται τα συνολικά χρονικά βήματα, αυξάνοντας τη διάρκεια του καθενός. Ωστόσο, όλα τα Recurrent Units δέχονται ως είσοδο μία χρονική σειρά διανυσμάτων και όχι διδιάστατων πινάκων. Δημιουργείται λοιπόν η ανάγκη για την μείωση των διαστάσεων αυτών των πινάκων.

Τώρα είναι η κατάλληλη στιγμή για να ερωτηθεί ο λόγος που περιγράφηκε πρώτα το υποσύστημα "Preciser". Η απάντηση είναι ότι θα

χρησιμοποιηθεί ως «εξαγωγέας» χαρακτηριστικών (feature extractor). Κάθε υπο-μετασχηματισμός τροφοδοτεί ένα εκπαιδευμένο CNN με σκοπό την εξαγωγή χαρακτηριστικών. Με αυτόν τον τρόπο είναι εφικτή η απεικόνιση ενός δι-διάστατου πίνακα σε ένα διάνυσμα χαρακτηριστικών του.

Ένας διαφορετικός εξαγωγέας χαρακτηριστικών είναι μοντέλο “Principal Component Analysis” (PCA) που δείχνει εξαιρετικά αποτελέσματα σε εφαρμογές όπως αναγνώριση προσώπου. Το κύριο χαρακτηριστικό ενός τέτοιου μοντέλου είναι η μείωση διαστάσεων, το οποίο παρομοίως μειώνει την πολυπλοκότητα.

Όσον αφορά την έξοδο, θα αντιμετωπιστεί ως ένα πρόβλημα multi-label ταξινόμησης. Η έξοδος είναι ένα διάνυσμα με μήκος όσο και ο αριθμός των χρονικών βημάτων. Αυτό γίνεται εφικτό με δύο τρόπους. Ο πρώτος είναι παίρνοντας τα διανύσματα εξόδου κάθε χρονικού βήματος και ο δεύτερος είναι παίρνοντας το διάνυσμα εξόδου μόνο του τελευταίου χρονικού βήματος.

6.3.3 RELATIVE LOCATOR version 2

Μία δεύτερη προσέγγιση για την εύρεση σχετικής θέσης ενός σημείου ενδιαφέροντος περιλαμβάνει την ανάλυση ενός μεγάλου σε διάρκεια τμήματος του μετασχηματισμού. Για όλους τους βασικούς μετασχηματισμούς θα χρησιμοποιηθούν παράθυρα διαρκείας των 100ms. Τα ελάχιστα δείγματα συχνότητας για τον μετασχηματισμό σε spectrogram υπολογίζονται στα 4411, το οποίο αναμένεται αφού η συχνότητα δειγματοληψίας όλων των τραγουδιών είναι 44100Hz. Ο τελικός μετασχηματισμός ενός τρίλεπτου τραγουδιού έχει σχήμα (1800, 4411).

Η ιδέα είναι να τροφοδοτηθεί ένα CNN με 100 δευτερόλεπτα κομματιού και να εφαρμόσει multi-label ταξινόμηση. Για να επιτευχθεί αυτό, το σύνολο των σημείων ενδιαφέροντος μετατρέπεται σε ένα hot-vector. Ένα hot-vector έχει μήκος ίδιο με του αντίστοιχου μετασχηματισμού και σε όποια θέση του ανήκει ένα σημείο ενδιαφέροντος συμπληρώνεται με τιμή 1. Αλλιώς με 0.

Ένα μέρος 100 δευτερολέπτων του μετασχηματισμού μαζί με το hot-vector που του αντιστοιχίζεται δημιουργούν ένα δείγμα εισόδου-εξόδου. Αυτό το μέρος δεν είναι απαραίτητο να ξεκινάει από την αρχή του τραγουδιού. Η διαδικασία επιλογής του αναλύεται σε επόμενο κεφάλαιο <<Δημιουργία Δεδομένων>>.

6.4 Υποσύστημα Β

Με τα το πέρας των προβλέψεων από το υποσύστημα Α και την αποθήκευση των αποτελεσμάτων σε κατάλληλη μορφή, έρχεται η στιγμή για τη σχεδίαση του δεύτερου υποσυστήματος. Πρόκειται για ένα σύστημα που ουσιαστικά χειρίζεται MIDI μηνύματα, στέλνει και δέχεται. Τα MIDI μηνύματα είναι <<χειρονομίες>> που ελέγχουν συσκευές ή λογισμικά συμβατά με MIDI, όπως γεννήτριες μουσικών ήχων ή λογισμικά deejay.

Το δεύτερο υποσύστημα ακολουθεί ένα καθιερωμένο αλγόριθμο ανεξαρτήτως του λογισμικού, στο οποίο είναι <<συνδεδεμένο>>. Η διαφορά ανάμεσα στα λογισμικά deejay βρίσκεται στην ερμηνεία του ίδιου MIDI μηνύματος. Π.χ. το Λογισμικό-Α για το μήνυμα-1 εκτελεί την Κίνηση-1 ενώ το Λογισμικό-Β εκτελεί την Κίνηση-2. Δεν πρόκειται για πρόβλημα αφού με την κατάλληλη διαμόρφωση μπορούν και τα δύο λογισμικά να εκτελούν την Κίνηση-1. Ο παρακάτω αλγόριθμος δέχεται μία λίστα σημείων ενδιαφέροντος για κάθε τραγούδι της play-list. Δεδομένων των σημείων και των κομματιών το τελικό mix υλοποιείται.

Αλγόριθμος

1. Προετοιμασία των δύο Decks
2. Στη βιβλιοθήκη των κομματιών κάνε highlight το πρώτο
3. Μετακίνησε το Fader-1 πάνω
4. Μετακίνησε το Fader-2 κάτω
5. Επανάλαβε έως ότου τελειώσει η play-list:
    a. Προετοίμασε το επόμενο κομμάτι.
    b. Περίμενε ένα χρονικό διάστημα, έως το τρέχων κομμάτι φτάσει στο mix_{point} του.
    c. Πάτα το κουμπί Play στο επόμενο Deck (φορτωμένο το επόμενο κομμάτι).

d. Πάτα το κουμπί beat-sync.
e. Μετακίνησε το επόμενο Fader πάνω.
f. Περίμενε ένα χρονικό διάστημα, μέχρι το τρέχων κομμάτι φτάσει στο $stopmix_{point}$.
g. Μετακίνησε το τρέχων Fader κάτω.
h. Τοποθέτησε τα BPM του επόμενου τραγουδιού στα αρχικά του.
i. Πάτα Cue στο τρέχων Deck.
j. Κάνε focus στη βιβλιοθήκη.
k. Επίλεξε το επόμενο τραγούδι.
l. Η επόμενη πλευρά γίνεται η τρέχουσα και η τρέχουσα η επόμενη.

Το υποσύστημα Β περιορίζεται από την μειωμένη πρόσβαση στα δεδομένα των λογισμικών deejay. Μία προφανής βελτίωση του υποσυστήματος θα ήταν η προσβασιμότητα στη τρέχουσα χρονική στιγμή όπως αναφέρθηκε προηγουμένως. Το βήματα που εκτελούν εντολές αναμονής (“Περίμενε ένα χρονικό διάστημα...”) θα μπορούσαν να αντικατασταθούν με εντολές τύπου “ Όταν το τραγούδι φτάσει στο mix_{point} ...”. Διαφορά αποτελεί ο αναγκαστικός υπολογισμός ενός χρονικού διαστήματος που εισαγάγει σφάλματα (πολύ μικρά), ενώ θα υπήρχε ακρίβεια γνωρίζοντας την τρέχουσα θέση του κομματιού κάθε χρονική στιγμή μέσω του ίδιου του λογισμικού deejay. Επίσης, το χρονικό διάστημα αναμονής το υποσύστημα Β βρίσκεται σε αδράνεια, ενώ στην άλλη περίπτωση θα ήταν δυνατή η εκτέλεση επιπλέον ενεργειών (προσθήκη εφέ στους ήχους, scratch κλπ.).

Σχήμα 6.1 – Ολικό Σύστημα

ΚΕΦΑΛΑΙΟ 7

ΔΗΜΙΟΥΡΓΙΑ DATA-SET

7.1 Εισαγωγή

Με σκοπό να είναι έτοιμο το data-set για εκπαίδευση (training) και testing, θα ακολουθηθεί μία διαδικασία τεσσάρων βημάτων. Όπως αναφέρθηκε σε προηγούμενο κεφάλαιο το Virtual DJ και το Rekordbox εξάγουν τις βάσεις δεδομένων τους σε μορφή XML. Επομένως το πρώτο βήμα χειρίζεται το XML αρχείο και το μετατρέπει σε CSV αποθηκεύοντας τα σημεία ενδιαφέροντος. Έπειτα, από κάθε τραγούδι, για το οποίο υπάρχουν διαθέσιμα σημεία ενδιαφέροντος, δημιουργούνται όλοι οι μετασχηματισμοί της εργασίας και αποθηκεύονται μαζί με τις εξόδους τους ως έχουν. Δηλαδή τα δεδομένα που αποθηκεύονται έχουν διαφορετικά μήκη μεταξύ τους. Σε επόμενο βήμα, το CSV αρχείο διαβάζεται από ένα σύστημα ανάλυσης δεδομένων. Σύμφωνα με τα αποτελέσματα του συστήματος δημιουργείται το τελικό data-set.

7.2 XML σε CSV

Μία βάση δεδομένων ενός dj-software αποθηκεύεται σε μορφή κατάλληλη για το ίδιο το λογισμικό. Εννοείται ότι τα δεδομένα της βάσης δεν είναι αμέσως διαχειρίσιμα, επομένως είναι απαραίτητη μία μετατροπή. Αυτό θα επιτευχθεί δημιουργώντας ένα αρχείο CSV με τα απαραίτητα δεδομένα, απορρίπτοντας την περιττή πληροφορία.

Ένα XML αρχείο δομείται όπως ένα δέντρο. Για πρόσβαση στα δεδομένα που θα χρησιμοποιηθούν πρέπει το δέντρο να διασχιστεί, έως ότου βρεθεί ένα φύλλο του δέντρου. Στο Virtual-DJ και στο Rekordbox ένα σημείο ενδιαφέροντος αποθηκεύεται σε φύλλο του δέντρου. Η διαφορά στα xml αρχεία βρίσκεται στις ετικέτες των τραγουδιών και των σημείων ενδιαφέροντος. Το Virtual-DJ χρησιμοποιεί τις ετικέτες “Song” και “Poi”, ενώ το Rekordbox χρησιμοποιεί “TRACK” και “POSITION_MARK”.

Δύο προγράμματα μετατρέπουν κάθε βάση σε ένα αρχείο CSV με την ίδια δομή, τα οποία ακολουθούν τον ίδιο αλγόριθμο:

α) Πάρε τη ρίζα του XML δέντρου

β) Για κάθε tag που αντιπροσωπεύει ένα τραγούδι:

β.1) Για κάθε tag που αντιπροσωπεύει P.O.I.:

β.1.1) Διάβασε και αποθήκευσε τα σημεία ενδιαφέροντος σε μία λίστα.

γ) Από την τελική λίστα αποθήκευσε το όνομα κάθε τραγουδιού μαζί με τα σημεία ενδιαφέροντος που του αντιστοιχίζονται σε ένα CSV αρχείο.

1. Αποθήκευση ολόκληρων μετασχηματισμών μαζί με τα αντιστοιχιζόμενα P.O.I.

