Σχολή Θετικών Επιστημών και Τεχνολογίας

Μεταπτυχιακή Εξειδίκευση στα Πληροφοριακά Συστήματα

Διπλωματική Εργασία

Κατηγοριοποίηση μουσικών κομματιών σε μουσικά είδη μέσω ανάλυσης ήχου με χρήση τεχνικών Βαθιάς Μάθησης

Παρασκευάς Παπαδόπουλος

Επιβλέπων καθηγητής: Δημήτρης Καστανιώτης

Πάτρα, Μάιος 2026

Κατηγοριοποίηση μουσικών κομματιών σε μουσικά είδη μέσω ανάλυσης ήχου με χρήση τεχνικών Βαθιάς Μάθησης

Παρασκευάς Παπαδόπουλος

Επιτροπή Επίβλεψης Διπλωματικής Εργασίας

Επιβλέπων Καθηγητής:
Δημήτρης Καστανιώτης
Μέλος ΣΕΠ, Ελληνικό Ανοιχτό Πανεπιστήμιο

Συν-Επιβλέπων Καθηγητής:
Γεώργιος Ρηγόπουλος
Μέλος ΣΕΠ, Ελληνικό Ανοιχτό Πανεπιστήμιο

Πάτρα, Μάιος 2026

Στους γονείς μου, Θεολόγο και Παναγιώτα.

Για την αγάπη και την στήριξή τους στην προσπάθειά μου.

Περίληψη

Η παρούσα Διπλωματική Εργασία εστιάζει στην αυτόματη ταξινόμηση μουσικών κομματιών σε μουσικά είδη μέσω τεχνικών Βαθιάς Μάθησης. Στόχος της μελέτης είναι ο έλεγχος της απόδοσης γνωστών αρχιτεκτονικών νευρωνικών δικτύων. Συγκεκριμένα, μελετάται η ικανότητά τους να αναγνωρίζουν πρότυπα που σχετίζονται με μουσικά είδη και να ταξινομούν μουσικά τραγούδια σύμφωνα με αυτά. Παράλληλα, επιδιώκεται σύγκριση των αρχιτεκτονικών ως προς την επίδοσή τους σε διαφορετικές συνθήκες εκπαίδευσης.

Η πειραματική διαδικασία βασίστηκε στο σύνολο δεδομένων GTZAN και οργανώθηκε σε συγκεκριμένες σειρές πειραμάτων. Οι αρχιτεκτονικές που δοκιμάστηκαν είναι οι MLP, CNN, RNN, LSTM, GRU καθώς και Transformer. Για την εκπαίδευση των μοντέλων Βαθιάς Μάθησης χρησιμοποιήθηκαν Mel Spectrograms και MFCCs, ενώ έγινε σύγκριση της απόδοσης κάθε αρχιτεκτονικής όταν εκπαιδεύεται με κάθε ένα από αυτά τα δεδομένα. Τα Mel Spectrograms και τα MFCCs εξήχθησαν μετά από τμηματοποίηση των δειγμάτων του GTZAN σε μικρότερα μέρη, ούτως ώστε να αυξηθεί το πλήθος των δειγμάτων. Επίσης έγινε σύγκριση της απόδοσης των μοντέλων για διαφορετικές συνθήκες εξαγωγής των χαρακτηριστικών, καθώς και για διαφορετικό αριθμό τμηματοποίησης των αρχικών δειγμάτων.

Η υλοποίηση των πειραμάτων έγινε με χρήση της γλώσσας προγραμματισμού Python. Η ανάπτυξη και η εκπαίδευση των μοντέλων κάθε αρχιτεκτονικής, βασίστηκε στη βιβλιοθήκη PyTorch. Περαιτέρω επεξεργασία των δεδομένων και παρουσίαση των διαγραμμάτων έγινε με χρήση βιβλιοθηκών όπως η NumPy, Matplotlib και TensorBoard.

Η αξιολόγηση της ικανότητας των μοντέλων να γενικεύουν, βασίστηκε σε καθιερωμένες μετρικές. Αρχικά παρουσιάζονται οι καμπύλες εκπαίδευσης κάθε μοντέλου, ενώ παρατίθενται και καμπύλες εκπαίδευσης όλων των μοντέλων σε κοινό διάγραμμα για κάθε σειρά πειραμάτων. Επίσης έγινε μέτρηση της πιστότητας του κάθε μοντέλου, υπολογίζοντάς την σε υποσύνολο του GTZAN το οποίο δε χρησιμοποιήθηκε για εκπαίδευση. Τέλος, κατασκευάστηκαν πίνακες σύγχυσης για κάθε μοντέλο.

Τα αποτελέσματα δείχνουν ότι η επιλογή της αναπαράστασης των ακουστικών χαρακτηριστικών, καθώς και ο τρόπος εξαγωγής τους, επηρεάζει σημαντικά την απόδοση κάθε αρχιτεκτονικής. Η εργασία συμβάλλει στην διαρκή ερευνητική προσπάθεια σύγκρισης παλαιότερων και νεότερων αρχιτεκτονικών Βαθιάς Μάθησης στον τομέα της ταξινόμησης

μουσικών ειδών. Παράλληλα, προσφέρει χρήσιμα συμπεράσματα για μελλοντικές κατευθύνσεις.

Λέξεις – Κλειδιά

Mel Spectrogram, MFCCs, Ταξινόμηση Μουσικών Ειδών, Βαθιά Μάθηση, PyTorch, Νευρωνικά δίκτυα.

Music Genre Classification using Deep Learning

Paraskevas Papadopoulos

Abstract

This thesis focuses on the automatic classification of music tracks into music genres using Deep Learning methods. The study's aim is to evaluate the performance of well-known neural network architectures. It particularly focuses on their ability to identify patterns related to music genres, in order to classify music songs. At the same time, we conduct a comparative analysis of the architectures with respect to their performance under various training conditions.

The experimental procedure was based on the GTZAN dataset and has been organized into specific experimental runs. The evaluated architectures include MLP, CNN, RNN, LSTM, GRU and Transformer. Our Deep Learning models were trained using Mel Spectrograms and MFCCs as input data. The mentioned input data were extracted after a segmentation of the GTZAN samples. The purpose of this procedure was to increase the size of the initial dataset. Finally, we compared the performance of each architecture when using different input data, applying different extraction conditions and using different segmentation strategies.

The experiments were implemented using Python. Model development and training were carried out using the PyTorch library. Additional data processing and visualization was performed using libraries such as NumPy, Matplotlib and TensorBoard.

The models' ability to perform was evaluated using established metrics. At first, we present the training curves of each model, along with comparative plots that include the training curves of all the architectures for each experiment run. Furthermore, we calculate the accuracy of each model using a subset of the GTZAN dataset. The subset was not used during training. Finally, we constructed confusion matrices for each model.

The results point out the fact that the choice of the acoustic features as well as the feature extraction methodology, play a significant role in the performance of each architecture. This work aims to contribute to the ongoing research on the comparison of neural network architectures. Both classic, as well as modern ones. Furthermore, it tries to provide useful insight on future research directions.

Keywords

Mel Spectrogram, MFCCs, Music Genre Classification, Deep Learning, PyTorch, Neural Networks.

Περιεχόμενα

Κατάλογος Εικόνων / Σχημάτων

Κατάλογος Πινάκων

Συντομογραφίες & Ακρωνύμια

ADC	Analog-to-digital converter
AI	Artificial intelligence
ANN	Artificial neural network
AST	Audio spectrogram transformer
CNN	Convolutional neural network
DAC	Digital-to-analog converter
DCT	Discrete cosine transform
DFT	Discrete Fourier transform
DL	Deep learning
DNN	Deep neural network
DTFT	Discrete time Fourier transform
FFT	Fast Fourier transform
GPU	Graphics processing unit
IDTFT	Inverse discrete-time Fourier transform
IFT	Inverse Fourier transform
GMM	Gaussian mixture model
GRU	Gated recurrent unit
KNN	K-Nearest neighbor
LSTM	Long short-term memory
MAE	Mean absolute error
MSE	Mean squared error
MFCC	Mel-Frequency Cepstral Coefficients
ML	Machine learning
MLP	Multi-layer perceptron
NLP	Natural language processing
ReLU	Rectified linear unit
RNN	Recurrent Neural Network
RMSE	Root mean square error
STFT	Short-time Fourier transform
SGD	Stochastic gradient descent
SVM	Support vector machines

ViT	Vision Transformer

1. Εισαγωγή

Η μουσική αποτελεί μια τέχνη άρρηκτα συνδεδεμένη με την πολιτισμική ιστορία του ανθρώπου. Θεωρείται μορφή επικοινωνίας με ιστορία αρκετά παλιά, εξίσου σημαντική με την ανθρώπινη γλώσσα. Μέχρι σήμερα, η ανθρώπινη καθημερινότητα είναι συνδεδεμένη με τη μουσική, καθώς κάθε άνθρωπος εκτίθεται σε αυτή με ποικίλες δραστηριότητες.

Τα μουσικά είδη, αποτελούν έναν από τους βασικότερους τρόπους που χρησιμοποιούνται για τη διάκριση της μουσικής. Μια τέχνη με τόσο μεγάλη ιστορία είναι λογικό να έχει να επιδείξει πολλά δείγματα και τα μουσικά είδη μας βοηθούν να «χαρτογραφούμε» το αχανές πεδίο της. Ειδικά στην ψηφιακή εποχή, με τεράστιες βάσεις δεδομένων που περιέχουν τραγούδια, η χρήση των μουσικών ειδών είναι καταλυτική για την αποδοτικότερη εύρεση των κατάλληλων μουσικών τραγουδιών από τους χρήστες. Στη σύγχρονη εποχή, όπου η τεχνολογία ηχογράφησης είναι πλέον προσιτή για το μέσο χρήστη, τα μουσικά είδη μπορούν βοηθούν τους συνθέτες να τακτοποιήσουν τη συλλογή τους, αλλά και να έχουν πρόσβαση σε μουσικά samples του είδους της αρεσκείας τους.

Σήμερα, με την κυριαρχία της Βαθιάς Μάθησης στην καθημερινότητά μας, η εφαρμογή της στην ταξινόμηση μουσικών ειδών παρουσιάζει ιδιαίτερο ερευνητικό ενδιαφέρον. Το ενδιαφέρον εστιάζεται στην ικανότητα των αρχιτεκτονικών αυτών να αναγνωρίζουν ηχητικά πρότυπα στα κομμάτια. Με όλο και περισσότερες πλατφόρμες μουσικής να χρησιμοποιούνται στο διαδίκτυο, η ταξινόμησή τους από μοντέλα Βαθιάς Μάθησης, είναι μια πολλά υποσχόμενη λύση σε προβλήματα που προκύπτουν από την χειροκίνητη ταξινόμηση.

1.1 Σκοπός της Διπλωματικής Εργασίας

Σκοπός της εν λόγω Διπλωματικής Εργασίας είναι η μελέτη της ικανότητας μερικών από τις βασικές αρχιτεκτονικές Βαθιάς Μάθησης, να ταξινομούν μουσικά τραγούδια σε μουσικά είδη. Έχει παρατηρηθεί ότι αρχιτεκτονικές όπως η CNN που χρησιμοποιείται στον τομέα του computer vision, αλλά και αναδρομικές αρχιτεκτονικές που χρησιμοποιούνται σε γλωσσικά μοντέλα, είναι ικανές να προσφέρουν εξαιρετικά αποτελέσματα και στον τομέα της ταξινόμησης μουσικών ειδών. Επιπροσθέτως, μια από τις πιο σύγχρονες αρχιτεκτονικές, αυτή του Transformer, που είναι συνδεδεμένη με κάποια από τα πιο προηγμένα γλωσσικά μοντέλα, φαίνεται ότι βρίσκει εφαρμογή στον συγκεκριμένο κλάδο.

Στόχος μας είναι να ερευνήσουμε την απόδοση και τη γενικότερη συμπεριφορά των μοντέλων των αρχιτεκτονικών αυτών στο συγκεκριμένο πρόβλημα.

Παράλληλα, η εργασία της ταξινόμησης των μουσικών κομματιών σε μουσικά είδη από τα μοντέλα Βαθιάς Μάθησης, εξαρτάται από πληθώρα παραγόντων. Τα μοντέλα συνήθως εκπαιδεύονται χρησιμοποιώντας οπτικοποιήσεις του ήχου. Κάποιες από τις πιο συνήθεις οπτικοποιήσεις για αυτόν το σκοπό είναι τα Mel Spectrograms και τα Mel-Frequency Cepstral Coefficients (MFCCs). Οι τρόποι που εξάγονται αυτές οι οπτικοποιήσεις, επίσης ποικίλουν. Στόχος μας είναι να μελετήσουμε τους βασικούς τρόπους που εξάγονται αυτά τα ακουστικά χαρακτηριστικά και το πως μπορεί να επηρεαστεί η απόδοση της κάθε αρχιτεκτονικής.

1.2 Ιστορική αναδρομή στην κατηγοριοποίηση μουσικών κομματιών ανά είδος με χρήση τεχνικών Βαθιάς Μάθησης

Η κατηγοριοποίηση μουσικών κομματιών ανά είδος έχει εξελιχθεί αρκετά τις τελευταίες δεκαετίες. Στα πρώιμα στάδιά της αποτελούσε μια διαδικασία που για την περάτωσή της βασιζόμασταν σε απλές χειροκίνητες μεθόδους. Παρ όλα αυτά, πλέον αυτοματοποιείται από μοντέλα Βαθιάς Μάθησης με υψηλή ακρίβεια.

Η μέθοδος αυτή έχει τις ιστορικές τις ρίζες στα συστήματα αναγνώρισης φωνής. Τα συστήματα αυτά χρησιμοποιούσαν οπτικοποιήσεις του ήχου όπως τα MFCCs με σκοπό να αναλύουν ηχητικά σήματα. Τα MFCCs με τη σειρά τους έχουν την ιστορική τους βάση στη μελέτη σεισμικών δονήσεων, με τον καιρό όμως βρήκαν εφαρμογή στην ανάλυση του ήχου. Στα τέλη της δεκαετίας του 90, ερευνητές χρησιμοποίησαν τεχνικές Machine Learning όπως Support Vector Machines μαζί με οπτικοποιήσεις του ήχου, για να ερευνήσουν την δυνατότητα αυτόματης κατηγοριοποίησης. Η έρευνα των Tzanetakis και Cook (2002) είναι μια από τις πρώτες προσπάθειες προς μια τέτοια κατεύθυνση, ενώ ακολουθήσαν κι άλλες.

Η ραγδαία ανάπτυξη των μοντέλων Βαθιάς Μάθησης τη δεκαετία του 2010, έφερε στο προσκήνιο την έρευνα για τη χρήση της τεχνολογίας αυτής για τον σκοπό της ταξινόμησης μουσικών ειδών. Ειδικά η ανάπτυξη της αρχιτεκτονικής CNN ήταν καταλυτική, καθώς η χρήση μοντέλων που χρησιμοποιούνταν κυρίως για computer vision, στον τομέα της ταξινόμησης μουσικών ειδών βελτίωσε κατά πολύ την ακρίβεια των μοντέλων.

Σήμερα, η έρευνα ακόμα συνεχίζεται, καθώς υβριδικά μοντέλα που μπορούν να χρησιμοποιούν πάνω από μια αρχιτεκτονικές, χρησιμοποιούνται γενικότερα για την

ταξινόμηση του ήχου, συμπεριλαμβανομένων των μουσικών τραγουδιών σε μουσικά είδη. Οι πιθανότητες συνδυασμών αρχιτεκτονικών και τρόπων λειτουργίας τους είναι μεγάλες σε πλήθος, συνεπώς η έρευνα παραμένει επίκαιρη. Παράλληλα, καθώς ο κλάδος είναι σε άνθιση και το ενδεχόμενο ανάπτυξης νέων αρχιτεκτονικών παραμένει ανοιχτό, η εφαρμογή τους στον εν λόγω κλάδο είναι υποσχόμενη. Κάτι αντίστοιχο έγινε με την πιο πρόσφατη αρχιτεκτονική του Transformer η οποία δοκιμάστηκε και βρήκε εφαρμογή και στην κατηγοριοποίηση ήχου, ενώ αρχικά είχε αναπτυχθεί για γλωσσικά μοντέλα.

1.3 Δομή της Διπλωματικής Εργασίας

Η δομή της Διπλωματικής Εργασίας αποσκοπεί στη σταδιακή παρουσίαση της μουσικής ταξινόμησης και του τρόπου με τον οποίο οι αρχιτεκτονικές Βαθιάς Μάθησης επιλύουν το συγκεκριμένο ζήτημα.

Το πρώτο κεφάλαιο, εισάγει τον αναγνώστη στο ζήτημα της ταξινόμησης μουσικών κομματιών σε μουσικά είδη. Παράλληλα τίθενται οι στόχοι της Διπλωματικής Εργασίας, ενώ γίνεται και μια ιστορική αναδρομή. Στην αναδρομή, πέρα από την εξέλιξη της τεχνολογίας, αναδεικνύεται το σύγχρονο τοπίο στον τομέα της αυτόματης ταξινόμησης, προσφέροντας το απαραίτητο εισαγωγικό πλαίσιο για την κατανόηση της παρούσας μελέτης

Στο δεύτερο κεφάλαιο παρουσιάζεται το θεωρητικό υπόβαθρο και οι βασικές έννοιες που χρησιμοποιούνται στην έρευνά μας. Το θεωρητικό υπόβαθρο ουσιαστικά χωρίζεται σε τρία μέρη. Αρχικά ορίζεται η έννοια του μουσικού είδους και παρατίθενται θεωρητικές προσεγγίσεις πάνω στο ζήτημα. Εφόσον καλούμαστε να ταξινομήσουμε σε μουσικά είδη, βλέπουμε τον τρόπο που αυτά διαχωρίζονται. Το δεύτερο μέρος επικεντρώνεται στα ακουστικά χαρακτηριστικά και τον τρόπο που αυτά ψηφιοποιούνται. Ο άνθρωπος αφουγκράζεται ακουστικά χαρακτηριστικά που τον βοηθούν να ταξινομήσει τα είδη. Αναλύουμε λοιπόν τα χαρακτηριστικά αυτά και βλέπουμε τις μεθόδους με τις οποίες αυτά μετασχηματίζονται σε ψηφιακή μορφή. Αυτό είναι απαραίτητο, καθώς αυτά τα χαρακτηριστικά θα αποτελέσουν τη βάση για την ταξινόμηση από υπολογιστικά μοντέλα. Τέλος, στο τρίτο μέρος αναλύουμε την απαραίτητη θεωρία πίσω από τα μοντέλα Βαθιάς Μάθησης, τα οποία είναι και τα εργαλεία που θα χρησιμοποιήσουμε για την ταξινόμηση.

Το τρίτο κεφάλαιο περιγράφει την απαραίτητη προετοιμασία που πραγματοποιούμε πριν την εκτέλεση των πειραμάτων. Η σύνθεση των μοντέλων, η επεξεργασία των δεδομένων

καθώς και τεχνολογίες που χρησιμοποιούμε για την έρευνά μας, αναφέρονται σε αυτό το σημείο.

Το τέταρτο κεφάλαιο περιγράφει τη στρατηγική και την οργάνωση των πειραμάτων μας. Αναφέρεται λεπτομερώς η σειρά τους και τα ερωτήματα που κάθε σειρά πειραμάτων καλείται να απαντήσει. Επιπροσθέτως, αναλύεται η διαδικασία των πειραμάτων καθώς και οι μέθοδοι και οι μετρικές που χρησιμοποιούμε.

Στο πέμπτο κεφάλαιο παραθέτουμε όλα τα αποτελέσματα των πειραμάτων. Το κεφάλαιο είναι οργανωμένο έτσι ώστε να παρουσιάζει τα αποτελέσματα σύμφωνα με την οργάνωση που περιγράφεται στο προηγούμενο κεφάλαιο. Σύντομα σχόλια για τη διαδικασία των πειραμάτων αλλά και τα αποτελέσματα, περιλαμβάνονται μαζί με αυτά.

Στο έκτο κεφάλαιο, είναι συγκεντρωμένα όλα τα συμπεράσματα που προκύπτουν από την αξιολόγηση των αποτελεσμάτων. Παρουσιάζονται τα πορίσματα για τα ερωτήματα που τέθηκαν στο τέταρτο κεφάλαιο, καθώς και μια αναφορά για την αξιολόγηση κάθε αρχιτεκτονικής.

2. Θεωρητικό υπόβαθρο

Στο κεφάλαιο αυτό θα εξεταστούν θεμελιώδεις έννοιες που σχετίζονται με τα μουσικά είδη, την επεξεργασία του ήχου και τα νευρωνικά δίκτυα. Παράλληλα, θα τεθούν οι απαραίτητες βάσεις για την κατανόηση των παραπάνω εννοιών.

Σκοπός της έρευνας είναι η μελέτη του τρόπου με τον οποίο ένα υπολογιστικό σύστημα «αντιλαμβάνεται» το μουσικό είδος από ένα αρχείο ήχου. Επομένως, κρίνεται σκόπιμο αρχικά να κατανοήσουμε το τι ορίζουμε ως μουσικό είδος και το πως τα είδη διαφέρουν μεταξύ τους. Δεδομένου ότι θα μελετήσουμε τη μέθοδο με την οποία ο υπολογιστής «αντιλαμβάνεται» τη μουσική και τον ήχο, είναι καλό πρώτα να κατανοήσουμε το πως ο άνθρωπος αντιλαμβάνεται τον ήχο. Κατόπιν, μπορούμε να δούμε πως μέσω υπολογιστικών συστημάτων ψηφιοποιούμε τα αρχεία ήχου και κατόπιν πως εξάγουμε χαρακτηριστικά που είναι ιδιαίτερα για κάθε μουσικό είδος.

Αυτά τα χαρακτηριστικά θα δίδονται ως δεδομένα στα νευρωνικά δίκτυα, τα οποία τα επεξεργάζονται ώστε να κατηγοριοποιήσουν το αρχείο ήχου σε είδος μουσικής. Είναι λοιπόν σημαντικό να δούμε τις βασικές αρχές λειτουργίας των νευρωνικών δικτύων, καθώς και μερικές βασικές αρχιτεκτονικές τους.

2.1 Μουσικά είδη

Η μουσική είναι μια μορφή επικοινωνίας που αποτυπώνει διαφορετικά συναισθήματα και αισθήματα (Davies, 2010). Κατά καιρούς, η μουσική έχει χρησιμοποιηθεί για την μεταβολή της διάθεσης, την έκφραση συναισθημάτων και τη ρύθμιση της συναισθηματικής κατάστασης (Chełkowska-Zacharewicz & Paliga, 2020). Τα μουσικά είδη είναι ένας πολύ συχνός και ενστικτώδης τρόπος που ο μέσος άνθρωπος επιλέγει για να ταξινομήσει τη μουσική (Bainbridge et al., 2003). Ενδιαφέρον όμως παρουσιάζει ο τρόπος και τα κριτήρια τα οποία οδηγούν στις διακρίσεις των ειδών. Ένα βασικό κριτήριο για τη διάκριση αυτή είναι, τα κοινά χαρακτηριστικά που φαίνεται να έχουν τα στοιχεία της κάθε ομάδας. Ένα μουσικό είδος χαρακτηρίζεται από τα κοινά χαρακτηριστικά (πχ ενορχήστρωση, ρυθμικά μοτίβα, αρμονικό περιεχόμενο) που υπάρχουν στα κομμάτια που το απαρτίζουν (Tzanetakis & Cook, 2002).

2.1.1 Ορισμός και έννοια των μουσικών ειδών

Ο Fabbri (1981), ορίζει το μουσικό είδος ως ένα σύνολο από μουσικά γεγονότα των οποίων η εξέλιξη διέπεται από ένα συγκεκριμένο πλαίσιο κοινωνικά αποδεκτών κανόνων. Ο ίδιος, χωρίζει τους κανόνες αυτούς σε πέντε κατηγορίες: τυπικοί και τεχνικοί κανόνες, σημειωτικοί κανόνες, κανόνες συμπεριφοράς, κοινωνικοί και ιδεολογικοί κανόνες, οικονομικοί και νομικοί κανόνες. Γενικά, ο όρος «μουσικό είδος» (genre) συχνά τείνει να συγχέεται στην καθομιλουμένη με άλλες συναφείς έννοιες, όπως το «ύφος» ή το «στυλ». Αυτή η εναλλακτική χρήση είναι ιδιαίτερα εμφανής στην περίπτωση των όρων «μουσικό είδος» και «μουσικό στυλ». Ωστόσο, στην επιστημονική και ερευνητική κοινότητα, είναι κρίσιμο να επισημανθεί ότι οι δύο αυτές έννοιες δεν ταυτίζονται εννοιολογικά. Η επιστημονική προσέγγιση απαιτεί τον σαφή διαχωρισμό τους, καθώς το «μουσικό είδος» ορίζεται ως μια διακριτή έννοια που δεν είναι συνώνυμη με το «μουσικό στυλ». Το πρώτο είναι ένας τύπος μουσικής που για τον οποιοδήποτε λόγο, σκοπό ή κριτήριο αναγνωρίζεται από μια κοινότητα, ενώ το δεύτερο αποτελεί μια επαναλαμβανόμενη διαρρύθμιση μουσικών χαρακτηριστικών που είναι αντιπροσωπευτικά για ένα άτομο (εκτελεστή ή συνθέτη), ένα συγκεκριμένο γκρουπ μουσικών ή ακόμα και μια χρονική περίοδο (Mckay & Fujinaga, 2006).

Σε κάθε μουσικό είδος, υπάρχουν συγκεκριμένα μοτίβα που μπορεί κανείς να παρατηρήσει. Το κάθε μουσικό είδος, χαρακτηρίζεται από στατιστικές ιδιότητες που σχετίζονται με την ενορχήστρωση, τη ρυθμική του δομή και τη μορφολογία (φόρμα) των τραγουδιών που το απαρτίζουν (Tzanetakis et al., 2001). Το tempo για παράδειγμα, είναι ένα τέτοιο χαρακτηριστικό το οποίο σύμφωνα με έρευνες φαίνεται πως σχετίζεται άμεσα με την ταξινόμηση μουσικών ειδών (Gouyon & Dixon, 2004).

Η διάκριση των ειδών όμως μπορεί να εξαρτάται και από παράγοντες που δεν εξαρτώνται από μουσικά χαρακτηριστικά. Όπως σημειώνουν οι Neumayer και Rauber (2007), στα στοιχεία που διαμορφώνουν το μουσικό είδος δεν περιλαμβάνονται μόνο τα μουσικά χαρακτηριστικά του, αλλά και οι στίχοι του. Ένας ακόμη μη μουσικός παράγοντας που επηρεάζει άμεσα την κατηγοριοποίηση, φαίνεται πως είναι το συναίσθημα. Σύμφωνα με τους Kim et al. (2010) η μουσική αναφέρεται ως η γλώσσα του συναισθήματος, επομένως είναι φυσικό για τους ανθρώπους να την κατηγοριοποιούν επηρεασμένοι από τα συναισθήματα που προκαλεί. Τα συναισθήματα που προκαλούνται, προφανώς ποικίλλουν από ακροατή σε ακροατή.

Η ταξινόμηση των μουσικών ειδών από τον άνθρωπο αποτελεί μια σύνθετη γνωστική διαδικασία, η οποία έχει διερευνηθεί και στο πλαίσιο της ανθρώπινης ψυχολογίας. Η Deliège (2001), έχει προτείνει τη θεωρία των "cues", σύμφωνα με την οποία οι άνθρωποι απομονώνουν μουσικά χαρακτηριστικά των κομματιών και τα οργανώνουν σε "cues". Η ίδια σημειώνει πως χρήσει των "cues", οι άνθρωποι κρίνουν τη μουσική ομοιότητα, καθώς επίσης σχηματίζουν «αποτυπώματα» που βοηθούν στην αντίληψη της μουσικής δομής και την εκτίμηση ομοιοτήτων που θα συμβάλουν στην κατηγοριοποίηση.

2.1.2 Ανάγκη και λειτουργία των μουσικών ειδών

Σύμφωνα με τους Pachet και Cazaly (2000), καθώς η διανομή της μουσικής γίνεται προοδευτικά όλο και περισσότερο μέσω ηλεκτρονικών μέσων, η ανάγκη για λεπτομερέστερα metadata γίνεται περισσότερο επιτακτική. Οι ίδιοι σημειώνουν πως αυτά τα metadata χρειάζονται, ώστε οι υπηρεσίες μουσικής διανομής να ανταπεξέλθουν τόσο στο τεράστιο μέγεθος των μουσικών καταλόγων, όσο και στην ανάγκη των χρηστών να έχουν πρόσβαση σε μουσικούς τίτλους βάση ομοιότητας. Ο Aigrain, (1999) επισημαίνει πως στην εποχή των ηλεκτρονικών music-on-demand υπηρεσιών η ύπαρξη metadata (συμπεριλαμβανομένου του είδους) είναι απαραίτητη, καθώς μπορεί να βοηθήσει στις μηχανές αναζήτησης των υπηρεσιών αυτών. Ενδιαφέρον επίσης, παρουσιάζει η έρευνα των Lee και Downie (2004) πάνω στον τρόπο και τη συμπεριφορά των ανθρώπων όταν αναζητούν μουσική. Σε έρευνά τους που συμπεριλάμβανε 427 ανθρώπους, παρατηρήθηκε πως το 62,7% αντέδρασε θετικά στην αναζήτηση μέσω μουσικού είδους με την δεύτερη επιλογή να είναι το «παρόμοιοι καλλιτέχνες», όπου η θετική απόκριση ήταν της τάξης του 59,3% .

Η ανάγκη για την ύπαρξη μουσικών ειδών, φαίνεται και από τον τρόπο που αυτά μπορούν να χρησιμοποιηθούν στην έρευνα. Αυτό καθίσταται ενδιαφέρον ειδικά όταν υπάρχει σύγκριση με χρήση αυτόματων classifiers μέσω σχετικού λογισμικού. Η μελέτη κοινών χαρακτηριστικών που εμφανίζονται στα μουσικά είδη μπορεί να συμβάλλει σε κοινωνιολογικές και ψυχολογικές έρευνες σχετικά με το πως οι άνθρωποι κατασκευάζουν την έννοια της μουσικής ομοιότητας, καθώς και πως αυτή συγκρίνεται με την «αντικειμενική» αλήθεια που προκύπτει από αυτόματους classifiers (Mckay & Fujinaga, 2005).

Η ύπαρξη μουσικών ειδών σε μια εποχή που είναι εφικτό ο διαχωρισμός να γίνει από λογισμικό, φαίνεται και σε απλά καθημερινά παραδείγματα. Τα συστήματα προτάσεων

μουσικής μπορούν να χρησιμοποιηθούν για να φιλτράρουν τεράστιες μουσικές βάσεις δεδομένων, τόσο γνωστής όσο και άγνωστης μουσικής στους χρήστες, καθώς και να τους προτείνουν άγνωστα κομμάτια με βάση τα είδη που είναι γνωστό ότι τους αρέσουν (Mckay & Fujinaga, 2005). Οι Bainbridge et al. (2003), επισημαίνουν το πόσο επίκαιρη παραμένει η ύπαρξη των ειδών, καθώς στην ψηφιακή εποχή που η πρόσβαση στην πληροφορία είναι πιο άμεση, οι χρήστες αναζητούν συνεχώς τρόπους να οργανώσουν την μουσική τους ώστε να έχουν πρόσβαση σε αυτή ανάλογα τη διάθεση που έχουν. Άλλη βοηθητική χρήση των μουσικών ειδών, αποτελεί το παράδειγμα λογισμικού για ηχογράφηση μουσικής, όπου χρησιμοποιούνται κατά κόρον samples. Ένα λογισμικό, θα μπορούσε να παρέχει την πληροφορία του είδους μουσικής του κάθε sample για διευκόλυνση του χρήστη (Wold et al., 1996).

Η σημασία των μουσικών ειδών διακρίνεται και στο αντίκτυπο που έχει σε πολιτισμικά και κοινωνικά θέματα, καθώς φαίνεται ότι μπορεί να συμβάλλει στη δημιουργία κοινωνικών ομάδων και γενικότερα στην κοινωνικοποίηση. Πολλά άτομα ταυτίζονται με το μουσικό είδος της προτίμησής τους, σε βαθμό που αυτό αντικατοπτρίζεται σε συμπεριφορές, όπως στον τρόπο που ντύνονται, καθώς εύκολα κανείς θα ξεχώριζε έναν λάτρη ακραίου metal από έναν λάτρη της rap (Mckay & Fujinaga, 2006). Οι άνθρωποι λοιπόν, παρατηρείται πως συνδέονται με τα είδη που τους αρέσουν σε βαθιά κοινωνικό επίπεδο. Έρευνα των North και Hargreaves (1997) σε 50 άτομα, έδειξε πως η πιθανότητα να αρέσει ένα τραγούδι σε κάποιον ακροατή, επηρεάζεται περισσότερο από το εάν το άτομο είναι λάτρης του είδους κομματιού, παρά από το αν το ίδιο το κομμάτι αρέσει στο άτομο.

2.1.3 Θεωρητικά ζητήματα και προκλήσεις

Ο εγγενής χαρακτήρας της μουσικής αποτελεί εμπόδιο για τον διαχωρισμό της σε είδη. Σύμφωνα με τον Foote (1999), η μουσική είναι μια μεγάλη και εξαιρετικά μεταβλητή (συνεπώς απαιτητική), κατηγορία ήχου.

Η κατηγοριοποίηση με βάση το είδος, μπορεί να είναι πολύ δύσκολη τόσο για τους ανθρώπους όσο και για τους υπολογιστές, καθώς πολύ σπάνια υπάρχουν ακριβή, σαφή και συνεπή κριτήρια που να καθορίζουν τα χαρακτηριστικά κάθε κατηγορίας (Mckay & Fujinaga, 2005). Οι ακροατές, συχνά μπορεί να ταξινομούν ηχογραφήσεις διαφορετικά μεταξύ τους, καθώς πολύ λίγα είδη ορίζονται συγκεκριμένα, ενώ συχνά παρατηρούνται «επικαλύψεις» μεταξύ των μουσικών ειδών (Mckay & Fujinaga, 2006). Σε έρευνα των Lippens et al. (2004), όπου 27 ακροατές κλήθηκαν να ταξινομήσουν 160 τραγούδια σε 6

πιθανά είδη, η σύγκλιση έφτανε το ποσοστό του 76%. Αξίζει να σημειωθεί όμως ότι στη συγκεκριμένη έρευνα το ένα είδος ήταν «λοιπά», κάτι που από μόνο του μπορεί να προκαλέσει ασάφεια (Mckay & Fujinaga, 2006).