Κάθε μετασχηματισμός έχει συγκεκριμένο σχήμα. Η διάσταση του χρόνου έχει $t_{samples}$ και της συχνότητας $f_{samples}$. Ο αριθμός των $f_{samples}$ καθορίζει την ανάλυση (resolution) του μετασχηματισμού, έχοντας ωστόσο ένα ελάχιστο, το οποίο εξαρτάται από το μέγεθος του παραθύρου. Ο αριθμός των $t_{samples}$ καθορίζεται επίσης από το μέγεθος του παραθύρου και το ποσοστό επικάλυψης (overlap). Οι συναρτήσεις παραθύρου σχεδόν πάντα μειώνονται προς το μηδέν στις οριακές συνθήκες. Χρησιμοποιώντας συνεχόμενα παράθυρα με 0% επικάλυψη για τη δημιουργία του μετασχηματισμού, θα εμφανίζεται απώλεια δεδομένων ή αλλιώς χαμηλότερη ανάλυση (resolution). Από την άλλη, ένα παράθυρο “Hann” με 50% επικάλυψη θα αποτελέσει καλύτερη ανάλυση του μετασχηματισμού. Μεταβλητή αποτελεί το μέγεθος (μήκος) του παραθύρου. Με 50% επικάλυψη μία στήλη του μετασχηματισμού <<διαρκεί>> όσο το μισό μήκος του παραθύρου.

Οι βασικές επιλογές για το μήκος παραθύρου είναι 20 ms και 200 ms, αναλόγως την εφαρμογή. Με σταθερό ρυθμό δειγματοληψίας το αντίστοιχα μήκη του παραθύρου είναι 882 και 8820 αντίστοιχα. Επομένως τα χτιστούν δύο data-set. Το πρώτο θα χρησιμοποιηθεί στα έργα των “Preciser” και “Locator” ενώ το δεύτερο μόνο στου “Locator”. Περαιτέρω έρευνα θα μπορούσε να περιλαμβάνει κι άλλα μήκη παραθύρου. Σ’αυτήν την εργασία θα γίνουν πειράματα με αυτές τις παραμέτρους.

Τα πρώτα βήματα για τη δημιουργία ενός data-set είναι ο υπολογισμός των μετασχηματισμών και της αποθήκευσης τους με τα αντίστοιχα σημεία ενδιαφέροντος. Πιο συγκεκριμένα, για κάθε τραγούδι στο data-set θα υπολογιστούν τα: spectrogram, mel-spectrogram, mfcc,

consant-q, και θα αποθηκευτούν ως numpy αρχεία. Η διάρκειές τους είναι ίδιες εξαιρώντας του constant-q. Έπειτα ένα αρχικοποιείται ένα διάνυσμα μηδενικών τιμών και μήκους $t_{samples}$. Το διάνυσμα παίρνει τιμή ένα όπου υπάρχει σημείο ενδιαφέροντος (hot-vector).

Παράδειγμα

Δίνονται: $song_{duration} = 152.55\,s, nfft = 512, window_{size} = 20\,ms, noverlap = 50\% * window_{size}, poi = 23.456$. Υπολογίστε τη θέση του P.O.I. στο μηδενικό διάνυσμα.

Λύση

$$n_{step} = window_{size} - noverlap = 50\% * window_{size} = 10\,ms$$

$$t_{samples} = song_{duration} * (1000\,ms\,/\,n_{step})$$

$$transform_{shape} = (15255, 512) => Zero_{vector}.length = 15255$$

$$POI_{position} = roundDown(\,poi * (1000\,ms\,/\,n_{step})) = 2345$$

Μία άλλη διαδικασία περιλαμβάνει την παραλλαγή των hot-vectors. Σκοπός είναι να ληφθούν υπόψη οι γειτονικές κλάσεις. Αυτό επιτυγχάνεται βάζοντας τιμή ένα στην κλάση του σημείου ενδιαφέροντος και μικρότερες τιμές στις γειτονικές. Όσο μία κλάση έχει μεγαλύτερη απόσταση από αυτήν του σημείου ενδιαφέροντος τόσο μικρότερη τιμή θα έχει. Η τιμή ακολουθεί τον τύπο:

$$value = \frac{1}{(1 + distance)}$$

με $distance \leq 5$. Η διαδικασία θα ονομαστεί distribution-mode.

2. Ανάλυση Δεδομένων και Δημιουργία

Σκοπός της ανάλυσης δεδομένων είναι η μελέτη των σημείων ενδιαφέροντος και της κατανομής τους σύμφωνα με δεδομένο υπο-μετασχηματισμό. Να σημειωθεί ότι ένα σημείο ενδιαφέροντος έχει συνεχή τιμή. Για τον υπολογισμό της κατανομής των σημείων πρέπει να ομαδοποιηθούν ανά χρονικά διαστήματα. Η κατανομή εξαρτάται από το υποσύστημα ("Locator", "Preciser"). Για το "Locator" θα γίνουν δύο ειδών ομαδοποιήσεις. Μία ανά 5 δευτερόλεπτα και μία ανά 100ms στη σειρά. Για τον "Preciser" οι ομαδοποιήσεις θα γίνουν ανά 10ms και 20ms.

Κάθε νευρωνικό δίκτυο έχει τουλάχιστον μία είσοδο και μία έξοδο με συγκεκριμένα σχήματα. Για παράδειγμα ένα convolutional νευρωνικό δίκτυο έχει στρώμα εισόδου που δέχεται πίνακες τεσσάρων διαστάσεων όπως π.χ.$(batch_{size}, 1000, 512, n_{channels})$ και ένα στρώμα εξόδου δύο διαστάσεων όπως $(batch_{size}, 1000)$. Επομένως, η αρχιτεκτονική εξυπηρετεί σε ταξινόμηση ανάμεσα σε 1000 κλάσεις. Ένα ορθό σύστημα εκπαίδευσης απαιτεί data-set με δείγματα που στο σύνολο τους ακολουθούν ομοιόμορφη κατανομή όσον αφορά τις κλάσεις. Σκοπός αυτού του βήματος είναι η μελέτη της κατανομής των κλάσεων, δεδομένου αριθμού κλάσεων $n_{classes}$, ένα $offset_{initial}$ και ενός $offset_{step}$.

Επειδή το στρώμα εισόδου λαμβάνει σταθερά σχήματα δεδομένων οι μετασχηματισμοί πρέπει να έχουν σταθερό σχήμα $(n_{rows}, n_{columns}, n_{channels})$. Ως αποτέλεσμα, ένα δείγμα εισόδου είναι ένα μέρος του μετασχηματισμού και όχι ολόκληρος ο μετασχηματισμός. Αυτό το μέρος εξαρτάται απο τον χρόνο. Έτσι, αν ένας μετασχηματισμός έχει σχήμα $(t_{sample}, f_{samples})$, ένα δείγμα εισόδου είναι υπο-μετασχηματισμός σχήματος $(n_{rows}, n_{columns})$ όπου:

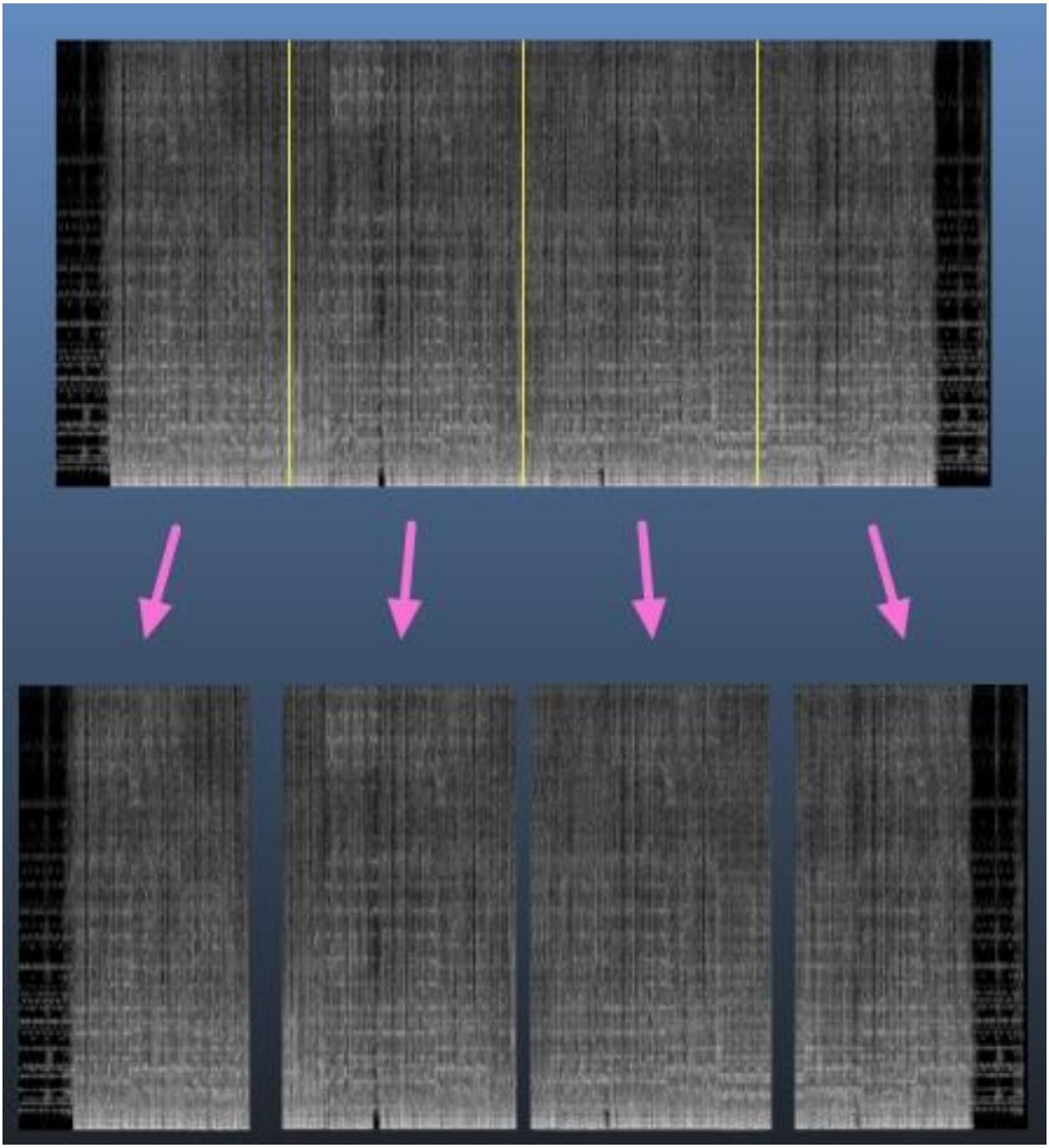
$$n_{rows} \leq t_{samples}$$

$$n_{columns} = f_{samples}$$

Επίσης, τα αντίστοιχα P.O.I. διανύσματα θα έχουν μήκος n_{rows}.

Το μόνο που απομένει είναι ο τρόπος με τον οποίο επιλέγονται οι υπο-μετασχηματισμοί. Το πρώτο πείραμα ήταν να δημιουργηθούν μη επικαλυπτόμενοι υπο-μετασχηματισμοί, όπως φαίνεται στην εικόνα 1. Το δεύτερο εισήγαγε δύο όρους: $offset_{initial}$ και $offset_{step}$ και χρησιμοποιεί επικαλυπτόμενους υπο-μετασχηματισμούς.