Η ταξινόμηση της μουσικής μπορεί να αλλάζει ανά τα χρόνια, καθώς η αγορά της μουσικής αλλάζει. Οι Pachet και Cazaly (2000) σημειώνουν πως όταν η πώληση της μουσικής γινόταν σε φυσική μορφή, η ταξινόμησή της πραγματοποιούνταν με τρόπο που θα ευνοούσε κάποιον πελάτη που περιηγείται σε ένα κατάστημα δίσκων. Οι ίδιοι αναφέρουν ότι αυτός ο τρόπος δεν είναι πια επίκαιρος. Παράλληλα, το μέσο διανομής της μουσικής, μπορεί να προκαλέσει αντιφάσεις όσον αφορά το είδος που αυτή κατηγοριοποιείται. Όταν η μουσική πωλούνταν σε δίσκους, μια κατηγορία είδους μπορεί να χρησιμοποιούνταν για όλα τα κομμάτια του, ενώ στην τρέχουσα εποχή που κομμάτια κυκλοφορούν μεμονωμένα, αυτός ο τρόπος κατηγοριοποίησης δεν είναι επίκαιρος (Pachet & Cazaly, 2000). Συνήθειες της ταξινόμησης σε είδη από παλαιότερα χρόνια μπορεί γενικά στη σημερινή εποχή να μην ευνοούν τους χρήστες. Οι Schuller et al. (2010) σημειώνουν πως παλαιότερα το μουσικό είδος συνδεόταν άμεσα με τον καλλιτέχνη, κάτι που σήμερα μπορεί να προκαλέσει ασάφεια στην ταξινόμηση μεμονωμένων τραγουδιών.

Οι Pachet & Cazaly (2000) αναφέρουν πως η δυσκολία στην ταξινόμηση των ειδών, έχει πολλές φορές σημασιολογικό και εννοιολογικό χαρακτήρα, διακρίνοντας συγκεκριμένες περιπτώσεις όπως πχ «γενεαλογικής» φύσης (για παράδειγμα η disco μπορεί να είναι υποσύνολο της pop) ή και «ιστορικής» περιόδου (για παράδειγμα η μπαρόκ μουσική ως είδος, υπάγεται στην κλασσική μουσική). Στις προκλήσεις της χρήσης των μουσικών ειδών, προστίθεται και το γεγονός ότι μουσικά είδη προστίθενται συνεχώς, ενώ παράλληλα είναι πιθανόν η αντίληψη των υπαρχόντων ειδών να μεταβάλλεται ανά τα χρόνια (Mckay & Fujinaga, 2006). Ο Brackett (2016) έχει μελετήσει εκτενώς την εξέλιξη των ειδών ανά τις δεκαετίες. Ένα χαρακτηριστικό παράδειγμα που παρουσιάζει είναι η περίπτωση του “rhythm and blues” (R&B) το οποίο σαν είδος αρχικά ξεπήδησε από τις soul/blues καταβολές τις αφροαμερικανικής κοινότητας των δεκαετιών του 60-70, όμως μετά τη δεκαετία του 90 και το γεγονός ότι οι DJ χρησιμοποιούσαν δίσκους από αυτό το είδος, κατέληξε να παραπέμπει περισσότερο σε ηλεκτρονική μουσική.

Παρ’ όλες τις προκλήσεις στην αναγνώριση των ειδών, υπάρχουν έρευνες που δείχνουν ότι οι άνθρωποι μπορούν να είναι αρκετά ικανοί στην αναγνώρισή τους. Οι Gjerdingen και Perrott (2008), παρατήρησαν ότι ακροατές μπορούν να διακρίνουν το είδος ενός τραγουδιού

ακόμα και σε διάστημα 250 ms. Αυτό το διάστημα, σημειώνουν, δεν είναι αρκετό ώστε να παρουσιαστεί εκτενώς η αρμονία, η μελωδία ή ο ρυθμός του τραγουδιού. Από ό,τι φαίνεται όμως, αυτά τα θεωρητικά χαρακτηριστικά δεν είναι απαραίτητα για τη διάκριση του είδους. Σύμφωνα με τους Martin et al. (1998), ακόμα και ακροατές που δεν είναι εκπαιδευμένοι μουσικοί ώστε να διακρίνουν θεωρητικά χαρακτηριστικά, αναγνωρίζουν επιτυχώς το είδος της μουσικής.

Προκλήσεις όμως, προκύπτουν ακόμα και στην χρήση λογισμικού για την αναγνώριση του μουσικού είδους. Στην ψηφιακή εποχή, η κατηγοριοποίηση μπορεί να γίνεται αυτόματα αλλά και χειροκίνητα. Στη δεύτερη περίπτωση, υπάρχει κίνδυνος η κατηγοριοποίηση να είναι ασαφής ή με είδη που δεν είναι ευρέως αποδεκτά (Schuller et al., 2010). Ζητήματα όμως, μπορεί να προκύψουν και στην έρευνα. Η χρήση νευρωνικών δικτύων για την αναγνώριση του είδους, δεν μας επιτρέπει να ξεχωρίσουμε ποια ακριβώς χαρακτηριστικά του δείγματος είναι αυτά που το χαρακτηρίζουν και είναι σημαντικά, καθώς δεν είναι εφικτό να «κοιτάξουμε εντός» του δικτύου (Wold et al., 1996).

2.1.4 Παραδείγματα μουσικών ειδών

Όπως αναλύθηκε στο παρόν κεφάλαιο, η διάκριση των μουσικών ειδών είναι ένα σύνθετο ζήτημα, καθώς πρόκειται για μια διαδικασία που δεν είναι ούτε απλή ούτε πλήρως αντικειμενική. Παρ' όλα αυτά, στην έρευνα είναι πολλές φορές η αναγκαία η διάκριση σε συγκεκριμένα είδη. Για τις ανάγκες τις έρευνάς μας λοιπόν, θα διακρίνουμε τα είδη όπως αυτά παρουσιάζονται στο GTZAN dataset το οποίο αποτελεί γνωστό σύνολο ηχητικών αρχείων. Το εν λόγω dataset περιλαμβάνει 10 είδη, ενώ είναι ένα dataset που χρησιμοποιείται ευρέως από την ακαδημαϊκή κοινότητα, ενώ έχει φτάσει να θεωρείται ορόσημο σε έρευνες ταξινόμησης μουσικών ειδών (W. Xu, 2024). Οι Tzanetakis και Cook (2002) λοιπόν, στο dataset διακρίνουν τα μουσικά είδη σε "classical", "country", "disco", "hip hop", "jazz", "rock", "blues", "reggae", "pop", "metal". Οι ίδιοι υποδιαιρούν κάποια από αυτά τα είδη σε υποκατηγορίες, πχ η jazz υποδιαιρείται σε "big band", "cool", "fusion", "piano", "quartet", "swing". Παρατηρούμε λοιπόν αυτό που λέγαμε αρχικά, ότι η διάκριση δεν είναι εύκολη. Γενικότερα πάντως, όταν το GTZAN χρησιμοποιείται στην έρευνα, παρατηρούμε ότι χρησιμοποιούνται τα 10 προαναφερθέντα βασικά είδη ως παραδείγματα για την ταξινόμηση των ειδών.

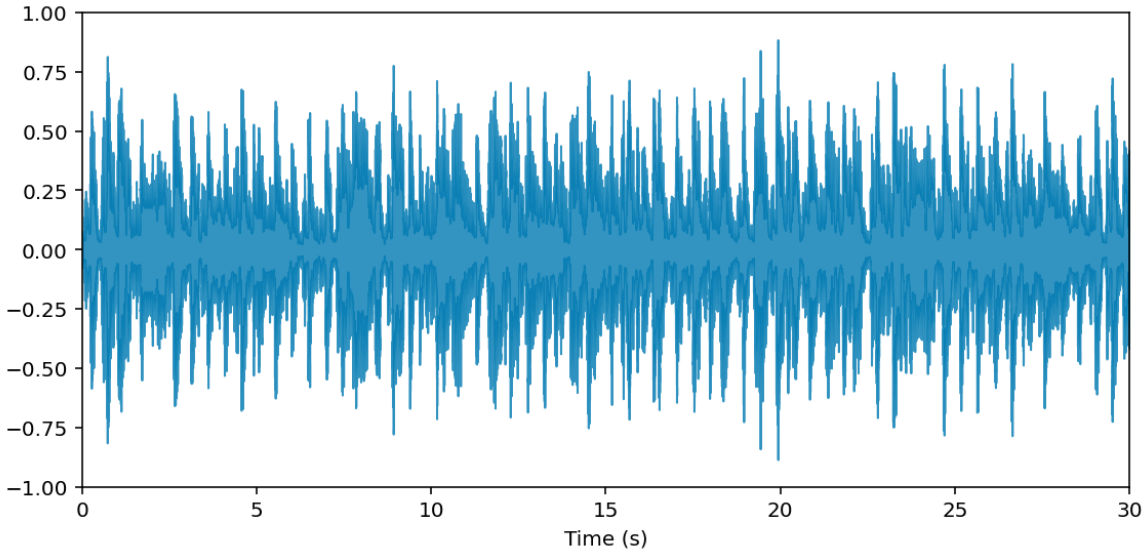
2.2 Χαρακτηριστικά ήχου και ψηφιακό σήμα

Η επιτυχής κατηγοριοποίηση μουσικών ειδών μέσω μεθόδων βαθιάς μάθησης βασίζεται κατά πολύ στην ποιότητα και την καταλληλόλητα της πληροφορίας που τροφοδοτείται στο σύστημα. Η απόδοση των μοντέλων επηρεάζεται άμεσα από την αναπαράσταση και τα χαρακτηριστικά του ηχητικού σήματος που χρησιμοποιούνται ως δεδομένα εισόδου (Zaman et al., 2023). Η μελέτη της φύσης του ήχου και των μηχανισμών της ανθρώπινης ακουστικής αντίληψης αποτελεί τον θεμέλιο λίθο για την επιλογή χαρακτηριστικών που θα συμβάλλουν στην επιτυχή ταξινόμηση των μουσικών ειδών. Άλλωστε κατά κάποιον τρόπο, χρήσει λογισμικού, ο άνθρωπος προσπαθεί να εξομοιώσει τον τρόπο που ο ίδιος αφουγκράζεται τον ήχο και τον κατηγοριοποιεί, συνεπώς πολλές προσεγγίσεις αξιοποιούν ψυχοακουστικές έννοιες και αντιληπτικές μετρικές για τη βελτίωση της αναπαράστασης της μουσικής πληροφορίας (Namgyal et al., 2024). Καθώς όμως η επεξεργασία καλείται να υλοποιηθεί από υπολογιστικά συστήματα, η φυσική υπόσταση του ήχου πρέπει απαραιτήτως να μεταφραστεί σε αυστηρές μαθηματικές δομές μέσω της ψηφιακής αναπαράστασης σήματος, χρησιμοποιώντας τυπικές τεχνικές δειγματοληψίας, κβαντισμού (quantization) και μετασχηματισμών στο πεδίο του χρόνου-συχνότητας (Božić & Horvat, 2024). Η διαδικασία αυτή μετασχηματίζει το φυσικό ακουστικό ερέθισμα σε κατάλληλα διαμορφωμένα δεδομένα εισόδου. Σε αυτά, τα νευρωνικά δίκτυα μπορούν στη συνέχεια να εντοπίσουν και να κωδικοποιήσουν τα πολύπλοκα μοτίβα που χαρακτηρίζουν κάθε μουσικό είδος.

2.2.1 Ήχος και κυματομορφή

Δεδομένου ότι η μουσική ανήκει στην ευρεία κατηγορία του ήχου, είναι λογικό να μελετήσουμε αρχικά το τι ακριβώς είναι ο ήχος. Ο άνθρωπος αντιλαμβάνεται τον ήχο με την ακοή, αλλά ως φυσικό φαινόμενο ο ήχος συνίσταται σε κύματα τα οποία παράγονται από μια πηγή, διαδίδονται σε ένα μέσο και φθάνουν κάπου όπου μπορούν να παρατηρηθούν (Christensen, 2019). Πιο συγκεκριμένα ήχος παράγεται από τη δόνηση ενός αντικειμένου, ενώ με τη σειρά τους οι δονήσεις αυτές κάνουν τα μόρια του αέρα να ταλαντώνονται, δημιουργώντας μεταβολές στην πίεση του αέρα, σχηματίζοντας διαμήκη κύματα (Rumsey & McCormick, 2021). Εάν οπτικοποιούσαμε το κύμα αυτό, θα διακρίναμε περιοχές υψηλότερης και χαμηλότερης πυκνότητας αέρα, που αντιστοιχούν στη διάδοση του ηχητικού κύματος.

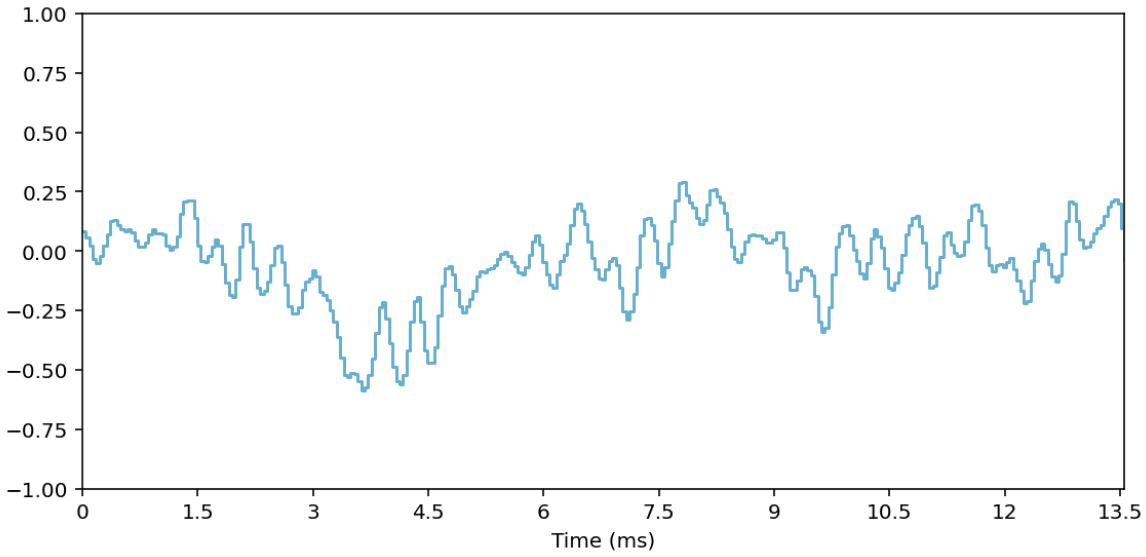
Μια αρκετά συχνή και χρήσιμη οπτικοποιημένη μορφή του ήχου, αποτελεί η κυματομορφή (waveform).



Εικόνα 1 Κυματομορφή ηχητικού δείγματος blues από το GTZAN dataset

Η κυματομορφή ουσιαστικά αποτελεί μια γραφική παράσταση πίεσης – χρόνου όπου διακρίνεται η μεταβολή της πίεσης του αέρα στη μονάδα του χρόνου ή πιο συγκεκριμένα η απόκλιση της πίεσης του αέρα από τη μέση τιμή του (Müller, 2021). Στην πράξη, η κυματομορφή ουσιαστικά αποτυπώνει την ταλάντωση του διαφράγματος ενός μικροφώνου, αναπαριστώντας γραφικά τη μετατόπισή του από μια θέση ισορροπίας συναρτήσει του χρόνου λόγω της αλλαγής στην πίεση του αέρα που προκαλείται από το ηχητικό κύμα (Christensen, 2019). Η Εικόνα 1 δείχνει την καταγεγραμμένη κυματομορφή ενός σήματος 30 δευτερολέπτων. Σε πολλά προγράμματα ηχογράφησης, είναι εφικτό να δει κανείς την κυματομορφή να διαγράφεται καθώς ηχογραφείται ένα μουσικό όργανο.

Για να κατανοήσουμε το σχήμα της κυματομορφής, θα πρέπει να ορίσουμε την έννοια του περιοδικού και απεριοδικού ήχου. Όπως φαίνεται και από την ετυμολογία των όρων, στην πρώτη περίπτωση έχουμε περιοδικότητα, ενώ στη δεύτερη όχι. Στην περίπτωση του μουσικού ήχου έχουμε ως επί τω πλείστον περιοδικά κύματα, ενώ στην περίπτωση του θορύβου απεριοδικά (Zain, 2024). Στην περίπτωση των περιοδικών κυμάτων, μπορούμε να έχουμε απλά ή σύνθετα περιοδικά κύματα. Εάν η πηγή του ήχου προκαλείται από απλή αρμονική διαταραχή του αέρα έχουμε απλό περιοδικό κύμα, εάν όμως έχουμε ήχο που είναι αποτέλεσμα μιας κυκλικής μεν αλλά ακανόνιστης διαταραχής, έχουμε σύνθετο περιοδικό κύμα (Hansen, 2018).



Εικόνα 2 Λεπτομέρεια κυματομορφής ηχητικού δείγματος του GTZAN dataset

Στην κυματομορφή, μπορούμε να διακρίνουμε κάποια φυσικά μεγέθη που σχετίζονται με το ηχητικό κύμα. Είναι ευκολότερο να διακρίνουμε τα μεγέθη αυτά, από ένα σχήμα απλού περιοδικού κύματος. Στην Εικόνα 2 όπου έχουμε μια λεπτομέρεια της κυματομορφής, μπορούμε να διακρίνουμε την ομοιότητά της με μια ημιτονοειδή καμπύλη, όπου τα μεγέθη αυτά θα διακρίνονταν ευκολότερα. Η μέγιστη απόσταση της κυματομορφής από τον οριζόντιο άξονα, μας δίνει το πλάτος (amplitude) του κύματος, ενώ η απόσταση δυο κορυφών της κυματομορφής, μας δίνει την περίοδο, το αντίστροφο της οποίας είναι η συχνότητα (frequency) του κύματος (Müller, 2021). Τα μεγέθη αυτά μας δίνουν πληροφορίες για το πως ακούγεται ο συγκεκριμένος ήχος. Ο ρυθμός με τον οποίο ταλαντώνεται η πηγή θα συνθέσει τη συχνότητα του ήχου ενώ η ποσότητα συμπίεσης ή αραίωσης του αέρα, συμβάλλει στην έντασή του (Rumsey & McCormick, 2021). Βλέπουμε λοιπόν ότι από την κυματομορφή, μπορούμε να εξάγουμε πληροφορίες για τη φύση του ήχου. Πιο συγκεκριμένα, αν η κυματομορφή προέρχεται από ένα μουσικό όργανο, όσο μεγαλύτερη η συχνότητα τόσο πιο ψηλή νότα ακούγεται, ενώ όσο πιο μεγάλο το πλάτος, τόσο δυνατότερα ακούγεται η νότα (Müller, 2021).

2.2.2 Ανθρώπινη αντίληψη ακουστικών χαρακτηριστικών

Δεδομένου ότι η κατηγοριοποίηση της μουσικής σε είδη ξεκινά από τον ίδιο τον άνθρωπο, κρίνεται σκόπιμο να εξηγηθεί ο τρόπος που ο άνθρωπος αντιλαμβάνεται τον ήχο. Η έρευνα έχει δείξει πως ο τρόπος που ο άνθρωπος ακούει τον ήχο, παρουσιάζει αρκετές ιδιαιτερότητες που φαίνεται πως σχετίζονται με τη φυσιολογία του σώματός του και την

ανάγκη να ξεχωρίζει συγκεκριμένους ήχους όπως την ίδια την ανθρώπινη φωνή (Rabiner & Schafer, 2011). Επίσης η αντίληψη του ήχου υπόκειται σε ψυχοακουστικούς παράγοντες ενώ μπορεί να είναι υποκειμενική από άνθρωπο σε άνθρωπο, καθώς το τι θεωρεί κανείς ευχάριστο στη μουσική ή το πόσο δυνατά θεωρεί ότι ακούγεται κάτι ποικίλλει (Sujatha, 2023).

Όταν μιλάμε για ένα απλό αρμονικό κύμα όπως αναφέρθηκε στο προηγούμενο υποκεφάλαιο, τότε ο ήχος που παράγεται, αντιστοιχεί σε μια συχνότητα (Hansen, 2018). Σύμφωνα με τον Müller (2021), κυματομορφή αυτού του ήχου μπορεί να παρασταθεί με μια ημιτονοειδή καμπύλη και αποτελεί τη βάση αυτού που ο άνθρωπος αντιλαμβάνεται ως μουσική νότα. Ο ίδιος, σημειώνει πως η συχνότητα που διακρίνεται από αυτή την ημιτονοειδή καμπύλη σχετίζεται με το τονικό ύψος (pitch) του ήχου. Η έννοια του ύψους του ήχου διακρίνεται εύκολα στις μουσικές νότες, οι οποίες είναι C, C#, D, D#, E, F, F#, G, G#, A, A#, B και μπορούν να ανήκουν σε διαφορετικές οκτάβες. Για να διακρίνουμε την οκτάβα που ανήκει μια νότα, δίπλα στο σύμβολό της έχουμε έναν αριθμό που δείχνει την οκτάβα. Έτσι λοιπόν η A2 αντιστοιχεί σε 110Hz, η A3 σε 220 Hz, η A4 σε 440 και η A5 σε 880Hz κτλ. (Christensen, 2019). Διακρίνουμε λοιπόν πως καθώς η ίδια νότα «ανεβαίνει» σε οκτάβες, η τιμή της συχνότητας είναι η διπλάσια της προηγούμενης. Σύμφωνα με τον Müller (2021), ο άνθρωπος, αντιλαμβάνεται την διαφορά του ύψους μεταξύ των νοτών διαφορετικής οκτάβας σαν ίδια, επομένως συμπεραίνουμε ότι η ανθρώπινη αντίληψη του ύψους ενός ήχου είναι λογαριθμική εκ φύσεως. Γενικότερα πάντως, ο ίδιος σημειώνει πως η αντίληψη του τονικού ύψους μπορεί να είναι υποκειμενική και να εξαρτάται από ψυχοακουστικούς παράγοντες.

Η ανθρώπινη αντίληψη φαίνεται πως παρουσιάζει ιδιαιτερότητα και όσον αφορά την ένταση του ήχου. Το ανθρώπινο αυτί φαίνεται πως αντιλαμβάνεται λογαριθμικά και τις αυξανόμενες τιμές πίεσης που δέχεται στο τύμπανο (Hansen, 2018). Το πόσο δυνατά αντιλαμβάνεται ο άνθρωπος τον ήχο είναι επίσης υποκειμενικό, αντίθετα με αντικειμενικά μεγέθη, όπως η ισχύς του ήχου και η ένταση του ήχου (sound power and sound intensity), τα οποία μπορούν να μετρηθούν με ακρίβεια (Müller, 2021). Η ισχύς του ήχου αποτελεί τον ρυθμό ενέργειας που διαδίδει η πηγή ανά μονάδα του χρόνου και μετριέται σε Watt (Sujatha, 2023). Αντίστοιχα, η ένταση ορίζεται ως η ισχύς του ήχου ανά μονάδα επιφάνειας (Müller, 2021). Ο ορισμός αυτών των δύο μεγεθών είναι σημαντικός για την κατανόηση του τρόπου που ο άνθρωπος αντιλαμβάνεται την ένταση.

Το ανθρώπινο αυτί παρουσιάζει αξιοσημείωτη ικανότητα αντίληψης ήχων σε ένα πολύ μεγάλο φάσμα ακουστικών εντάσεων. Για μια συχνότητα 1000Hz, η χαμηλότερη ένταση στην οποία μπορεί να ακούσει το αυτί, δηλαδή το όριο ακοής (threshold of hearing), είναι τα 10-12 W/m2 (Moore, 2013). Δεδομένου λοιπόν ότι στα 10 W/m2 φθάνουμε σε αυτό που ορίζεται ως όριο πόνου (threshold of pain), διακρίνουμε την ανάγκη να χρησιμοποιείται μια λογαριθμική κλίμακα για την ένταση, η οποία είναι συνήθως τα decibel ή απλούστερα dB (Müller, 2021). Τα decibel ουσιαστικά είναι μια κλίμακα ένδειξης του πόσο μεγαλύτερο είναι ένα μέγεθος από ένα άλλο (Rumsey & McCormick, 2021). Καθώς ο τύπος υπολογισμού τους είναι

$$dB(I) = 10 \cdot \log_{10}\left(\frac{I}{I_{TOH}}\right) \quad (1)$$

Όπου I είναι η ένταση του ήχου που μελετάμε, ενώ I_{TOH} είναι η ένταση του ορίου ακοής που ορίστηκε παραπάνω. Διακρίνουμε λοιπόν πως το dB είναι μια λογαριθμική κλίμακα που ουσιαστικά μας δείχνει πόσο μεγαλύτερη είναι η ένταση που μετράμε, συγκριτικά με το όριο της ανθρώπινης ακοής. Στη θέση του ορίου ακοής, μπορεί να έχουμε ένα άλλο σημείο αναφοράς, εάν όμως έχουμε αυτό το όριο, τότε η κλίμακα αναφέρεται ως dB SPL εκ του "Sound pressure level" (Moore, 2013). Εύκολα υπολογίζεται ότι διπλασιασμός του τετραγώνου της πίεσης του αέρα, οδηγεί σε αύξηση της κλίμακας κατά 3 dB που είναι το όριο που αντιλαμβάνεται ο άνθρωπος ως αλλαγή στην ένταση, ενώ αύξηση κατά 10 dB για την ανθρώπινη αντίληψη, σημαίνει διπλασιασμό της προηγούμενης έντασης (Sujatha, 2023).

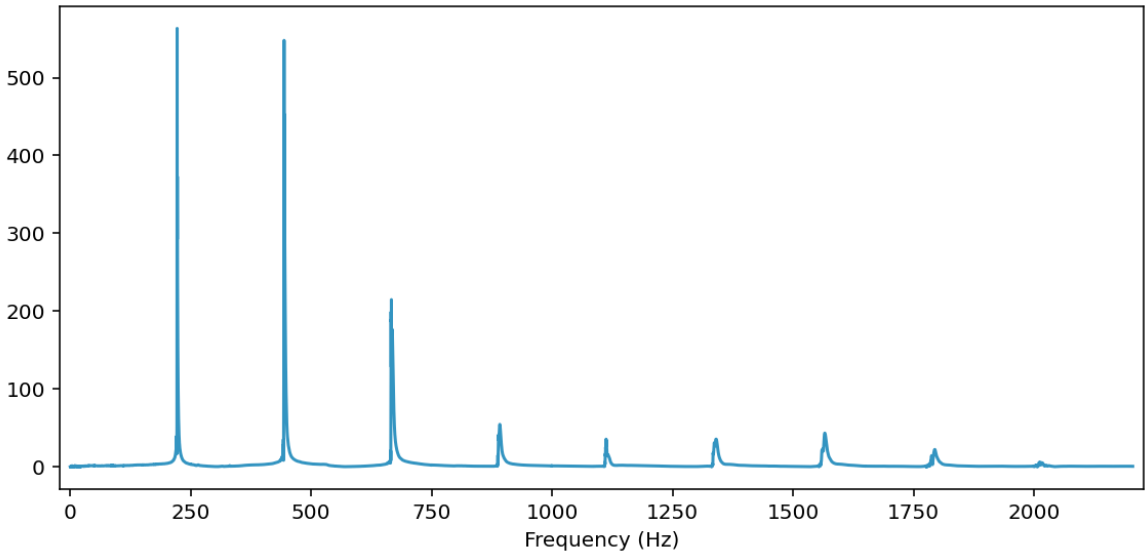
Η κλίμακα decibel, παρουσιάζει μια αντικειμενική μέτρηση της έντασης, αναφέραμε όμως ότι η αντίληψη της έντασης από τον άνθρωπο είναι υποκειμενική. Για την υποκειμενική μελέτη της έντασης του ήχου, εισάγεται η έννοια της ακουστότητας (loudness), καθώς το ανθρώπινο αυτί δεν είναι το ίδιο ευαίσθητο σε όλες τις συχνότητες (Rumsey & McCormick, 2021). Οι Fletcher και Muson, το 1933 παρουσίασαν μετά από πειράματα ένα διάγραμμα με έντασης – συχνότητας όπου παρουσιάζονται καμπύλες ίσης ακουστότητας, ενώ κατόπιν ακολούθησαν και άλλα πειράματα (Moore, 2013). Σε αυτά τα διαγράμματα φαίνεται πως ενώ ένας ήχος μπορεί να έχει δυνατότερη ένταση από έναν άλλο, ο άνθρωπος μπορεί να τον ακούει το ίδιο δυνατά, μόνο και μόνο επειδή έχουν διαφορετική συχνότητα. Η ακουστότητα μετριέται σε phon, με τα 0 phon να είναι η ακουστότητα που αντιλαμβάνεται ο άνθρωπος μια συχνότητα των 1000 Hz σε ένταση ίση με το όριο της ακοής (Rabiner & Schafer, 2011).

Παράλληλα, η αντίληψη της έντασης από τον άνθρωπο εξαρτάται και από άλλους παράγοντες όπως η διάρκεια του ήχου ενώ σημαντικοί παράγοντες είναι η ηλικία και η γενικότερη κατάσταση της υγείας του ακουστικού συστήματος (Moore, 2013).

Έχοντας σχολιάσει τον τρόπο που ο άνθρωπος αντιλαμβάνεται το ύψος ενός ήχου και την έντασή του, φθάνουμε σε ένα τρίτο χαρακτηριστικό που αφορά τον τρόπο που αυτός αντιλαμβάνεται τον ήχο, το οποίο είναι η χροιά ή ηχόχρωμα (timbre). Η χροιά είναι το ακουστικό χαρακτηριστικό που δίνει στον ακροατή τη δυνατότητα να ξεχωρίσει δυο διαφορετικούς ήχους στους οποίους όμως διακρίνει το ίδιο ύψος και την ίδια ακουστότητα (Sethares, 2005). Επίσης, η χροιά επιτρέπει στον ακροατή να ξεχωρίσει έναν ίδιο μουσικό τόνο παιγμένο από διαφορετικά μουσικά όργανα (Müller, 2021). Αντίθετα με το ύψος ή την ακουστότητα, η χροιά είναι ένα χαρακτηριστικό το οποίο μπορεί να χαρακτηριστεί ως «πολυδιάστατο», υπό την έννοια ότι εξαρτάται από πολλούς παράγοντες (Moore, 2013). Ο Schouten (1968, όπ.αναφ. στους.Müller et al., 2011) συνοψίζει αυτούς τους παράγοντες σε 5 κυρίαρχους: (1) κατά πόσο ο ήχος θυμίζει μουσική ή θόρυβο, (2) πως μεταβάλλονται οι συχνότητες με το χρόνο, (3) πως μεταβάλλεται η ένταση με το χρόνο (4) πως μεταβάλλονται οι συχνότητες συγκριτικά με την κύρια αρμονική συχνότητα και (5) την έναρξη του ήχου εν συγκρίσει με την υπόλοιπή του διάρκεια. Αυτοί οι παράγοντες, μπορούν να εξεταστούν με τα εξής τρία χαρακτηριστικά: την περιβάλλουσα του ήχου (sound envelope), το αρμονικό του υπόβαθρο και την διαμόρφωση συχνότητας/έντασης (frequency/amplitude modulation).

Η περιβάλλουσα του ήχου περιγράφει τα διαδοχικά στάδια των δυναμικών ενός ήχου από την αφετηρία του, μέχρι να σβήσει. Συνολικά αποτελείται από 4 μέρη: (1) attack: Η «ατάκα» του ήχου, πρόκειται για το σύντομο διάστημα στο οποίο ο ήχος αρχικά εντείνεται, (2) decay: η σύντομη ελάττωση της έντασης αμέσως μετά το attack, (3) sustain: το μέρος όπου ο ήχος διαρκεί και σβήνει ελάχιστα, (4) release: το μέρος όπου ο ήχος εν τέλει σβήνει (Müller, 2021). Ο τρόπος που αυτές οι φάσεις θα εξελιχθούν, χαρακτηρίζουν την περιβάλλουσα και επηρεάζουν τη χροιά. Για παράδειγμα μια νότα παιγμένη στο πιάνο χαρακτηρίζεται από απότομο attack και σταδιακό σβήσιμο, ενώ αν την ηχογραφούσαμε και την ακούγαμε να αναπαράγεται ανάποδα, δε θα θύμιζε πιάνο (Moore, 2013). Το αρμονικό υπόβαθρο, σχετίζεται με τις συχνότητες που συνθέτουν έναν ήχο. Προηγουμένως αναφερθήκαμε σε κυματομορφές που αποτελούνται από μια ημιτονοειδή καμπύλη, αλλά στην πραγματικότητα, οι περισσότεροι ήχοι είναι αποτέλεσμα συνδυασμού μοτίβων

δονήσεων που έχουν ως αποτέλεσμα ένα σύνθετο κύμα (Rumsey & McCormick, 2021). Δεδομένου η ανάλυση αφορά τη μουσική, θα παραδειγματιστούμε πάνω σε αυτή. Έστω λοιπόν ένα μουσικό όργανο το οποίο παίζει μια νότα. Η νότα αυτή έχει μια «κυρίαρχη» συχνότητα f, αλλά ο ήχος που ακούγεται είναι σύνθεση αυτής και των συχνοτήτων 2f, 3f κλπ (Sethares, 2005). Οι συχνότητες αυτές συνήθως καλούνται αρμονικές και η κατανομή της έντασής τους είναι χαρακτηριστική της χροιάς των μουσικών οργάνων, καθώς για παράδειγμα το κλαρινέτο έχει τις αρμονικές που αντιστοιχούν σε περιττά πολλαπλάσια εντονότερες (Müller et al., 2011). Στην Εικόνα 3, βλέπουμε της αρμονικές που εμφανίζονται καθώς μια ηλεκτρική κιθάρα παίζει την νότα A3 που αντιστοιχεί στα 220 Hz. Η κορυφή της κυρίαρχης συχνότητας είναι χαρακτηριστική, ενώ βλέπουμε ότι υπάρχουν αντίστοιχα κορυφές για τη διπλάσια, την τριπλάσια και τις υπόλοιπες συχνότητες, ενώ κάθε αρμονική φαίνεται να έχει διαφορετική ένταση. Τέλος, ο τρόπος που πολλά όργανα παίζονται μπορεί να εμπεριέχει το εκφραστικό ύφος του εκτελεστή. Για παράδειγμα, συχνά παρατηρούμε καθώς το βιολί παίζει μια νότα, να έχουμε σταδιακές αυξομειώσεις έντασης ή της συχνότητας (tremolo και vibrato αντίστοιχα), κάτι το οποίο χαρακτηρίζει τη χροιά του (Müller, 2021).



Εικόνα 3 Αρμονικές που εμφανίζονται καθώς ηλεκτρική κιθάρα παίζει τη νότα A3

Τα παραπάνω συνοψίζουν κάποια από τα βασικότερα στοιχεία του τρόπου που ο άνθρωπος αντιλαμβάνεται τη μουσική. Η έρευνα έχει δείξει ότι χαρακτηριστικά σαν αυτά που αναφέρθηκαν αλλά και άλλα (όπως πχ ο ρυθμός) είναι εφικτό να εξαχθούν από αρχεία ήχου

και να χρησιμοποιηθούν για σκοπούς όπως η ταξινόμηση μουσικών ειδών (Müller et al., 2011).