Ο πρώτος τρόπος, παρ'όλο που γεννάει επαρκή αριθμό δεδομένων για training και testing, η κατανομή τους δεν ήταν ομοιόμορφη. Οι κλάσεις 1, 2 και 3 που αντιπροσωπεύουν τα χρονικά διαστήματα $[0-100), [100-200), [200-300)$ σε ms αντίστοιχα, με παράθυρο 200ms, εμφανίζονται περισσότερες φορές από τις υπόλοιπες κλάσεις, κάνοντας τη μέθοδο να απορριφθεί. Αυτό συμβαίνει επειδή υπάρχουν πολλά σημεία ενδιαφέροντος στην αρχή του κομματιού τις περισσότερες φορές.



Σχήμα 7.2.1. Ολόκληρος μετασχηματισμός (άνω) και 4 συνεχόμενοι υπο-μετασχηματισμοί (κάτω)

.

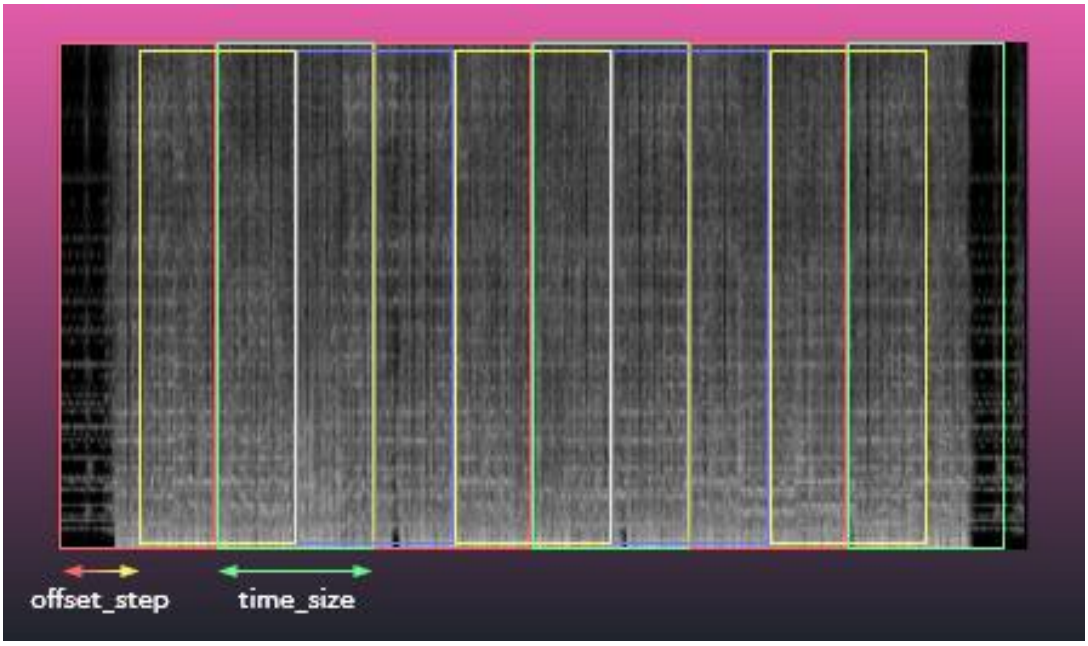
Ο δεύτερος τρόπος χρησιμοποιεί επικαλυπτόμενους υπο-μετασχηματισμούς. Κάθε υπο-μετασχηματισμός είναι σχετικός χρονικά από τον προηγούμενο κατά ένα $offset_{step}$. Ένας υπο-μετασχηματισμός δίνεται από:

$$subtransform_i = transform[t: t + time_{size}, :]$$

Ο επόμενος:

$$subtransform_{i+1} = transform[t + offset_{step}: t + time_{size} + offset_{step}, :]$$

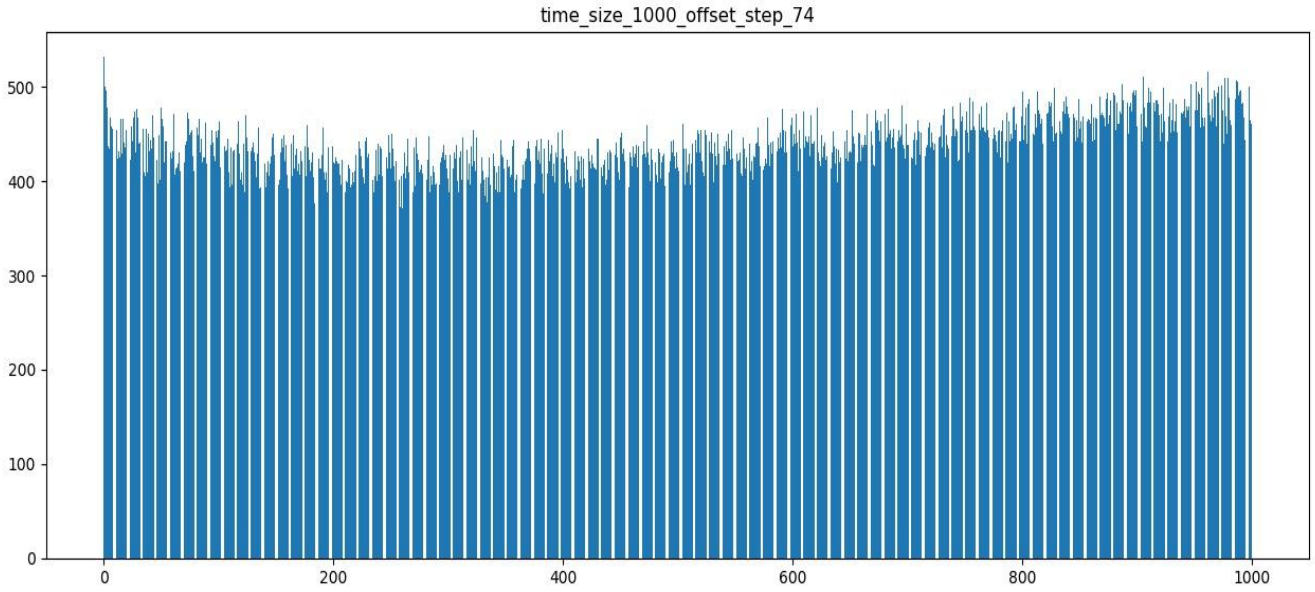
Όσο περισσότερα δεδομένα δημιουργούνται έτσι η κατανομή τείνει να γίνει ομοιόμορφη. Η εικόνα 2 δείχνει την επιλογή υπο-μετασχηματισμών.



Σχήμα 7.2.2 – Ολόκληρος μετασχηματισμός και όλοι οι επιλεγμένοι υπο-μετασχηματισμοί.

Ο πρώτος υπο-μετασχηματισμός που επιλέχθηκε αρχικά ξεκίναγε από την αρχή του κομματιού και αυτή ήταν η αιτία της αυξημένης συχνότητας των κλάσεων 1, 2 και 3. Η τρίτη σκέψη ήταν να εισαχθεί ο όρος $offset_{initial}$. Επομένως ο πρώτος υπο-μετασχηματισμός ξεκινάει

από ένα σημείο εντός του διαστήματος $[0 - offset_{initial}]$. Το σημείο επιλέγεται τυχαία μέσω της βιβλιοθήκης random της Python. Οι υπόλοιποι υπο-μετασχηματισμοί υπολογίζονται με τη χρήση του $offset_{step}$. Όλες οι κλάσεις είναι πλέον ισορροπημένες. Ωστόσο, η κατανομή εξαρτάται και από την τιμή του $offset_{step}$. Αφού υπολογιστούν όλες οι κατανομές για κάθε $offset_{step}$ εντός του διαστήματος $[50, 100)$, συμπεραίνεται ότι για $n_{classes}$ ίσο με 500 ή 1000, καλές τιμές του $offset_{step}$, είναι 72 και 74 αντίστοιχα. Το συμπέρασμα βγήκε μελετώντας την τυπική απόκλιση και τον συντελεστή απόκλισης κάθε κατανομής.



Σχήμα 7.2.3. Κατανομή 500 κλάσεων.. Data-set με μέγεθος παραθύρου 10 ms.

Τελικά, μαζί με τους μετασχηματισμούς των τραγουδιών και τα αντίστοιχα σημεία, αποθηκεύονται και τα $offsets$ των επιλεγμένων κατανομών σε διαφορετικό υπο-φάκελο με δομή ονομασίας:

$$song_{name} + "offset" + offset$$

ΚΕΦΑΛΑΙΟ 8

ΑΥΞΗΣΗ ΔΕΔΟΜΕΝΩΝ

8.1 Εισαγωγή

Η υλοποίηση της εργασίας απαιτεί ειδικευμένο data-set προσαρμοσμένο στο σκοπό της. Η διαδικασία δημιουργίας του data-set δεν είναι αυτοματοποιημένη και ως συνέπεια η σημείωση των σημείων ενδιαφέροντος απαιτεί πολύ χρόνο για μεγάλο αριθμό τραγουδιών. Τα τραγούδια στο σύνολο τους που έχουν σημειωθεί χειροκίνητα είναι 600. Αυτό σημαίνει για το υπο-σύστημα “Preciser” με ένα P.O.I. ανά δύο μπάρες, ότι έχει στη διάθεση του περίπου 30.000 σημεία για εκπαίδευση, επικύρωση και δοκιμή. Ωστόσο για το υποσύστημα “Relative Locator” τα δείγματα στο data-set είναι μόνο 600. Δεν επαρκεί αυτό το μέγεθος για να βγει κάποιο συμπέρασμα. [20] Η εκπαίδευση βαθέων νευρωνικών δικτύων γενικά απαιτεί μεγάλο αριθμό δειγμάτων εκπαίδευσης. Όταν ο αριθμός τους είναι μικρός, τα μοντέλα γίνονται ευαίσθητα σε overfitting, προκαλώντας μεγάλα σφάλματα γενίκευσης στα δείγματα δοκιμής.

Το πρώτο βήμα στην επίλυση του προβλήματος των δειγμάτων ήταν να προστεθούν άλλα 400 τραγούδια για τον “Relative Locator”. Συνολικά υπάρχουν 1000 δείγματα το οποίο δεν είναι αρκετό. Το επόμενο βήμα είναι data augmentation, γεννώντας νέα δεδομένα. *Οι τεχνικές data augmentation όπως η αναστροφή ή η περικοπή μίας εικόνας, που μεγεθύνουν τα δείγματα του του data-set, έχουν χρησιμοποιηθεί για την αύξηση της απόδοσης της γενίκευσης των βαθιών νευρωνικών δικτύων.

Στην επιβλεπόμενη μάθηση, μία πρακτική είναι η αύξηση των δεδομένων θέτοντας την ίδια ετικέτα σε παραγόμενα δεδομένα από την ίδια πηγή.

Για προβλήματα ταξινόμησης εικόνας οι παραδοσιακές μέθοδοι παραγωγής δεδομένων είναι η περικοπή ή περιστροφή των εικόνων του data-set. Τα spectrograms είναι δι-διάστατα με την πρώτη διάσταση να περιγράφει το χρόνο και τη δεύτερη τη συχνότητας. Έπειτα μία τέτοια είσοδος αντιστοιχίζεται σε μία συνεχή μεταβλητή χρόνου (P.O.I.). Όπως φαίνεται στην εικόνα 1, ένα σημείο ενδιαφέροντος εντοπίζεται στον οριζόντιο άξονα. Περιστρέφοντας 180 μοίρες ως προς τον άξονα του χρόνου το σημείο ενδιαφέροντος βρίσκεται στο ίδιο μέρος. Επομένως με αυτόν τον τρόπο διπλασιάζεται το data-set. Άλλη τεχνική είναι η κάλυψη συχνότητας, δηλαδή η κάλυψη συγκεκριμένου διαστήματος συχνότητας με μία σταθερή τιμή (μηδενική ενδεικτικά).

Αυτή η προσέγγιση εφαρμόζεται στους ίδιους τους μετασχηματισμούς. Αντί αυτού, μπορούν να δημιουργηθούν νέα αρχεία ήχου που με τη σειρά τους θα παράγουν μετασχηματισμούς. Η μετατροπή ενός κομματιού σε ένα καινούριο, προσθέτοντας ηχητικά εφέ ή ήχους κλπ, είναι η δημιουργία ενός ρεμίξ (remix). Ένα δημοφιλές είδος ρεμίξ στην χιπ χοπ και RNB μουσική είναι το “Chopped N’ Screwed”. Πρόκειται ουσιαστικά για μία τραβηγμένη ως προς τον χρόνο

έκδοση του αυθεντικού κομματιού. Μπορεί να το φανταστεί κανείς σα να παίζει ένα βινύλιο 12” με ταχύτητα 20 r.p.m. (στροφές ανά λεπτό), αντί για 33 και 1/3 που είναι η καθιερωμένη περιστροφική ταχύτητα. Άλλοι τύποι από ρεμίξ δημιουργούνται μετακινώντας το pitch, προσθέτοντας reverb κ.α. Αυτές οι μέθοδοι μπορούν να εφαρμοστούν για να αυξηθεί το μέγεθος του data-set.

8.2 Τέντωμα Χρόνου

Το τέντωμα χρόνου είναι η διαδικασία αλλαγής της ταχύτητας και συνεπώς της διάρκειας ενός ηχητικού σήματος χωρίς να μετακινηθεί το pitch. Ο απλούστερος τρόπος αλλαγής της διάρκειας ή και του pitch γίνεται μέσω μετατροπής του ρυθμού δειγματοληψίας. Είναι μία μαθηματική λειτουργία που αποτελεσματικά ξαναχτίζει μία συνεχή κυματομορφή από τα δείγματα της, με τα δείγματα να σχηματίζουν την κυματομορφή αλλά με διαφορετικό ρυθμό. Όταν τα δείγματα παίζονται στον κανονικό τους ρυθμό το ηχητικό κλιπ ακούγεται πιο αργό ή πιο γρήγορο. Κανονικά, επιβραδύνοντας τον ήχο χαμηλώνει το pitch και επιταχύνοντας το ανεβαίνει το pitch. Για να επιτευχθεί αυτό θα χρησιμοποιηθεί η βιβλιοθήκη "libROSA" της Python. Η συνάρτηση που παρέχει ονομάζεται effects.time_stretch και ο αλγόριθμος που ακολουθεί εφαρμόζει τέντωμα χρόνου χωρίς να αλλάξει το pitch. Ορίσματα:

Parameters:

y:np.ndarray [shape=(n,)]

audio time series

rate:float > 0 [scalar]

Stretch factor. If *rate > 1*, then the signal is sped up. If *rate < 1*, then the signal is slowed down.

kwargs:additional keyword arguments.

See *librosa.decompose.stft* for details.

Returns:

y_stretch:np.ndarray [shape=(round(n/rate),)]

audio time series stretched by the specified rate

Το rate ή το ποσοστό τεντώματος θα έχει τιμές $[0.75, 1.25]$, τριπλασιάζοντας έτσι το μέγεθος του data-set.

Μετά το τέντωμα, τα σημεία ενδιαφέροντος του κομματιού δε βρίσκονται στις ίδιες θέσεις. Έστω ένα τραγούδι ότι διαρκεί τ δευτερόλεπτα. Με rate 0.75 το τραγούδι μεταβάλει την ταχύτητα του

κατά 75% της αρχικής. Επομένως το νέο τραγούδι διαρκεί $\frac{\tau}{0.75}$. Έτσι όλα τα σημεία ενδιαφέροντος μετατοπίζονται σε:

$$poi_{new} = \frac{poi_{original}}{0.75}$$

Απόδειξη

Το αρχικό τραγούδι είναι ένας μονοδιάστατος πίνακας με N στοιχεία και ρυθμό δειγματοληψίας sr. Η διάρκεια του τραγουδιού σε δευτερόλεπτα είναι:

$$t = \frac{N}{sr}$$

Τεντώνοντας το τραγούδι αλλάζει ο αριθμός των δειγμάτων στον μονοδιάστατο πίνακα κρατώντας σταθερό το ρυθμό δειγματοληψίας. Το νέο τραγούδι θα έχει:

$$N_{new} = \frac{N}{rate}$$

Έτσι η νέα διάρκεια είναι:

$$t_{new} = \frac{N_{new}}{sr_{new}} = \frac{N}{sr * rate} = \frac{t}{rate}, \quad sr_{new} = sr$$

Αυτή η εξίσωση εφαρμόζεται σε κάθε αρχικό τραγούδι. Τα νέα σημεία ενδιαφέροντος υπολογίζονται:

$$poi\_new = \frac{poi}{rate}$$

8.3 Μεταβολή Pitch

[21] Το pitch είναι μία ηχητική αίσθηση με την οποία οι ακροατές αναθέτουν μουσικές νότες σε σχετικές θέσεις στη μουσική κλίμακα, βασισμένοι κυρίως στην αντίληψη της συχνότητας δόνησης. Το pitch είναι στενά συνδεδεμένο με τη συχνότητα, αλλά δεν είναι ισοδύναμα. Η συχνότητα είναι αντικειμενικό, επιστημονικό χαρακτηριστικό που μπορεί να μετρηθεί. Το pitch είναι η αντικειμενική αντίληψη του κάθε ανθρώπου για ένα ηχητικό κύμα, που δε μπορεί να μετρηθεί άμεσα

[22] Η μεταβολή του pitch δεν αλλάζει το μήκος του σήματος. Σε πρακτικές εφαρμογές, αυτό επιτυγχάνεται αλλάζοντας το μήκος του ήχου κι έπειτα εφαρμόζοντας αλλαγή του ρυθμού δειγματοληψίας για την αλλαγή του pitch. Επικρατεί σύγχυση για τις έννοιες <<μεταβολή pitch>> και <<μεταβολή συχνότητας>>. Μία πραγματική μεταβολή συχνότητας θα μετατοπίσει το φάσμα του ήχου, ενώ το pitch θα το διαστέλλει, διατηρώντας την αρμονική σχέση του ήχου.

Για να επιτευχθεί μεταβολή pitch, θα χρησιμοποιηθεί η συνάρτηση του “libROSA” effects.pitch_shift, που μεταβάλλει το pitch της κυματομορφής κατά n_{steps} semitones. Ορίσματα:

Parameters:

y:np.ndarray [shape=(n,)]

audio time series

sr:number > 0 [scalar]

audio sampling rate of *y*

n_steps:float [scalar]

how many (fractional) half-steps to shift *y*

bins_per_octave:float > 0 [scalar]

how many steps per octave

res_type:string

Resample type. Possible options: ‘kaiser_best’, ‘kaiser_fast’, and ‘scipy’, ‘polyphase’, ‘fft’. By default, ‘kaiser_best’ is used.

See **core.resample** for more information.

kwargs: additional keyword arguments.

See *librosa.decompose.stft* for details.

Returns: **y_shift:np.ndarray [shape=(n,)]**

The pitch-shifted audio time-series

[23] Ένα ημιτόνιο, ή αλλιώς μισό βήμα ή μισός τόνος, είναι το μικρότερο μουσικό διάστημα, που χρησιμοποιείται κοινώς σε Δυτικά τονική μουσική, και θεωρείται το πιο παράφωνο όταν ακούγεται αρμονικά. Ορίζεται από το διάστημα ανάμεσα σε δύο γειτονικές νότες στην κλίμακα 12 νοτών. Για παράδειγμα, η νότα C είναι γειτονική με τη C#. Το διάστημα μεταξύ τους είναι ένα ημιτόνιο.

Οι τιμές του n_{steps} που επιλέγονται είναι [-6, -3, 3, 6] πενταπλασιάζοντας το αρχικό data-set. Το σύνολο των τραγουδιών με σημεία ενδιαφέροντος από 1000 πλέον είναι 7000.

ΚΕΦΑΛΑΙΟ 9

ΦΟΡΤΩΜΑ DATA-SET

9.1 Εισαγωγή

Η δομή των data-sets βοηθάει στην ισορροπία της χρονικής και χωρικής πολυπλοκότητας. Κάθε data-set παρέχει ένα φάκελο με το όνομα “offsets_list_dataset.csv”. Αυτός ο φάκελος περιέχει γραμμές από IDs και ονόματα δειγμάτων. Τα ονόματα δειγμάτων έχουν τη δομή: όνομα_τραγουδιού + “_offset_X.npy”. Με αυτόν τον τρόπο ολόκληρο το data-set μπορεί να φορτωθεί σ’ένα πρόγραμμα αφού είναι απλά μία λίστα από strings με αποτέλεσμα να απαιτείται λίγη μνήμη. Πιο διευκρινιστικά, για ένα παράθυρο των 100ms το αρχείο “offsets_list_dataset.csv” του data-set περιέχει 116.844 γραμμές και το μέγεθος του είναι 7.9 MB. Τότε, το μόνο που μένει είναι για να γίνει διαχειρίσιμο το data-set και τα δείγματα του αρκεί να αρχικοποιηθεί ένα αντικείμενο της κλάσης SkaterbotDataGenerator. Αυτό το αντικείμενο παρέχει <<γεννήτριες>> (generators) οι οποίες είναι συναρτήσεις όπως οι “train_flow” και “test_flow”. Σκοπός τους είναι το εύκολο φόρτωμα

των δεδομένων κατά το run-time χρησιμοποιώντας μικρό ποσοστό της μνήμης.

Ο δεύτερος τρόπος διαχείρισης των δειγμάτων των data-sets είναι το φόρτωμα μετασχηματισμού (ολόκληρου) και των σημείων ενδιαφέροντος, επιτρέποντας στον χρήστη να τα διαχειριστεί όπως θέλει. Για να γίνει εφικτό αυτό πρέπει να διαβαστεί το αρχείο “samples_list.csv”. Για κάθε δείγμα στη λίστα είναι αναγκαία η πρόσβαση στους υπο-φακέλους “inputs” και “outputs_raw”, έτσι ώστε να διαβαστούν οι μετασχηματισμοί και τα σημεία. Τότε ο χρήστης μπορεί να διαχειριστεί τα δεδομένα όπως θέλει.

9.2 Διαδικασία του “flow”

Όλες οι *γεννήτριες* ακολουθούν την ίδια διαδικασία με σκοπό την τροφοδότηση του μοντέλου με mini-batches αποδοτικά κατά το training, testing ή predicting. Το πρώτο βήμα είναι η δημιουργία ενός αντικειμένου της κλάσης “SkaterbotDataGenerator”. Ο constructor αρχικοποιεί μεταβλητές του αντικειμένου, όπως τα μεγέθη των διαστάσεων εισόδου και εξόδου, και διαβάζει ένα CSV αρχείο που περιλαμβάνει το data-set σε strings. Έτσι δεν υπάρχουν παραβάσεις μνήμης. Το data-set αποθηκεύεται στη μνήμη ως λίστα της Python με strings. Έπειτα το data-set χωρίζεται σε train-set και test-set.