Τελειώνοντας την αναφορά στην ανθρώπινη αντίληψη ακουστικών χαρακτηριστικών, αξίζει να αναφερθούμε στο εύρος συχνοτήτων που μπορεί να συλλάβει το ανθρώπινο αυτί. Γενικώς σημειώνεται ότι το συχνοτικό εύρος το οποίο συλλαμβάνει το ανθρώπινο αυτί είναι μεταξύ 20Hz έως 20kHz, όμως αυτό που γενικά συμβαίνει είναι πως ο άνθρωπος δεν περιορίζεται σε αυτά τα όρια, απλά φαίνεται πως εκτός των ορίων αυτών η ένταση πρέπει να είναι σχετικά μεγάλη ώστε να υπάρξει κάποιο ερέθισμα (Rumsey & McCormick, 2021). Δεδομένου λοιπόν ότι το συχνοτικό εύρος εξαρτάται από την ένταση του ερεθίσματος, σε συγκριτικές έρευνες του ακουστικού εύρους διαφόρων ειδών συνηθίζεται οι ήχοι να ακούγονται σε ένταση 60 dB SPL (Heffner & Heffner, 2007). Σε αυτή την ένταση, το συχνοτικό εύρος του ανθρώπινου αυτιού φαίνεται ότι κυμαίνεται μεταξύ 31 Hz και 17.6 kHz (Jackson et al., 1999). Συγκριτικά με το άνθρωπο πάντως, οι σκύλοι και οι γάτες φαίνεται πως στα 60 dB δε συλλαμβάνουν χαμηλές συχνότητες που ακούει ο άνθρωπος, όμως μπορούν να ακούσουν υψηλότερες, ενώ παρόμοια συμπεριφορά συναντάμε και σε άλλα θηλαστικά (Heffner & Heffner, 2007).

2.2.3 Ψηφιοποίηση ηχητικού σήματος

Όπως αναλύθηκε στα προηγούμενα υποκεφάλαια, ο ήχος και συγκεκριμένα η μουσική, περιλαμβάνουν χαρακτηριστικά τα οποία ο άνθρωπος μπορεί να εντοπίσει και να διακρίνει ή και να ταξινομήσει τι είναι αυτό που ακούει. Αναλύθηκε επίσης, το ότι στοιχεία που αφορούν έναν ήχο, μπορούν να εξαχθούν από την κυματομορφή. Αυτή μπορεί να προκύψει από ένα μικρόφωνο το οποίο μετατρέπει την πίεση των μορίων του αέρα στο διάφραγμά του σε ένα ηλεκτρικό σήμα (Rumsey & McCormick, 2021). Το πρόβλημα εδώ είναι πως αυτό το σήμα είναι συνεχές, ενώ οι υπολογιστές μπορούν να αποθηκεύσουν έναν πεπερασμένο αριθμό δεδομένων, γι' αυτό απαιτούνται κάποιες διαδικασίες (δειγματοληψία, κβαντισμός), ώστε από το άπειρο πλήθος τιμών του συνεχούς σήματος να διατηρηθούν όσο το δυνατόν περισσότερες πληροφορίες μέσα από ένα σύνολο διακριτών τιμών (Christensen, 2019). Στην περίπτωση του συνεχούς σήματος λέμε πως έχουμε αναλογικό σήμα ενώ όταν έχουμε διακριτές τιμές έχουμε ψηφιακό σήμα, με τη μετατροπή του ενός σήματος στο άλλο να είναι εφικτή, χρήσει digital-to-analog ή analog-to-digital μετατροπείς, γνωστούς και ως DAC/ADC converters (Sujatha, 2023).

Για την κατανόηση της ψηφιοποίησης του ηχητικού σήματος, κρίνεται απαραίτητη η περιγραφή της διαδικασίας της δειγματοληψίας (sampling) και του κβαντισμού (quantization). Ξεκινάμε λοιπόν από τον ορισμό του πρώτου. Όπως αναφέρθηκε, δεν είναι εφικτό να υπάρξει αποθηκευτικός χώρος στον υπολογιστή για άπειρη ποσότητα δεδομένων, επομένως πρέπει να πάρουμε έναν πεπερασμένο αριθμό αυτών. Πώς όμως θα γίνει η επιλογή του πόσα δεδομένα θα πάρουμε; Η διαδικασία της δειγματοληψίας περιλαμβάνει ακριβώς αυτό. Σύμφωνα με τους A. Oppenheim et al. (1996), ένα αναλογικό σήμα είναι εφικτό να αναπαρασταθεί αλλά και να αναδομηθεί πλήρως από τα δείγματα ενός ψηφιακού σήματος. Οι ίδιοι σημειώνουν ότι αυτή η διαδικασία θυμίζει τη λειτουργία του βίντεο όπου ουσιαστικά παρουσιάζονται ακίνητες εικόνες στη σειρά, όμως με τον κατάλληλο ρυθμό προβολής τους, ο θεατής δεν καταλαβαίνει ότι πρόκειται για διακριτές εικόνες. Για τη δειγματοληψία, ορίζουμε το μέγεθος του ρυθμού ή συχνότητας δειγματοληψίας (sample rate) ως τον αριθμό δειγμάτων που λαμβάνουμε ανά δευτερόλεπτο, μέγεθος που μετριέται σε Hz (Müller, 2021). Το ερώτημα που τίθεται εδώ είναι το ποια είναι η κατάλληλη τιμή του ρυθμού δειγματοληψίας. Αποδεικνύεται, ότι εάν η υψηλότερη συχνότητα που περιλαμβάνεται σε ένα αναλογικό σήμα είναι fmax, τότε αυτό μπορεί πλήρως να αναδομηθεί από μια δειγματοληψία ρυθμού ίσου ή μεγαλύτερου από 2·fmax (Proakis & Manolakis, 2013). Η ελάχιστη αυτή τιμή του απαιτούμενου ρυθμού δειγματοληψίας, λέγεται συχνότητα Nyquist προς τιμήν του ηλεκτρολόγου μηχανολόγου Harry Nyquist (1889-1976) και κατά την αναδόμηση του αναλογικού σήματος, μόνο συχνότητες μικρότερες από το ήμισυ αυτής θα αναπαρασταθούν επαρκώς (Steiglitz, 1996). Αυτό το δεδομένο λαμβάνεται υπόψιν κατά την ψηφιοποίηση αναλογικών σημάτων, έτσι ώστε να έχουμε επαρκή αναπαράστασή τους κατά τη σύνθεσή τους από ψηφιακά σήματα. Ενδεικτικά, ο ρυθμός δειγματοληψίας ενός ψηφιακού CD μουσικής, είναι 44.100 Hz (Steiglitz, 1996). Αυτή η τιμή δεν είναι τυχαία, αν αναλογιστούμε ότι ήδη αναφέρθηκε πως ο άνθρωπος φέρεται γενικώς να ακούει ήχους έως 20.000 Hz, μια τιμή περίπου ίση με το μισό της τιμής του ρυθμού δειγματοληψίας του CD. Στην περίπτωση που ο ρυθμός δειγματοληψίας είναι μικρός, η αναπαραγωγή από το ψηφιακό μέσο δεν θα παρέχει όλη την πληροφορία του αρχικού σήματος και θα έχουμε το φαινόμενο του aliasing (Müller, 2021). Για την κατανόηση του φαινομένου, παραπέμπουμε σε ένα οπτικό φαινόμενο, γνωστό ως wagon wheel effect. Πολύ συχνά στον κινηματογράφο, βλέπουμε τις ζάντες ενός αυτοκινήτου να κινούνται ανάποδα από την προβλεπόμενη φορά, πράγμα που οφείλεται στο ότι ο ρυθμός παρουσίασης των εικόνων του βίντεο είναι μικρός συγκριτικά με την

ταχύτητα του οχήματος (Purves et al., 1996). Αυτή η οφθαλμαπάτη είναι μια ατέλεια που οφείλεται στα κινηματογραφικά τεχνικά μέσα και αντίστοιχες ηχητικές ατέλειες έχουμε στην περίπτωση του aliasing στα ακουστικά μέσα.

Είδαμε ότι κατά τη διαδικασία της δειγματοληψίας παίρνουμε έναν συγκεκριμένο αριθμό δειγμάτων από το συνεχές αναλογικό σήμα. Στην περίπτωση της ψηφιοποίησης της κυματομορφής, αυτές αποτελούν διακριτές τιμές του άξονα χ που αντιστοιχεί στο χρόνο. Αυτές οι τιμές αντιστοιχούν σε μια τιμή πλάτους του κύματος που μπορεί να είναι οποιαδήποτε. Όμως ένα υπολογιστικό σύστημα έχει πεπερασμένο αριθμό από bit για να της αποθηκεύσει, επομένως με μια διαδικασία που ονομάζουμε κβάντωση (quantization) αντιστοιχούμε τις άπειρες πιθανές τιμές που αντιστοιχούν οι τιμές που προκύπτουν από τη δειγματοληψία, σε έναν πεπερασμένο αριθμό τιμών (Christensen, 2019). Σε ένα ψηφιακό σήμα, οι τιμές του άξονα χ γίνονται διακριτές μέσω της δειγματοληψίας, ενώ οι τιμές του άξονα y γίνονται διακριτές μέσω της κβάντωσης, όπου ουσιαστικά έχουμε διακριτές τιμές για το πλάτος της κυματομορφής (Sujatha, 2023). Σύμφωνα με τους Proakis και Manolakis (2013), ο κβαντισμός ενός σήματος περιλαμβάνει την επιλογή ενός αριθμού από bit, τα οποία και θα χρησιμοποιηθούν για τις διακριτές τιμές του πλάτους. Οι ίδιοι σημειώνουν πως ο αριθμός αυτός συνήθως είναι 16-bit, μια τιμή που συναντάμε στα audio CD καθώς αποδεικνύεται πως με αυτήν μπορούμε να καλύψουμε ένα δυναμικό εύρος που υπάρχει στις περισσότερες μουσικές ηχογραφήσεις. Με τα 16 αυτά bit, μπορούμε να έχουμε 65536 πιθανές διακριτές τιμές για το πλάτος της κυματομορφής (Müller, 2021). Η τιμή των bit που επιλέγονται για τον κβαντισμό συνήθως καλείται «bit-depth», έννοια η οποία συνδέεται με το δυναμικό εύρος μιας ηχογράφησης, καθώς μεγαλύτερο bit depth σημαίνει μεγαλύτερο εφικτό δυναμικό εύρος που μπορεί να αναπαραχθεί από το ψηφιακό σήμα (Melchior, 2019). Το δυναμικό εύρος είναι μια διαισθητική έννοια που αφορά την αναλογία της στάθμης των δυνατότερων και των πιο απαλών ήχων μιας πηγής (Proakis & Manolakis, 2013).

Είδαμε λοιπόν πως ένα σήμα μπορεί να ψηφιοποιηθεί. Η διαδικασία αυτή πραγματώνεται με ειδικούς μετατροπείς ADC, ενώ η αντίστροφη διαδικασία είναι επίσης εφικτή με τους μετατροπείς DAC (Sujatha, 2023). Η γενική ροή επεξεργασίας του ήχου πάντως, περιλαμβάνει τη λήψη του σήματος από μια συσκευή όπως το μικρόφωνο, τη μετατροπή του σήματος σε ψηφιακό μέσω ADC, την επεξεργασία από έναν υπολογιστή, τη μετατροπή του ψηφιακού σήματος σε αναλογικό μέσω DAC και τέλος την αναπαραγωγή από ένα μέσο όπως τα ηχεία (Christensen, 2019).

2.2.4 Εξαγωγή χαρακτηριστικών περιγραφών ήχου

Έως τώρα, έχουμε δει ότι ο ήχος έχει συγκεκριμένα χαρακτηριστικά, τα οποία ο άνθρωπος αφουγκράζεται και αντιλαμβάνεται με δικό του τρόπο, που εξαρτάται από τη φυσιολογία του. Τίθεται όμως το ερώτημα του πως εξάγουμε αυτά τα χαρακτηριστικά, καθώς επίσης ποια από αυτά τα χαρακτηριστικά θα χρησιμεύσουν ως δεδομένα εισόδου για ένα σύστημα τεχνητής νοημοσύνης, ώστε να εκτελέσει εργασίες όπως η κατηγοριοποίηση ήχου. Είναι αλήθεια πως ένα ηχητικό σήμα εμπεριέχει πολλή πληροφορία και ο άνθρωπος μπορεί από το σήμα αυτό να εξάγει ποικιλία ακουστικών χαρακτηριστικών (audio features), που μπορεί να χρησιμοποιήσει για σκοπούς που τον εξυπηρετούν Οι Knees και Schedl (2016), αναφέρουν πως τα ακουστικά χαρακτηριστικά που οι υπολογιστές χρησιμοποιούν ώστε να μοντελοποιήσουν την ανθρώπινη ακουστική αντίληψη, μπορούν να κατηγοριοποιηθούν με διάφορους τρόπους όπως (1) το επίπεδο σχετικότητάς τους (level of abstraction), (2) το χρονικό τους εύρος, (3) το μουσικό χαρακτηριστικό που περιγράφουν (νότα, αρμονία κλπ) και (4) το πεδίο που περιγράφεται το σήμα (πχ χρόνου ή συχνότητας). Τα χαρακτηριστικά της τελευταίας κατηγορίας θα μας απασχολήσουν κυρίως.

Το πεδίο όπου περιγράφεται το σήμα, είναι συνήθως είτε χρονικό (time domain) ή συχνοτικό (frequency domain), όπου το πρώτο είναι η κυματομορφή, ενώ το δεύτερο συνήθως προκύπτει από μετασχηματισμό Fourier (Knees & Schedl, 2016). Εφαρμόζοντας μετασχηματισμό Fourier σε ηχητικό σήμα συγκεκριμένης διάρκειας, μπορούμε να πάρουμε ένα διάγραμμα όπου θα φαίνονται οι συχνότητες που εμφανίζονται σε αυτή τη διάρκεια, καθώς και το μέτρο της καθεμίας (Christensen, 2019). Ενδεικτικά, χαρακτηριστικά τα οποία μπορούν να προκύψουν από το time domain είναι τα “amplitude envelope”, “root-mean-square energy”, “zero-crossing rate”, ενώ από το frequency domain μπορεί να έχουμε τα “band energy ratio”, “spectral centroid”, “spectral flux” (Knees & Schedl, 2016). Κάθε ένα από αυτά τα ακουστικά χαρακτηριστικά, μπορεί να χρησιμοποιηθεί ως δεδομένο εισόδου για αλγόριθμους μηχανικής μάθησης για διάφορους σκοπούς, ενώ αναλύθηκε ήδη το ότι η περιβάλλουσα του ήχου που σχετίζεται με το “amplitude envelope”, είναι χαρακτηριστική της χροιάς του ήχου. Για παράδειγμα ο Klapuri (1999), χρησιμοποιώντας το συγκεκριμένο χαρακτηριστικό του time domain, κατασκεύασε σύστημα αυτόματης ανίχνευσης μουσικών γεγονότων σε τραγούδια, υπολογίζοντας πχ για πόση διάρκεια εμφανίζεται το κάθε μουσικό όργανο σε ένα τραγούδι.

Δουλεύοντας στο πεδίο του χρόνου ή των συχνοτήτων ξεχωριστά, δεν έχουμε τη δυνατότητα να γνωρίζουμε το πότε εμφανίζονται οι συχνότητες. Για να επιτευχθεί αυτό, μπορούμε να «τεμαχίσουμε» τον χρόνο και να μεταφέρουμε το καθένα από αυτά τα τμήματα στο πεδίο των συχνοτήτων, κάτι που θα δώσει μια καλύτερη διαισθητική αντίληψη του ήχου, αλλά και χρησιμοποιείται στη συντριπτική πλειοψηφία των προηγμένων συστημάτων ανάλυσης ήχου (Lerch, 2022). Σύμφωνα με τους Knees και Schedl (2016), πρακτικά, αυτό επιτυγχάνεται μέσω της διαδικασίας του short-time Fourier transform (STFT) κατά το οποίο πραγματοποιείται μετασχηματισμός Fourier σε τμήματα του ηχητικού σήματος. Οι ίδιοι σημειώνουν πως η ένωση των αποτελεσμάτων των μετασχηματισμών, τοποθετημένων στον άξονα του χρόνου, είναι ουσιαστικά ένα φασματογράφημα (spectrogram). Η επεξεργασία των δεδομένων που προκύπτουν από το STFT, μπορεί να δώσει και άλλα δεδομένα όπως τα Mel-Spectrograms, Constant-Q transform τα οποία αποτελούν με τη σειρά τους audio features. Σε όποιο πεδίο και να δουλεύουμε, το αποτέλεσμα της εξαγωγής των δεδομένων μπορεί να οργανωθεί σε τανυστές (tensors) που εμπεριέχουν τα αριθμητικά δεδομένα των ακουστικών χαρακτηριστικών (Lerch, 2022).

Δεδομένου ότι αυτά τα αριθμητικά δεδομένα αποτελούν δεδομένα εισόδου για αλγορίθμους τεχνητής νοημοσύνης, ακολουθεί μια σύντομη αναφορά στις τεχνολογίες αυτές και το πως κατηγοριοποιούνται. Η τεχνητή νοημοσύνη (artificial intelligence – AI) είναι ένας ευρύς όρος που αφορά την τεχνολογία που επιτρέπει στις μηχανές να εκτελούν εργασίες οι οποίες κανονικά θα απαιτούσαν ανθρώπινη νοημοσύνη (Soori et al., 2023). Στόχος του AI, είναι να προσδώσει στις μηχανές νοημοσύνη αντίστοιχου επιπέδου με την ανθρώπινη (Perumal et al., 2024). Ο όρος «τεχνητή νοημοσύνη» χρησιμοποιείται γενικότερα όταν οι μηχανές εκτελούν εργασίες συνυφασμένες με την ανθρώπινη δραστηριότητα όπως η εκμάθηση ή η επίλυση προβλημάτων, συνεπώς δεδομένου ότι η μάθηση είναι σημαντικό μέρος του AI, συμπεραίνουμε πως η μηχανική μάθηση (machine learning – ML) είναι υποτομέας του AI (Shinde & Shah, 2018). Η μηχανική μάθηση (ML), είναι ο τομέας που επικεντρώνεται στην υλοποίηση συστημάτων που λύνουν προβλήματα, αφού εκπαιδεύονται από οργανωμένα δεδομένα που αφορούν ένα συγκεκριμένο πρόβλημα (Janiesch et al., 2021). Αυτό επιτυγχάνεται κάνοντας επαναληπτικά χρήση συγκεκριμένων αλγορίθμων ώστε τα συστήματα να βρουν μοτίβα στα δεδομένα που χρησιμοποιούνται για την εκπαίδευσή τους (Bishop, 2006). Τα τελευταία χρόνια όμως, σε τομείς που χρησιμοποιείται μηχανική

μάθηση, φαίνεται πως οι τεχνικές βαθιάς μάθησης (Deep Learning - DL) αποδίδουν καλύτερα (Perumal et al., 2024). Η βαθιά μάθηση είναι μέθοδος που συμπεριλαμβάνει τη χρήση νευρωνικών δικτύων για να αναλύσει μεγάλη ποσότητα δεδομένων και να αναγνωρίσει μοτίβα σε αυτά (Soori et al., 2023). Συνολικά, η τεχνητή νοημοσύνη θεωρείται ένας τομέας που περιλαμβάνει και τη μηχανική μάθηση, η οποία με τη σειρά της ως τομέας, περιλαμβάνει και τη βαθιά μάθηση (Perumal et al., 2024). Όσον αφορά την ταξινόμηση μουσικών ειδών, γνωστοί αλγόριθμοι μηχανικής μάθησης είναι οι K-nearest Neighbor (K-NN), Gaussian mixture model (GMM) και οι Support vector machines (SVM), ενώ η προσέγγιση της ταξινόμησης χρήσει MFCCs με SVM είναι αρκετά συχνή (Lerch, 2021). Ταξινόμηση μέσω μεθόδων βαθιάς μάθησης συναντάμε κατά τη χρήση CNN ή RNN νευρωνικών δικτύων τα οποία έχουν ως δεδομένα εισόδου Mel Spectrograms (W. Xu, 2024).

Είδαμε πως το ηχητικό σήμα μπορεί να ληφθεί από μια ηχογράφηση με μικρόφωνο και κατόπιν μπορεί να ψηφιοποιηθεί. Για την εξαγωγή των ακουστικών χαρακτηριστικών όμως, είναι απαραίτητη μια διαδικασία που καλείται “windowing”, με την οποία χωρίζουμε το σήμα σε διακριτά “time frames” (Knees & Schedl, 2016). Κίνητρο για αυτή τη διαδικασία, είναι το γεγονός ότι η αντιληπτική ικανότητα του ανθρώπινου αυτιού έχει διάρκεια περίπου 10 ms, επομένως οτιδήποτε συμβαίνει σε μικρότερο χρονικό διάστημα, δεν είναι εφικτό να γίνει αντιληπτό από τον άνθρωπο (Lerch, 2021). Δεδομένου λοιπόν ότι μέσω της δειγματοληψίας έχουμε πλέον διακριτές τιμές στον άξονα του χρόνου, είναι εύκολο να συμπεράνουμε ότι τα time frames θα αποτελούνται από συγκεκριμένο αριθμό δειγμάτων. Κριτήριο για την επιλογή μεγέθους είναι η προαναφερθείσα διάρκεια αντιληπτικής ικανότητας του ανθρώπινου αυτιού, ενώ οι ενδεικτικές τιμές για αυτά είναι κυμαίνονται μεταξύ 256-8192 samples (Knees & Schedl, 2016). Η τομή του αρχικού σήματος σε frames, εισάγει ένα νέο πρόβλημα, καθώς είναι οριακά απίθανο τα άκρα των frames να συμπέσουν με την περίοδο του αρχικού κύματος (McFee, 2023). Αυτό το πρόβλημα εκδηλώνεται στα διαγράμματα του frequency domain ως επιπρόσθετες πλασματικές εμφανίσεις συχνοτικών εξάρσεων και είναι γνωστό ως spectral leakage (Mitra & Kaiser, 1993). Για να αποφευχθεί το εν λόγω πρόβλημα, εφαρμόζουμε σε κάθε frame μια windowing function, η οποία θα μειώσει την ένταση του κύματος στα άκρα, ενώ για τη διαδικασία αυτή συνήθως επιλέγεται η συνάρτηση του Hann (Knees & Schedl, 2016). Αυτή η διαδικασία, με τη σειρά της, εισάγει το πρόβλημα της απώλειας πληροφορίας στα άκρα του κάθε frame, όπου ελαττώθηκε ή και

μηδενίστηκε η ένταση, αναλόγως τη window function που επιλέχθηκε. Για την αποφυγή αυτού του προβλήματος, τα time frames επικαλύπτονται μεταξύ τους αντί να είναι διαδοχικά (Backstrom, 2019). Το κατά πόσο επικαλύπτονται τα frames, ορίζεται από μια παράμετρο που λέγεται hop length και ουσιαστικά αποτελεί τον σταθερό αριθμό samples μεταξύ της αφετηρίας ενός frame από την αφετηρία του επόμενου (McFee, 2023).

Συνοπτικά, πραγματοποιώντας τα παραπάνω βήματα, έχουμε μια ολοκληρωμένη ροή (pipeline) για την εξαγωγή των χαρακτηριστικών ήχου. Αρχικά ηχογραφούμε το σήμα, το ψηφιοποιούμε, το χωρίζουμε σε επικαλυπτόμενα frames και από εκεί μπορούμε να πάρουμε τα ακουστικά χαρακτηριστικά που επιθυμούμε, οργανωμένα σε διανύσματα, πίνακες ή γενικά τανυστές (Knees & Schedl, 2016).

2.2.5 Μετασχηματισμός Fourier

Ο μετασχηματισμός Fourier είναι ένα από τα σημαντικότερα εργαλεία στην ανάλυση και την επεξεργασία ήχου, καθώς μας επιτρέπει να μεταβαίνουμε από το πεδίο του χρόνου (time domain), δηλαδή τη μεταβολή της κυματομορφής στο χρόνο, στο πεδίο των συχνοτήτων (frequency domain), δηλαδή την παρουσίαση των συχνοτήτων που τη συνθέτουν (Christensen, 2019). Η συγκεκριμένη μέθοδος παίρνει το όνομά της από τον Jeane Baptiste Fourier, ο οποίος με αφορμή τη μελέτη της διάδοσης της θερμότητας, κατά το 1807 υποστήριξε πως οποιοδήποτε περιοδικό σήμα, μπορεί να αναλυθεί σε ένα σύνολο ημιτονοειδών καμπυλών (A. Oppenheim et al., 1996). Σύμφωνα με τον Smith (2013), ο γενικός όρος του μετασχηματισμού Fourier, μπορεί να χωριστεί σε τέσσερεις κατηγορίες, αναλόγως το είδος του σήματος:

- Απεριοδικό-Συνεχές σήμα: Σε αυτή την περίπτωση αναφερόμαστε σε μετασχηματισμό Fourier (Fourier transform).
- Περιοδικό-Συνεχές σήμα: Σε αυτή την περίπτωση αναφερόμαστε σε σειρές Fourier (Fourier series)
- Απεριοδικό-Διακριτό σήμα: Σε αυτή την περίπτωση χρησιμοποιούμε τον μετασχηματισμό Fourier διακριτού χρόνου - Discrete time Fourier transform (DTFT).
- Περιοδικό-Διακριτό σήμα: Στην τελευταία αυτή περίπτωση αναφερόμαστε απλά σε διακριτό μετασχηματισμό Fourier – Discrete Fourier transform (DFT).

Όπως αναφέρθηκε και σε προηγούμενα υποκεφάλαια, ένα ηχητικό σήμα που προέρχεται από κάποιο μουσικό όργανο, μπορεί να είναι σύνολο διάφορων συχνοτήτων, έχοντας πάντα όμως μια κύρια συχνότητα. Μέσω του μετασχηματισμού Fourier, μπορούμε να αναλύσουμε μια ηχογράφηση ενός μουσικού οργάνου (και όχι μόνο) στις αρμονικές που συνθέτουν τη χροιά του (Lenssen & Needell, 2013). Βλέπουμε λοιπόν πως ένα τέτοιο εργαλείο είναι αρκετά σημαντικό καθώς μπορεί να μας δώσει πληροφορίες που σχετίζονται με την ανθρώπινη αντίληψη του ήχου.

Όπως εξηγεί ο Müller (2021), η βασική ιδέα πίσω από την ανάλυση Fourier, είναι η σύγκριση ενός σήματος με διάφορες πραγματικές τιμές συχνοτήτων, ώστε να διαπιστωθεί σε τι βαθμό η εν λόγω συχνότητα φαίνεται να υπάρχει εντός του σήματος. Για κάθε τιμή συχνότητας, παίρνουμε μια τιμή μέτρου (magnitude) d_f και μια τιμή φ_f που αντιστοιχεί στη φάση της εν λόγω συχνότητας. Μεγαλύτερες τιμές d_f δείχνουν εντονότερη παρουσία της συχνότητας εντός του σήματος, ενώ η τιμή της φάσης δείχνει πρακτικά την φάση που πρέπει να έχει η εξίσωση της ημιτονοειδούς καμπύλης της εν λόγω συχνότητας που συμβάλει στο σήμα. Οι Briggs και Henson, (1995) εξηγούν πως η αναπαράσταση μιας κυματομορφής που δίνεται με την μορφή εξίσωσης $g(t)$, στο πεδίο των συχνοτήτων, δίνεται από την παρακάτω εξίσωση Fourier:

$$\hat{g}(f) = \int_{-\infty}^{+\infty} g(t)\, e^{-i2\pi f t} dt \tag{2}$$

Ενώ μπορεί να υπάρξει και η αντίστροφη διαδικασία και από το πεδίο των συχνοτήτων να επανέλθουμε στο πεδίο του χρόνου με μια διαδικασία γνωστή ως αντίστροφος μετασχηματισμός Fourier – Inverse Fourier transform (IFD)

$$g(t) = \int_{-\infty}^{+\infty} \hat{g}(f)\, e^{i2\pi f t} df \tag{3}$$

Παρατηρούμε χαρακτηριστικά πως στην πρώτη περίπτωση η ολοκλήρωση γίνεται ως προς το χρόνο, ενώ στη δεύτερη ως προς τη συχνότητα. Το ολοκλήρωμα της σχέσης (2) δεν θα δώσει απευθείας μια συνάρτηση από την οποία θα δούμε το μέτρο κάθε συχνότητας στο σήμα. Ο μετασχηματισμός Fourier αναπαριστά το σήμα ως σύνολο μιγαδικών συντελεστών (Fourier coefficients), όπου κάθε συντελεστής αποτελεί μια πολική αναπαράσταση πλάτους και φάσης μιας συχνότητας (Müller, 2021). Ουσιαστικά κάθε ένας από αυτούς τους συντελεστές, αντιστοιχεί σε μια τιμή συχνότητας και είναι ένας μιγαδικός αριθμός. Από το

πραγματικό και το φανταστικό μέρος του μιγαδικού αριθμού μπορεί χρήσει τύπων να υπολογιστεί το μέτρο και η φάση της κάθε συχνότητας όπως αυτή υπάρχει στην κυματομορφή.

Οι τύποι (2) και (3), αφορούν αναλογικό σήμα με συνεχείς τιμές. Όπως αναφέρθηκε όμως σε προηγούμενα υποκεφάλαια, το αναλογικό σήμα ψηφιοποιείται και σε αυτή τη μορφή αποθηκεύεται στους υπολογιστές. Δεδομένου λοιπόν ότι πλέον ο χρόνος δεν παίρνει συνεχείς τιμές αλλά διακριτές όπως αυτές προκύπτουν από τη δειγματοληψία, θα πρέπει να προσαρμοστεί ο μετασχηματισμός Fourier. Παρατηρώντας τον τύπο (2), όπως αναφέρουν και οι Briggs & Henson (1995), εμφανίζονται μερικά προβλήματα όταν έχουμε διακριτές τιμές: Τα όρια του ολοκληρώματος φαίνεται σαν να υποδηλώνουν άπειρο χρόνο ο οποίος όπως αναφέρθηκε είναι συνεχής, ενώ οι τιμές της συχνότητας είναι επίσης συνεχείς. Για αυτά τα προβλήματα και την τελική προσαρμογή του τύπου (2), ο Müller (2021) αναφέρει τις εξής διορθώσεις:

- Αντί για συνεχείς τιμές χρόνου έχουμε N τιμές, ίσες με τον αριθμό δειγμάτων.
- Αντί για ολοκλήρωμα, έχουμε άθροισμα των διακριτών αυτών τιμών.
- Ο χρόνος δεν είναι άπειρος, αλλά είναι εφάμιλλος της διάρκειας της ηχογράφησης που έχουμε.
- Οι τιμές της συχνότητας επίσης δεν είναι άπειρες, αλλά διακριτές και ίσες με M. Συνηθίζεται να επιλέγουμε M = N, καθώς αυτό είναι υπολογιστικά πιο αποδοτικό.

Εν τέλει ο τύπος γίνεται ως εξής:

$$\hat{x}(k/N) = \sum_{n=0}^{N-1} x(n)\, e^{-i2\pi n\frac{k}{N}} \tag{4}$$

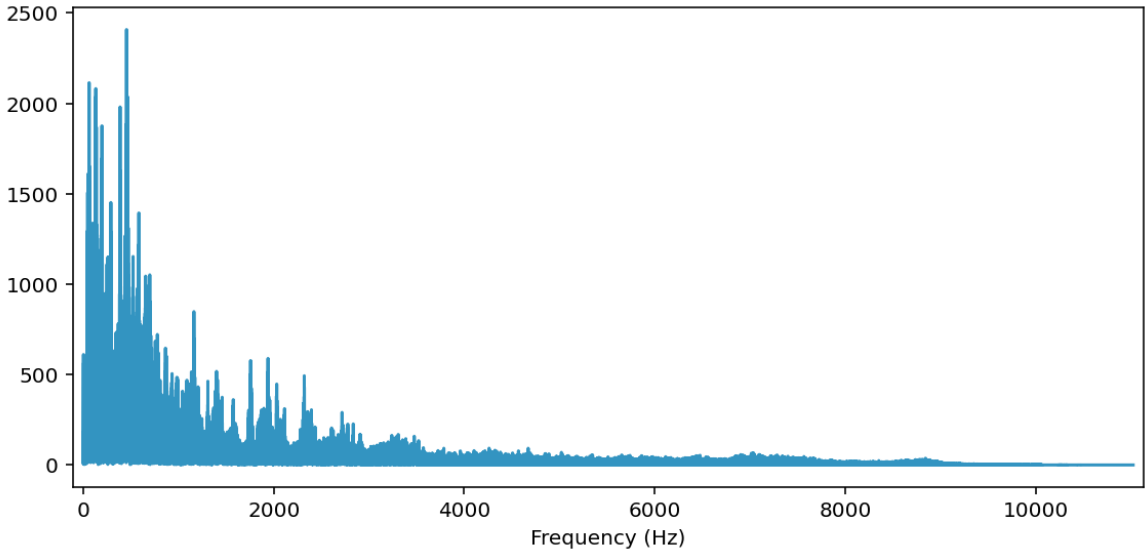
Ο τύπος (4) είναι ο τύπος της DFT. Στον τύπο (4) η συχνότητα ορίστηκε ως k/N όπου k = [0, N-1]. Ουσιαστικά η συχνότητα παίρνει πλέον διακριτές τιμές οι οποίες αντιστοιχούν σε ένα συγκεκριμένο εύρος ίσο με:

$$\mathrm{F}(k) = \frac{k}{N} \cdot s_r \tag{5}$$

Όπου s_r είναι ο ρυθμός δειγματοληψίας. Οι διακριτές αυτές τιμές συχνότητας που έχουν συγκεκριμένο εύρος, συχνά ονομάζονται “frequency bins” (McFee, 2023). Τα ψηφιακά συστήματα επεξεργασίας ήχου χρησιμοποιούν τον DFT και όχι τον DTFT, καθώς ο

δεύτερος θα χρειαζόταν άπειρες ημιτονοειδείς καμπύλες για την ανάλυση του σήματος, γεγονός που καθιστά την υλοποίησή του από αλγόριθμο αδύνατη (Smith, 2013).

Όπως σημειώνει ο Müller (2021), ένα βασικό μειονέκτημα του DFT είναι η υλοποίησή του, καθώς ένας αλγόριθμος που θα υλοποιούσε τον εν λόγω μετασχηματισμό θα είχε πολυπλοκότητα της τάξης του $O(N^2)$ λόγω των πράξεων που απαιτεί. Ο ίδιος αναφέρει πως λύση σε αυτό το πρόβλημα έδωσαν ο Gauss και ο Fourier 200 χρόνια πριν με τον αλγόριθμο που είναι γνωστός ως Ταχύς Μετασχηματισμός Fourier – Fast Fourier Transform (FFT). Οι Cooley και Tukey (1965) υλοποίησαν τον FFT, βελτιώνοντας την πολυπλοκότητα καθώς ο αλγόριθμος είναι της τάξης του $O(K \cdot \log K)$, δίνοντας μια αποδοτική εκτέλεση του DFT που χρησιμοποιείται μέχρι σήμερα στα συστήματα επεξεργασίας ήχου.