Μία γεννήτρια “flow” αρχικοποιεί ένα μετρητή με μηδενική τιμή και ανακατώνει το data-set έτσι ώστε τα δεδομένα να εισέρχονται με τυχαία σειρά. Τα δεδομένα εισέρχονται ως mini-batches σταθερού μεγέθους. Τα mini-batches χρησιμοποιούνται για εκπαίδευση, επικύρωση ή δοκιμή (training, validation ή testing). Η γεννήτρια συνεχίζει να επιστρέφει mini-batches αφού χρησιμοποιηθούν από το μοντέλο-νευρωνικό δίκτυο έως ότου εξαντληθούν τα δεδομένα

9.3 Mini-batch

Κάθε mini-batch περιέχει ένα ζευγάρι εισόδου και εξόδου. Η είσοδος είναι ένας υπο-μετασχηματισμός. Ο υπο-μετασχηματισμός εξήχθη από τον υπολογισμένο μετασχηματισμό ενός τραγουδιού. Το μήκος-διάρκεια του υπο-μετασχηματισμού είναι “time_size”. Η έξοδος είναι ένα διάνυσμα μεγέθους “time_size” με τα σημεία ενδιαφέροντος εκφρασμένα ως απλά hot-vector ή σε distribution-mode ή ένας αριθμός με τρία δεκαδικά ψηφία. Αυτό εξαρτάται από το έργο προς υλοποίηση.

ΚΕΦΑΛΑΙΟ 10

ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΠΑΙΔΕΥΣΗΣ - ΔΟΚΙΜΗΣ - ΠΡΟΒΛΕΨΗΣ

10.1 Εισαγωγή

Το κεφάλαιο 6 παρουσιάζει δύο τρόπους φορτώματος για την εκπαίδευση και την δοκιμή του νευρωνικού δικτύου. Ανεξαρτήτως του νευρωνικού, μία εποχή (epoch) αποτελείται από δύο βήματα. Το πρώτο αφορά την εκπαίδευση και το δεύτερο την δοκιμή . Τα βήματα επαναλαμβάνονται για οποιοδήποτε αριθμό εποχών.

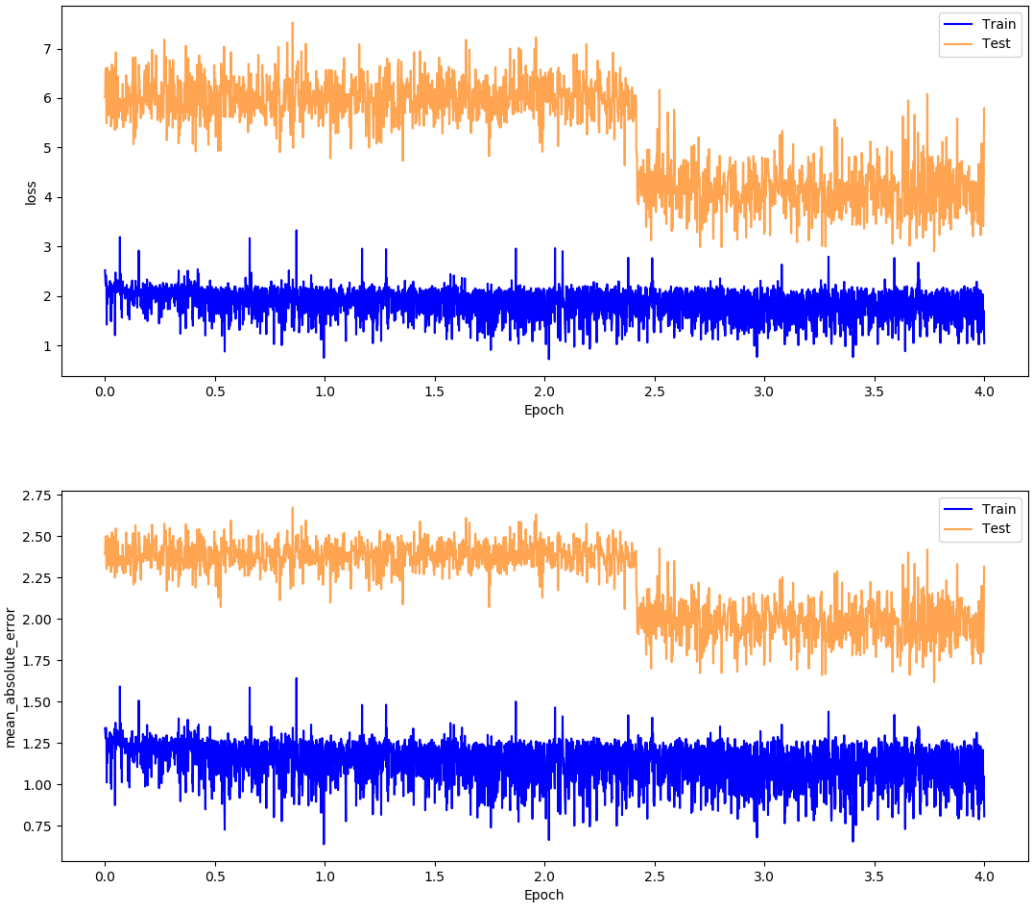
Κάθε μοντέλο-νευρωνικό κατασκευάζεται μέσω των Tensorflow και Keras. Το Tensorflow είναι “μία end-to-end open source” πλατφόρμα για μηχανική μάθηση. Έχει ένα περιεκτικό , ευέλικτο οικοσύστημα εργαλείων, βιβλιοθηκών και πόρων της κοινότητας που επιτρέπει στους ερευνητές να για να εξελίξουν το “state-of-the-art” στη

μηχανική μάθηση και στους developers να χτίσουν εύκολα και να αναπτύξουν εφαρμογές που χρησιμοποιούν μοντέλα μιχανικής μάθησης. Το Keras είναι ένα Application Programming Interface (API) για το χτίσιμο και την εκπαίδευση μοντέλων deep-learning.

Κατά την εκπαίδευση, ένα mini-batch φορτώνεται στη μνήμη και χρησιμοποιείται από το μοντέλο του πειράματος. Για την επιτάχυνση της διαδικασίας, χρησιμοποιείται μία κάρτα γραφικών (GPU) GeForce GTX 1050 με 3GB of VRAM. Λόγω της χωρητικότητας της GPU όλα τα μοντέλα θα χρησιμοποιούν $batch_{size} \in \{4, 8, 32\}$ Το υπόλοιπο υπολογιστικό σύστημα περιλαμβάνει έναν επεξεργαστή i7-8750H @ 2.20GHz και 16GB RAM.

10.2 Preciser version 1 - Μοντέλα

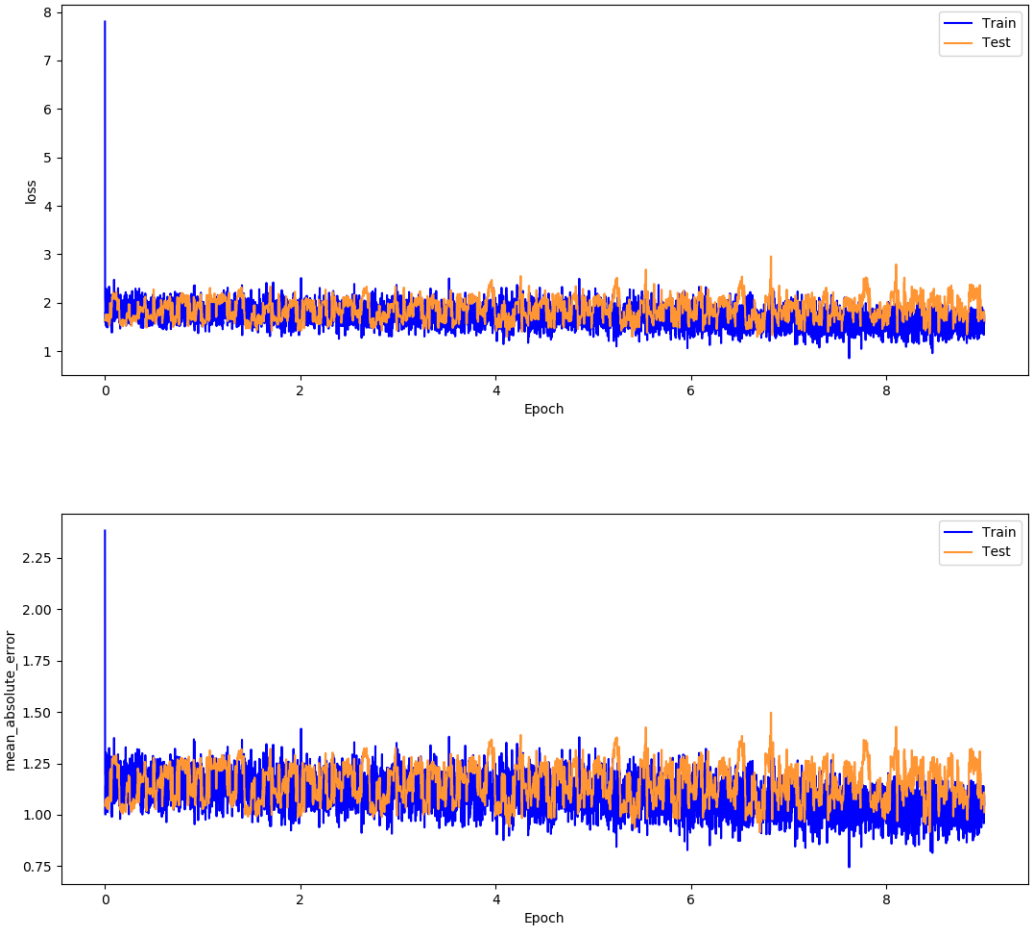
- $Architecture - ID = 1$
  $transform_{type} = spectrogram,$
  $transform_{shape} = (500, 512)$
  $batch_{size} = 1$
  $Optimizer = SGD$ with:
  a) $learning_{rate} = 0.001$
  b) $decay = 0.000001$
  c) $momentum_{nesterov} = 0.9$
  $loss = Mean\ Squared\ Error$



Σχήμα 10.2.1 – Πείραμα 1

Το μοντέλο εκπαιδεύτηκε με 850720 δείγματα και δοκιμάστηκε με 142041.

- $Architecture - ID = 1$
  $transform_{type} = melspectrogram,$
  $transform_{shape} = (500, 256)$
  $batch_{size} = 1$
  $Optimizer = SGD$ with:
  a) $learning_{rate} = 0.0001$
  b) $decay = 0.000001$
  c) $momentum_{nesterov} = 0.9$
  $loss = Mean\ Squared\ Error$

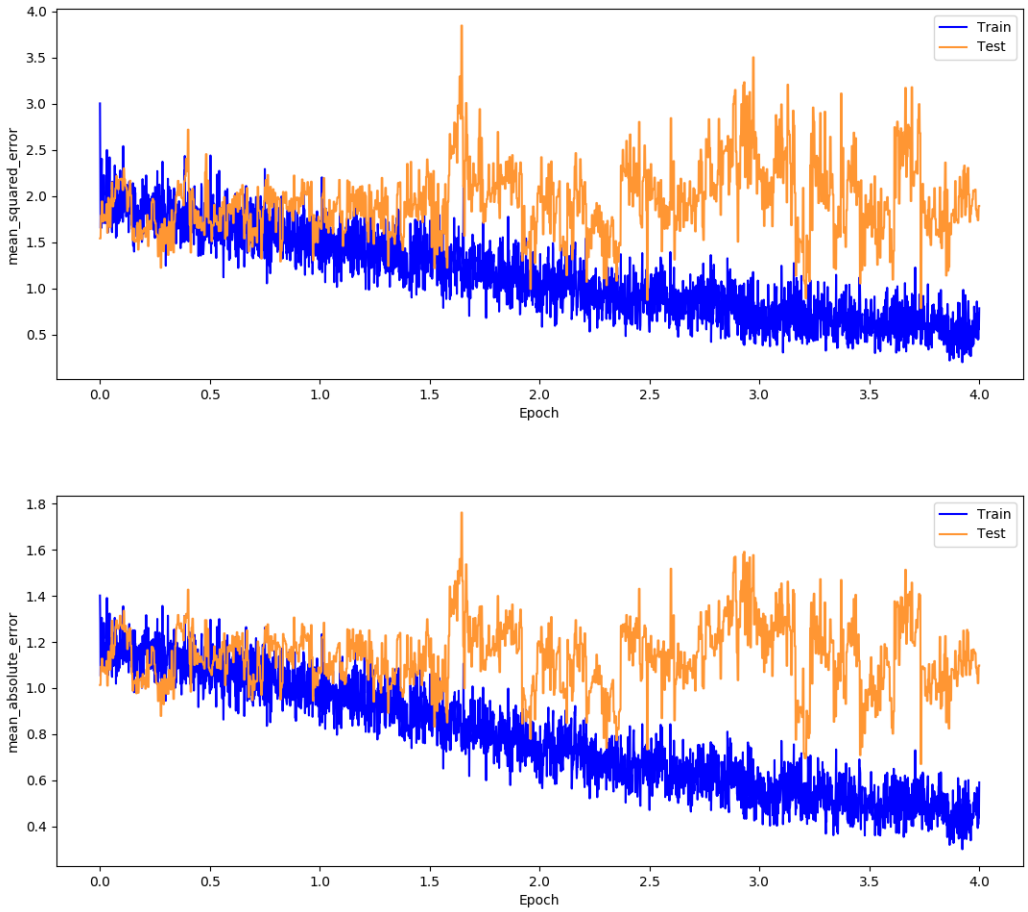


Σχήμα 10.2.2 – Πείραμα 2

Το μοντέλο αυτό εκπαιδεύτηκε για 9 epochs. Παρατηρώντας τα γραφήματα, είναι φανερό πως το δίκτυο παρουσιάζει overfitting

αφού το σφάλμα εκπαίδευσης μειώνεται αργά και το σφάλμα δοκιμής κυμαίνεται στη τιμή 1.2 Χρησιμοποιήθηκαν 67224 δείγματα εκπαίδευσης και 28494 δείγματα δοκιμής .

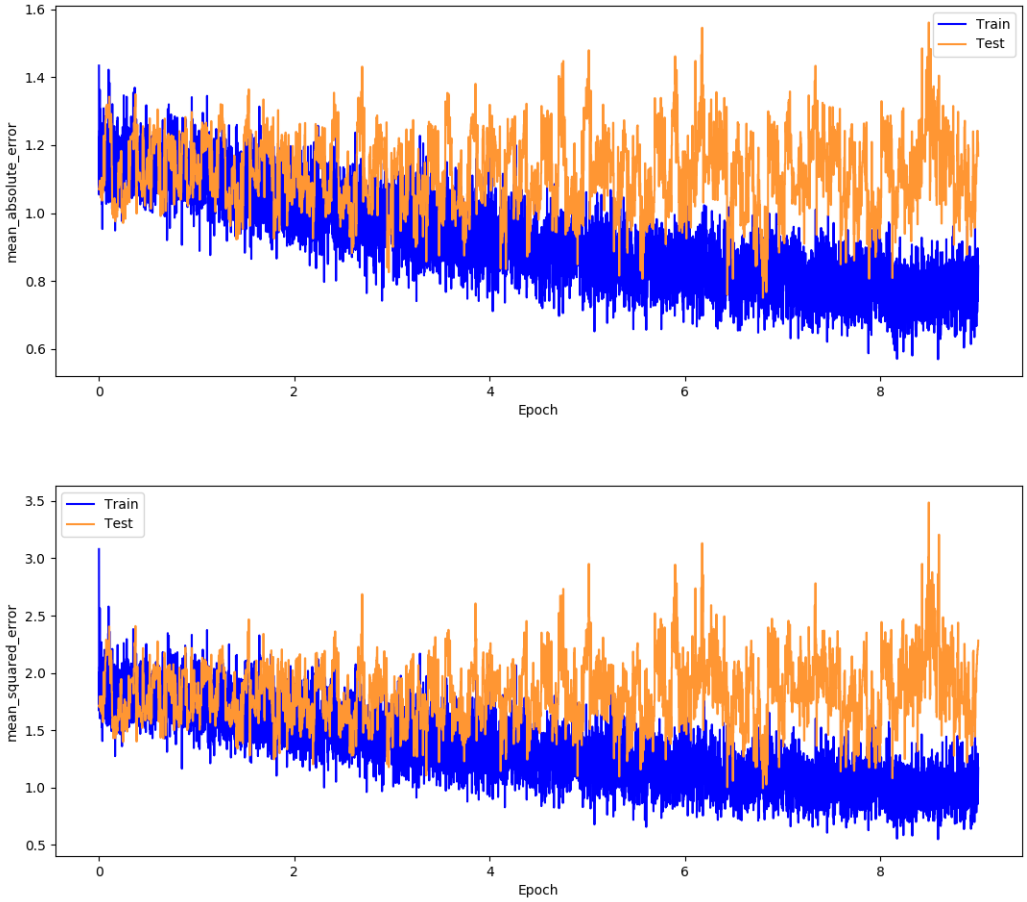
- $Architecture - ID = 3$
  $transform_{type} = melspectrogram,$
  $transform_{shape} = (500, 256)$
  $batch_{size} = 1$
  $Optimizer = SGD$ with:
  a) $learning_{rate} = 0.001$
  b) $decay = 0.000001$
  c) $momentum_{nesterov} = 0.9$
  $loss = Mean\ Squared\ Error$



Σχήμα 10.2.3 – Πείραμα 3

Σ'αυτό το μοντέλο γίνεται πιο αισθητή η ύπαρξη του overfitting. Χρησιμοποιήθηκαν 67224 δείγματα εκπαίδευσης και 28494 δείγματα δοκιμής

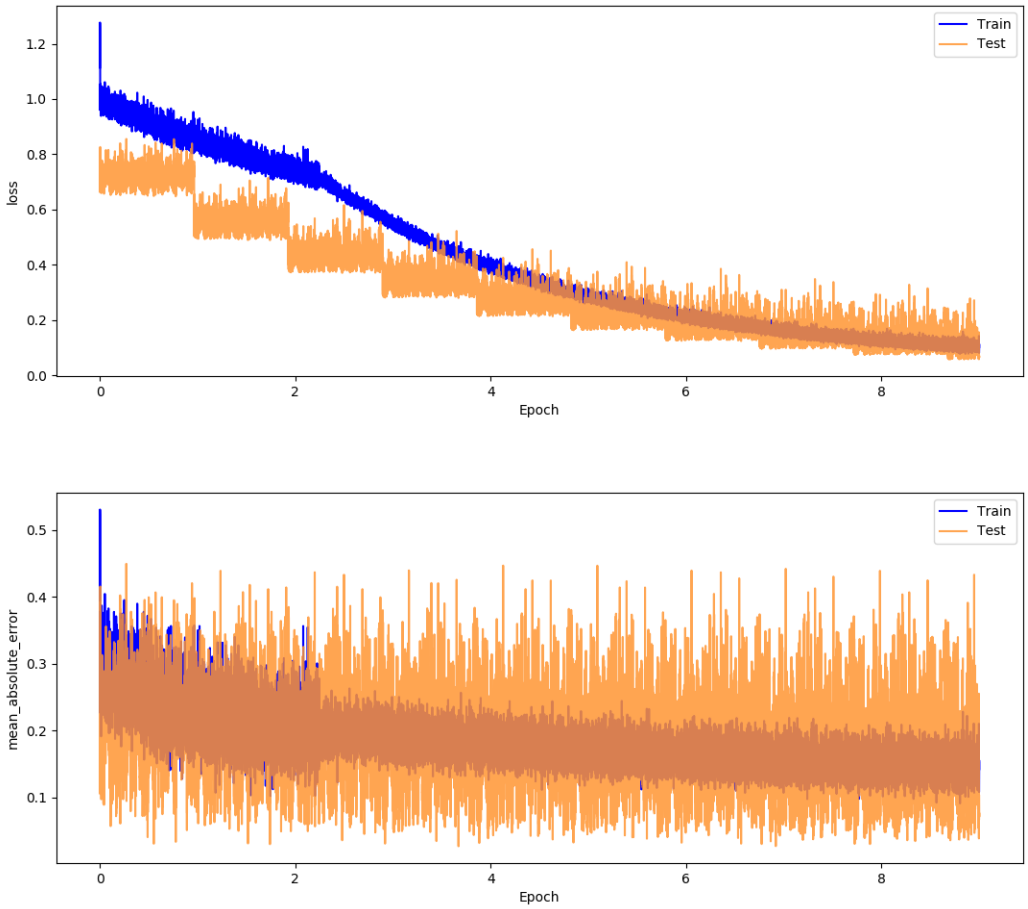
- $Architecture - ID = 6$
  $transform_{type} = melspectrogram,$
  $transform_{shape} = (500, 256)$
  $batch_{size} = 1$
  $Optimizer = SGD$ with:
  a) $learning_{rate} = 0.001$
  b) $decay = 0.000001$
  c) $momentum_{nesterov} = 0.9$
  $loss = Mean\ Squared\ Error$



Σχήμα 10.2.4 – Πείραμα 4

Χρησιμοποιήθηκαν 67224 δείγματα εκπαίδευσης και 28494 δείγματα δοκιμής

- $Architecture - ID = 8$
  $transform_{type} = Constant - Q,$
  $transform_{shape} = (100, 108)$
  $batch_{size} = 32$
  $Optimizer = SGD$ with:
  a) $learning_{rate} = 0.001$
  b) $decay = 0.000001$
  c) $momentum_{nesterov} = 0.9$
  $loss = Mean\ Squared\ Error$