Εικόνα 4 Εφαρμογή DFT σε ακουστικό δείγμα του GTZAN dataset. Το πεδίο των συχνοτήτων (frequency domain)

Μέχρι στιγμής, είδαμε πως μπορούμε με τον μετασχηματισμό Fourier να μεταβαίνουμε από το πεδίο του ήχου στο πεδίο των συχνοτήτων. Στην Εικόνα 4 βλέπουμε την εφαρμογή του DFT στην κυματομορφή της Εικόνας 1 και τη μετάβαση αυτής στο πεδίο συχνοτήτων. Με την διάκριση των δυο αυτών πεδίων, παρατηρούμε πως προκύπτει ένας συμβιβασμός στην πληροφορία που παίρνουμε. Όπως αναφέρει η Hubbard (2010), η κυματομορφή απεικονίζει την εξέλιξη στο χρόνο, αποκρύπτοντας πληροφορίες για τη συχνότητα, ενώ ο μετασχηματισμός Fourier αποκαλύπτει τις συχνότητες αλλά δεν αποκαλύπτει τη χρονική πληροφορία Η ίδια σημειώνει πως με λίγα λόγια στην περίπτωση της μουσικής να μπορούμε να γνωρίζουμε είτε ποιες νότες (συχνότητες) παίζονται, ή πότε αυτές παίζονται, αλλά όχι

και τα δύο ταυτόχρονα. Στις Εικόνες 3 και 4 για παράδειγμα, βλέπουμε ποιες συχνότητες εμφανίζονται, αλλά δεν ξέρουμε πότε γίνεται αυτό. Μια πολύ σημαντική μέθοδος η οποία δίνει τη δυνατότητα να γνωρίζουμε το πότε εμφανίζεται η κάθε συχνότητα είναι ο Σύντομος μετασχηματισμός Fourier – Short-time Fourier transform (STFT), με τον οποίο εφαρμόζουμε DFT σε διαδοχικά χρονικά τμήματα (Knees & Schedl, 2016).

Η κεντρική ιδέα πίσω από τον STFT, είναι ο διαχωρισμός του ηχητικού σήματος σε επικαλυπτόμενα frames, στα οποία θα εφαρμοστεί ξεχωριστά ο DFT, αφού έχει προηγηθεί η εφαρμογή μιας windowing function (Müller, 2021). Η διαδικασία αυτή έχει ήδη περιγραφεί σε προηγούμενο υποκεφάλαιο και είναι ίδια και εδώ. Όπως λοιπόν εξηγεί ο Müller (2021), ο STFT σε ένα σήμα $x(n)$ όπου εφαρμόζουμε windowing function $w(n)$, δίνεται από τον τύπο:

$$S(m,k) = \sum_{n=0}^{N-1} x(n+mH) \cdot w(n) \cdot e^{-i2\pi n\frac{k}{N}} \tag{6}$$

Όπου n είναι το ο αριθμός του sample, N ο συνολικός αριθμός samples, H το hop length, k είναι το frequency bin και m είναι το time frame στο οποίο εφαρμόζουμε τον DFT. Όπως αναφέρθηκε, ο μετασχηματισμός Fourier δίνει coefficients σε μιγαδική μορφή, που περιέχουν πληροφορίες για το μέτρο της έντασης και τη φάση μιας συχνότητας, ή εν προκειμένω ενός frequency bin. Εδώ λοιπόν παρατηρούμε ότι μας δίνονται αυτές οι πληροφορίες συναρτήσει και του time frame. Στην περίπτωση του DFT λοιπόν, η πληροφορία που εξάγουμε μπορεί να παρασταθεί σαν ένα διάνυσμα (τανυστής 1ου βαθμού) με στοιχεία όσα και τα frequency bins, καθώς έχουμε ένα Fourier coefficient για κάθε frequency bin (Deshpande et al., 2021). Αντίστοιχα, στην περίπτωση του STFT, η πληροφορία που εξάγουμε παρίσταται με έναν πίνακα (τανυστής 2ου βαθμού) όπου ο ένας βαθμός του τανυστή αντιστοιχεί στο frequency bin και ο άλλος στο time frame, καθώς έχουμε ένα Fourier coefficient για κάθε frequency bin του κάθε time frame (Liu et al., 2022).

Ο McFee (2023) εξηγεί τους μαθηματικούς τύπους για τον υπολογισμό των frequency bins και των time frames αυτού του πίνακα, οι οποίοι είναι οι εξής:

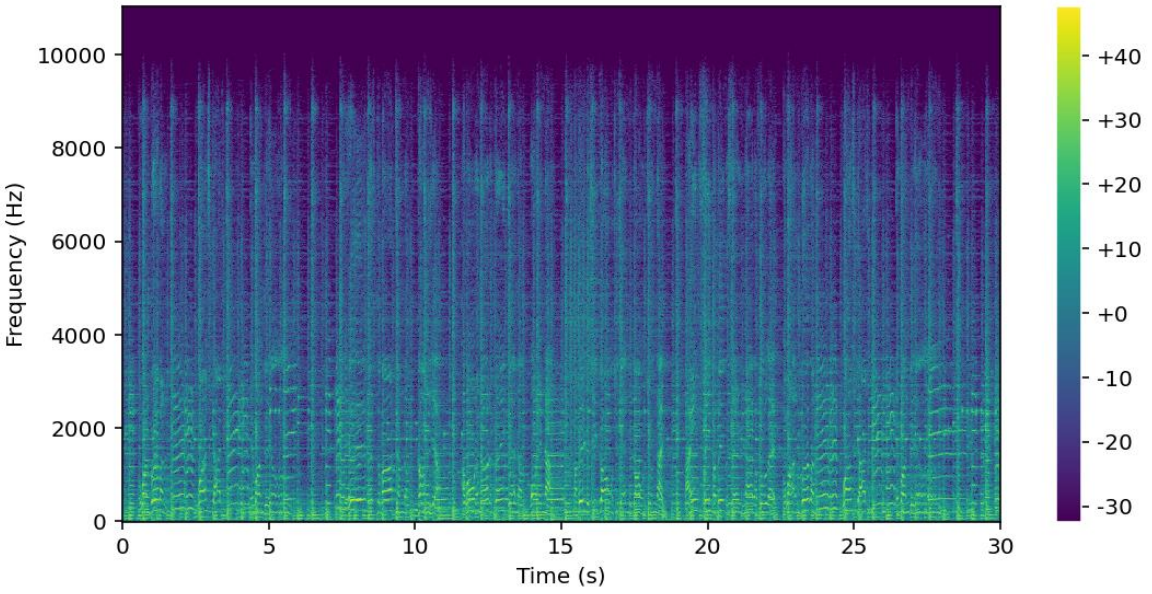
$$N_{frequency\ bins} = \left\lfloor \frac{N_F}{2} \right\rfloor + 1 \tag{7}$$

Όπου N_F είναι το ο αριθμός δειγμάτων του κάθε time frame.

$$N_{time\ frames} = \left\lfloor \frac{N - N_F}{N_H} \right\rfloor + 1 \quad (8)$$

Όπου N ο συνολικός αριθμός δειγμάτων και N_Hτο hop size.

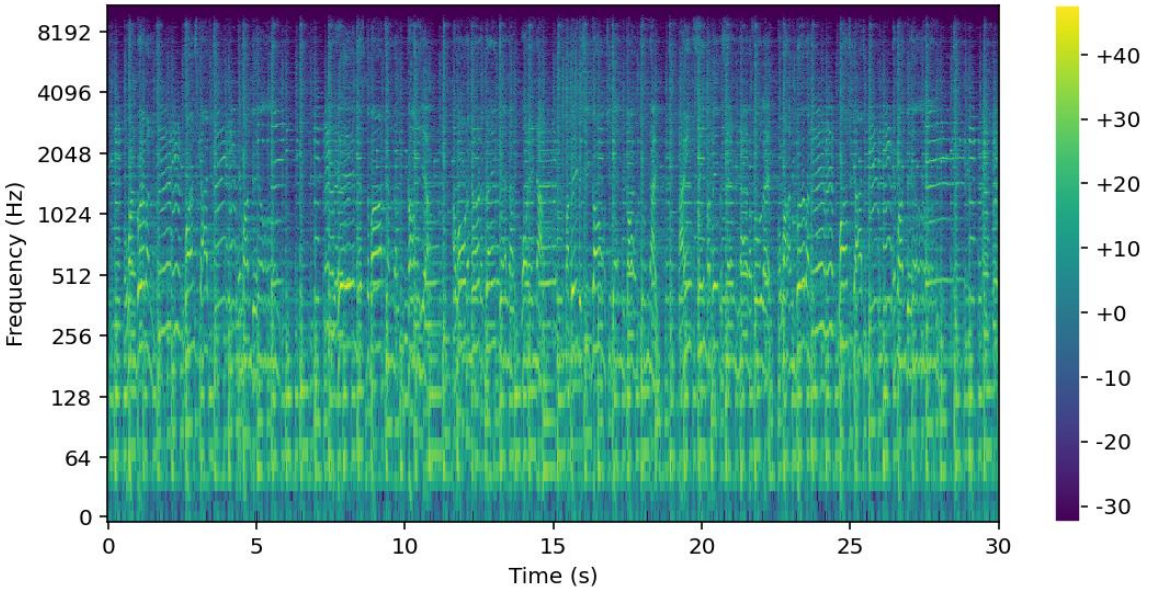
Ένα ενδιαφέρον φαινόμενο κατά την επιλογή του μεγέθους των frames είναι το γεγονός ότι η επιλογή ενός μεγαλύτερου frame θα μας δώσει περισσότερες πληροφορίες για τις συχνότητες, καθώς θα δούμε περισσότερες συχνότητες που εμφανίζονται, όμως δεν θα έχουμε καλή πληροφόρηση για το πότε αυτές εμφανίζονται, ενώ ένα μικρότερο frame θα έχει το αντίστροφο αποτέλεσμα (Müller, 2021). Η επιλογή του μεγέθους του frame και το κατά πόσο θα έχουμε περισσότερη πληροφορία για το «ποιες συχνότητες» ή το «πότε εμφανίζονται, αναφέρεται ως "time-frequency trade-off" και είναι κάτι που κανείς καλείται να αποφασίσει κατά τη δημιουργία του εκάστοτε συστήματος (McFee, 2023). Το γεγονός αυτό, ότι δεν υπάρχει καμία windowing function που θα προσφέρει ταυτόχρονο εντοπισμό συχνοτήτων και χρόνου με απόλυτη ακρίβεια, είναι μια έκφραση της αρχής απροσδιοριστίας του Heisenberg (Strang & Nguyen, 1996).



Εικόνα 5 Το φασματογράφημα (spectrogram) ηχητικού δείγματος blues από το GTZAN dataset

Ο Mitra (1998), αναφέρει ότι στην πράξη, αυτό που κυρίως μας ενδιαφέρει είναι το μέγεθος της συχνότητας όπως παρουσιάζεται από τον STFT. Ο ίδιος, συνεχίζει λέγοντας ότι αυτή η παρουσίαση γίνεται με το φασματογράφημα (spectrogram), το οποίο όμως θα είναι ένα διάγραμμα τριών μεταβλητών (χρόνος, συχνότητα, ένταση συχνότητας) που για να παρουσιάσουμε σε δύο διαστάσεις, εμφανίζουμε την ένταση συχνότητας με διαφορετικό

χρώμα. Στην Εικόνα 5, βλέπουμε το spectrogram της κυματομορφής που είχαμε στην Εικόνα 1. Παρατηρούμε χαρακτηριστικά την εμφάνιση των συχνοτήτων και την έντασή τους στην κλίμακα dB, καθώς μεγαλύτερη ένταση σημαίνει εντονότερο χρώμα. Κατά την παρουσίαση του spectrogram, κάτι που συνήθως γίνεται είναι η επεξεργασία των αξόνων, καθώς όπως αναφέρθηκε σε προηγούμενο υποκεφάλαιο, η αντίληψη της συχνότητας και της έντασης του ήχου από τον άνθρωπο είναι λογαριθμικές. Αυτό που συνηθίζεται λοιπόν είναι η ένταση να παρουσιάζεται σε άξονα ο οποίος είναι βαθμονομημένος σε λογαριθμική κλίμακα decibel (Christensen, 2019).



Εικόνα 6 Log-frequency spectrogram ηχητικού δείγματος blues από το GTZAN dataset.

Ο άξονας που περιέχει τα frequency bins επίσης αρκετές φορές παρουσιάζεται σε λογαριθμική κλίμακα, περίπτωση στην οποία το φασματογράφημα πολλές φορές αναφέρεται ως "log-frequency spectrogram" (Müller, 2021). Αυτές οι μετατροπές πράγματι παρουσιάζουν διαγράμματα όπου οι ακουστικές μεταβολές και τα γενικότερα ακουστικά φαινόμενα γίνονται οπτικώς πιο διακριτά. Στην Εικόνα 6 για παράδειγμα, βλέπουμε εμφανή διαφορά με την Εικόνα 5. Η κατανομή των συχνοτήτων είναι πολύ διαφορετική, πράγμα που αποτελεί κατά κάποιον τρόπο μια «οπτικοποιημένη» εκδοχή της λογαριθμικής αντίληψης της συχνότητας από τον άνθρωπο που ήδη αναλύθηκε.

2.2.6 Mel Spectrograms

Με τη χρήση του STFT είδαμε πως μπορούμε να οπτικοποιήσουμε τα ακουστικά φαινόμενα και να μεταβάλλουμε τους άξονες των διαγραμμάτων ώστε να ανταποκρίνονται καλύτερα

στην ανθρώπινη αντίληψη. Επίσης αναφέραμε ότι αυτά τα δεδομένα μπορούν να αναπαρασταθούν με εικόνες, αλλά και να αποθηκευτούν αριθμητικά ως τανυστές 2ου βαθμού, ώστε να αποτελέσουν δεδομένα εισόδου για νευρωνικά δίκτυα. Η «οπτικοποίηση» των ακουστικών φαινομένων όμως, δε σταματά εκεί καθώς υπάρχουν κι άλλες μέθοδοι να εξάγουμε ακουστικά χαρακτηριστικά με τον τρόπο που ο άνθρωπος τα αφουγκράζεται.

Το Mel Spectogram (ή log-Mel Spectogram) είναι μια από τις πιο δημοφιλείς μεθόδους εξαγωγής ακουστικών χαρακτηριστικών, ώστε να τροφοδοτήσουμε νευρωνικά δίκτυα (Purwins et al., 2019). Στο Mel Spectogram, αντί για τα frequency bins που έχουμε στον άξονα των συχνοτήτων, έχουμε τις επονομαζόμενες "mel bands" (Toyoshima et al., 2023). Ο Lerch (2021), αναφέρει ότι το συγκεκριμένο φασματογράφημα αναπαριστά αρκετά πιστά τη χροιά και το ύψος των μουσικών ήχων και υπολογίζεται κατόπιν ομαδοποίησης των frequency bins που προκύπτουν από το φασματογράφημα του STFT, σε επικαλυπτόμενες συχνοτικές ομάδες (mel bands). Ο ίδιος αναφέρει ότι με τον τρόπο αυτό έχουμε μια καλύτερη εξομοίωση του τρόπου που ακούει το ανθρώπινο αυτί σύμφωνα με την κλίμακα mel, ενώ κατά μήκος του άξονα συχνοτήτων, η ανάλυση (resolution) παρατηρούμε ότι μειώνεται λογαριθμικά. Οι Stevens et al. (1937), εισήγαγαν τον όρο "Mel" εκ του "melody" ως μέτρο ύψους ενός ήχου ή μιας νότας. Οι ίδιοι κατέληξαν στην κλίμακα mel, διεξάγοντας ψυχοακουστικά πειράματα για να εντοπίσουν ίσες διαφορές στο ύψος των ήχων. Στην κλίμακα Mel, η σχέση Hertz και Mel είναι κατά προσέγγιση γραμμική για συχνότητες κάτω των 500 Hz ενώ κατόπιν η διαφορά των Hz που αντιστοιχούν σε ίσες τιμές Mel και αντίληψη του ύψους, μεγαλώνει σταδιακά (Knees & Schedl, 2016). Οι Chakroborty et al. (2008), περιγράφουν τις μετατροπές από Hz σε mel και αντιστρόφως. Συγκεκριμένα, αναφέρουν πως για να μετατρέψουμε τα Hz σε Mels έχουμε:

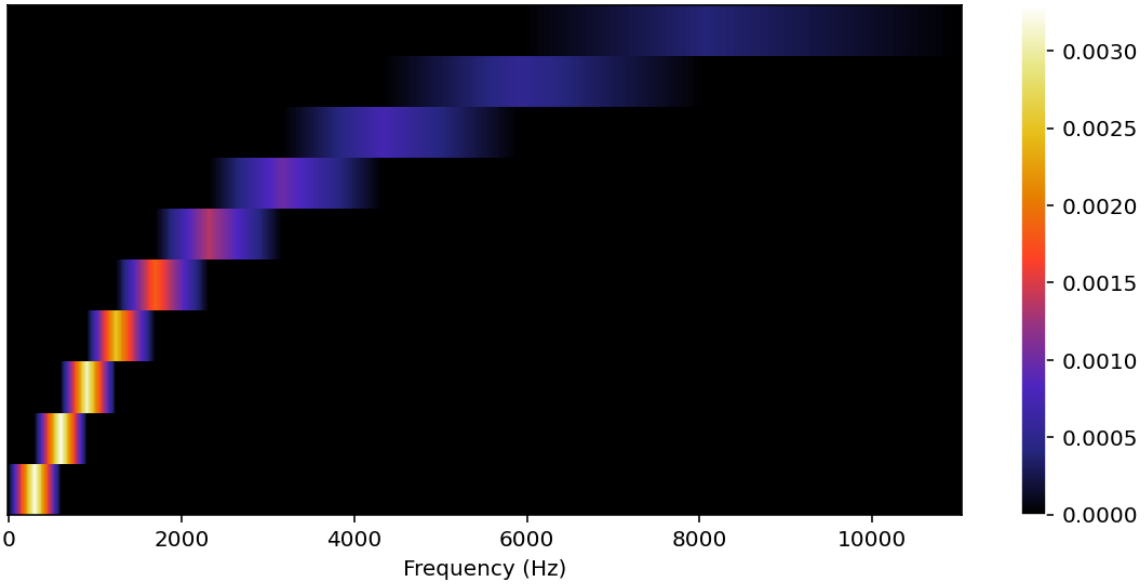
$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{9}$$

Όπου f είναι η συχνότητα σε Hz και m είναι το ύψος του ήχου σε Mels. Αντίστροφα, για μετατροπή από Mels σε Hz, έχουμε:

$$f = 700\left[10\frac{m}{2595} - 1\right] \tag{10}$$

Αξίζει να σημειωθεί ότι στην κλίμακα Mel, τα 1000 Hz αντιστοιχούν σε 1000 Mels, ενώ μια νότα ύψους 500 Mels (το μισό του 1000) αντιστοιχεί σε 390 Hz, όμως μια νότα 2000 Mels (το διπλάσιο του 1000) αντιστοιχεί σε 3429 Hz (Rabiner & Schafer, 2011).

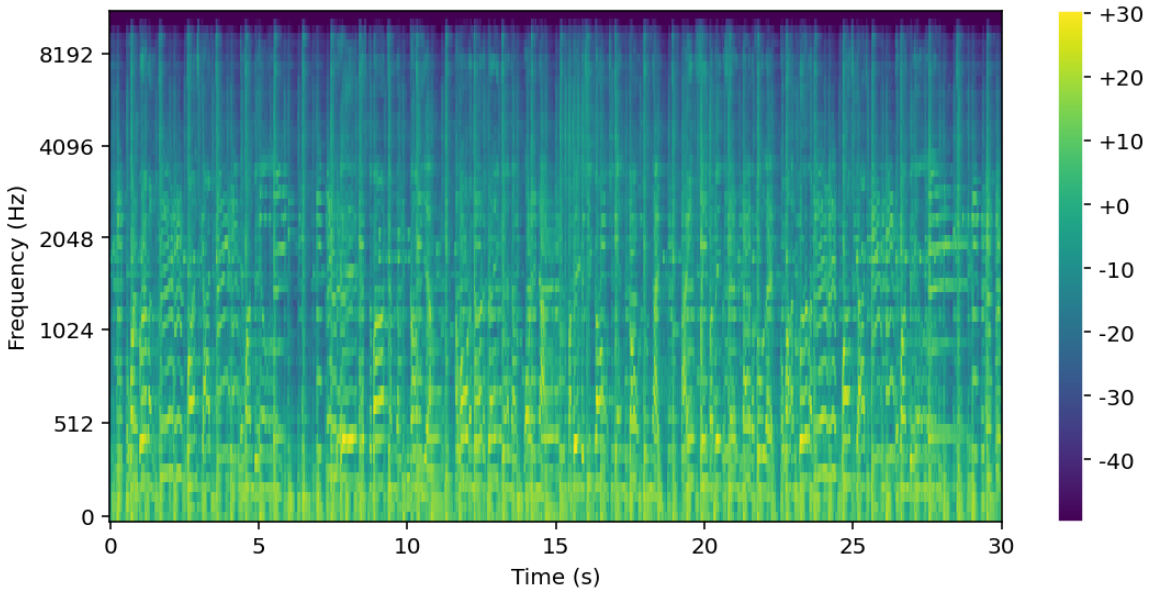
Η διαδικασία εξαγωγής του Mel Spectrogram περιλαμβάνει κάποια συγκεκριμένα βήματα. Θα πρέπει αρχικά να εφαρμοστεί STFT ώστε να πάρουμε το spectrogram, να γίνει μετατροπή του άξονα της έντασης σε dB και να γίνει μετατροπή των συχνοτήτων στην κλίμακα Mel, επιλογή κατάλληλων υπερπαραμέτρων όπως ο αριθμός των mel banks, ενώ τέλος πιθανώς χρειαστεί κάποια κανονικοποίηση των αριθμητικών δεδομένων (Donahue et al., 2018). Η ίδια η μετατροπή όμως των συχνοτήτων στην κλίμακα Mel, απαιτεί κάποια βήματα από μόνη της. Αρχικά απαιτείται η επιλογή του αριθμού των Mel bands, κάτι που σχετίζεται με την έννοια των “critical bands”. Η ιδέα των “critical bands” εισήχθη από τον Fletcher και αφορά το εύρος ενός φάσματος ακουστικών συχνοτήτων στο οποίο ο άνθρωπος αντιλαμβάνεται έναν ήχο (Fastl & Zwicker, 2007). Μια αποδοτική μέθοδος για την οργάνωση των συχνοτήτων σε critical bands επιτυγχάνεται χρήση της κλίμακας mel, καθώς αντιστοιχούμε ένα εύρος συχνοτήτων γύρω από μια «κεντρική συχνότητα», ενώ φαίνεται πως για συχνότητες κάτω των 500 Hz αυτό το εύρος είναι μικρό ενώ στη συνέχεια διευρύνεται (Rabiner & Schafer, 2011). Η επιλογή του αριθμού αυτών των critical bands που ουσιαστικά είναι οι mel bands που αναφέρθηκαν πριν, είναι πάντα σχετική και εξαρτάται από την περίπτωση, ενώ είναι αντικείμενο ευρείας μελέτης. Οι Xuedong. Huang et al. (2001) εξηγούν λεπτομερώς τη διαδικασία αντιστοιχίας των frequency bins σε mel banks, που περιλαμβάνει τη χρήση τριγωνικών φίλτρων.



Εικόνα 7 Mel Filters. Αντιστοιχία συχνοτήτων σε 10 διαφορετικά mel bands

Αρχικά χρειάζεται να μετατρέψουμε τις ακραίες τιμές συχνότητας σε mel, να χωρίσουμε την κλίμακα που προκύπτει σε ίσα μέρη και κατόπιν χρήσει των τριγωνικών φίλτρων

βρίσκουμε τις επικαλυπτόμενες mel bands. Δεδομένου ότι είναι επικαλυπτόμενες, κάθε frequency bin μπορεί να συνεισφέρει σε 2 bands. Ο O'Shaughnessy (2000) εξηγεί το πως αυτό συμβαίνει βάση μιας κλίμακας ισχύος που αντιστοιχεί σε κάθε συχνότητα και περιλαμβάνεται στα τριγωνικά φίλτρα που εφαρμόζονται. Στην Εικόνα 7 βλέπουμε πως η εφαρμογή αυτών των φίλτρων μπορεί να δημιουργήσει αντιστοιχία με συχνότητες. Στον κατακόρυφο άξονα της εικόνας, διακρίνονται οι 10 διαφορετικές mel bands και με εντονότερα χρώματα βλέπουμε πως οι συχνότητες του οριζόντιου άξονα συνεισφέρουν σε κάθε μια από αυτές. Εμφανής είναι επίσης, η επικάλυψη μεταξύ τους.



Εικόνα 8 Το Mel spectrogram ηχητικού δείγματος blues από το GTZAN dataset.

Στην Εικόνα 8 βλέπουμε πως διαμορφώνεται το Mel spectrogram του ηχητικού δείγματος της κυματομορφής από την εικόνα 1. Παρατηρούμε ότι παραπέμπει στην Εικόνα 6, απλά η κατανομή των συχνοτήτων είναι διαφορετική. Στην πράξη, το Mel Spectrogram μπορεί να εξαχθεί χρήσει λογισμικού που χρησιμοποιεί τανυστές 2ου βαθμού, δηλαδή πίνακες. Όπως αναφέρουν οι Masuyama et al. (2025), τα τριγωνικά φίλτρα μπορούν να παρασταθούν ως ένας πίνακας με γραμμές τα mel bands και στήλες τα αντίστοιχα frequency bins. Δεδομένου ότι το spectrogram όπως προκύπτει από τον STFT είναι ένας πίνακας με γραμμές τα frequency bins και στήλες τα time frames, όπως γνωρίζουμε από τη γραμμική άλγεβρα, είναι εφικτό το γινόμενο των πινάκων αυτών. Η πράξη αυτή μάλιστα, θα μας δώσει το Mel spectrogram ως έναν πίνακα με γραμμές τα mel bands και στήλες τα time frames.

2.2.7 Mel-Frequency Cepstral Coefficients (MFCCs)

Σε προηγούμενα υποκεφάλαια, είδαμε το πως μπορούμε να «οπτικοποιήσουμε» τον ήχο και τη μουσική υπό το πρίσμα της ανθρώπινης αντίληψης και πως προκύπτουν τα φασματογραφήματα που παρουσιάζουν ακουστικά χαρακτηριστικά. Συγκεκριμένα, είδαμε πως αυτά παρουσιάζονται σε spectrograms, log-frequency spectrograms και Mel spectrograms και πως μπορούν να οργανωθούν σε αριθμητικές δομές μέσω τανυστών, ώστε να αποτελέσουν δεδομένα εισόδου για περαιτέρω επεξεργασία. Πέρα από τα προαναφερθέντα φασματογραφήματα, μια εξαιρετικά διαδεδομένη μέθοδος αναπαράστασης ακουστικών χαρακτηριστικών είναι τα Mel-Frequency Cepstral Coefficients (MFCCs). Μάλιστα, για δεκαετίες τα MFCCs χρησιμοποιούνται ευρέως ως ο βασικός τρόπος αναπαράστασης ακουστικών χαρακτηριστικών για εφαρμογές ακουστικής ανάλυσης (Purwins et al., 2019). Όσον αφορά τη χρήση τους στην επεξεργασία ήχου, τα MFCCs αρχικά χρησιμοποιούνταν στην ανάλυση της ανθρώπινης ομιλίας, αλλά με τον καιρό παρατηρήθηκε ότι μπορούν να αναπαραστήσουν τη χροιά του ήχου γενικότερα (Knees & Schedl, 2016). Η πρώτη εξαγωγή και χρήση τους έγινε το 1980 από τους Davis και Mermelstein οι οποίοι πρότειναν μια λεπτομερή μέθοδο υπολογισμού τους για χρήση σε επεξεργασία σημάτων ανθρώπινης φωνής (Davis & Mermelstein, 1980). Την επόμενη δεκαετία όμως, άρχισαν να χρησιμοποιούνται και για ταξινόμηση ήχων, με χαρακτηριστική την περίπτωση του Foote που παρουσίασε μια μέθοδο ταξινόμησης μουσικών ήχων (J. T. Foote, 1997). Σήμερα, τα MFCCs χρησιμοποιούνται περισσότερο από κάθε άλλο ακουστικό χαρακτηριστικό στα βασικά συστήματα επεξεργασίας ήχου, καθώς έχουν αποδειχθεί ιδιαίτερα αξιόπιστα και χρήσιμα για πληθώρα εργασιών (Lerch, 2022).

Μια αρχική προσέγγιση για την κατανόηση των MFCCs, θα μπορούσε να γίνει αναλύοντας τις λέξεις που υπάρχουν στον τίτλο. Ο όρος “Mel-Frequency” αναφέρεται στην κλίμακα Mel που αναλύθηκε σε προηγούμενο υποκεφάλαιο, ενώ ο όρος “coefficients” σημαίνει συντελεστές. Ο άγνωστος όρος είναι ο όρος “Cepstral”. Οι Bogert et al. (1963), στη μελέτη τους πάνω στην ηχώ των σεισμικών δονήσεων εισήγαγαν μια λίστα από ασυνήθιστους όρους όπως ο όρος “cepstrum” που δεν είναι τίποτα άλλο από ένα λογοπαίγνιο πάνω στον όρο “spectrum” που σημαίνει φάσμα. Οι ίδιοι στην έρευνά τους παρέθεσαν μια σειρά όρων όπως “alanysis” αντί για “analysis” που αφορά την ανάλυση του cepstrum, “liftering” αντί για “filtering” που σχετίζεται με το φιλτράρισμα, “quefrency” αντί για “frequency” που αφορά συχνότητες, και “rahmonic” αντί για “harmonic” το οποίο σχετίζεται με τις

αρμονικές που συνθέτουν ένα σύνθετο κύμα. Το "cepstral" λοιπόν προέρχεται από τον όρο "cepstrum". Για την κατανόηση αυτών των όρων, βοηθά το να δει κανείς πως το ίδιο το cepstrum υπολογίζεται.

Σε προηγούμενα υποκεφάλαια είδαμε πως προκύπτει το log-spectrum από τον STFT και τη λογαρίθμιση του άξονα της έντασης. Το cepstrum γενικά υπολογίζεται από την εφαρμογή κάποιας μορφής του μετασχηματισμού Fourier στο log-spectrum (Young et al., 2006). Πιο συγκεκριμένα, για σήματα που προέρχονται από δειγματοληψία, το cepstrum υπολογίζεται ως ο αντίστροφος διακριτός μετασχηματισμός Fourier – Inverse discrete-time Fourier transform (IDTFT) του log-spectrum όπως αυτό έχει προκύψει από έναν DTFT ενός σήματος (Rabiner & Schafer, 2011). Αντί λοιπόν να λέμε ότι έχουμε το "spectrum ενός spectrum", λέμε ότι έχουμε ένα cepstrum. Η μεταβλητή του οριζόντιου άξονα στο cepstrum αντιστοιχεί στην quefrency, όπως ακριβώς η μεταβλητή του οριζόντιου άξονα στο spectrum, αντιστοιχεί στη frequency, δηλαδή τη συχνότητα (Bogert et al., 1963). Όπως και στην περίπτωση του spectrum όπου συχνότητες που εμφανίζονται σε αυτό, είναι αυτές που συνθέτουν την κυματομορφή, έτσι και εδώ οι quefrencies συνθέτουν το λογαριθμισμένο spectrum. Αποδεικνύεται ότι οι χαμηλές quefrencies αντιστοιχούν σε μέρη του αρχικού σήματος που αφορούν αργές ταλαντώσεις, ενώ οι υψηλές, σε γρήγορες ταλαντώσεις (Rabiner & Schafer, 2011). Αυτό όπως θα σημειωθεί παρακάτω είναι αρκετά σημαντικό.

Για την καλύτερη κατανόηση της εφαρμογής του cepstrum στη μουσική και στη συγκρότηση των MFCCs, προτείνεται η μελέτη της εφαρμογής του στην ανάλυση της ομιλίας. Οι Rabiner & Schafer (1978), εξηγούν πως οι ήχοι της ομιλίας παράγονται από παλμούς του αέρα που διαπερνούν τη γλωττίδα (το άνοιγμα των φωνητικών χορδών), ενώ ο αέρας εν συνεχεία ταξιδεύει δια μέσου της φαρυγγο-λαρυγγικής κοιλότητας (γνωστή και ως φωνητική οδός). Οι ίδιοι εξηγούν πως η όλη διαδικασία θυμίζει τα πνευστά όργανα όπου ο αέρας διαπερνά το όργανο και όπως εκεί, οι συχνότητες και τα χαρακτηριστικά του ήχου, διαμορφώνονται από τη φωνητική οδό η οποία διαφέρει από άνθρωπο σε άνθρωπο. Οι αντηχήσεις αυτές στις κοιλότητες όπως η φωνητική οδός, προσδίδουν στη χροιά του ήχου και λέγονται "formants" (Xuedong. Huang et al., 2001). Εν τέλει, αυτό που φαίνεται είναι πως η ομιλία, μπορεί να παρουσιαστεί σαν μια συνέλιξη (convolution) δύο σημάτων: της απόκρισης των παλμών από τη φωνητική οδό και του παλμού της γλωττίδας (Noll, 1967). Αν λοιπόν έχουμε ένα σήμα ομιλίας $x(t)$, τότε εάν το σήμα του αέρα που περιοδικά

διαπερνά τη γλωττίδα κατά την ομιλία είναι $s(t)$ και η απόκριση της φωνητικής οδού είναι $h(t)$, τότε ισχύει:

$$x(t) = s(t) * h(t) \tag{11}$$

Ενώ αν εφαρμόσουμε μετασχηματισμό Fourier για να μεταβούμε στο πεδίο των συχνοτήτων, θα ισχύει:

$$X(f) = S(f) \cdot H(f) \tag{12}$$

Όπου παρατηρούμε ότι αντί για την πράξη της συνέλιξης που είχαμε στον τύπο (11), έχουμε πλέον γινόμενο. Καθώς λοιπόν κατόπιν εφαρμόζουμε λογαρίθμιση, θα ισχύει

$$\log X(f) = \log S(f) + logH(f) \tag{13}$$

Η όλη αντιμετώπιση των σημάτων που είναι αποτέλεσμα συνέλιξης και η εν τέλει αντιμετώπισή τους ως αθροίσματα μέσω της χρήσης λογαρίθμου, είναι αποτέλεσμα της μελέτης του A. V Oppenheim (1965). Ο ίδιος, μελετώντας μη γραμμικά συστήματα επεξεργασίας σημάτων, εισήγαγε μια μεθοδολογία σύμφωνα με την οποία σήματα που είναι αποτέλεσμα συνέλιξης μπορούν να μετασχηματιστούν σε κατάλληλο πεδίο, όπου η συνέλιξη αντιστοιχίζεται σε άθροισμα, ενώ τα συστήματα που άγονται σε αυτήν την αντιστοίχιση ονομάστηκαν “homomorphic systems”.