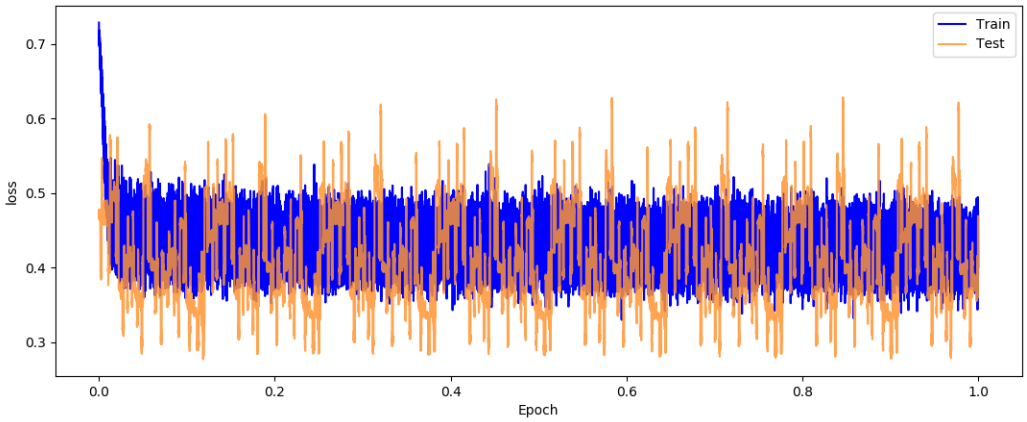


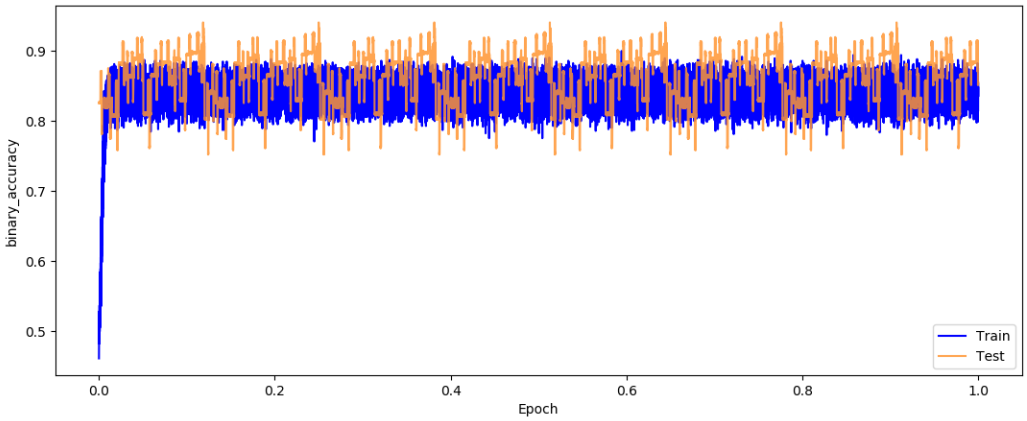
Σχήμα 10.2.5 – Πείραμα 5

Κομμάτια ενός Constant-Q μετασχηματισμού διάρκειας 1.161 δευτερολέπτων αποδίδουν επιθυμητά αποτελέσματα. Το μοντέλο χρησιμοποιεί 181390 δείγματα εκπαίδευσης και79438 δοκιμής . Το πρώτο γράφημα δείχνει σύγκλιση της απώλειας εκπαίδευσης σε μία μέση τιμή 0.1 ενώ το σφάλμα δοκιμής μετά από 10 εποχές συγκλίνει σε μία μέση τιμή 0.15. Επίσης κατά τη δεύτερη εποχή φαίνεται από την απώλεια εκπαίδευσης ότι το σύστημα αποκτά ευστάθεια, αφού έχουν μειωθεί τα πλάτη των ταλαντώσεων.

10.3 Relative Locator version 1 - Μοντέλα

- $Architecture - ID = 3$
  $transform_{type} = constant - q,$
  $transform_{shape} = (86, 100, 108)$
  $Optimizer = SGD$ with:
  a) $learning_{rate} = 0.001$
  b) $decay = 0.000001$
  c) $momentum_{nesterov} = 0.9$
  $loss = Binary\ Cross\ Entropy$



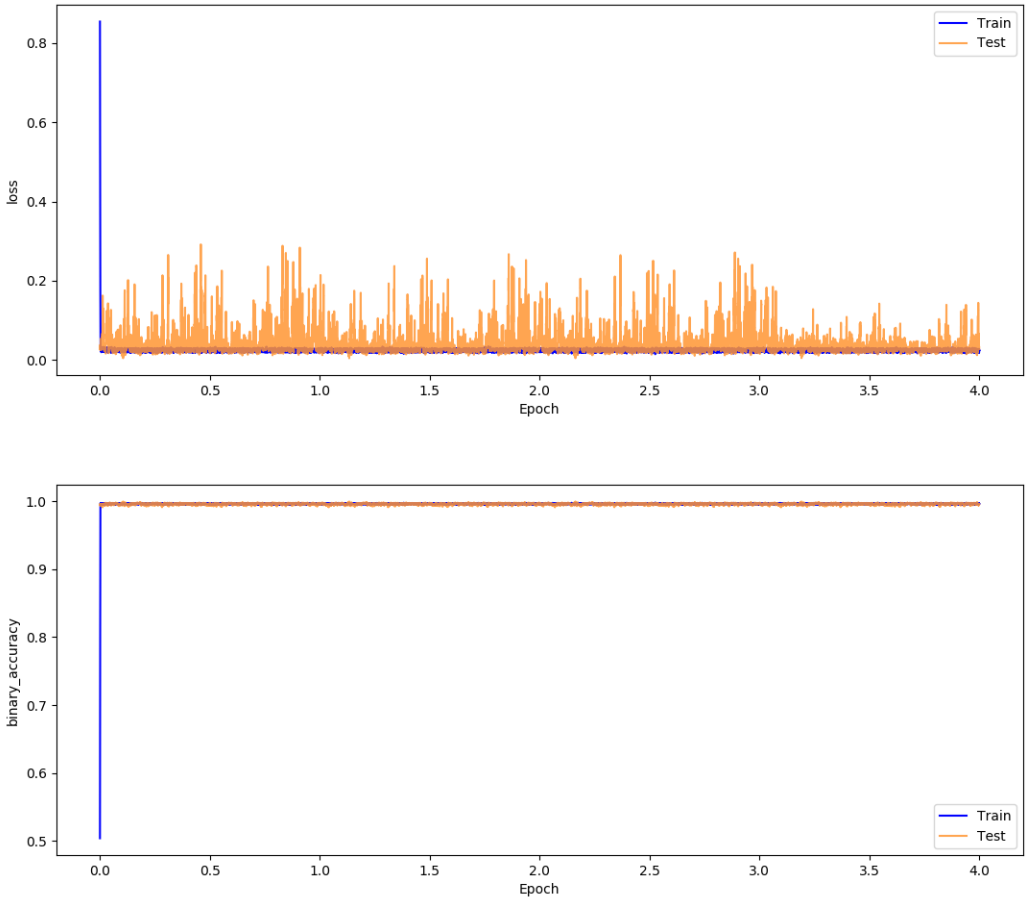


Σχήμα 10.2.6 – Πείραμα 6

Χρησιμοποιήθηκαν 50030 δείγματα εκπαίδευσης και 21920 δοκιμής .

10.3 Relative Locator version 2 - Μοντέλα

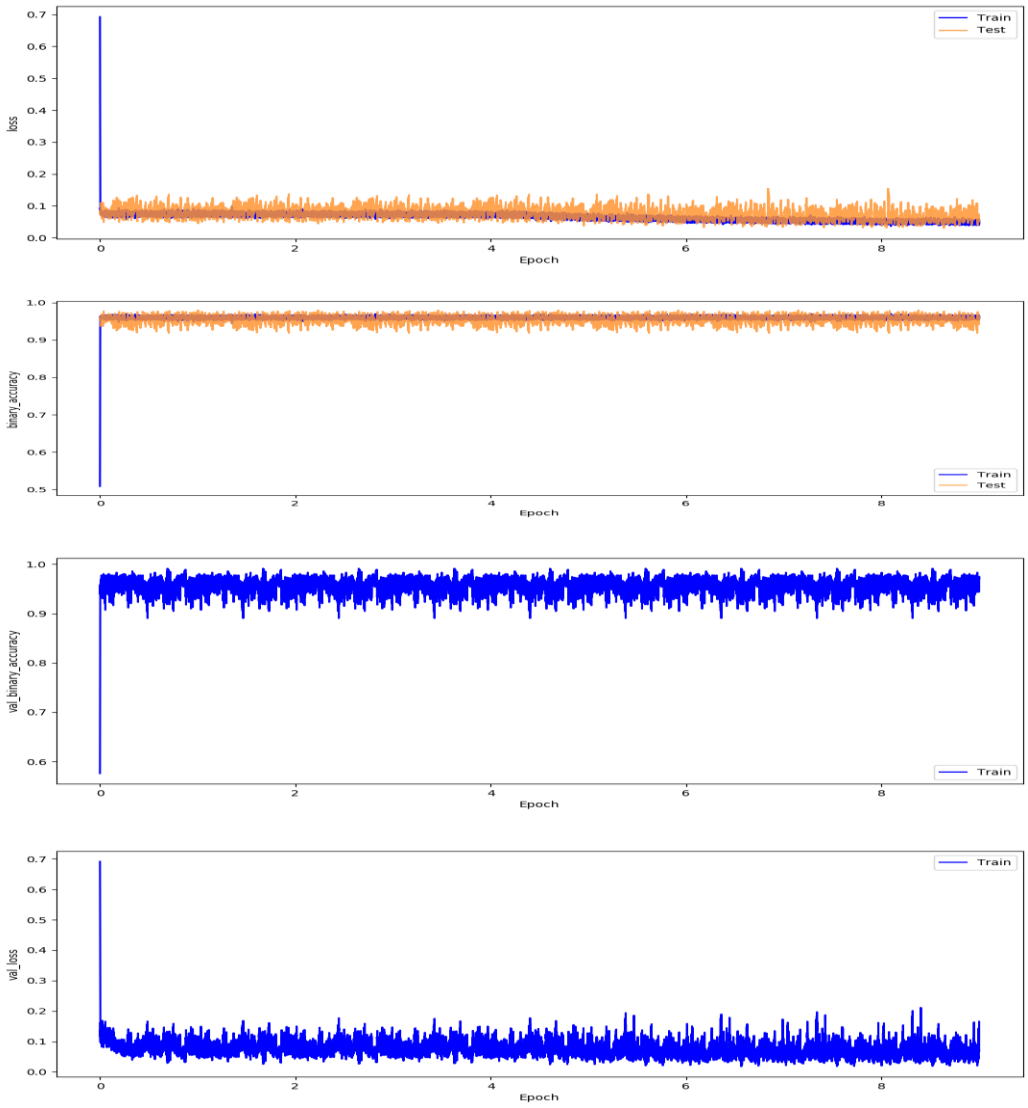
- $Architecture - ID = 3$
  $transform_{type} = melspectrogram,$
  $transform_{shape} = (1000, 256)$
  $Optimizer = SGD$ with:
  a) $learning_{rate} = 0.001$
  b) $decay = 0.000001$
  c) $momentum_{nesterov} = 0.9$
  $loss = Binary\ Cross\ Entropy$



Σχήμα 10.3.1 – Πείραμα 7

Μετά την τρίτη εποχή το μοντέλο ξεκινάει να γενικεύει περισσότερο αφού σταθεροποιείται το σφάλμα δοκιμής .

- $Architecture - ID = 6$
  $transform_{type} = melspectrogram,$
  $transform_{shape} = (1000, 256)$
  $Optimizer = Adam$ with:
  a) learning rate = 0.0001
  $loss = Binary\ Cross\ Entropy$



Σχήμα 10.3.2 – Πείραμα 8

Αυτό το μοντέλο εκπαιδεύτηκε με hot-vectors σε distribution-mode. Όλα τα προηγούμενα μοντέλα εφαρμόζουν multi-label ταξινόμηση με hot-vectors.