Βλέπουμε λοιπόν πως τα formants που είναι χαρακτηριστικά της χροιάς του ήχου, βρίσκονται στο 2ο μέρος της εξίσωσης (13) και συνεπώς αυτό το μέρος αποτελεί τη χρήσιμη πληροφορία. Ο διαχωρισμός αυτός επιτυγχάνεται μέσω της προαναφερθείσας διαδικασίας του “liftering” (A. V. Oppenheim & Schafer, 2004). Τα formants λοιπόν βρίσκονται στις χαμηλότερες τιμές της quefrency του cepstrum, οι οποίες αντιστοιχούν στα peaks των «αργών» ταλαντώσεων του διαγράμματος του log-spectrum (Rabiner & Schafer, 1978). Αυτές οι «αργές» ταλαντώσεις είναι ουσιαστικά η περιβάλλουσα (envelope) του log spectrum, μια έννοια που αναλύθηκε σε προηγούμενο υποκεφάλαιο. Εν τέλει λοιπόν με αυτή τη διαδικασία, στα MFCCs παραμετροποιείται η περιβάλλουσα αυτού του φασματογραφήματος (Müller, 2021). Γενικότερα, οι χαμηλές τιμές του quefrency αντιστοιχούν στη δομή της φωνητικής οδού και κατ’ επέκτασιν στα formants και τη χροιά του ήχου, ενώ οι ψηλότερες, αντιστοιχούν στους παλμούς του αέρα που διαπερνούν τη γλωττίδα (A. V. Oppenheim & Schafer, 2004).

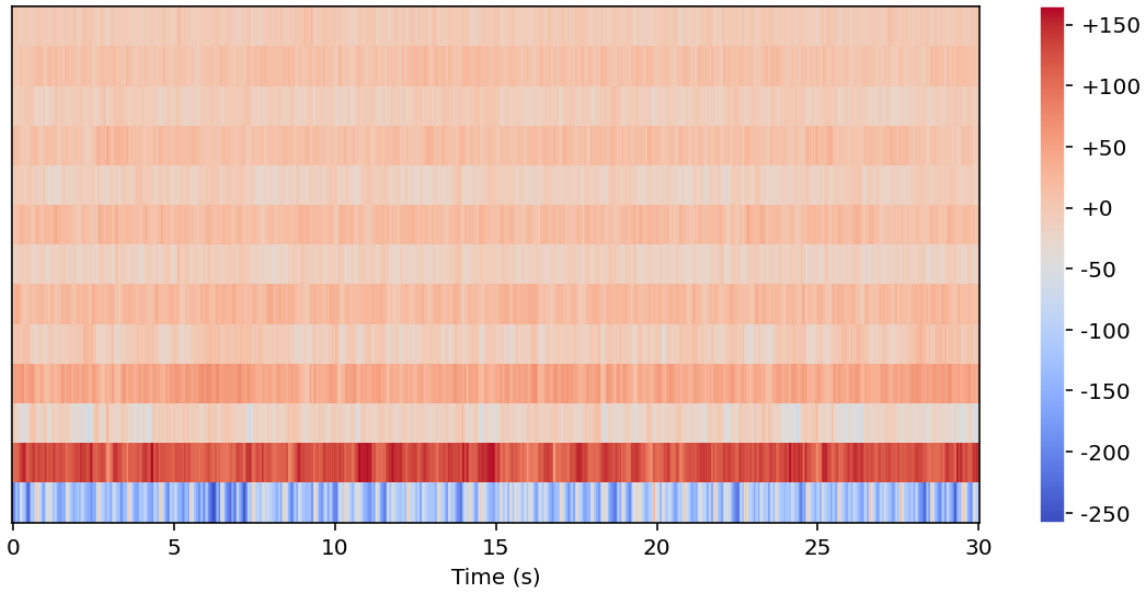
Τα παραπάνω αποτελούν τη θεωρητική προσέγγιση της εξαγωγής των formants που ουσιαστικά εμπεριέχουν πληροφορία για τη χροιά του ήχου. Στην πράξη, η διαδικασία γίνεται συνήθως με κάποια συγκεκριμένα βήματα που θα αναλύσουμε. Οι Davis και Mermelstein (1980), περιγράφουν πως το σήμα υφίσταται τα εξής βήματα για την εξαγωγή των MFCCs: windowing, DFT, λογαρίθμιση της έντασης των συχνοτήτων, μετατροπή των συχνοτήτων σε κλίμακα mel και τέλος διακριτό μετασχηματισμό συνημίτονου. Βλέπουμε λοιπόν πως αντί για αντίστροφο μετασχηματισμό Fourier στο log spectrum, έχουμε τον διακριτό μετασχηματισμό συνημίτονου – Discrete cosine transform (DCT), ενώ οι συχνότητες παρουσιάζονται στην κλίμακα Mel, γεγονός που δικαιολογεί το "Mel" στα MFCCs. Οι Sahidullah και Saha (2012), αναλύουν λεπτομερώς τα πλεονεκτήματα του DCT έναντι άλλων μετασχηματισμών και αναφέρουν πως αυτός είναι η συνηθέστερη επιλογή ως τελικό βήμα, ενώ δεδομένου ότι τα MFCCs θα εξαχθούν σε μορφή πίνακα, προσθέτουν πως είναι σύνηθες ένα ακόμα βήμα στη διαδικασία εξαγωγής τους. Αυτό είναι η προσθήκη μηδενικών (zero padding) μετά το windowing του σήματος, καθώς οι διαστάσεις των πινάκων πρέπει να είναι συγκεκριμένες ούτως ώστε να είναι εφικτός ο πολλαπλασιασμός τους στα επόμενα βήματα. Εν τέλει, η διαδικασία αυτή θα δώσει έναν πίνακα (τανυστή 2ου βαθμού) όπου οι γραμμές αντιστοιχούν στα coefficients (συντελεστές) και οι στήλες στα time frames όπως αυτά προκύπτουν από το windowing του πρώτου βήματος. Οι H. Kim et al. (2005), παρουσιάζουν τον τύπο του DCT ως:

$$c_i = \sum_{j=1}^{N_f} \left\{ \log(E_j) \cos\left[i \left(j - \frac{1}{2} \right) \frac{\pi}{N_f} \right] \right\} \ (1 \le i \le N_c) \qquad (14)$$

Το c_iείναι το i-οστό MFCC. Με τον όρο E_j αναπαριστούμε την ισχύ του σήματος εντός του συγκεκριμένου mel band, όπως αυτή ορίζεται από το τετράγωνο του πλάτους του σήματος και αναφέρεται στη βιβλιογραφία ως "spectral energy". N_f είναι ο συνολικός αριθμός των τριγωνικών φίλτρων mel που εφαρμόζουμε, N_c είναι ο συνολικός αριθμός των MFCC που θα πάρουμε για κάθε time frame. Ήδη από τη σχέση (14), μπορούμε να εντοπίσουμε κάποια προτερήματα του DCT. Συγκεκριμένα, εντοπίζουμε πως το DCT μας δίνει απευθείας συντελεστές σε μορφή πραγματικού αριθμού και όχι μιγαδικού όπως είδαμε ότι κάνει ο DFT, από όπου υπολογίζαμε το πραγματικό μέρος. Επίσης, η ύπαρξη του όρου E_j σηματοδοτεί ότι η συμβολή της κάθε συχνότητας στην τελική spectral energy του κάθε mel band υπολογίζεται ανεξάρτητα από το σε πόσες mel bands συμβάλλει η κάθε συχνότητα. Η

σημαντικότητα αυτής της ιδιότητας είναι διακριτή καθώς είδαμε σε προηγούμενο υποκεφάλαιο ότι τα τριγωνικά φίλτρα που εφαρμόζουμε για τον υπολογισμό της κλίμακας mel, επικαλύπτονται και μια συχνότητα μπορεί να συμβάλλει σε 2 mel bands.

Το ερώτημα που προκύπτει στη συνέχεια, είναι το πόσα MFCCs θα πρέπει να υπολογίσουμε, ώστε να έχουμε μια καλή αναπαράσταση των ακουστικών χαρακτηριστικών. Οι Knees και Schedl (2016) σημειώνουν πως ο τυπικός αριθμός MFCCs που συνήθως υπολογίζεται είναι μεταξύ 13 και 25. Ο Lerch (2022) αναφέρει πως ειδικά για προβλήματα ταξινόμησης, οι απαραίτητες πληροφορίες βρίσκονται ήδη στα πρώτα MFCCs και ο απαραίτητος αριθμός που χρειάζεται είναι μεταξύ 4 και 20. Ο Eronen (2001), στην έρευνά του για την αναγνώριση μουσικών οργάνων σε μουσικά σήματα, παρατηρεί ότι ο ιδανικός αριθμός ήταν 12 MFCCs, ενώ περισσότερα ή λιγότερα MFCCs έριχναν την απόδοση των μοντέλων του. Οι Eronen et al. (2006), σε μελέτη μοντέλων αναγνώρισης ήχων ενός τυπικού αστικού περιβάλλοντος, επίσης χρησιμοποίησαν 11-12 MFCCs, ενώ τα MFCCs ως δεδομένα εισόδου είχαν την καλύτερη απόδοση συγκριτικά με άλλες απεικονίσεις ακουστικών χαρακτηριστικών. Σε μελέτη αναγνώρισης του ρεφραίν από κομμάτια, ο Eronen (2007) επίσης χρησιμοποιεί 12 MFCCs, ενώ τονίζει πως σημασία έχουν και άλλες υπερπαράμετροι όπως ο αριθμός mel bands και το window size.



Εικόνα 9 Κατανομή για 13 MFCCs ακουστικού δείγματος blues του GTZAN dataset

Εν τέλει, τα MFCC απεικονίζονται ως ένα heatmap όπου στον κάθετο άξονα έχουμε τα MFCCs και στον οριζόντιο άξονα τα time frames. Ο αριθμός που επιλέξαμε για τα MFCCs

είναι διακριτός στον κάθετο άξονα, καθώς παρατηρούμε αντίστοιχο αριθμό τετραγώνων. Στην Εικόνα 9 όπου έχουμε τα MFCCs της κυματομορφής της Εικόνας 1, αυτό είναι διακριτό. Παρατηρούμε επίσης ότι αυτή η απεικόνιση διαφέρει από τα προηγούμενα φασματογραφήματα που είδαμε. Επιπροσθέτως, ενώ στα προηγούμενα φασματογραφήματα μπορεί κανείς να διακρίνει τη φυσική σημασία τους, καθώς είναι εμφανές για παράδειγμα το πότε εμφανίζεται ένας δυνατός ήχος, εδώ δε συμβαίνει αυτό. Ο Lerch (2022), μέσω της διαδικασίας εξαγωγής των MFCCs, εξηγεί το τί βλέπουμε σε ένα τέτοιο διάγραμμα. Σημειώνει πως η DCF εφαρμόζεται σε ένα λογαριθμισμένο φασματογράφημα σε κλίμακα mel, όπου ουσιαστικά γίνεται μια σύγκριση μεταξύ του φασματογραφήματος και ενός συνόλου από προκαθορισμένες συνημιτονοειδείς συναρτήσεις οι οποίες είναι επίσης προσαρμοσμένες στην κλίμακα Mel. Το χρώμα του κάθε MFCC ουσιαστικά δείχνει το βαθμό αντιστοίχισης της συνάρτησης με το φασματογράφημα, δηλαδή το κατά πόσο αυτή «υπάρχει» σε αυτό.

2.3 Νευρωνικά Δίκτυα και Βαθιά Μάθηση

Στα προηγούμενα κεφάλαια, αναλύσαμε το πως ορίζονται και διακρίνονται τα μουσικά είδη, καθώς επίσης το πως τα χαρακτηριστικά των ειδών μπορούν να διακριθούν και να εξαχθούν από ένα ηχητικό σήμα. Ο άνθρωπος, αφού αφουγκραστεί τα ηχητικά χαρακτηριστικά, βάσει της εμπειρίας του τα διακρίνει σε είδη. Η βαθιά μάθηση αποτελεί τη μεθοδολογία βάσει της οποίας μιμούμαστε την ανθρώπινη λειτουργία και συμπεριφορά με τους υπολογιστές, ενώ έχουμε φτάσει σε σημείο που μοντέλα βαθιάς μάθησης ξεπερνούν ακόμα και την ανθρώπινη απόδοση (Chelladurai & Sujatha, 2023). Στη συνέχεια, θα δούμε αυτούς τους μηχανισμούς με τους οποίους το ηχητικό σήμα μπορεί να ταξινομηθεί σε είδη.

Αφού λοιπόν το σήμα αναπαρασταθεί με μια κατάλληλη μορφή, τότε αυτή η μορφή τροφοδοτεί κάποιο μοντέλο βαθιάς μάθησης το οποίο βασίζεται σε αρχιτεκτονικές όπως CNN, RNN αλλά και Transformer (Zaman et al., 2023). Η βαθιά μάθηση αποτελεί έναν κλάδο της μηχανικής μάθησης που στηρίζεται στα τεχνητά νευρωνικά δίκτυα – artificial neural networks (ANN) (Janiesch et al., 2021). Η εξέλιξη των ANN σε βαθιά νευρωνικά δίκτυα με καλύτερη δυνατότητα εκμάθησης συνοψίζεται ως βαθιά μάθηση (Lecun et al., 2015). Η βαθιά μάθηση λοιπόν, αποτελεί αναφορά στα βαθιά νευρωνικά δίκτυα, ενώ ο όρος «βαθιά» αφορά στον αριθμό των επιπέδων (layers) που έχει το δίκτυο και συγκεκριμένα των κρυφών επιπέδων (hidden layers) που αυτό έχει (Shinde & Shah, 2018). Τα νευρωνικά δίκτυα γενικότερα αποτελούν μια τεχνολογική επανάσταση, πράγμα που οφείλεται σε

διάφορους λόγους όπως η διαθεσιμότητα υπολογιστικής ισχύος λόγω τεχνολογικής προόδου (πχ GPUs), η διαθεσιμότητα τεράστιας ποσότητας δεδομένων καθώς και η δυνατότητά τους να τα διαχειρίζονται (L. Zhang et al., 2018).

2.3.1 Εισαγωγή στα νευρωνικά δίκτυα

Τα νευρωνικά δίκτυα είναι έννοια άρρηκτα συνδεδεμένη με τη βαθιά μάθηση. Αρχικά, η βαθιά μάθηση (DL) εντάσσεται σε μια ιεραρχική δομή, καθώς αποτελεί κλάδο της μηχανικής μάθησης, που με τη σειρά της αποτελεί κλάδο της τεχνητής νοημοσύνης (Perumal et al., 2024). Τα νευρωνικά δίκτυα λοιπόν, αποτελούν τον πυρήνα της βαθιάς μάθησης όπου χρησιμοποιούνται για εκπαίδευση μοντέλων που κάνουν χρήση αρχιτεκτονικών με πολλαπλά επίπεδα, γεγονός που ξεδιπλώνει την πλήρη υπολογιστική τους ισχύ, η οποία παλαιότερα περιοριζόταν στη χρήση μόλις ενός ή δύο επιπέδων και μικρό αριθμό δεδομένων (L. Zhang et al., 2018). Παρόμοια με τον ανθρώπινο εγκέφαλο, τα νευρωνικά δίκτυα είναι σχεδιασμένα να αναγνωρίζουν πρότυπα και να κατηγοριοποιούν πληροφορίες, μαθαίνοντας να εκτελούν εργασίες μέσα από τη δυνατότητα να συσχετίζουν νέα δεδομένα με κάτι που ήδη γνωρίζουν (Chelladurai & Sujatha, 2023).

Όπως αναφέρθηκε ήδη, ένα νευρωνικό δίκτυο αποτελείται από επίπεδα. Οι Shao et al. (2003) σημειώνουν ότι δομικό χαρακτηριστικό των επιπέδων είναι οι νευρώνες, ενώ το κάθε επίπεδο αποτελείται από έναν αριθμό νευρώνων. Οι ίδιοι σημειώνουν πως το πρώτο επίπεδο ονομάζεται επίπεδο εισόδου (input layer), ενώ το τελικό ονομάζεται επίπεδο εξόδου (output layer), καθώς επίσης υπάρχουν τα ενδιάμεσα ή κρυφά επίπεδα (hidden layers). Δομικό στοιχείο των δικτύων λοιπόν όπως υποδηλώνει και το όνομά τους, είναι ο νευρώνας (Benois-Pineau & Zemmari, 2021). Οι McCulloch και Pitts (1943) στα μέσα του 20ου αιώνα εισήγαγαν την έννοια του νευρώνα εμπνεόμενοι από τη νευροφυσιολογία του εγκεφάλου και τον έθεσαν ως λογική μονάδα σε ένα δίκτυο, παρόμοιο με αυτό των συνάψεων. Οι Benois-Pineau και Zemmari (2021) εξηγούν ότι από μαθηματικής σκοπιάς, ο νευρώνας δέχεται δεδομένα εισόδου $(x_1, x_2, \dots, x_p)$, εφαρμόζει μια συνάρτηση ενεργοποίησης (activation function) σε έναν γραμμικό συνδυασμό z των δεδομένων, ο οποίος εξαρτάται από ένα διάνυσμα από βάρη (weights) που τίθεται ως $(w_1, w_2, \dots, w_p)$ και από μια πόλωση (bias) b. Ο μαθηματικός τύπος που περιγράφει τη διαδικασία δίνεται ως:

$$y = f(z) = f\left(\sum_{i=1}^{p} w_i x_i + b\right) \tag{15}$$

Όπου y είναι η έξοδος του κάθε νευρώνα και f κάποια συνάρτηση ενεργοποίησης. Ένας νευρώνας λοιπόν είναι μια μονάδα επεξεργασίας πληροφορίας που αποτελείται από 3 βασικά στοιχεία: συνδέσεις με άλλους νευρώνες, ένα μέρος που προσθέτει και υπολογίζει το σταθμισμένο άθροισμα των δεδομένων εισόδου σε αυτόν και τέλος μια συνάρτηση ενεργοποίησης η οποία περιορίζει το εύρος των τιμών της εξόδου του νευρώνα (Haykin, 2009). Κατά αναλογία με τις συνάψεις του ανθρώπινου εγκεφάλου, έτσι και οι συνδέσεις μεταξύ των νευρώνων ενισχύονται ή εξασθενούν μέσω βαρών, τα οποία σταθμίζουν τα σήματα εισόδου που αθροίζονται και στη συνέχεια μετασχηματίζονται από μια συνάρτηση ενεργοποίησης (Janiesch et al., 2021).

Η γενική λειτουργία των νευρωνικών δικτύων είναι πως δέχονται κάποια αριθμητικά δεδομένα εισόδου, τα δεδομένα αυτά θα υποστούν επεξεργασία από τους νευρώνες και εν τέλει φτάνοντας στο output layer, παίρνουμε έναν αριθμό ο οποίος ερμηνεύεται ανάλογα το πρόβλημα, καθώς για παράδειγμα θα μπορούσε να είναι ένα «ναι» ή «όχι» σε ένα πρόβλημα ταξινόμησης (Leskovec et al., 2020). Για να είναι εφικτό το δίκτυο να δώσει σωστό αποτέλεσμα, θα πρέπει τα βάρη και οι πολώσεις που αντιστοιχούν σε κάθε νευρώνα να έχουν κάποιες συγκεκριμένες τιμές, πράγμα που επιτυγχάνεται με τη διαδικασία της εκπαίδευσης (Géron, 2023). Κατά τη διαδικασία της εκπαίδευσης (training), εκθέτουμε το δίκτυο σε data γνωρίζοντας το αποτέλεσμα που θα πρέπει να πάρουμε και εν τέλει με διάφορες μεθόδους, τα βάρη και οι πολώσεις ρυθμίζονται ώστε να έχουν τις κατάλληλες τιμές (Leskovec et al., 2020).

Γενικά, κάθε νευρώνας του δικτύου συνδέεται με καθέναν από τους νευρώνες του προηγούμενου επιπέδου ενώ δεν συνδέεται με τους νευρώνες του επιπέδου στο οποίο ανήκει (Benois-Pineau & Zemmari, 2021). Αυτό είναι κάτι που ισχύει στις περισσότερες αρχιτεκτονικές αλλά μπορεί να μην ισχύει σε αρχιτεκτονικές που υπάρχει αυτοτροφοδότηση όπως στα RNN. Ο αριθμός των νευρώνων σε ένα hidden layer δίνει το βάθος (width) του στο layer αυτό (Goodfellow et al., 2017). Με βάση την αρχιτεκτονική, τα νευρωνικά δίκτυα μπορούν γενικά να κατηγοριοποιηθούν σε νευρωνικά δίκτυα πρόσθιας τροφοδότησης (feedforward) και σε αναδρομικά/επαναλαμβανόμενα (recurrent/recursive) νευρωνικά δίκτυα, ενώ συναντάμε και συνδυασμούς των δύο περιπτώσεων (L. Zhang et al., 2018). Οι Benois-Pineau και Zemmari (2021) αναφέρουν πως κάθε νευρωνικό δίκτυο πρέπει να έχει τουλάχιστον δύο επίπεδα: ένα για είσοδο και ένα για έξοδο και αυτή η αρχιτεκτονική μπορεί και να είναι αρκετή για κάποιες πράξεις. Οι ίδιοι αναφέρουν όμως ότι

ένα βαθύ νευρωνικό δίκτυο οφείλει να έχει τουλάχιστον ένα κρυφό επίπεδο. Ο αριθμός των επιπέδων, μας δίνει το βάθος (depth) του δικτύου ενώ ο ίδιος ο όρος «βαθιά μάθηση» προκύπτει ακριβώς από αυτή την έννοια (Goodfellow et al., 2017).

2.3.2 Forward Propagation και συναρτήσεις ενεργοποίησης

Η διαδικασία του Forward Propagation αφορά τον υπολογισμό και την αποθήκευση ενδιάμεσων μεταβλητών ενός νευρωνικού δικτύου, με σειρά από το επίπεδο εισόδου προς το επίπεδο εξόδου (A. Zhang et al., 2024). Αναφέραμε ήδη πως ο κάθε νευρώνας δέχεται αριθμητικά δεδομένα τα οποία υφίστανται υπολογισμό σύμφωνα με τη σχέση (15). Η διαδικασία του Forward Propagation αποτελεί τον υπολογισμό των δεδομένων κάθε νευρώνα του δικτύου έως την τελική έξοδο. Στη φάση αυτή, οι τιμές των weights είναι σταθερές και ο υπολογισμός γίνεται από το input layer προς το output, αντίθετα με τη διαδικασία του Backpropagation που γίνεται με σκοπό τη ρύθμιση των τιμών των weights για καλύτερο αποτέλεσμα (Haykin, 2009). Οι Leskovec et al. (2020) σημειώνουν πως κατά τον μαθηματικό υπολογισμό του Forward Propagation αλλά και γενικότερα στα νευρωνικά δίκτυα γίνεται χρήση γραμμικής άλγεβρας και οργάνωση των δεδομένων, των βαρών κλπ. σε διανύσματα, πίνακες ή γενικότερα τανυστές. Οι ίδιοι, προσθέτουν πως αυτό δε γίνεται μόνο χάριν συντομογραφίας, αλλά και λόγω απόδοσης καθώς οι υπολογισμοί πράξεων γραμμικής άλγεβρας μπορούν να εκτελεστούν πολύ γρήγορα από επεξεργαστές GPU χάριν της αρχιτεκτονικής τους.

Οι συναρτήσεις ενεργοποίησης (activation functions) καθορίζουν το εάν ένας νευρώνας θα ενεργοποιηθεί ή όχι ελέγχοντας το σταθμισμένο άθροισμα των δεδομένων όπου έχει προστεθεί η πόλωση (A. Zhang et al., 2024). Θα λέγαμε λοιπόν ότι λειτουργούν αντίστοιχα με έναν διακόπτη φωτός με ON και OFF (Chelladurai & Sujatha, 2023). Αυτό είναι κάτι που διακρίνεται και στη σχέση (15).

Ο υπολογισμός του σταθμισμένου αθροίσματος και της πόλωσης υποδηλώνει γραμμικότητα στους υπολογισμούς λόγω της μορφής της εξίσωσης υπολογισμού. Ένας από τους ρόλους της συνάρτησης ενεργοποίησης που θα εφαρμοστεί, είναι να εισάγει μη γραμμικότητα (Goodfellow et al., 2017). Χωρίς τις συναρτήσεις ενεργοποίησης, οι υπολογισμοί του νευρωνικού δικτύου άγονται σε μια γραμμική συνάρτηση, έτσι λοιπόν με αυτές εισάγουμε τη μη γραμμικότητα και είναι εφικτή η επίλυση περισσότερων και πιο περίπλοκων προβλημάτων (Géron, 2023). Οι συναρτήσεις ενεργοποίησης λοιπόν, θα

λέγαμε ότι είναι μη γραμμικοί μετασχηματισμοί των σταθμισμένων δεδομένων εισόδου σε ένα νευρώνα (Polson & Sokolov, 2023).

Ειδικά για προβλήματα ταξινόμησης ενδείκνυται οι νευρώνες του τελευταίου επιπέδου να παίρνουν τιμές στο διάστημα (0, 1) και οι συναρτήσεις ενεργοποίησης μας δίνουν αυτή τη δυνατότητα (Bishop, 2006). Συνήθως σε προβλήματα ταξινόμησης, οι νευρώνες του τελευταίου επιπέδου αντιστοιχούν στις κλάσεις στις οποίες ταξινομούμε τα δεδομένα εισόδου, επομένως η συνάρτηση ενεργοποίησης θα μετασχηματίσει τις τιμές τους σε πιθανότητα τα δεδομένα αυτά να αντιστοιχούν σε κάθε κλάση (Zaman et al., 2023).

Στη συνέχεια θα παρουσιάσουμε κάποιες συναρτήσεις ενεργοποίησης. Ο Haykin (2009) ορίζει τη συνάρτηση κατωφλίου (threshold function) ως εξής:

$$f(x) = \begin{cases} 1 & if\ x \geq 0 \\ 0 & if\ x < 0 \end{cases} \tag{16}$$

Ενώ διευκρινίζει ότι η συγκεκριμένη συνάρτηση στη μηχανική είναι γνωστή στους μηχανικούς ως “Heaviside function”. Οι Benois-Pineau και Zemmari (2021) αναφέρουν την ίδια συνάρτηση ως συνάρτηση βήματος (step function).

Οι L. Zhang et al. (2018) αναφέρουν ότι πολύ συχνά χρησιμοποιούμενη ιστορικά συνάρτηση είναι η σιγμοειδής συνάρτηση (sigmoid function) η οποία ορίζεται ως:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{17}$$

Η μεταβλητή x είναι πιθανόν να πολλαπλασιάζεται με έναν συντελεστή λ ο οποίος είναι γνωστός ως “slope parameter” και επηρεάζει την κλίση της σιγμοειδούς (Haykin, 2009). Η σιγμοειδής συνάρτηση γνωστή και ως “squashing function”, δέχεται οποιονδήποτε πραγματικό αριθμό και τον «συμπιέζει» στο διάστημα μεταξύ του 0 και του 1 (A. Zhang et al., 2024). Η συγκεκριμένη συνάρτηση έχει αρκετά προτερήματα συγκριτικά με τη συνάρτηση κατωφλίου, με βασικό την αποδοτικότερη ενημέρωση των βαρών κατά την εκπαίδευση (Leskovec et al., 2020). Ο Géron (2023) αναφέρει πως ειδικά για προβλήματα ταξινόμησης, η σιγμοειδής συνάρτηση μπορεί να χρησιμοποιηθεί εκτενώς μετά το τελευταίο επίπεδο του δικτύου, καθώς το αποτέλεσμά της ερμηνεύεται ως πιθανότητα. Προσθέτει πως για ένα πρόβλημα δυαδικής ταξινόμησης (binary classification), μπορεί στο τελευταίο επίπεδο να υπάρχει ένας νευρώνας και το αποτέλεσμα της σιγμοειδούς να στρογγυλοποιείται σε 0 ή 1, ενώ για περισσότερες κλάσεις, κάθε νευρώνας του επιπέδου

εξόδου, αντιστοιχεί σε μια κλάση. Γενικά, τα δεδομένα όπως αυτά υπολογίζονται και προκύπτουν από το τελευταίο επίπεδο, συνήθως λέγονται "logits", ενώ η εφαρμογή της σιγμοειδούς σε αυτά τα κανονικοποιεί και τα μετατρέπει σε πιθανότητες (Goodfellow et al., 2017).

Σύμφωνα με τους Leskovec et al. (2020), παρόμοια λειτουργία με τη σιγμοειδή έχει η συνάρτηση της υπερβολικής εφαπτομένης που δίνεται από τον τύπο:

$$f(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{18}$$

Η υπερβολική εφαπτομένη επίσης «συμπιέζει» τις τιμές εισόδου των δεδομένων, απλά με το να τις αναθέτει στο διάστημα μεταξύ -1 και 1. (A. Zhang et al., 2024). Η βασική της διαφορά από τη σιγμοειδή είναι ότι τείνει να παρουσιάζει υψηλότερες τιμές κλίσης (Benois-Pineau & Zemmari, 2021).

Οι A. Zhang et al. (2024) αναφέρουν ότι μια επίσης διαδεδομένη συνάρτηση ενεργοποίησης είναι η rectified linear unit (ReLU), που δίνεται από τον τύπο:

$$f(x) = ReLU(x) = \max(x, 0) \tag{19}$$

Οι ίδιοι σημειώνουν πως η εν λόγω συνάρτηση απλά παραλείπει τις αρνητικές τιμές, αντικαθιστώντας τες με μηδέν. Τα περισσότερα σύγχρονα νευρωνικά δίκτυα χρησιμοποιούν τη ReLU ως συνάρτηση ενεργοποίησης (Goodfellow et al., 2017). Οι κύριοι λόγοι που η ReLU έχει αντικαταστήσει τις υπόλοιπες συναρτήσεις ενεργοποίησης είναι το γεγονός ότι η παράγωγός της μένει σταθερή, πράγμα που κάνει πιο αποδοτική την εκπαίδευση από άποψης χρόνου, ενώ η παράγωγός της υπολογίζεται με απλές μαθηματικές πράξεις χωρίς εκθετικά (Leskovec et al., 2020). Παρ' όλα αυτά, η ReLU έχει με τη σειρά της μειονεκτήματα, γεγονός που έχει οδηγήσει στο να προταθούν βελτιώσεις της ή και άλλες συναρτήσεις ενεργοποίησης (Perumal et al., 2024).

Οι Maas et al. (2013), έχουν προτείνει τη Leaky ReLU. Οι He et al. (2015) έχουν προτείνει τη Parametric Rectified Linear Units (PReLU). Οι B. Xu et al. (2015) παρουσιάζουν τη Randomized Leaky ReLU (RReLU). Οι Jin et al. (2016), προτείνουν την S-shaped ReLU (SReLU). Ενώ οι Clevert et al. (2015) αναφέρουν τις Exponential Linear Units (ELUs). Τέλος, οι Hendrycks και Gimpel (2016) προτείνουν τα Gaussian Error Linear Units (GELUs). Τα παραπάνω είναι μερικά ενδεικτικά παραδείγματα.

Τέλος, θα κάνουμε μια αναφορά στη Softmax. Η εν λόγω συνάρτηση ενεργοποίησης συνήθως συναντάται στο επίπεδο εξόδου (L. Zhang et al., 2018). Συνήθως χρησιμοποιείται για τον υπολογισμό πιθανοτήτων που σχετίζονται με την κατανομή multinoulli, ή αλλιώς κατηγορική κατανομή (Goodfellow et al., 2017). Οι Benois-Pineau και Zemmari (2021) αναφέρουν πως για διάνυσμα $y = (y_1, y_2, \dots, y_k)$ με θετικές πραγματικές τιμές, η softmax μετασχηματίζει τις τιμές του y σε ένα διάνυσμα $s = (p_1, p_2, \dots, p_k)$ όπου:

$$p_i = \frac{e^{y_i}}{\sum_{j=1}^{k} e^{y_j}} \tag{20}$$

Με p την πιθανότητα. Το άθροισμα των τιμών που προκύπτουν από τη softmax είναι 1, ενώ η ίδια τείνει να ωθεί τη μεγαλύτερη συνιστώσα του διανύσματος προς τη μονάδα και τις υπόλοιπες προς το μηδέν (Leskovec et al., 2020). Η softmax συνήθως χρησιμοποιείται μαζί με την συνάρτηση κόστους cross-entropy, ενώ συνήθως οι τιμές που υπολογίζει αντιστοιχούν στα δεδομένα κάθε νευρώνα του επιπέδου εξόδου, δίνοντας την πιθανότητα η τάξη που αντιστοιχεί στον κάθε νευρώνα να είναι το σωστό αποτέλεσμα ενός προβλήματος ταξινόμησης (Benois-Pineau & Zemmari, 2021). Κάτι αντίστοιχο δηλαδή με αυτό που είδαμε και στη σιγμοειδή. Γενικότερα, η διαδικασία επεξεργασίας των δεδομένων ώστε να έχουν άθροισμα 1 ονομάζεται κανονικοποίηση (normalization) και αποτελεί σημαντικό κομμάτι της υλοποίησης ενός μοντέλου βαθιάς μάθησης (A. Zhang et al., 2024).

2.3.3 Συναρτήσεις κόστους (Loss functions) και αξιολόγηση μοντέλων

Κατά την ανάπτυξη και εκπαίδευση των μοντέλων, απαιτείται ένα τυπικό μέτρο που να ποσοτικοποιεί την απόδοσή τους και να επιτρέπει τη σύγκριση διαφορετικών προσεγγίσεων, ρόλο που επιτελεί στη μηχανική μάθηση και συγκεκριμένα στη βελτιστοποίηση η συνάρτηση κόστους (A. Zhang et al., 2024). Μέρος της βελτιστοποίησης ενός αλγορίθμου βαθιάς μάθησης αποτελεί η τροποποίηση της μεταβλητής x μιας συνάρτησης $f(x)$, που είναι η συνάρτηση κόστους, ώστε αυτή να ελαχιστοποιηθεί (Goodfellow et al., 2017). Όσο καλύτερο είναι ένα μοντέλο βαθιάς μάθησης λοιπόν, τόσο χαμηλότερη η τιμή της συνάρτησης κόστους, ενώ με την ελαχιστοποίησή της επιτυγχάνουμε τις ιδανικές τιμές των βαρών (weights) και άλλων παραμέτρων που συντελούν στην απόδοση του μοντέλου (Chelladurai & Sujatha, 2023). Συνοπτικά, οι συναρτήσεις κόστους ποσοτικοποιούν τη διαφορά μεταξύ των προβλέψεων ενός μοντέλου από τις πραγματικές τιμές, ενώ η εκπαίδευσή του περιλαμβάνει την τροποποίηση τιμών της όπως το βάρος και η πόλωση (A. Zhang et al., 2024). Εναλλακτικά, υπάρχουν προσεγγίσεις

που χρησιμοποιούν συναρτήσεις οφέλους (utility functions) όπου ο σκοπός είναι η μεγιστοποίησή τους, αλλά η όλη ιδέα παραμένει ίδια (Bishop, 2006). Συνήθως διαφορετικές συναρτήσεις κόστους αντιστοιχούν σε διαφορετικού είδους προβλήματα και προβλέψεις, γενικά όμως διακρίνουμε δυο περιπτώσεις: τις συναρτήσεις που έχουμε κάποιο πρόβλημα παλινδρόμησης (regression) όπου το μοντέλο προβλέπει μια αριθμητική τιμή και τα προβλήματα ταξινόμησης (classification) όπου το μοντέλο προβλέπει εάν τα δεδομένα που έχει στην είσοδό του ανήκουν σε κάποια κλάση (Leskovec et al., 2020). Γενικότερα η επιλογή της συνάρτησης κόστους είναι πολύ σημαντικό μέρος της υλοποίησης ενός συστήματος βαθιάς μάθησης, ενώ η συνάρτηση πρέπει να είναι παραγωγίσιμη ως προς τις παραμέτρους που σχετίζονται με την εκπαίδευσή του (Purwins et al., 2019).

Στη συνέχεια θα παρουσιάσουμε μερικές συναρτήσεις κόστους. Οι Goodfellow et al. (2017) ορίζουν το μέσο τετραγωνικό σφάλμα (mean squared error – MSE) με τον τύπο:

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(\hat{Y}_i - Y_i)^2 \tag{21}$$

Όπου $\hat{Y}_i$ είναι η τιμή που προέβλεψε το μοντέλο, Y_i η πραγματική τιμή και m ο συνολικός αριθμός δειγμάτων. Οι ίδιοι εξηγούν το πως ουσιαστικά το $\hat{Y}_i$ μπορεί να εκφραστεί ως συνάρτηση των βαρών. Το MSE επίσης αναφέρεται ως L2 Loss (Elharrouss et al., 2025). Το MSE γενικότερα χρησιμοποιείται για προβλήματα παλινδρόμησης (Leskovec et al., 2020). Παραλλαγή του MSE είναι η ρίζα του μέσου τετραγωνικού σφάλματος (root square mean error - RMSE) που όπως υποδηλώνει και το όνομά της, υπολογίζεται βρίσκοντας την τετραγωνική ρίζα του MSE, αφού το υπολογίσουμε (Géron, 2023). Άλλη συνάρτηση κόστους είναι το μέσο απόλυτο σφάλμα (mean absolute error - MAE) γνωστό και ως L1 Loss και δίνεται από τον τύπο:

$$MAE = \frac{1}{m}\sum_{i=1}^{m}\left|\hat{Y}_i - Y_i\right| \tag{22}$$

Όπου οι μεταβλητές είναι ίδιες με της αντίστοιχες της σχέσης (21) για το MSE (Elharrouss et al., 2025). Το MAE είναι επίσης μια συνάρτηση που χρησιμοποιείται για προβλήματα παλινδρόμησης, όπως και η RMSE (Géron, 2023). Προβλήματα που προκύπτουν από συναρτήσεις όπως η προηγούμενη, λύνονται με τη χρήση του Huber Loss, μια συνάρτηση κόστους επίσης χρήσιμη για προβλήματα παλινδρόμησης (Leskovec et al., 2020). Οι

Elharrouss et al. (2025) σημειώνουν ότι το Huber Loss είναι γνωστό και ως Smooth L1 Loss και δίνεται από τον τύπο:

$$L_h(y, \hat{y}) = \begin{cases} \frac{1}{2}(y - \hat{y})^2 & \varepsilon\acute{\alpha}\nu\ |y - \hat{y}| < 1 \\ |y - \hat{y}| - \frac{1}{2} & \sigma\varepsilon\ \acute{\alpha}\lambda\lambda\eta\ \pi\varepsilon\rho\acute{\iota}\pi\tau\omega\sigma\eta \end{cases} \quad (23)$$

Όπου y και $\hat{y}$ είναι οι πραγματικές και οι προβλεπόμενες τιμές αντίστοιχα.

Στα προβλήματα ταξινόμησης, η έξοδος του συστήματος βαθιάς μάθησης εκφράζει τις πιθανότητες τα δεδομένα εισόδου να αντιστοιχούν σε μια κλάση, ενώ στα προβλήματα παλινδρόμησης είχαμε την πρόβλεψη μιας τιμής (Leskovec et al., 2020). Συνεπώς οι συναρτήσεις κόστους στην περίπτωση ταξινόμησης θα πρέπει να είναι ανάλογα προσαρμοσμένες για τους υπολογισμούς. Για προβλήματα ταξινόμησης λοιπόν, αρκετά συχνή είναι η συνάρτηση της cross-entropy loss (Géron, 2023). Οι A. Zhang et al. (2024) σημειώνουν τον τύπο υπολογισμού της ως:

$$L_{CE} = -\sum_{i=1}^{m} y_i \log(\hat{y}_i) \quad (24)$$

Όπου m ο αριθμός των κλάσεων και y και $\hat{y}$ είναι οι πραγματικές και οι προβλεπόμενες τιμές αντίστοιχα. Δεδομένου ότι αναφερόμαστε σε πιθανότητες, οι τιμές των y και $\hat{y}$ είναι μεταξύ 0 και 1 (Elharrouss et al., 2025). Ειδικά όμως για την περίπτωση που έχουμε δύο κλάσεις, μπορούμε να χρησιμοποιήσουμε την συνάρτηση binary cross-entropy loss (Géron, 2023). Ο τύπος για τον υπολογισμό της συνάρτησης αυτής είναι:

$$L_{BCE} = -[y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - \hat{y})] \quad (25)$$

Όπου οι μεταβλητές είναι αντίστοιχες με αυτές που τέθηκαν και στη σχέση (24) για τον υπολογισμό της cross-entropy loss (Elharrouss et al., 2025). Όπως αναφέρθηκε σε προηγούμενο κεφάλαιο, συχνά για την περίπτωση ταξινόμησης εφαρμόζουμε τη συνάρτηση ενεργοποίησης softmax. Επειδή λοιπόν και η cross-entropy loss είναι μια συνάρτηση κόστους συνυφασμένη με την ταξινόμηση, συχνά οι δυο τους υπολογίζονται μαζί, ταυτόχρονα με σχέσεις που προκύπτουν συνδυάζοντας τους τύπους υπολογισμού τους (A. Zhang et al., 2024).

Οι παραπάνω ήταν κάποιες ενδεικτικές συναρτήσεις κόστους. Πέραν αυτών, έχουν προταθεί και άλλες όπως Gaussian Wasserstein Distance, Focal Loss, Dice Loss, Jaccard

Loss, Perceptual Loss, Cycle consistency Loss, Categorical Loss, Contrastive Loss κ.α., κάθε μια εκ των οποίων μπορεί να είναι και πιο εξειδικευμένη για συγκεκριμένα πλαίσια εφαρμογής (Elharrouss et al., 2025).

Από τη λειτουργία των συναρτήσεων κόστους προκύπτει ότι αποτελούν μέσο αξιολόγησης των μοντέλων βαθιάς μάθησης, ωστόσο, επειδή συμβάλλουν ταυτόχρονα στην εκπαίδευση και στη βελτιστοποίησή τους, υφίσταται και η χρήση πρόσθετων μετρικών αξιολόγησης (Benois-Pineau & Zemmari, 2021). Οι μετρικές αυτές χρησιμοποιούνται αποκλειστικά για την εκτίμηση της απόδοσης των μοντέλων προκειμένου να προσδιοριστεί το πόσο καλά ένα μοντέλο μπορεί να εκπαιδευθεί από τα δεδομένα που του δίνονται, καθώς και να εντοπιστούν περιοχές που χρήζουν βελτίωσης (Zaman et al., 2023). Ο Lerch (2022), σημειώνει κάποιους όρους που συναντάμε στην περίπτωση ταξινόμησης με 2 κλάσεις, αλλά είναι απαραίτητοι για τον υπολογισμό των μετρικών που θα εξηγηθούν:

- TP (True Positive/Αληθώς θετικά): Θετικά δείγματα που ορθώς έχουν ταυτοποιηθεί ως θετικά.
- TN (True Negative/Αληθώς αρνητικά): Αρνητικά δείγματα που ορθώς έχουν ταυτοποιηθεί ως αρνητικά.
- FP (False Positive/Ψευδώς θετικά): Θετικά δείγματα που λανθασμένα έχουν ταυτοποιηθεί ως αρνητικά.
- FN (False Negative/Ψευδώς αρνητικά): Θετικά δείγματα που λανθασμένα έχουν ταυτοποιηθεί ως θετικά.

Οι Perumal et al. (2024) βάσει των παραπάνω όρων αναφέρουν τις εξής μετρικές: πιστότητα (accuracy), ακρίβεια (precision), ανάκληση (recall) και μέτρο F1 (F1-Score). Αντίστοιχα αναφέρουν ότι οι τύποι του υπολογισμού της καθεμίας είναι:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{26}$$

$$Precision = \frac{TP}{TP + FP} \tag{27}$$

$$Recall = \frac{TP}{TP + FN} \tag{28}$$

$$F1 - Score = 2 \cdot \frac{Recall \cdot Precision}{Recall + Precision} \tag{29}$$

Για την πιστότητα συγκεκριμένα, που είναι και μια αρκετά συχνή μετρική, βλέπουμε πως ουσιαστικά αποτελεί το ποσοστό των σωστών προβλέψεων προς όλες τις προβλέψεις (Zaman et al., 2023).

Μια ακόμα συνήθης και χρήσιμη μετρική είναι η μήτρα σύγχυσης (confusion matrix). Εναλλακτικά, αναφέρεται και ως πίνακας σύγχυσης. Ο Géron (2023) αναφέρει πως η γενική ιδέα της μήτρας σύγχυσης ξεκινά από την καταγραφή σειρών με την αληθή κλάση και στηλών με τις κλάσεις που προβλέφθηκαν. Ο ίδιος, συμπληρώνει πως κάθε στοιχείο της μήτρας δείχνει τον αριθμό δειγμάτων μιας αληθούς κλάσης που ταξινομήθηκαν σε κάθε προβλεπόμενη κλάση. Συνεπώς, συμπεραίνουμε ότι τα διαγώνια στοιχεία της μήτρας είναι αντιστοιχούν στις σωστές προβλέψεις, Ένας τέλειος ταξινομητής, θα έχει καταχωρήσεις μόνο στη διαγώνιο και μάλιστα ο αριθμός τους θα ταυτίζεται με το άθροισμα των δειγμάτων, ενώ ένας μη τέλειος ταξινομητής θα έχει καταχωρήσεις εκτός της διαγώνιου, από τις οποίες μπορούμε να βγάλουμε συμπεράσματα για τάσεις στις προβλέψεις του μοντέλου (Lerch, 2022). Η μήτρα σύγχυσης είναι ιδιαίτερα χρήσιμη μετρική στη μελέτη ταξινόμησης μουσικών ειδών, καθώς μπορούμε να διακρίνουμε ποια είδη μπερδεύει το μοντέλο μεταξύ τους (Knees & Schedl, 2016).

2.3.4 Εκπαίδευση μοντέλων

Τα νευρωνικά δίκτυα εκπαιδεύονται καθώς εκτίθενται σε μεγάλες ποσότητες δεδομένων, τις οποίες επεξεργάζονται μέσω του επαναληπτικού αλγορίθμου οπισθοδιάδοσης (backpropagation), ο οποίος κάνει χρήση της συνάρτησης κόστους (Chelladurai & Sujatha, 2023). Τυπικά, πριν το ξεκίνημα της εκπαίδευσης, χωρίζουμε τα δεδομένα σε 2 μέρη: το training set που θα χρησιμοποιηθεί για την εκπαίδευση και το test set το οποίο χρησιμοποιείται για αξιολόγηση (A. Zhang et al., 2024). Μια συνήθης αναλογία διαχωρισμού είναι να χρησιμοποιήσουμε το 80% των δεδομένων για το training set και το 20% για το test set (Leskovec et al., 2020). Η γενική στρατηγική, περιλαμβάνει την αφετηρία με ένα ανεκπαίδευτο δίκτυο στο οποίο δίνουμε ως δεδομένα εισόδου το training set, συγκεντρώνουμε τα δεδομένα εξόδου και χρήσει της συνάρτησης κόστους υπολογίζουμε το σφάλμα, ενώ προσπαθούμε να τροποποιήσουμε τις παραμέτρους της συνάρτησης κόστους ώστε το σφάλμα να ελαττωθεί ή και να ελαχιστοποιηθεί (Shao et al., 2003). Κατά τη διαδικασία, παρακολουθούμε την επίδοση του μοντέλου και στα δύο σύνολα δεδομένων, ενώ μια ενδιαφέρουσα αναλογία είναι πως η εκπαίδευση στο training set μοιάζει με την προσπάθεια ενός μαθητή που προετοιμάζεται μελετώντας παλαιότερα

θέματα εξετάσεων, ενώ το test set αντιστοιχεί στις τελικές εξετάσεις. (A. Zhang et al., 2024). Πολλές φορές συνηθίζεται ο διαχωρισμός των δεδομένων σε 3 μέρη: το training set, το evaluation ή validation set και το test set, πράγμα που βοηθά στην περίπτωση που θέλουμε να συγκρίνουμε ήδη εκπαιδευμένα μοντέλα εκθέτοντάς τα σε δεδομένα που δεν έχουν αξιοποιηθεί ποτέ (Géron, 2023). Το validation set συνήθως προκύπτει ως υποσύνολο του test set, όμως αυτά τα 2 σύνολα χρησιμοποιούνται ξεχωριστά (Goodfellow et al., 2017).

Είδαμε λοιπόν πως η συνάρτηση κόστους είναι μια συνάρτηση πολλών μεταβλητών όπως τα βάρη και η πόλωση κάθε επιπέδου του δικτύου. Επίσης αναφέρθηκε το ότι ιδεατά αυτή η συνάρτηση θα είχε τη μικρότερη δυνατή τιμή, πράγμα που σημαίνει ότι το σφάλμα της πρόβλεψης είναι το μικρότερο δυνατό. Από τα μαθηματικά, γνωρίζουμε ότι μια συνάρτηση μπορεί να είναι παραγωγίσιμη ως προς κάποια μεταβλητή και στο σημείο που η παράγωγος μηδενίζεται, έχουμε τοπικό ελάχιστο ή τοπικό μέγιστο, με την πρώτη περίπτωση να σημαίνει ότι η τιμή της συνάρτησης εκεί είναι μικρότερη συγκριτικά με τα γειτονικά σημεία (Goodfellow et al., 2017). Ο Cauchy (1847) έδειξε πως σε μία συνάρτηση $f(x)$ μπορούμε να προσεγγίσουμε το σημείο αυτό μεταβάλλοντας το x με μικρά βήματα προς την αντίθετη κατεύθυνση από το πρόσημο της παραγώγου. Η διαδικασία αυτή ονομάζεται κάθοδος κλίσης (gradient descent), ενώ το βήμα το οποίο κάνουμε για να προσεγγίσουμε το σημείο του τοπικού ελάχιστου, ονομάζεται learning rate. Το learning rate είναι μια από τις υπερπαραμέτρους που ρυθμίζουμε και η επιλογή της θέλει πολύ προσοχή, καθώς μια μικρή τιμή σημαίνει ότι θα χρειαστεί πολύς χρόνος και επαναλήψεις για να ελαχιστοποιηθεί η συνάρτηση κόστους, αλλά μια μεγάλη τιμή σημαίνει ότι μπορεί να μη βρεθεί καν η τιμή ελαχιστοποίησής της (Chelladurai & Sujatha, 2023). Εκτός από την κάθοδο κλίσης, μπορούμε να χρησιμοποιήσουμε και τη στοχαστική κάθοδο κλίσης (stochastic gradient descent-SGD) για τον υπολογισμό της κλίσης (L. Zhang et al., 2018). Η βασική διαφορά αυτής της μεθόδου είναι πως αντί να χρησιμοποιεί ολόκληρο το σύνολο των δεδομένων, χρησιμοποιεί ένα τυχαία επιλεγμένο υποσύνολό τους, καθώς με αυτόν τον τρόπο είναι εφικτή η εκπαίδευση των μοντέλων όταν τα δεδομένα δεν χωράνε στη μνήμη, κάτι αρκετά σύνηθες στα μοντέλα βαθιάς μάθησης (Polson & Sokolov, 2023).

Η διαδικασία του backpropagation και του υπολογισμού της κλίσης, ουσιαστικά περιλαμβάνει τον υπολογισμό της κλίσης ως προς τα βάρη μεταξύ ενός επιπέδου και του προηγούμενου εφαρμόζοντας τον κανόνα της αλυσίδας στην παραγώγιση, προχωρώντας ανά επίπεδο προς τα πίσω, μέχρι να φτάσουμε στο επίπεδο εισόδου (Lecun et al., 2015). Οι

L. Zhang et al. (2018) εξηγούν πως ουσιαστικά η διαδικασία περιλαμβάνει τον υπολογισμό της κλίσης της συνάρτησης κόστους ως προς τα βάρη μεταξύ του τελευταίου κρυφού επιπέδου και του επιπέδου εξόδου. Οι ίδιοι συμπληρώνουν πως εν συνεχεία υπολογίζεται η κλίση ως προς τα βάρη μεταξύ των προηγούμενων επιπέδων, εφαρμόζοντας κανόνα αλυσίδας και παραγωγίζοντας αναδρομικά. Κατά το στάδιο αυτό, τα βάρη τροποποιούνται ώστε η συνάρτηση κόστους να μειωθεί, ενώ η όλη διαδικασία επαναλαμβάνεται για ένα συγκεκριμένο αριθμό επαναλήψεων που επιλέγουμε και ονομάζεται “epochs” (Goodfellow et al., 2017). Κατά την εκπαίδευση, παράγοντες όπως πχ ο αριθμός νευρώνων, η συνάρτηση ενεργοποίησης, το learning rate επιλέγονται χειροκίνητα, όχι αυτόματα και αποτελούν τις υπερπαραμέτρους (hyperparameters) του μοντέλου (Perumal et al., 2024). Η χρήση αυτού του όρου μεταξύ άλλων, γίνεται για να υπάρχει σαφής διαχωρισμός από τις παραμέτρους του μοντέλου. Οι παράμετροι του μοντέλου, είναι μεταβλητές που δε ρυθμίζονται χειροκίνητα, αλλά από διαδικασίες όπως το gradient descent, ενώ ένα χαρακτηριστικό παράδειγμα είναι τα βάρη του μοντέλου (Goodfellow et al., 2017). Σε γενικές γραμμές, οι υπερπαράμετροι είναι μεταβλητές που επιτελούν 2 εργασίες: τον έλεγχο της δομής του δικτύου και τη ρύθμιση της εκπαίδευσής του (Chelladurai & Sujatha, 2023). Για την επιλογή των υπερπαραμέτρων και την αξιολόγησή τους χρησιμοποιείται το validation set ή το test set, ανάλογα το σύνολο στο οποίο το μοντέλο δεν έχει εκτεθεί για τη ρύθμισή τους (Goodfellow et al., 2017).

Οι Rumelhart et al. (1986) ήταν από τους πρώτους που εφάρμοσαν τη διαδικασία του backpropagation στο πλαίσιο των νευρωνικών δικτύων, αυξάνοντας τη δημοτικότητά του. Στο παρελθόν και ειδικά στις δεκαετίες του ’60- ’70 είχε επίσης χρησιμοποιηθεί σε έρευνες σε προβλήματα ελαχιστοποίησης (Polson & Sokolov, 2023). Οι Lecun et al. (2015) σημειώνουν ότι στα τέλη της δεκαετίας του ’90, η ευρύτερη κοινότητα της μηχανικής μάθησης αγνοούσε τα νευρωνικά δίκτυα και τεχνικές όπως το backpropagation, θεωρώντας πως με αυτή τη μέθοδο τα μοντέλα θα κατέληγαν σε τοπικά ελάχιστα και θα εγκλωβίζονταν σε αυτά με αποτέλεσμα να μην αποδίδουν σωστά. Οι ίδιοι συμπληρώνουν ότι στην πράξη, αυτό το πράγματι υπαρκτό πρόβλημα δε φαίνεται να είναι ανασταλτικός παράγοντας, καθώς τα μοντέλα καταλήγουν σε λύσεις που φαίνεται να αποδίδουν. Οι Raina et al. (2009), σχεδόν μια δεκαετία αργότερα, παρουσίασαν μια από τις πρώτες πολύ σημαντικές εφαρμογές αυτών των τεχνικών, πάνω στην αναγνώριση φωνής κάνοντας χρήση GPU, κάτι που τους επέτρεψε να εκπαιδεύουν μοντέλα 10 με 20 φορές γρηγορότερα.

Η εκπαίδευση ενός μοντέλου συνοδεύεται συχνά από προκλήσεις που καθιστούν δύσκολη την υλοποίησή της.. Η διαδικασία, μπορεί να κοστίσει αρκετά οικονομικά, ενώ σε αρκετές περιπτώσεις η έλλειψη επαρκούς ποσότητας δεδομένων περιορίζει την απόδοση του μοντέλου (Janiesch et al., 2021). Μέσω της τεχνικής του transfer learning, μπορούμε να χρησιμοποιούμε ήδη εκπαιδευμένα μοντέλα με ρυθμισμένες τιμές για τα βάρη, ή και να τα τροποποιούμε ανάλογα την περίπτωση που θέλουμε να τα χρησιμοποιήσουμε (Pouyanfar et al., 2019). Η ιδέα του transfer learning έχει ρίζες στην εκπαιδευτική ψυχολογία και τη θεωρία της γενίκευσης όπου υποστηρίζεται πως η μεταφορά γνώσης μέσω της γενίκευσης των εμπειριών (Perumal et al., 2024). Κατά την εφαρμογή του, είναι πιθανό να χρησιμοποιήσουμε ένα μοντέλο ως έχει, αλλά είναι επίσης πιθανό να τροποποιήσουμε κάποια από τα επίπεδα, περίπτωση στην οποία θα χρειαστεί κάποια επανεκπαίδευση του μοντέλου στα δεδομένα που έχουμε (Pouyanfar et al., 2019). Ειδικά για την περίπτωση της ταξινόμησης, εάν το μοντέλο από το οποίο θα κάνουμε transfer learning είναι ρυθμισμένο να ταξινομεί σε διαφορετικό αριθμό κλάσεων, θα χρειαστεί σίγουρα να τροποποιήσουμε το επίπεδο εξόδου και ίσως κάποια πριν από αυτό, ώστε να έχουμε τον επιθυμητό αριθμό κλάσεων (Tsalera et al., 2021).

2.3.5 Βελτιστοποίηση και κανονικοποίηση στην εκπαίδευση

Η διαδικασία της εκπαίδευσης ενός μοντέλου μπορεί να αποτελέσει πολλές φορές πρόκληση, τόσο όσον αφορά τους υπολογισμούς, αλλά και όσον αφορά την ποιότητα των δεδομένων που χρησιμοποιούνται για αυτή (Perumal et al., 2024). Προσεχώς θα δούμε τρόπους για να αντιμετωπίσουμε πιθανές δυσκολίες που συναντώνται στην εκπαίδευση ενός μοντέλου βαθιάς μάθησης. Όπως αναφέρθηκε σε προηγούμενο υποκεφάλαιο, κατά τη διαδικασία της εκπαίδευσης, γίνεται προσπάθεια ελαχιστοποίησης κάποιας συνάρτησης κόστους. Η όλη διαδικασία του gradient descent αποτελεί έναν αλγόριθμο βελτιστοποίησης (optimization) που στόχο έχει να φτάσει σε μια καλύτερη δυνατή τιμή (Bishop, 2006). Εκτός από το gradient descent, αναφέρθηκε ακόμα ένας αλγόριθμος βελτιστοποίησης, ο SGD. Οι Goodfellow et al. (2017) εξηγούν πως ο SGD είναι μια καλή μέθοδος, όμως φαίνεται να αργεί χρονικά Συμπληρώνουν λοιπόν πως αν εκμεταλλευθούμε τον αλγόριθμο του momentum ο οποίος υπολογίζει έναν εκθετικά φθίνοντα κινητό μέσο όρο των προηγούμενων κλίσεων, μπορούμε να καταλήξουμε γρηγορότερα στο αποτέλεσμα. Έτσι προκύπτει ο αλγόριθμος SGD with momentum. Μια ακόμα προσέγγιση σε αλγορίθμους βελτιστοποίησης είναι η ύπαρξη ενός learning rate του οποίου η τιμή προσαρμόζεται

(adaptative learning rate) κατά τη διαδικασία (Benois-Pineau & Zemmari, 2021). Σε αυτή την κατεύθυνση κινείται ο AdaGrad ο οποίος προσαρμόζει ξεχωριστά το learning rate για κάθε παράμετρο του μοντέλου, μεταβάλλοντας την κλίμακά του αντιστρόφως ανάλογα με την τετραγωνική ρίζα του αθροίσματος των τετραγώνων όλων των προηγούμενων κλίσεων (Duchi et al., 2011). Εξέλιξη του AdaGrad είναι ο RMSProp ο οποίος προτάθηκε από τους Tielman και Hinton σε ένα μάθημα του coursera το 2012 (A. Zhang et al., 2024). Πλέον, διαδεδομένος και σύγχρονος αλγόριθμος βελτιστοποίησης είναι ο Adam ο οποίος προσαρμόζει το learning rate κατά την εκπαίδευση, ενώ κάνει και χρήση του momentum, αποτελώντας γενικότερη εξέλιξη προηγούμενων αλγόριθμων (Kingma & Ba, 2014).

Ένα από τα σημαντικότερα προβλήματα που μπορεί να παρουσιάσει ένα μοντέλο είναι αυτό του “overfitting”, όπου το μοντέλο προσαρμόζεται στα δεδομένα του training set σε βαθμό που τα «απομνημονεύει» (Perumal et al., 2024). Ο βασικός στόχος ενός μοντέλου που αποκαλείται και γενίκευση (generalization), είναι το να αποδίδει ορθά όταν τροφοδοτείται με δεδομένα τα οποία δεν έχει ξαναέρθει σε επαφή (Goodfellow et al., 2017). Ο Géron (2023) παρομοιάζει το πρόβλημα του overfitting με την ανθρώπινη προκατάληψη, λέγοντας ότι είναι σαν να θεωρούμε πως όλοι οι οδηγοί ταξί είναι κλέφτες επειδή ενδεχομένως σε μια κούρσα εξαπατηθήκαμε από έναν. Συμπληρώνει, πως η υπερβολική γενίκευση οδηγεί στην προκατάληψη, ενώ αντίστοιχα στην περίπτωση του DL και του overfitting το μοντέλο αποδίδει καλά στο train set αλλά όχι στο test set, με αποτέλεσμα να μη «γενικεύει» καλά, όπως αυτό ορίστηκε πριν. Στις συναρτήσεις κόστους, ορίστηκε μια τιμή «σφάλματος» ή «κόστους» μεταξύ της προβλεπόμενης τιμής και της πραγματικής, όταν χρησιμοποιούσαμε το training set. Αν ορίσουμε ένα αντίστοιχο σφάλμα και για τις τιμές του test set, τότε στην περίπτωση του overfitting παρατηρούμε βελτίωση του σφάλματος μόνο στο training set, αλλά όχι στο test set (Goodfellow et al., 2017). Συγκρίνοντας την απόδοση του μοντέλου στο training set και στο test set μπορούμε να συμπεράνουμε αν υφίσταται overfitting, ενώ με τις τεχνικές κανονικοποίησης (regularization) μπορούμε να αντιμετωπίσουμε τη συμπεριφορά αυτή (Leskovec et al., 2020). Αντίστοιχα ορίζεται η έννοια του “underfitting” όπου το σφάλμα στο training set δε βελτιώνεται (Goodfellow et al., 2017).

Προσεχώς θα αναλύσουμε μερικές τεχνικές κανονικοποίησης. Μια βασική τεχνική είναι αυτή του early stopping. Στη συγκεκριμένη μέθοδο, δεδομένου ότι όπως αναφέραμε η εκπαίδευση είναι μια επαναλαμβανόμενη διαδικασία σε epochs, αυτό που κάνουμε είναι να παρατηρούμε το σημείο που ξεκινάει το overfitting και να σταματάμε την εκπαίδευση εκεί,

δηλαδή νωρίτερα από πριν (Leskovec et al., 2020). Σε αρκετές περιπτώσεις overfitting, μπορούμε να διακρίνουμε και σημεία όπου το σφάλμα στο test set αρχίζει και να αυξάνεται αντί να μειώνεται, περίπτωση στην οποία μπορούμε να σταματήσουμε τη διαδικασία σε εκείνο το σημείο (Bishop, 2006). Το early stopping είναι από τις συνηθέστερες τεχνικές, ενώ είναι σχετικά απλή δεδομένου ότι δεν επηρεάζουμε υπερπαραμέτρους της διαδικασίας εκπαίδευσης, πέραν του αριθμού των epochs ο οποίος τροποποιείται εύκολα (Goodfellow et al., 2017). Μια άλλη τεχνική είναι αυτή του dropout, η οποία ειδικά σε συνδυασμό με χρήση ReLU φαίνεται να είναι μια αρκετά αποδοτική λύση κανονικοποίησης σε νευρωνικά δίκτυα που επεξεργάζονται μουσικά αρχεία (Sigtia & Dixon, 2014). Οι Srivastava et al. (2014) πρότειναν την τεχνική αυτή όπου ουσιαστικά κατά την εκπαίδευση «ακυρώνονται» κάποιοι νευρώνες με το να μηδενίζεται οι έξοδός τους. Η τεχνική αυτή λειτουργεί καθώς εμποδίζει την εξάρτηση νευρώνων από συγκεκριμένα μοτίβα, ενώ εισάγει τυχαιότητα και ενώ οι λόγοι αυτοί τυπικά παραμένουν προς εξέταση, η ίδια η μέθοδος πρακτικά είναι αποδοτική (A. Zhang et al., 2024).

Τεχνική κανονικοποίησης είναι επίσης η μέθοδος του weight decay το οποίο λειτουργεί ως συντελεστής ποινής στη συνάρτηση κόστους (Pouyanfar et al., 2019). Η μέθοδος επιδρά ουσιαστικά κατά την εκπαίδευση με το να «ενθαρρύνει» τις τιμές του βάρους να μένουν μικρότερες σε μέτρο (G. Zhang et al., 2018). Η μέθοδος αύτη ουσιαστικά μας επιτρέπει να ελέγξουμε τη πολυπλοκότητα του μοντέλου και εξαρτάται από ένα συντελεστή λ (Bishop, 2006). Μικρότερες τιμές του λ περιορίζουν λιγότερο, ενώ μεγαλύτερες περιορίζουν περισσότερο, συνήθως όμως έχουμε τιμές μεταξύ 0 και 1 (A. Zhang et al., 2024). Η μέθοδος αυτή συναντάται συχνά και ως L2 regularization καθώς συνδέεται με την Ευκλείδεια απόσταση L2 από τη γραμμική άλγεβρα (A. Zhang et al., 2024). Εκτός αυτής υφίσταται και η L1 regularization η οποία συνδέεται με την απόσταση L1, γνωστή και ως απόσταση Manhattan, συνήθως όμως προτιμάται η προηγούμενη (Leskovec et al., 2020). Εκτός από το weight decay, υφίσταται και η τεχνική του learning rate decay καθώς και γενικότερα αλγόριθμοι που ρυθμίζουν το learning rate κατά την εκπαίδευση (Pouyanfar et al., 2019).

Πολύ διαδεδομένη τεχνική κανονικοποίησης είναι επίσης το data augmentation, το οποίο ουσιαστικά είναι μια μέθοδος να αυξηθεί η ποσότητα των διαθέσιμων δεδομένων (Perumal et al., 2024). Κάτι τέτοιο είναι αρκετά χρήσιμο, καθώς ένας τρόπος να περιοριστεί το overfitting είναι το να χρησιμοποιηθούν περισσότερα δεδομένα στην εκπαίδευση (A. Zhang et al., 2024). Οι Leskovec et al. (2020) εξηγούν πως σε αυτή την τεχνική, ουσιαστικά

εφαρμόζουμε τροποποιήσεις στα υπάρχοντα δεδομένα, ώστε με τεχνητό τρόπο να εκθέτουμε το μοντέλο σε καινούρια δεδομένα κατά την εκπαίδευση. Συμπληρώνουν πως αν για παράδειγμα στα δεδομένα μας έχουμε τη φωτογραφία μιας γάτας, μπορούμε να την περιστρέψουμε ή να την καθρεφτίσουμε.

Εκτός των προαναφερόμενων μεθόδων, υπάρχουν κι άλλες στρατηγικές οι οποίες συμβάλλουν στην καλύτερη απόδοση του μοντέλου, επεμβαίνοντας στην εκπαίδευσή του. Ενδεικτικά, οι Ioffe και Szegedy (2015) σχετικά πρόσφατα εισήγαγαν την τεχνική του batch normalization. Το σκεπτικό αυτής της στρατηγικής είναι η κανονικοποίηση των τιμών που συνδέονται με ένα νευρώνα, έτσι ώστε μεγαλύτερες τιμές να μην επηρεάζουν τους υπολογισμούς περισσότερο (Goodfellow et al., 2017). Η κανονικοποίηση των τιμών γίνεται σε ομάδες νευρώνων σε διάφορους συνδυασμούς (πχ ανά επίπεδο), συνεπώς η εν λόγω μέθοδος εμφανίζεται σε διάφορες μορφές (A. Zhang et al., 2024).

Επίσης αξίζει να αναφερθεί κανείς στον τρόπο που αρχικοποιούνται οι τιμές του βάρους. Όπως έχει αναφερθεί, ουσιαστικά ένα νευρωνικό δίκτυο αρχικά έχει τυχαίες τιμές βάρους, οι οποίες κατόπιν της επαναληπτικής διαδικασίας του backpropagation, θα πάρουν τις τελικές τους τιμές. Αποδεικνύεται ότι η αρχική ρύθμιση των τιμών αυτών είναι σημαντική, καθώς αν πχ τις θέταμε όλες ίσες με μηδέν ή με μια κοινή τιμή, η διαδικασία της εκπαίδευσης δεν θα απέδιδε, επομένως υφίστανται συγκεκριμένες κατανομές τιμών που τίθενται αρχικά στα βάρη (Chelladurai & Sujatha, 2023).

2.3.6 Πολυεπίπεδα νευρωνικά δίκτυα (MLP)

Έχοντας δει τις βασικές ιδιότητες των νευρωνικών δικτύων, μπορούμε να αναφερθούμε σε μερικές από τις βασικές αρχιτεκτονικές τους. Η πρώτη αρχιτεκτονική στην οποία θα αναφερθούμε είναι αυτή του πολυεπίπεδου αντίληπτρου (Multi-Layer Perceptron - MLP). Το MLP είναι ουσιαστικά ένας τύπος ANN που αποτελεί τη θεμελιώδη αρχιτεκτονική βάση για τα βαθιά νευρωνικά δίκτυα (deep neural networks- DNN) και στο οποίο η πληροφορία διαδίδεται ευθέως χωρίς βρόχους ανατροφοδότησης (Sarker, 2021).

Το MLP ουσιαστικά αποτελείται από 3 μέρη: το επίπεδο εισόδου, τα κρυφά επίπεδα που μπορεί να είναι ένα ή περισσότερα και το επίπεδο εξόδου (Perumal et al., 2024). Πρόκειται για ένα πλήρως συνδεδεμένο (fully connected) δίκτυο, πράγμα που σημαίνει ότι κάθε νευρώνας ενός επιπέδου συνδέεται με όλους τους νευρώνες του προηγούμενου επιπέδου (Haykin, 2009). Πρακτικά αυτό που συμβαίνει είναι πως το επίπεδο εισόδου δέχεται τα

δεδομένα και τα κανονικοποιεί, τα κρυφά επίπεδα επεξεργάζονται τα δεδομένα και οι τελικές αποφάσεις ή προβλέψεις ορίζονται στο επίπεδο εξόδου (Perumal et al., 2024). Η διαδικασία αυτή λαμβάνει χώρα στο επίπεδο εξόδου, μέσω συναρτήσεων ενεργοποίησης όπως η ReLU, αντίστροφη εφαπτομένη, σιγμοειδής, softmax, ανάλογα το πρόβλημα (Sarker, 2021). Τα ενδιάμεσα κρυφά επίπεδα αποτελούνται από νευρώνες που όπως αναλύθηκε σε προηγούμενο υποκεφάλαιο ουσιαστικά αθροίζουν τα βάρη συνυπολογίζοντας την πόλωση και εφαρμόζουν μια συνάρτηση ενεργοποίησης.