Παράδειγμα μορφής εξόδου

P.O.I. = [0.60, 14.33, 49.76, 89.04, 99.0]

Το διάνυσμα εξόδου για την εκπαίδευση, επικύρωση και δοκιμή του νευρωνικού δικτύου θα είναι:

$[0, 0.1482, 0.167, 0.25, 0.33, 0.5\ , 1\ , 0.33, 0.25, 0.2, 0.167, 0.1482, 0, 0, \dots]$

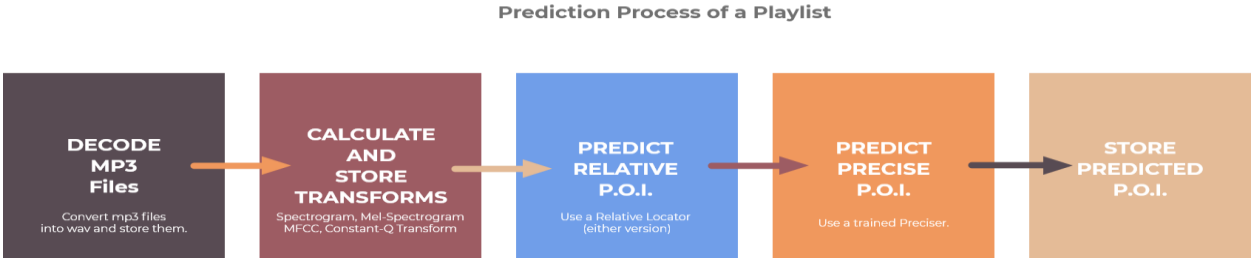
(σημείο ενδιαφέροντος που φαίνεται → 0.60)

ΚΕΦΑΛΑΙΟ 11

ΠΡΟΒΛΕΨΗ

11.1 Εισαγωγή

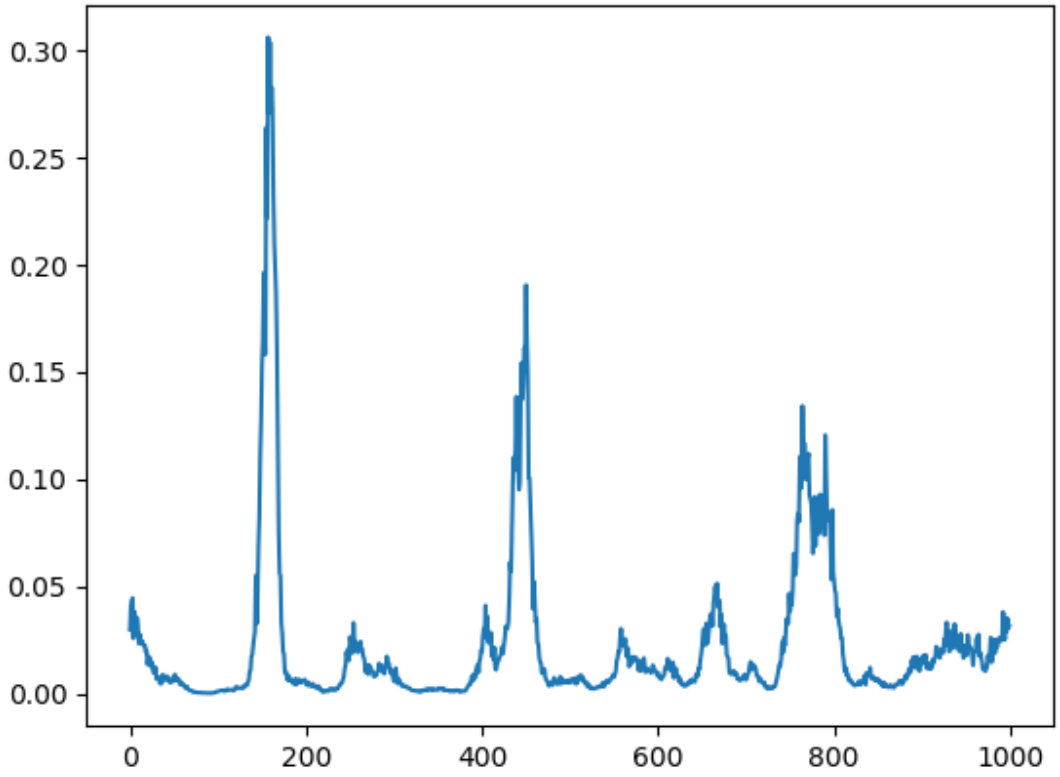
Η διαδικασία πρόβλεψης αποτελείται από μία γραμμή υπο-εργασιών με σκοπό την απόκτηση των σημείων ενδιαφέροντος αρχείων mp3. Τα βήματα που ακολουθούνται επιτυγχάνουν την αποκωδικοποίηση των αρχείων mp3, δημιουργώντας και τη δημιουργία όλων των πιθανών μετασχηματισμών. Έπειτα προβλέπονται σχετικά σημεία ενδιαφέροντος και σύμφωνα με αυτά προβλέπονται τα ακριβή σημεία ενδιαφέροντος, που τελικά αποθηκεύονται σε ένα αρχείο CSV.



Σχήμα 11.1.1 – Σύστημα Πρόβλεψης (Data Pipeline)

11.2 Επιλογή σχετικών σημείων ενδιαφέροντος

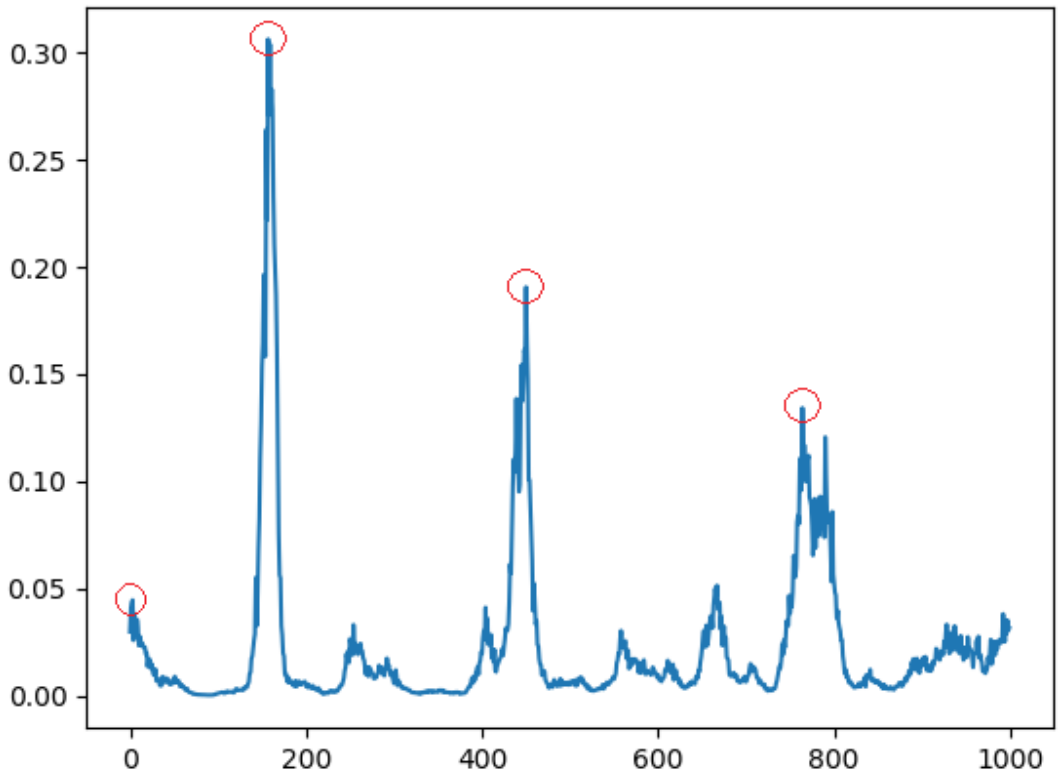
Από όλα τα πειράματα που έγιναν τα καλύτερα αποτελέσματα προβλέψεων δόθηκαν από το σύστημα που εκπαιδεύτηκε με διανύσματα εξόδου σε distribution mode. Ένα διάνυσμα πρόβλεψης μοιάζει ως εξής:



Σχήμα 11.2.1 – Διάνυσμα Πρόβλεψης

Πρόκειται για ένα διάνυσμα με 1000 στοιχεία. Κάθε στοιχείο ένα χρονικό διάστημα. Η τιμή του στοιχείου αντιπροσωπεύει την πιθανότητα το χρονικό διάστημα να περιέχει σημείο ενδιαφέροντος. Για συντομία το χρονικό διάστημα με σημείο ενδιαφέροντος θα καλείται ως σχετικό σημείο ενδιαφέροντος. Από το διάνυσμα πρόβλεψης πρέπει να εξαχθούν 4 σημεία ενδιαφέροντος.

Για αρχή θα γίνει έρευνα σχετικών σημείων όπου υπάρχουν κορυφές όπως φαίνεται και στο παραπάνω γράφημα. Υπάρχουν όμως πολλές κορυφές, το οποίο σημαίνει ότι θα απορριφθούν πολλά σχετικά σημεία. Οι αλγόριθμοι επιλογής σχετικών σημείων λαμβάνει υπόψη την ελάχιστη διάρκεια μίας ενότητας ενός κομματιού και ένα κατώφλι. Ο πρώτος εντοπίζει το σχετικό cue point μέσα στα πρώτα 10 δευτερόλεπτα του κομματιού. Έπειτα ο δεύτερος αλγόριθμος εντοπίζει τα σχετικά σημεία με τη μεγαλύτερη πιθανότητα σε ένα διάστημα 10 δευτερολέπτων και η πιθανότητα τους είναι πάνω από 5%.



Σχήμα 11.2.2 – Πρόβλεψη Σημείων Ενδιαφέροντος Τραγουδιού

Αναφορές - Βιβλιογραφία

[1] - https://en.wikipedia.org/wiki/Music
[2] - https://en.wikipedia.org/wiki/Sound
[3] - https://en.wikipedia.org/wiki/Pulse-code_modulation
[4] - https://en.wikipedia.org/wiki/Short-time_Fourier_transform
[5] - Speech Emotion Recognition from Spectrograms with Deep Convolutional Neural Network
[6] - https://en.wikipedia.org/wiki/Mel_scale
[7] - Theory-and-Applications-of-Digital-Speech-Processing
[8] - https://en.wikipedia.org/wiki/Mel-frequency_cepstrum
[9] - Calculation of a constant Q spectral transform
[10] - https://www.midi.org/
[11] - The Complete MIDI 1.0 Detailed Specification
[12] - Recent Advances in Convolutional Neural Networks
[13] -https://en.wikipedia.org/wiki/Convolutional_neural_network
[14] - Understanding Long Short-Term Memory Recurrent Neural Networks – a tutorial-like introduction
[15] - Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies
[16] - https://en.wikipedia.org/wiki/Long_short-term_memory
[17] - https://en.wikipedia.org/wiki/Gated_recurrent_unit
[18] - SDSS DR7 superclusters Principal component analysis
[19] - Face Recognition Machine Vision System Using Eigenfaces
[20] - RETHINKING DATA AUGMENTATION: SELF-SUPERVISION AND SELF-DISTILLATION
[21] - https://en.wikipedia.org/wiki/Pitch_(music)
[22] - http://blogs.zynaptiq.com/bernsee/time-pitch-overview/
[23] -https://en.wikipedia.org/wiki/Semitone