Ο Rosenblatt (1958) στα μέσα του 20ου αιώνα εισήγαγε την έννοια του νευρώνα. Οι πρώιμες χρήσεις του νευρώνα είχαν πολλούς περιορισμούς στις υπολογιστικές δυνατότητες, οι οποίοι όμως ξεπερνούνται με την εισαγωγή του νευρώνα στη δομή του νευρωνικού δικτύου και τη συγκρότηση ενός MLP (Haykin, 2009). Ουσιαστικά, ένα νευρωνικό δίκτυο είναι μια συνάρτηση πολλών μεταβλητών που της δίνουμε δεδομένα σε μορφή αριθμών και παράγει ένα αποτέλεσμα (Goodfellow et al., 2017). Τα δεδομένα μας μπορεί να είναι αριθμητικά ή ακόμα και λέξεις που μέσω τεχνικών τις μετατρέπουμε σε αριθμητικά δεδομένα. Τα δεδομένα μας μπορεί να είναι ακόμα και εικόνες, οι οποίες δίνονται με τη μορφή τανυστών όπου κάθε διάσταση μπορεί να είναι η ένταση ενός pixel για κάποιο από τα 3 βασικά χρώματα, τα οποία με τη σειρά τους αναπαρίστανται από μια διάσταση (A. Zhang et al., 2024). Δεδομένου λοιπόν ότι τα δεδομένα είναι αριθμητικά, η συνάρτηση που συνίσταται από το νευρωνικό δίκτυο, μπορεί να είναι γραμμική ή μη γραμμική. Η μη γραμμικότητα είναι απαραίτητη, καθώς όλα τα προβλήματα στη φύση δεν είναι προβλήματα αναλογίας, τα οποία θα μπορούσαν να λύνονται με μια γραμμική συνάρτηση (A. Zhang et al., 2024). Όπως έχουμε εξηγήσει, η μη γραμμικότητα εφαρμόζεται με τη χρήση συναρτήσεων ενεργοποίησης. Ο Géron (2023) εξηγεί πως οι MLP μπορούνε να χρησιμοποιηθούν τόσο για προβλήματα παλινδρόμησης, όσο και για προβλήματα ταξινόμησης. Ο ίδιος συμπληρώνει πως ειδικά στη 2η περίπτωση συνίσταται στο τελευταίο επίπεδο η χρήση μιας συνάρτησης ενεργοποίησης που θα μετατρέψει τα δεδομένα εξόδου σε πιθανότητες.

2.3.7 Συνελικτικά νευρωνικά δίκτυα (CNN)

Τα συνελικτικά νευρωνικά δίκτυα (convolutional neural networks-CNN) είναι μια ιδιαίτερα αποτελεσματική κατηγορία DNN που χρησιμοποιείται ευρέως σε τομείς όπως εντοπισμό αντικειμένων, αναγνώριση φωνής, υπολογιστική όραση (computer vision), κατηγοριοποίηση εικόνων και βιοπληροφορική (Perumal et al., 2024). Τα δίκτυα αυτά,

αποτελούν νευρωνικά δίκτυα εμπρόσθιας τροφοδότησης, που χρησιμοποιούν συνελικτικές δομές για να εξάγουν χαρακτηριστικά από τα δεδομένα εισόδου που τα τροφοδοτούν (Li et al., 2022). Τα μοντέλα που βασίζονται στην αρχιτεκτονική CNN μαθαίνουν απευθείας από τα δεδομένα εισόδου, χωρίς την ανάγκη για εξαγωγή χαρακτηριστικών από τον άνθρωπο (Sarker, 2021). Οι Janiesch et al. (2021) εξηγούν το παραπάνω, με το παράδειγμα της περίπτωσης αναγνώρισης αντικειμένων, όπου η αρχιτεκτονική των CNN επιτρέπει την ιεραρχική μάθηση χαρακτηριστικών. Πιο συγκεκριμένα, συμπληρώνουν πως τα πρώτα επίπεδα του δικτύου είναι υπεύθυνα για την εξαγωγή βασικών χαρακτηριστικών όπως ακμές και γωνίες, ενώ στα επόμενα επίπεδα αυτά τα στοιχεία συνδυάζονται σταδιακά σε πιο σύνθετες αναπαραστάσεις, επιτρέποντας στο μοντέλο να αναγνωρίζει ολόκληρες αναπαραστάσεις, όπως πχ ζώα, σπίτια ή αυτοκίνητα. Συνεπώς, το CNN αποτελεί εξέλιξη απλούστερων αρχιτεκτονικών όπως το MLP (Sarker, 2021).

Οι Goodfellow et al. (2017) εξηγούν ότι τα CNN συγκριτικά με άλλες αρχιτεκτονικές, έχουν τρία προτερήματα: διαμοιρασμό παραμέτρων (parameter sharing), αραιές αλληλεπιδράσεις (sparse interactions), ισοδύναμες αναπαραστάσεις (equivariant representations) οι οποίες προκύπτουν από τον αντιμεταθετικό χαρακτήρα της συνάρτησης που προκύπτει από το νευρωνικό δίκτυο. Όπως αναλύουν, ο διαμοιρασμός παραμέτρων σημαίνει πως αντίθετα με δίκτυα τύπου MLP όπου κάθε νευρώνας έχει ένα δικό του πίνακα με βάρη, εδώ έχουμε διαμοιρασμό, πράγμα που οδηγεί σε γρηγορότερους υπολογισμούς. Επίσης, εξηγούν ότι σε δίκτυα τύπου MLP είδαμε ότι κάθε νευρώνας ενός επιπέδου αλληλεπιδρά με κάθε νευρώνα γειτονικού επιπέδου, πράγμα που υλοποιείται με πολλαπλασιασμό πινάκων. Η αρχιτεκτονική των CNN είναι τέτοια που ακριβώς επειδή δεν είναι πλήρως συνδεδεμένο δίκτυο όπως το MLP, η τελική συνάρτηση που συνιστά το δίκτυο, έχει πολύ λιγότερες παραμέτρους (Benois-Pineau & Zemmari, 2021). Γενικότερα, τα προτερήματα αυτά οδηγούν σε λιγότερες παραμέτρους, πράγμα που κάνει το μοντέλο γρηγορότερο και ευκολότερο στην εκπαίδευση (Pouyanfar et al., 2019).

Η αρχιτεκτονική των CNN είναι επηρεασμένη από την ανθρώπινη οπτική αντίληψη (Li et al., 2022). Πιο αναλυτικά, η ανάπτυξη και η χρήση τους στην επεξεργασία εικόνων, αντλεί την έμπνευσή της από τις γνώσεις μας πάνω στη βιολογία του ανθρώπινου οπτικού συστήματος και τον τρόπο που επεξεργαζόμαστε τα οπτικά ερεθίσματα (Choi et al., 2016). Γενικότερα η λειτουργία των CNN παραπέμπει στα κύτταρα του οπτικού φλοιού τα οποία ευαισθητοποιούνται από μικρά τμήματα μιας εικόνας και όχι από όλη την εικόνα

ταυτόχρονα (L. Zhang et al., 2018). Αντίστοιχα, το CNN δεν επεξεργάζεται όλες τις παραμέτρους ξεχωριστά όπως ένα πλήρως συνδεδεμένο δίκτυο, αλλά έχει διαμοιραζόμενα βάρη (Pouyanfar et al., 2019).

Στη συνέχεια, θα περιγράψουμε την αρχιτεκτονική ενός τυπικού δικτύου CNN, καθώς και τον τρόπο που επεξεργάζονται τα δεδομένα. Δομικά, η αρχιτεκτονική των CNN αποτελείται από 2 μέρη: το μέρος που εξάγει χαρακτηριστικά από τα δεδομένα που εισάγουμε και το μέρος που εκτελεί την ταξινόμηση (Perumal et al., 2024). Σε ένα τυπικό δίκτυο CNN έχουμε μια σειρά από συνελικτικά επίπεδα (convolutional layers) που ακολουθούνται από συγκεντρωτικά επίπεδα (pooling layers), ενώ τα τελευταία επίπεδα είναι πλήρως συνδεδεμένα όπως έχουμε σε ένα MLP (Zaman et al., 2023). Τα CNN δέχονται δεδομένα εισόδου οργανωμένα σε τανυστές τριών διαστάσεων (H, W, D) όπου εάν αυτά αναπαριστούν μια φωτογραφία, το H είναι το ύψος, το W πλάτος, ενώ το D είναι ο αριθμός των καναλιών των χρωμάτων, τα οποία τυπικά είναι 3 για μια έγχρωμη εικόνα (Schlüter & Böck, 2014). Κάθε συνελικτικό επίπεδο εφαρμόζει μια σειρά από φίλτρα που ονομάζονται kernel τα οποία έχουν επίσης ένα μέγεθος που καθορίζεται από το ύψος και το πλάτος τους σε pixel (A. Zhang et al., 2024). Τα kernel επίσης αναπαρίστανται από τανυστές, ενώ το ύψος και το πλάτος τους πρέπει να είναι μικρότερο από αυτό του αρχείου που αποτελεί δεδομένο εισόδου στο δίκτυο (Pouyanfar et al., 2019). Τα kernel σαρώνουν την εικόνα και κάθε ένα είναι ρυθμισμένο έτσι ώστε να αναζητά ένα χαρακτηριστικό, οπότε και στο τέλος θα εξάγει έναν τανυστή με τα αποτελέσματα που ονομάζεται "feature map" αυτού του χαρακτηριστικού (Benois-Pineau & Zemmari, 2021). Οι L. Zhang et al. (2018) εξηγούν ότι κατά τη σάρωση τα kernel εκτελούν την πράξη της συνέλιξης (convolution) μεταξύ πίνακα με τα βάρη του δικτύου και τα pixel της εικόνας. Επιπροσθέτως, συμπληρώνουν ότι ουσιαστικά, ο feature map που προκύπτει δείχνει πόσο "έντονη" είναι η παρουσία αυτού του χαρακτηριστικού στο εν λόγω κανάλι της εικόνας. Οι Pouyanfar et al. (2019) παρουσιάζουν τον τύπο της συνέλιξης ως:

$$h^k = f(W^k * x + b^k) \tag{30}$$

Όπου h είναι το εκάστοτε feature map ως τανυστής, Wκαι b είναι οι τανυστές με τα βάρη και τις πολώσεις αντίστοιχα. Η f είναι κάποια συνάρτηση ενεργοποίησης που μπορεί να εφαρμοστεί για εισαγωγή μη γραμμικότητας, ενώ ο δείκτης k αναφέρεται στο εκάστοτε kernel που υπολογίζεται. Το μέγεθος των kernel καθώς και ο τρόπος που θα σαρώσουν την εικόνα, ορίζονται από υπερπαραμέτρους όπως πχ το "stride", ενώ ένα kernel ενδέχεται γύρω

του να έχει επιπλέον pixel (padding) που δεν αντιστοιχούν σε κάποια αριθμητική τιμή, ώστε να σαρώνει πληρέστερα στα άκρα της εικόνας (A. Zhang et al., 2024).

Μετά το συνελικτικό επίπεδο, κάθε feature map που προκύπτει αποτελεί δεδομένο εισόδου για τα pooling layers τα οποία υποβιβάζουν την ανάλυσή τους (Schlüter & Böck, 2014). Με τον όρο ανάλυση αναφερόμαστε στον αριθμό pixel του ύψους και του πλάτους. Η μείωση της ανάλυσης δεν οδηγεί σε απώλεια σημαντικής πληροφορίας, ίσα ίσα επιτρέπει στο μοντέλο να "επικεντρωθεί" στα σημαντικά χαρακτηριστικά αυξάνοντας την πιστότητα του μοντέλου (Zaman et al., 2023). Επιπροσθέτως, η διαδικασία αυτή επιταχύνει την εκπαίδευση και περιορίζει το overfitting (Pouyanfar et al., 2019). Οι A. Zhang et al. (2024) αναφέρουν ότι το pooling επιτυγχάνεται με ένα pooling window το οποίο επίσης ορίζεται από υπερπαραμέτρους. Επίσης, προσθέτουν ότι το pooling window σε κάθε σαρώνει έναν αριθμό από pixel και δίνει το μέσο όρο της έντασής τους, ώστε εν τέλει εάν σαρώσει μια εικόνα, να εξάγει μια εικόνα μικρότερων διαστάσεων. Στη σχέση (30), είδαμε πως το convolution layer ακολουθείται από μια συνάρτηση ενεργοποίησης. Εναλλακτικά, αυτή η συνάρτηση μπορεί να τοποθετηθεί μετά το pooling layer. Οι W. Xu (2024) στη χρήση μοντέλων CNN για ταξινόμηση μουσικών ειδών, αναφέρουν πως το convolution layer και το pooling layer συνήθως ακολουθούνται από μια συνάρτηση ενεργοποίησης σιγμοειδή ή ReLU, η οποία χρησιμοποιείται για να εισάγει μη γραμμικότητα στη συνολική συνάρτηση. Εν συνεχεία, το τελικό MLP δίκτυο που βρίσκεται στο πέρας της αρχιτεκτονικής, δέχεται τα δεδομένα που προκύπτουν και σχηματίζει μια αφηρημένη αναπαράσταση της πληροφορίας (Pouyanfar et al., 2019). Αξίζει να σημειωθεί ότι πριν το MLP δίκτυο, είναι δυνατόν να υπάρχουν πάνω από μια αλληλουχίες convolution layer - pooling layer - activation function, ανάλογα την αρχιτεκτονική του μοντέλου (Sarker, 2021). Ως τελικό επίπεδο στα μοντέλα ταξινόμησης, υπάρχει μια συνάρτηση ενεργοποίησης που θα μετατρέψει τα δεδομένα σε πιθανότητες αντιστοιχίας με κάποια κλάση (Zaman et al., 2023).

Όπως αναφέρθηκε ήδη, τα CNN χρησιμοποιούνται σε διάφορες εφαρμογές. Ο Sarker (2021) αναφέρει ότι παραδείγματα αποτελούν εφαρμογές σχετικές με οπτική αναγνώριση, ανάλυση εικόνων ιατρικού περιεχομένου, διαχωρισμό εικόνων σε τμήματα, αλλά και σε επεξεργασία φυσικής γλώσσας (natural language processing - NLP). Η χρήση τους είναι αρκετά συνδεδεμένη με τη χρήση με εικόνες και όπως είδαμε, η αρχιτεκτονική τους είναι τέτοια ώστε να δέχεται αρχεία εικόνας ως δεδομένα εισόδου με αρκετή λειτουργικότητα. Η εφαρμογή της πράξης της συνέλιξης τους δίνει τη δυνατότητα να εντοπίζουν τη θέση των

διάφορων χαρακτηριστικών στις εικόνες, πράγμα που τους επιτρέπει να βρίσκουν μοτίβα στην εμφάνιση αυτών των χαρακτηριστικών (Chelladurai & Sujatha, 2023). Ταυτόχρονα όμως, τα CNN φαίνεται ότι έχουν τη δυνατότητα να χρησιμοποιηθούν και σε εφαρμογές επεξεργασίας ήχου, δεδομένου ότι τους δίνεται μια κατάλληλη οπτική αναπαράστασή του (Shuvaev et al., 2017). Στην επεξεργασία ήχου, χρησιμοποιούνται για αναγνώριση φωνής, ταξινόμηση ήχων, μουσικές προτάσεις, διαχωρισμό περιεχομένων σήματος από αυτό και πολλές άλλες εφαρμογές (Zaman et al., 2023). Γενικά, τα CNN έχουν σχεδιαστεί έτσι ώστε τα αντίστοιχα μοντέλα να εντοπίζουν έντονα οπτικά χαρακτηριστικά μιας εικόνας, κάτι που έχει εφαρμογή και στις ηχητικές αναπαραστάσεις, ακόμα κι αν αυτές δεν έχουν απαραίτητα την ίδια τοπολογία με τις περισσότερες εικόνες (Choi et al., 2016). Ο όρος «τοπολογία» εδώ, αναφέρεται στη χωρική και δομική οργάνωση των δεδομένων. Συνήθως στην επεξεργασία ήχου με μοντέλα αρχιτεκτονικής CNN τροφοδοτούμε την είσοδό τους όχι με την ίδια την κυματομορφή, αλλά κάποια αναπαράσταση αυτής όπως φασματογραφήματα ή MFCCs (Zaman et al., 2023).

Η αρχιτεκτονική που περιεγράφηκε νωρίτερα, αποτελεί μια τυπική αρχιτεκτονική CNN. Υπάρχουν διάφορες παραλλαγές που έχουν παρουσιαστεί με τη μορφή επώνυμων μοντέλων DL. Η πρώτη εφαρμογή των CNNs υλοποιήθηκε για την αρχιτεκτονική του LeNet το 1998 και αφορούσε την αναγνώριση OCR και χαρακτήρων σε κείμενα (Shinde & Shah, 2018). Οι Krizhevsky et al. (2012) παρουσίασαν την αρχιτεκτονική του AlexNet για ταξινόμηση εικόνων. Αρκετά χαρακτηριστική είναι και η αρχιτεκτονική του VGG (Visual Geometry Group) επίσης για εφαρμογή στον κλάδο της υπολογιστικής όρασης (Simonyan & Zisserman, 2014). Άλλες γνωστές αρχιτεκτονικές είναι οι Inception, ResNet (Residual Networks), WideResNet, FractalNet, SqueezeNet, InceptionResNet, Xception (Extreme Inception), MobileNet, DenseNet (Dense Convolutional Network), SENet (Squeeze-and-Excitation Network), Efficientnet (Perumal et al., 2024).

2.3.8 Επαναληπτικά νευρωνικά δίκτυα (RNN) – δίκτυο LSTM – μονάδα GRU

Τα RNN είναι μια γενική κατηγορία από DNN, στα οποία συναντάμε εσωτερική μνήμη, κάτι που τους δίνει τη δυνατότητα να δέχονται τα δεδομένα με συγκεκριμένη σειρά (Perumal et al., 2024). Πρόκειται για μια ευρέως διαδεδομένη αρχιτεκτονική DNN, στην οποία όμως τροφοδοτούμε την είσοδο με τα δεδομένα εξόδου (Mandic & Chambers, 2001). Καινοτομία λοιπόν των RNN είναι ότι τα δεδομένα που δέχονται έχουν μια συγκεκριμένη σειρά, ενώ δε χρειάζεται να είναι συγκεκριμένου μεγέθους (Chelladurai & Sujatha, 2023).

Τα μοντέλα αυτά μπορούν να δεχθούν δεδομένα των οποίων η σειρά έχει σημασία (όπως πχ ένα κείμενο) και δεδομένα στα οποία και ο χρόνος έχει σημασία (όπως πχ η τιμή μιας μετοχής που εξαρτάται από την ημέρα), με τα πρώτα να είναι γνωστά ως "sequential data" και τα δεύτερα ως "time series data" (Sarker, 2021). Αυτή η δυνατότητα είναι αρκετά σημαντική σε γλωσσικά μοντέλα, καθώς η σειρά που είναι δοσμένες οι λέξεις μιας πρότασης έχουν σημασία για το νόημά της (Perumal et al., 2024). Άλλες αρχιτεκτονικές δικτύων δεν μπορούν να διαχειριστούν δεδομένα όταν η μορφή τους είναι δοσμένη ως σειρά (Zaman et al., 2023). Τα RNN αντιμετωπίζουν αυτή τη δυσκολία με το να δέχονται τα δεδομένα ανά τμήματα και για κάθε ένα τμήμα να παράγουν μια έξοδο (A. Zhang et al., 2024). Σε γενικό πλαίσιο, η αρχιτεκτονική του δικτύου περιλαμβάνει το επίπεδο εισόδου, ένα κρυφό επίπεδο και ένα επίπεδο εξόδου, με το δεύτερο να καλείται "hidden state" και ουσιαστικά να λειτουργεί ως μνήμη του δικτύου (Pouyanfar et al., 2019). Το RNN λοιπόν, μπορεί να παράγει διαφορετικές εξόδους δεδομένων, ανάλογα τη στιγμή και τη σειρά των δεδομένων που του δίνονται (Chelladurai & Sujatha, 2023). Όπως και στις υπόλοιπες αρχιτεκτονικές, τα RNN εκπαιδεύονται από τα δεδομένα εισόδου, όμως το γεγονός ότι αυτοτροφοδοτούνται παράλληλα με τα δεδομένα εξόδου τους, τους δίνει τη δυνατότητα να έχουν κάποιου είδους "μνήμη", επεμβαίνοντας έτσι στην επιρροή των δεδομένων εισόδου τους (Sarker, 2021). Γενικότερα, στα RNN η διαδικασία εξαγωγής δεδομένων γίνεται σε χρονικά βήματα (time steps), όπου σε κάθε βήμα δίνεται ένα τμήμα των δεδομένων ως είσοδος, γίνεται επεξεργασία των δεδομένων συνυπολογίζοντας τα περιεχόμενα της μνήμης και τελικά παράγεται μια έξοδος (Mandic & Chambers, 2001). Αυτό σημαίνει ότι αντίθετα με άλλες αρχιτεκτονικές όπου είσοδος και έξοδος είναι ανεξάρτητες, στα RNN η έξοδος εξαρτάται άμεσα από τη σειρά που δίνονται τα δεδομένα (Sarker, 2021).

Τα RNN λοιπόν, επεξεργάζονται τα δεδομένα εισόδου τμηματικά, ενώ είναι σε θέση να διατηρούν ιστορικό από την έξοδο των προηγούμενων βημάτων ως "hidden state" (Lecun et al., 2015). Για την εκπαίδευση των RNN χρησιμοποιείται επίσης ο αλγόριθμος του backpropagation (W. Xu, 2024). Σε αυτά τα δίκτυα μπορούμε να ορίσουμε την έξοδο ενός χρονικού βήματος ως "hidden unit" και να δεχθούμε ότι αυτή είναι ταυτόχρονα είσοδος σε ένα άλλο unit το οποίο όμως είναι το ίδιο το δίκτυο (Lecun et al., 2015). Σχηματικά, ουσιαστικά έχουμε ένα hidden state το οποίο δέχεται μια είσοδο και παράγει μια έξοδο (A. Zhang et al., 2024). Η έξοδος αυτή, μπορεί να ληφθεί απομονωμένα, αλλά μπορεί και να τροφοδοτήσει ξανά το hidden state, ενώ αυτή η διαδικασία επαναλαμβάνεται όσες φορές

θέλουμε, απλά τυπικά αναλόγως το βήμα το hidden state μπορεί να αριθμηθεί διαφορετικά (Benois-Pineau & Zemmari, 2021). Αυτό μπορεί να φανεί και από τη σχέση που δίνει την έξοδο ενός hidden unit στο χρόνο:

$$s_t = f(U \cdot x_t + W \cdot s_{t-1} + b) \tag{31}$$

Όπου f είναι συνάρτηση ενεργοποίησης (συνήθως αντίστροφη εφαπτομένη ή σιγμοειδής), s_t είναι το hidden state μιας στιγμής εκφρασμένο ως τανυστής, τα U και W είναι πίνακες βαρών, το b είναι διάνυσμα πολώσεων, ενώ x είναι τα δεδομένα εισόδου (Leskovec et al., 2020). Οι Lecun et al. (2015), αναφέρουν πως συνοπτικά μπορούμε να δούμε τα RNN σαν δίκτυα συνδεδεμένα με τον εαυτό τους, πράγμα που βοηθά στην κατανόηση του πως λειτουργεί εν προκειμένω το backpropagation. Κατόπιν, συμπληρώνουν πως υπό αυτό το πρίσμα, τα RNN είναι στην ουσία πολύ βαθιά νευρωνικά δίκτυα στα οποία όλα τα επίπεδα μοιράζονται τα ίδια βάρη. Ταυτόχρονα, το "hidden unit" ουσιαστικά λειτουργεί ως μνήμη στο RNN και όπως φαίνεται από τη σχέση (31), συνυπολογίζεται για τον υπολογισμό της εξόδου του επόμενου βήματος στο χρόνο (Leskovec et al., 2020). Όπως αναφέρθηκε νωρίτερα, οι υπολογισμοί του RNN γίνονται σε time steps. Σύμφωνα με τους L. Zhang et al. (2018), σε κάθε time step, χρήσει της σχέσης (31) υπολογίζεται το τρέχον hidden state ενώ είναι δυνατόν να έχουμε και μια έξοδο που ορίζεται από αυτό το hidden state που υπολογίστηκε. Οι ίδιοι, συμπληρώνουν ότι αν για παράδειγμα το RNN επεξεργάζεται κείμενο μπορεί να δέχεται σε κάθε time step μια λέξη του κειμένου, οπότε ο αριθμός των time steps και των αντίστοιχων εξόδων που παίρνουμε, εξαρτάται από τον αριθμό λέξεων του κειμένου. Ως έξοδο του RNN λοιπόν, είναι εφικτό να πάρουμε έναν τανυστή με όλα τα hidden states, για κάθε time step που έλαβε χώρα και είτε να τα χρησιμοποιήσουμε όλα, ή μόνο το τελευταίο (A. Zhang et al., 2024).

Τα RNN μπορεί να είναι αρκετά ισχυρά δυναμικά συστήματα, αλλά η εκπαίδευσή τους πολλές φορές είναι δύσκολη (Lecun et al., 2015). Λόγω της αρχιτεκτονικής τους, τα απλά RNN δίκτυα αντιμετωπίζουν το πρόβλημα των εξαφανιζόμενων κλίσεων (vanishing gradients), το οποίο καθιστά την εκπαίδευση με μακροσκελείς σειρές δεδομένων δύσκολη (A. Zhang et al., 2024). Είδαμε σε προηγούμενο κεφάλαιο, ότι κατά την διαδικασία της εκπαίδευσης με τον αλγόριθμο του backpropagation, υπολογίζεται μέσω του κανόνα αλυσίδας, η παράγωγος της συνάρτησης κόστους ως προς το κάθε βάρος. Ο κανόνας αλυσίδας εξ ορισμού, περιλαμβάνει ένα σύνολο μερικών παραγώγων. Το πρόβλημα των vanishing gradients προκύπτει όταν αυτές οι μερικές παράγωγοι είναι πολύ μικρές, με

αποτέλεσμα τα βάρη να ενημερώνονται με τέτοιον τρόπο που η τιμή τους αλλάζει ελάχιστα (Benois-Pineau & Zemmari, 2021). Για την αντιμετώπιση αυτού του προβλήματος, έχουν προταθεί βελτιώσεις με τη μορφή εναλλακτικών αρχιτεκτονικών, όπως οι LSTM και GRU (Sarker, 2021). Αντίστοιχα, ορίζεται το πρόβλημα των exploding gradients όπου οι μερικές παράγωγοι παίρνουν μεγάλες τιμές (Leskovec et al., 2020).

Το προαναφερθέν πρόβλημα των RNN να λειτουργήσουν με μεγάλες σειρές δεδομένων, οφείλεται ακριβώς στα vanishing gradients. Παρακάτω θα αναλύσουμε τις αρχιτεκτονικές LSTM και GRU που προτείνονται ως λύση αυτού του προβλήματος. Οι Hochreiter και Schmidhuber (1997) εισήγαγαν την αρχιτεκτονική του LSTM (Long-Short term memory) για την αντιμετώπιση του προβλήματος των vanishing gradients εισάγοντας στη δομή κελί μνήμης (memory cell) και πύλες (gates) που ελέγχουν τη ροή δεδομένων. Πιο συγκεκριμένα, υφίσταται μια πύλη λήθης (forget gate) που ελέγχει ποια δεδομένα θα σβηστούν από τη μνήμη, μια πύλη εισόδου (input gate) που ελέγχει ποια δεδομένα θα εισέλθουν στη μνήμη και μια πύλη εξόδου (output gate) που ελέγχει ποια δεδομένα θα εξαχθούν (Sarker, 2021). Σε ένα LSTM η αρχιτεκτονική είναι πιο περίπλοκη, καθώς δεν έχουμε απλά το δίκτυο να συνδέεται με τον εαυτό του, αλλά έχουμε 2 καταστάσεις του δικτύου: το hidden state που υπάρχει και στο απλό RNN και το cell state (L. Zhang et al., 2018). Σε κάθε χρονικό βήμα, το δίκτυο πριν τροφοδοτηθεί με τα δεδομένα του hidden state, "αποφασίζει" τι δεδομένα θα απορρίψει από το cell state, χρησιμοποιώντας την forget gate (A. Zhang et al., 2024). Κατόπιν, μέσω της input gate, αποφασίζει ποια από αυτά τα δεδομένα θα αποθηκευτούν στο cell state για τα επόμενα χρονικά βήματα, ενώ μέσω της output gate επιλέγει ποια από τα αποθηκευμένα δεδομένα θα συμβάλλουν στην έξοδο (Perumal et al., 2024). Το LSTM θεωρείται από τις πιο επιτυχείς RNN αρχιτεκτονικές, καθώς κατά πολύ λύνει το πρόβλημα της εκπαίδευσης τέτοιων δικτύων. Η πολυπλοκότητα του LSTM ανά χρονικό βήμα και ανά βάρος είναι O(1) (Shinde & Shah, 2018).

Οι Chung et al. (2014) πρότειναν μια αρχιτεκτονική για τη βελτίωση των LSTM, όπου υπάρχουν λιγότερες παράμετροι και αντί για τις προαναφερθείσες πύλες, εδώ υφίσταται μια πύλη επαναφοράς (reset gate) και μια πύλη ενημέρωσης (update gate). Η αρχιτεκτονική αυτή έγινε γνωστή ως GRU (Gated recurrent unit) και λειτουργεί πιο προσαρμοστικά στις μεγάλες ακολουθίες δεδομένων, χωρίς απαραίτητα να απορρίπτει πληροφορία από προηγούμενα μέρη της ακολουθίας (Sarker, 2021). Το GRU αποτελεί μια ελαφρώς πιο απλοποιημένη παραλλαγή του LSTM δομικά, η οποία συχνά είναι αποδοτικότερη και

ταχύτερη στους υπολογισμούς (Chung et al., 2014). Η αρχιτεκτονική του GRU αντικαθιστά τις input gate και forget gate με την reset gate, ενώ δεν έχει ξεχωριστό cell state για μνήμη (Perumal et al., 2024). Τα μέρη του GRU του επιτρέπουν να διατηρήσει επιλεκτικά μέρη της μνήμης του και να τα χρησιμοποιήσει για τον υπολογισμό της εξόδου κάθε βήματος (A. Zhang et al., 2024). Η update gate καθορίζει το ποσοστό της παρελθοντικής πληροφορίας που θα διατηρηθεί και θα συνδυαστεί με τα δεδομένα εισόδου του εκάστοτε βήματος, ενώ η reset gate καθορίζει το ποσοστό της παρελθοντικής πληροφορίας που θα απορριφθεί (Perumal et al., 2024). Σύμφωνα με τους A. Zhang et al. (2024) το τρέχον περιεχόμενο της μνήμης σε ένα βήμα υπολογίζεται από την πληροφορία που επιτρέπει η reset gate, σε συνδυασμό με το hidden state και την τρέχουσα είσοδο. Οι ίδιοι συμπληρώνουν ότι η τελική κατάσταση της μνήμης που θα ανατροφοδοτήσει το δίκτυο διαμορφώνεται από το τρέχον περιεχόμενο της μνήμης και το προηγούμενο hidden state, ενώ μια πύλη εξόδου (output gate) μπορεί να χρησιμοποιηθεί για να οδηγήσει το σήμα αυτό και ως έξοδο του εν λόγω βήματος.

Συνήθεις εφαρμογές των RNN αποτελούν τα προβλήματα πρόβλεψης, μετάφραση κειμένου, NLP, σύνοψη κειμένου, αναγνώριση φωνής κ.α. (Sarker, 2021). Σύμφωνα με τους Zaman et al. (2023), τα RNN είναι σε θέση να επεξεργαστούν ηχητικά σήματα και για παράδειγμα να ταξινομήσουν το ακουστικό περιεχόμενο μιας ηχογράφησης βλέποντας αν περιέχει μουσική ή ομιλία. Οι ίδιοι, αναφέρουν επίσης ότι λόγω του τρόπου λειτουργίας τους, τα RNN μπορούν να διακρίνουν την εξέλιξη του ηχητικού σήματος στο χρόνο, καθώς το δέχονται ως δεδομένα εισόδου με μια σειριακή μορφή.

2.3.9 Νευρωνικό δίκτυο Transformer

Στα χρόνια της άνθησης της βαθιάς μάθησης μετά το 2010, τα MLP, CNN, και RNN δίκτυα κυριάρχησαν και ενώ υπήρξε πρόοδος σε τεχνικές βελτιστοποίησης, οι αρχιτεκτονικές αυτές παρέμειναν πιστές στη βασική τους δομή, γεγονός που δείχνει ότι η πρόοδος κατά πολύ βασίστηκε στην αύξηση της υπολογιστικής ισχύος και των διαθέσιμων δεδομένων (A. Zhang et al., 2024). Οι Vaswani et al. (2017) προς το τέλος της δεκαετίας εισήγαγαν την αρχιτεκτονική του Transformer για εφαρμογές NLP. Έκτοτε, η προσέγγιση αυτή αποτελεί μια από τις θεμελιώδεις αρχιτεκτονικές στη βαθιά μάθηση (Perumal et al., 2024). Ειδικά από το 2020 και μετά η βασική αρχιτεκτονική που επιλέγεται για εφαρμογές NLP και όχι μόνο, είναι αυτή του Transformer με παραδείγματα μοντέλων όπως το BERT, GPT-2 και GPT-3 (A. Zhang et al., 2024). Οι Transformers λοιπόν είναι ένα είδος DNN το οποίο

χρησιμοποιεί την έννοια του attention (προσοχής), γεγονός που τους επιτρέπει να επεξεργάζονται μακροσκελείς σειρές δεδομένων μεταβλητού μήκους (Zaman et al., 2023). Το δεύτερο βασικό χαρακτηριστικό των Transformers, είναι η χρήση του positional embedding (Božić & Horvat, 2024). Η έννοια του attention χρησιμοποιείται και σε RNN δίκτυα στα οποία όμως τα δεδομένα εισάγονται διαδοχικά, ενώ στους Transformers η επεξεργασία τους γίνεται παράλληλα, διατηρώντας όμως τη σημασία τους στη σειρά χάριν του positional embedding (A. Zhang et al., 2024).

Πριν την περαιτέρω ανάλυση των Transformers, κρίνεται σκόπιμο το να εξηγηθεί η έννοια του attention. Η ιδιότητα αυτή, δίνει σε ένα μοντέλο να δώσει επιλεκτικά περισσότερη «προσοχή», ενισχύοντας τη σημασία συγκεκριμένων τμημάτων των δεδομένων εισόδου (A. Zhang et al., 2024). Για να το πετύχει αυτό, ορίζει και χρησιμοποιεί τρία στοιχεία ονόματι “query”, “key” και “value” (Božić & Horvat, 2024).Ο Transformer εφαρμόζουν ένα συγκεκριμένο είδος attention, μέσω του οποίου υπολογίζεται η σχέση των στοιχείων μιας σειράς δεδομένων εισόδου μεταξύ τους και με τον εαυτό τους (A. Zhang et al., 2024). Πρακτικά, στην περίπτωση του NLP, όπου τα δεδομένα εισόδου είναι προτάσεις, αυτό μπορεί να δώσει τη σχέσεις μεταξύ των λέξεων, πράγμα που σχετίζεται άμεσα με την ερμηνεία της πρότασης (Xiao & Zhu, 2023). Τα query, key και value επιτρέπουν την παραμετροποίηση αυτών των συσχετίσεων (Božić & Horvat, 2024). Επιπλέον όμως, οι Transformers εφαρμόζουν multi-head attention, μια τεχνική που δίνει τη δυνατότητα να τα query, key και value να χρησιμοποιήσουν και διαφορετικούς συσχετισμούς για τα δεδομένα εισόδου, διαμορφώνοντας έτσι διαφορετικά επίπεδα (layers) συσχετισμών (Xiao & Zhu, 2023). Οι Vaswani et al. (2017) στην αρχική τους έρευνα και παρουσίαση του multi-head attention συμπεριέλαβαν 8 τέτοια επίπεδα.

Το positional embedding σχετίζεται με τον τρόπο που οι τα δεδομένα εισόδου διαμορφώνονται για να υποστούν επεξεργασία από το δίκτυο. Για την εξήγηση του embedding γενικότερα, θα χρησιμοποιηθεί το παράδειγμα όπου τα δεδομένα εισόδου είναι κείμενο, καθώς οι Transformers εφαρμόστηκαν εκεί αρχικά όπως ήδη ειπώθηκε. Οι Transformers αναπαριστούν τις λέξεις ως “embeddings” (Xiao & Zhu, 2023). Γενικότερα, τα embeddings είναι ένας τρόπος αριθμητικής αναπαράστασης των λέξεων (οπότε και λέγονται “word embeddings”) και πιο συγκεκριμένα διανυσματικής αναπαράστασης, οπότε και αντί για λέξη, έχουμε ένα διάνυσμα (Géron, 2023). Οι A. Zhang et al. (2024) εξηγούν ότι το διάνυσμα αυτό έχει οποιονδήποτε πεπερασμένο αριθμό διαστάσεων, ενώ όσες

περισσότερες διαστάσεις, τόσο «αναλυτικότερη» είναι η περιγραφή της λέξης. Επίσης προσθέτουν, ότι σε μια φράση οι λέξεις αναφέρονται ως "tokens" καθένα από τα οποία είναι ένα διάνυσμα, με τις δικές του "embedding dimensions". Συνεπώς μια φράση μπορεί να αναπαρασταθεί ως τανυστής όπου κάθε λέξη είναι ένα token το οποίο αναπαρίσταται ως διάνυσμα. Επιπροσθέτως, σε αυτά τα embeddings, προστίθεται ένα "positional embedding" του οποίου ρόλος είναι να διατηρεί τη σειρά των tokens στην αλληλουχία δεδομένου ότι δεν έχουμε αρχιτεκτονική όπως πχ των RNN όπου τα δεδομένα δίνονται με συγκεκριμένη σειρά και έτσι η αλληλουχία και η σημασία της σειράς των λέξεων διατηρείται (Vaswani et al., 2017). Πολύ σημαντικό, επίσης, είναι το γεγονός ότι αυτά τα embeddings είναι παραμετροποιημένες αναπαραστάσεις τις οποίες το μοντέλο είναι σε θέση να μάθει (Xiao & Zhu, 2023),

Οι Vaswani et al. (2017), στην αρχική αρχιτεκτονική του Transformer, περιλάμβαναν έναν κωδικοποιητή (encoder) και έναν αποκωδικοποιητή (decoder). Τα δύο αυτά μέρη δέχονται τα δεδομένα ήδη ως embeddings στα οποία έχει προστεθεί το positional embedding, πράγμα που σημαίνει ότι πριν την είσοδο, πρέπει να έχει γίνει η σχετική επεξεργασία (Perumal et al., 2024). Οι Božić και Horvat (2024) περιγράφουν ότι οι encoder και decoder αποτελούνται από μια σειρά όμοιων επιπέδων τα οποία έχουν 2 υποεπίπεδα: ένα multihead attention και ένα πλήρως συνδεδεμένο δίκτυο εμπρόσθιας διάδοσης (feed-forward network). Επιπροσθέτως, αναφέρουν ότι τα επίπεδα αυτά συνδέονται με residual connections ενώ περιλαμβάνουν κανονικοποίηση. Ο decoder είναι λίγο πιο σύνθετος καθώς περιλαμβάνει και ένα υποεπίπεδο "masked multi-head attention", ενώ δέχεται και τα δεδομένα του encoder (Perumal et al., 2024).

Η αρχιτεκτονική του Transformer αρχικά σχεδιάστηκε για NLP, αργότερα όμως εφαρμόστηκε σε εικόνες (Zaman et al., 2023). Οι Dosovitskiy et al. (2020) παρουσίασαν μια εκδοχή του Transformer γνωστή ως "Vision Transformer" (ViT), όπου χρησιμοποίησαν την αρχιτεκτονική αυτή και έδειξαν ότι μπορεί να χρησιμοποιηθεί για ταξινόμηση εικόνων. Επειδή στην περίπτωση των εικόνων δεν έχουμε λέξεις, χρειάζεται να γίνουν κάποιες αντιστοιχίες για τα δεδομένα εισόδου. Οι Xiao και Zhu (2023) εξηγούν ότι μια εικόνα, πριν εισέλθει στον encoder, χωρίζεται σε τμήματα (patches) τα οποία αντιστοιχούν στα tokens. Πιο συγκεκριμένα, μια εικόνα μεγέθους $H \times W \times C$ θα χωριστεί σε $\frac{HW}{P^2}$ τμήματα, όπου το κάθε τμήμα έχει σχήμα $P \times P \times C$. Το H αντιστοιχεί σε ύψος το W σε πλάτος, το C είναι το κανάλι του χρώματος, ενώ το P είναι η πλευρά του τμήματος. Έτσι λοιπόν αντί για

tokens/λέξεις, έχουμε patches και αντί για embedding dimensions, ουσιαστικά έχουμε τις διαστάσεις, δηλαδή το κάθε pixel του patch. Σαφώς, οι διαστάσεις του patch πρέπει να είναι τέτοιες, ώστε να μπορεί να χωριστεί σε έναν ακέραιο αριθμό τμημάτων. Οι Dosovitskiy et al. (2020) επίσης, προτείνουν έναν εναλλακτικό τρόπο χωρισμού της εικόνας, χρήσει των feature maps ενός convolution layer. Πιο συγκεκριμένα, το convolution layer θα παράγει έναν αριθμό από features ο οποίος αντιστοιχεί στον συνολικό αριθμό των pixels ενός patch για όλα τα κανάλια. Το layer αυτό όπως είδαμε, θα παράγει έναν τανυστή όπου για κάθε feature, έχουμε έναν feature map. Ο flattened feature map αντιστοιχεί στον αριθμό των τμημάτων ενώ το κάθε feature αντιστοιχεί σε ένα pixel του τμήματος ή ένα "embedding dimension". Η αρχιτεκτονική του ViT περιλαμβάνει μόνο τον encoder ο οποίος ουσιαστικά θα χρησιμοποιήσει τα embeddings για να φτιάξει αντίστοιχες πρότυπες αναπαραστάσεις μέσω του μηχανισμού του attention (Xiao & Zhu, 2023). Στη συνέχεια αυτές μπορούν να τροφοδοτήσουν ένα MLP και έναν ταξινομητή ώστε να γίνει ταξινόμηση (Dosovitskiy et al., 2020).

Η αρχιτεκτονική του Transformer έχει χρησιμοποιηθεί και σε ηχητικά σήματα χρησιμοποιώντας φασματογραφήματα ως αναπαραστάσεις (Zaman et al., 2023). Δεδομένου ότι ο ήχος όπως είδαμε μπορεί να αναπαρασταθεί οπτικά και υπάρχει ήδη ένα μοντέλο όπως το ViT για ταξινόμηση εικόνων, φαίνεται πως κάτι τέτοιο είναι εφικτό. Οι Gong et al. (2021) βασισμένοι σε αυτό το σκεπτικό παρουσίασαν το μοντέλο του AST (Audio Spectrogram Transformer). Το σκεπτικό στο μοντέλο αυτό είναι παρόμοιο με το ViT, μόνο που ως δεδομένα εισόδου έχουμε Mel Spectrograms τα οποία αντίστοιχα τμηματοποιούνται για να γίνουν embeddings και να τροφοδοτήσουν τον encoder του Transformer. Χαρακτηριστικό είναι το ότι όπως αναφέρεται, το εν λόγω μοντέλο χρησιμοποιεί ήδη διαμορφωμένα βάρη από το ViT και κάνει χρήση τεχνικών όπως Transfer Learning στις οποίες αναφερθήκαμε σε προηγούμενο υποκεφάλαιο. Οι εφαρμογές των Transformer στη μουσική δεν περιορίζονται μόνο σε ταξινόμηση, καθώς υπάρχουν μοντέλα τα οποία χρησιμοποιούν την αρχιτεκτονική αυτή ακόμα και για τη σύνθεση μουσικής όπως το Music Transformer το οποίο έχει τη δυνατότητα να βρει μοτίβα σε ένα κομμάτι και να συνεχίσει την εκτέλεση (C.-Z. A. Huang et al., 2018). Άλλα ενδεικτικά μοντέλα είναι το FastSpeech, Wave2Midi2Wave, RobuTrans, Jukebox κ.α. (Božić & Horvat, 2024).

3. Προετοιμασία και σχεδίαση συστήματος

Στο κεφάλαιο αυτό παρουσιάζεται η προετοιμασία των δεδομένων και των βασικών στοιχείων του συστήματος για την πειραματική διαδικασία. Για την υλοποίηση ενός συστήματος αναγνώρισης μουσικών ειδών, είναι δυνατή η αξιοποίηση διαφορετικών αρχιτεκτονικών βαθιάς μάθησης. Θα παρουσιάσουμε λοιπόν τη δομή του κάθε μοντέλου το οποίο αντιστοιχεί σε κάθε αρχιτεκτονική. Παράλληλα, ένα από τα σημαντικότερα βήματα για την ανάπτυξη ενός τέτοιου συστήματος, είναι η διαχείριση των δεδομένων που θα χρησιμοποιηθούν για την εκπαίδευσή του. Κρίνεται σκόπιμο λοιπόν το να περιγράψουμε την όλη διαδικασία επεξεργασίας των δεδομένων αυτών.

Η υλοποίηση του συστήματος γίνεται με χρήση της γλώσσας προγραμματισμού Python. Η συγκεκριμένη γλώσσα χρησιμοποιείται ευρέως για την υλοποίηση μοντέλων βαθιάς μάθησης, καθώς διαθέτει αρκετά σύγχρονες βιβλιοθήκες ακριβώς για αυτό το σκοπό. Η βιβλιοθήκη που θα χρησιμοποιήσουμε για την πραγματοποίηση των πειραμάτων είναι η βιβλιοθήκη PyTorch. Οι Paszke et al. (2019) παρουσίασαν την PyTorch ως μια βιβλιοθήκη ανοιχτού κώδικα για ML και DL, με χρήση για κατασκευή νευρωνικών δικτύων. Η PyTorch αναπτύχθηκε από τη “Meta AI” (πρώην Facebook AI Research) και αποτελεί ένα από τα πιο δημοφιλή εργαλεία στην έρευνα και τη βιομηχανία του DL.

Το περιβάλλον εργασίας όπου θα γίνουν τα πειράματα είναι το Jupyter. Το εν λόγω περιβάλλον είναι αρκετά διαδεδομένο για χρήση με ML. Βασικό χαρακτηριστικό του είναι ότι επιτρέπει στον χρήστη να εκτελέσει κώδικα Python τμηματικά, σε αποκαλούμενα κελιά (cells). Έτσι είναι ευκολότερη η δοκιμή των μοντέλων, ενώ παράλληλα είναι εφικτό να σημειώνονται γραπτά σχόλια ανάμεσα στα κελιά. Η προετοιμασία των δεδομένων λοιπόν θα πρέπει να είναι τέτοια ώστε να γίνεται εισαγωγή των δεδομένων στο περιβάλλον Jupyter.

3.1 Προετοιμασία και επεξεργασία δεδομένων

Η απόδοση ενός μοντέλου DL, είναι άρρηκτα συνδεδεμένη με τα δεδομένα που θα χρησιμοποιηθούν για την εκπαίδευσή του. Η ποσότητα αλλά και η ποιότητα των δεδομένων είναι δύο σημαντικότατοι παράγοντες που συντελούν στην υλοποίηση ενός μοντέλου που θα φέρνει εις πέρας την οποιαδήποτε εργασία. Για της ανάγκες των πειραμάτων, η όλη διαδικασία της επεξεργασίας των δεδομένων γίνεται σε ένα python script ονόματι “preprocess.py”.

3.1.1 Το dataset GTZAN

Για την εκπαίδευση των μοντέλων που θα χρησιμοποιηθούν στα πειράματα, θα αξιοποιηθεί το σύνολο δεδομένων «GTZAN». Το συγκεκριμένο σύνολο διατίθεται δωρεάν στο διαδίκτυο, ενώ κάποια frameworks γλωσσών προγραμματισμού προσφέρουν έτοιμα datasets με αυτό. Στα δικά μας πειράματα, δε θα χρησιμοποιηθούν έτοιμα datasets, καθώς θα τα κατασκευάσουμε εμείς για καλύτερο έλεγχο. Στο GTZAN λοιπόν, έχουμε 10 φακέλους καθένας εκ των οποίων αντιστοιχεί σε ένα είδος μουσικής. Τα είδη που περιλαμβάνονται είναι τα “blues”, “classical”, “country”, “disco”, “hiphop”, “jazz”, “metal”, “pop”, “reggae”, “rock”. Κάνε ένα από αυτά τα είδη αντιστοιχεί στις 10 κλάσεις στις οποίες κάθε μοντέλο καλείται να ταξινομήσει το ηχητικό αρχείο που του δίνεται. Το GTZAN, για κάθε κλάση έχει 100 αρχεία ήχου διάρκειας 30 δευτερολέπτων. Τα αρχεία αυτά είναι προσεκτικά επιλεγμένα χαρακτηριστικά παραδείγματα του μουσικού είδους στο οποίο αντιστοιχούν.

3.1.2 Διαχωρισμός ηχητικών αρχείων σε τμήματα

Το GTZAN dataset έχει συνολικά 1000 αρχεία ήχου. Ο αριθμός αυτός προφανώς προκύπτει από τα 100 αρχεία που έχει καθεμιά από τις 10 κλάσεις του. Ένα dataset που αποτελείται από 1000 δείγματα είναι ένα σχετικά μικρό dataset, από το οποίο δεν είναι πολύ πιθανό να προκύψουν καλά εκπαιδευμένα μοντέλα. Για το λόγο αυτό, συχνά συναντάται στην έρευνα η τμηματοποίηση των αρχικών δειγμάτων, σε μικρότερης διάρκειας ηχητικά αρχεία. Με τον τρόπο αυτό, αυξάνεται το μέγεθος του συνόλου δεδομένων.

Στο αρχείο preprocess.py λοιπόν, μεταξύ άλλων γίνεται η τμηματοποίηση του κάθε δείγματος. Μέσω συγκεκριμένης υπερπαραμέτρου, μπορούμε να ορίσουμε σε πόσα τμήματα επιθυμούμε να χωρίσουμε το κάθε ηχητικό αρχείο. Για παράδειγμα, χωρίζοντας το κάθε αρχείο ήχου σε 10 τμήματα, το μέγεθος του dataset αυτομάτως από 1000 δείγματα, αυξάνεται στα 10.000. Ο αριθμός αυτός είναι ένας σχετικά συχνός αριθμός τμηματοποίησης στη χρήση του GTZAN και συναντάται αρκετά στην έρευνα.

Η τμηματοποίηση των αρχείων ήχου, σαφώς συνεπάγεται μείωση της διάρκειας του κάθε δείγματος. Εάν διαχωρίσουμε το κάθε δείγμα σε 10 μέρη, τότε το κάθε ένα από αυτά θα έχει διάρκεια 3 δευτερόλεπτα. Αυτό ισχύει σαφώς, δεδομένου ότι το χωρίζουμε ισομερώς και η αρχική διάρκεια είναι 30 δευτερόλεπτα. Το ερώτημα που προκύπτει σε αυτό το σημείο, είναι το ποια είναι η ελάχιστη διάρκεια που χρειάζεται να έχει ένα αρχείο ήχου, ώστε να περιέχει την απαραίτητη πληροφορία για την ταυτοποίηση του μουσικού είδους

που ανήκει. Συνεπώς, εύκολα συμπεραίνει κανείς ότι η επιλογή του αριθμού τμημάτων είναι αρκετά σημαντική και αποτελεί από μόνη της υπερπαράμετρο που δύναται να επηρεάσει την απόδοση των μοντέλων.

3.1.3 Εξαγωγή χαρακτηριστικών

Το κάθε τμήμα που προκύπτει από τα αρχικά αρχεία ήχου, χρησιμοποιείται για εξαγωγή χαρακτηριστικών. Τα ακουστικά χαρακτηριστικά που θα χρησιμοποιηθούν για τα πειράματα είναι τα Mel Spectrograms και τα MFCCs. Η βιβλιοθήκη PyTorch χρησιμοποιεί συγκεκριμένες μεθόδους για την εξαγωγή αυτών των χαρακτηριστικών. Επιπλέον όμως χρειάζεται η μετατροπή του μεγέθους της έντασης σε κλίμακα decibel, κάτι που επίσης γίνεται με μέθοδο της βιβλιοθήκης PyTorch.

Στις μεθόδους που χρησιμοποιούνται για την εξαγωγή των χαρακτηριστικών, μπορούμε να ρυθμίσουμε τις υπερπαραμέτρους που θα χρησιμοποιηθούν για τη διαδικασία υπολογισμού τους. Όσον αφορά την κλίμακα mel στα ακουστικά χαρακτηριστικά, θα χρησιμοποιήσουμε 128 mel banks. Η τιμή αυτή εμφανίζεται σε αρκετά παραδείγματα στη βιβλιογραφία, ενώ είναι και η προεπιλεγμένη τιμή για βιβλιοθήκες της Python. Χρησιμοποιούμε την ίδια τιμή για τα MFCCs αλλά και για τα Mel Spectrograms. Επιπλέον, για τα MFCCs χρησιμοποιούμε τα 13 πρώτα cepstral coefficients, καθώς όπως έχει αναφερθεί είναι αρκετά για να περιλάβουν την πληροφορία της χροιάς.

Στη συνέχεια, η PyTorch μας δίνει τη δυνατότητα να επιλέξουμε ρυθμίσεις για την FFT. Για αυτές τις ρυθμίσεις θα δοκιμάσουμε διαφορετικά σετ τιμών τα οποία θα αναλυθούν παρακάτω. Σε κάθε περίπτωση όμως, τα MFCCs και τα Mel Spectrograms εξάγονται ως τανυστές με τις εξής διαστάσεις: (batch_size, mfccs, time_frames) και (batch_size, mels, time_frames) αντίστοιχα. Όπου batch_size είναι το σύνολο των δειγμάτων του τανυστή, mfccs είναι τα cepstral coefficients, mels είναι οι διαφορετικές mel_bands και time_frames είναι τα διαφορετικά frames του χρόνου όπως αυτά προκύπτουν από την FFT. Δεδομένων όσων αναφέρθηκαν, σε όλα τα πειράματα τα mfccs θα είναι 13, ενώ τα mels θα είναι 128. Τέλος, αναφέρουμε πως αυτή είναι και η διάταξη των διαστάσεων (shape) των τανυστών, οι οποίοι αποτελούν τα δεδομένα εισόδου για τα μοντέλα διαφορετικών αρχιτεκτονικών. Κάποιες αρχιτεκτονικές δέχονται διαφορετικές διαστάσεις τανυστών. Σε αυτή την περίπτωση, θα τροποποιήσουμε τις διαστάσεις χρησιμοποιώντας μεθόδους της PyTorch, ενώ επίσης αυτό θα αναφέρεται.

Ειδικά για την περίπτωση των δεδομένων που θα χρησιμοποιηθούν με το μοντέλο AST, η διαδικασία είναι λίγο διαφορετική. Σε αυτή την περίπτωση, θα χρησιμοποιηθεί ένα pre-trained μοντέλο από την πλατφόρμα του Hugging Face. Το μοντέλο αυτό τροφοδοτείται με δεδομένα που εξάγονται με συγκεκριμένο τρόπο και όχι μέσω των μεθόδων της PyTorch. Στην περίπτωση αυτή λοιπόν, χρησιμοποιούμε την κλάση `ASTFeatureExtractor` για να εξάγουμε τα ακουστικά χαρακτηριστικά. Η κλάση αυτή ανήκει στη βιβλιοθήκη Transformers, η οποία ουσιαστικά περιλαμβάνει μοντέλα της πλατφόρμας Hugging Face. Τα δεδομένα αυτά, είναι σαφώς ρυθμισμένα έτσι ώστε να μπορούν να χρησιμοποιηθούν με μοντέλα που έχουν υλοποιηθεί μέσω της PyTorch.

3.1.4 Οργάνωση και αποθήκευση δεδομένων εισόδου

Η πειραματική διαδικασία εκτελείται στο περιβάλλον Jupyter. Είναι λοιπόν απαραίτητη η κατάλληλη επεξεργασία των δεδομένων ούτως ώστε να γίνεται εισαγωγή τους στο περιβάλλον εργασίας. Για το λόγο αυτό, τα ακουστικά χαρακτηριστικά που εξάγονται από τα αρχεία, αποθηκεύονται με κατάλληλο τρόπο σε συγκεκριμένες μορφές αρχείων.

Τα δεδομένα πριν την αποθήκευσή τους, οργανώνονται με κατάλληλο τρόπο, έτσι ώστε να είναι εύκολη η μετέπειτα διαχείρισή τους. Για την οργάνωση των δεδομένων, χρησιμοποιούμε τα λεξικά της Python. Πιο συγκεκριμένα, χρησιμοποιούμε ένα λεξικό με 3 καταχωρήσεις. Κάθε καταχώρηση έχει ως κλειδί μια συμβολοσειρά και ως τιμή μια λίστα. Τα κλειδιά είναι:

- “mapping”: Αντιστοιχεί στις κλάσεις όπως αυτές προκύπτουν από το GTZAN.
- “melspec” ή “mfcc”: Αντιστοιχεί στους τανυστές που αποτελούν την αριθμητική αναπαράσταση του Mel Spectrogram ή των MFCCs αντίστοιχα.
- “labels”: Περιέχει αριθμούς που αντιστοιχούν στην κλάση του κάθε τανυστή.

Στο preprocces.py εμπεριέχεται μεταξύ άλλων, η όλη διαδικασία αποθήκευσης των δεδομένων. Για την αποθήκευση, αρχικά χρησιμοποιούμε τη βιβλιοθήκη os της Python. Η εν λόγω βιβλιοθήκη μας επιτρέπει να αλληλοεπιδρούμε με το λειτουργικό σύστημα. Εμείς τη χρησιμοποιούμε για να «διασχίσουμε» αναδρομικά τους φακέλους που περιέχουν το GTZAN dataset. Καθώς γίνεται αυτό, χρησιμοποιήσουμε τα ονόματα φακέλων για να συμπληρώσουμε τη λίστα που αντιστοιχεί στο κλειδί “mapping”. Κατόπιν, στη λίστα του δεύτερου ζεύγους αποθηκεύονται όλοι οι τανυστές που προκύπτουν από την εξαγωγή του Mel Spectrogram ή των MFCCs. Παράλληλα με αυτή τη διαδικασία, στη λίστα του τρίτου

ζεύγους, αποθηκεύεται ένας αριθμός που αντιστοιχεί στην κλάση του κάθε τανυστή. Οι αριθμοί της λίστας "labels", αντιστοιχούν στους δείκτες της λίστας "mapping". Αν για παράδειγμα στη θέση "5" της λίστας mapping έχουμε την κλάση "jazz", τότε εάν κάποιος τανυστής έχει εξαχθεί από jazz τραγούδι, του αντιστοιχεί το label "5".

Το λεξικό αυτό, εν τέλει αποθηκεύεται σε μορφή αρχείου JSON. Η εν λόγω μορφή αποθήκευσης έχει αρκετά πλεονεκτήματα και γι' αυτό την προτιμούμε. Αρχικά, είναι μια μορφή που μπορεί να διαβαστεί από άνθρωπο και αυτός . Μετά την αποθήκευση, είναι εφικτό κανείς να ανοίξει το αρχείο JSON και να έχει μια γενική εποπτεία του τι αποθήκευσε. Επιπλέον, είναι μια μορφή που αναγνωρίζεται από πληθώρα λογισμικών. Δεν περιοριζόμαστε μόνο στην Python και δυνητικά μπορούμε να επεξεργαστούμε τα αρχεία αυτά και με άλλα εργαλεία ή γλώσσες. Βασικό μειονέκτημα αυτής της μορφής όμως, αποτελεί το μεγάλο μέγεθος των αρχείων. Επιπλέον, όταν μετέπειτα φορτώνουμε τα αρχεία αυτά στη μνήμη για επεξεργασία, τα αρχεία φορτώνονται ολόκληρα. Υπάρχουν περιπτώσεις στις οποίες η μνήμη δεν αρκεί για τη φόρτωση των αρχείων. Σε αυτή την περίπτωση, αποθηκεύουμε τα δεδομένα σε μορφή .pt. Αυτά τα αρχεία είναι αρχεία που αναγνωρίζονται από την Python και είναι αρκετά μικρότερα. Σε περιπτώσεις που κρίθηκε αναγκαίο και κυρίως στην αποθήκευση δεδομένων που θα χρησιμοποιηθούν με το μοντέλο AST, χρησιμοποιήθηκαν αρχεία pt.

3.2 Αρχιτεκτονική δομή των μοντέλων

Στο συγκεκριμένο κεφάλαιο θα αναλύσουμε την αρχιτεκτονική του κάθε μοντέλου που χρησιμοποιούμε στα πειράματα. Δεδομένου ότι θέλουμε να διακρίνουμε τη συμπεριφορά της κάθε αρχιτεκτονικής πάνω στην ταξινόμηση των μουσικών ειδών, θα υλοποιήσουμε μια σχετικά απλή μορφή της κάθε αρχιτεκτονικής. Η υλοποίηση της κάθε αρχιτεκτονικής είναι τέτοια ώστε να προβλέπεται από τη βιβλιογραφία.

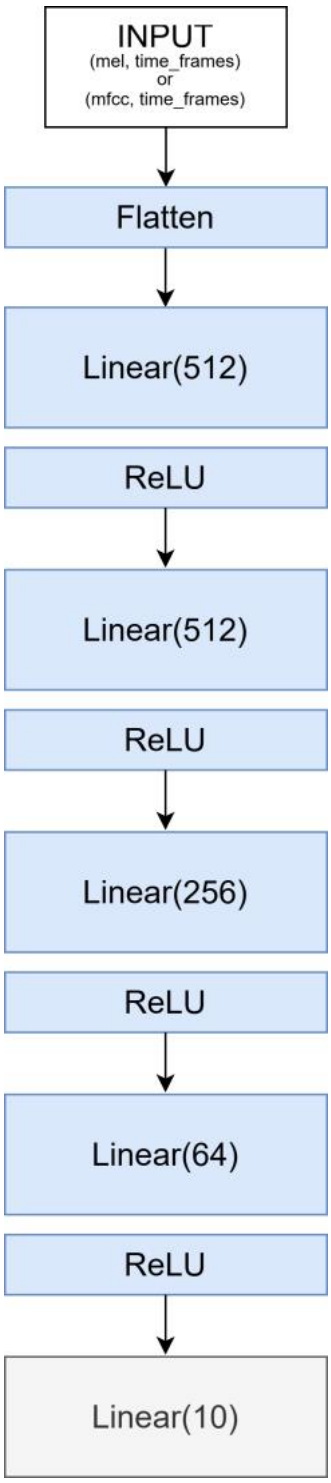
Η κάθε αρχιτεκτονική υλοποιείται σε κελιά του εκάστοτε Jupyter notebook. Παρουσιάζεται ως κλάση και το εκάστοτε μοντέλο μπορεί να δημιουργηθεί ως αντικείμενο αυτής της κλάσης. Η PyTorch προσφέρει έτοιμες κλάσεις για τα επίπεδα των περισσότερων γνωστών αρχιτεκτονικών και όλων όσων θα χρησιμοποιήσουμε. Παράλληλα, προτείνει συγκεκριμένη διαδικασία για την δημιουργία των κλάσεων, την οποία και ακολουθήσαμε.

3.2.1 Αρχιτεκτονική του μοντέλου MLP

Για την αρχιτεκτονική MLP θα χρησιμοποιήσουμε την κλάση `torch.nn.Linear` της PyTorch η οποία μας δίνει ένα πλήρως συνδεδεμένο επίπεδο νευρωνικού δικτύου. Κάθε επίπεδο θα ακολουθηθεί από μια συνάρτηση ενεργοποίησης. Για το σκοπό αυτό επιλέξαμε τη ReLU η οποία επίσης προσφέρεται ως κλάση στην PyTorch.

Όπως αναλύθηκε σε προηγούμενο κεφάλαιο, τα δεδομένα εξάγονται ως τανυστές δύο διαστάσεων, της μορφής (mfcc, time_frame) ή (mels, time_frame). Ένα επίπεδο νευρωνικού δικτύου όμως δεν μπορεί να δεχθεί τανυστές 2 διαστάσεων, επομένως μετατρέπουμε τις δυο διαστάσεις σε μια. Για το σκοπό αυτό, χρησιμοποιούμε τη μέθοδο `flatten` της PyTorch, που έχει σχεδιαστεί για αυτή τη λειτουργία. Το επίπεδο εισόδου λοιπόν θα έχει αριθμό τιμών εισόδου ίσο με το γινόμενο των 2 διαστάσεων του αρχικού τανυστή. Αν για παράδειγμα τροφοδοτούμε το δίκτυο με Mel Spectrograms διαστάσεων (128, 130), τότε θα έχουμε $128 \cdot 130 = 16640$ τιμές εισόδου.

Σχεδιάζουμε την αρχιτεκτονική του MLP με τέτοιον τρόπο ώστε περιοδικά να έχουμε λιγότερο αριθμό νευρώνων σε κάθε επίπεδο. Αρχικά ξεκινάμε με 512, κατόπιν 256 και εν τέλει 64. Το τελικό επίπεδο εξόδου, θα πρέπει να έχει αριθμό νευρώνων ίσο με τον αριθμό κλάσεων, δηλαδή 10.



Σχήμα 1 Η αρχιτεκτονική του δικτύου MLP.

Στο Σχήμα 1 διακρίνουμε την αρχιτεκτονική των μοντέλων MLP όπως αυτή περιεγράφηκε. Τέλος αναφέρουμε ότι τα μοντέλα που δημιουργούνται από την κλάση αυτή, έχουν δύο ορίσματα. Αρχικά μια λίστα με τις διαστάσεις του τανυστή που δέχονται ως δεδομένα εισόδου και δεύτερον έναν ακέραιο αριθμό που αναπαριστά τον αριθμό κλάσεων για την

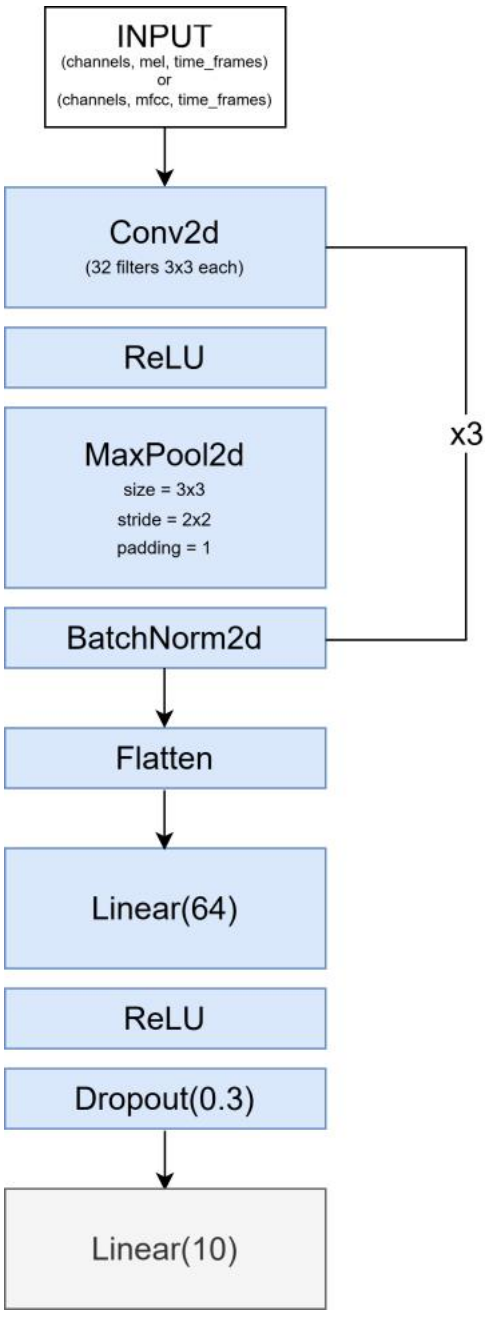
ταξινόμηση. Το δεύτερο όρισμα, έχει ως προεπιλεγμένη τιμή το 10, καθώς αυτός είναι ο αριθμός που χρησιμοποιούμε στα πειράματα.

3.2.2 Αρχιτεκτονική του μοντέλου CNN

Η αρχιτεκτονική του μοντέλου CNN αποτελείται από δύο βασικά μέρη. Στο πρώτο έχουμε διαδοχικά συνελικτικά και συγκεντρωτικά επίπεδα. Εν συνεχεία, τα δεδομένα που προκύπτουν από το πρώτο μέρος, περνούν σε ένα πλήρως συνδεδεμένο δίκτυο που ουσιαστικά είναι ένα MLP. Στους 10 νευρώνες του επιπέδου εξόδου αυτού του δικτύου παίρνουμε τις τιμές που αφορούν στην ταξινόμηση. Γενικότερα, η σχεδίαση του δικτύου μπορεί να γίνει με διάφορους τρόπους. Οι Khasgiwala και Tailor (2021) έχουν προτείνει μια σχετικά απλή αρχιτεκτονική για τη μελέτη της συμπεριφοράς των CNN όταν χρησιμοποιούνται για ταξινόμηση μουσικών ειδών. Η δική μας αρχιτεκτονική έχει ως αφετηρία τις επιλογές υπερπαραμέτρων που τίθενται εκεί.

Το πρώτο μέρος μιας αρχιτεκτονικής CNN, αποτελείται από διαδοχικές συνελικτικές ενότητες (convolution blocks). Εμείς χρησιμοποιούμε τρεις. Κάθε μια έχει ένα convolution layer το οποίο θα σαρώσει τα διαγράμματα παράγοντας 32 feature maps. Η σάρωση θα γίνει με kernel πλευράς 3 ενώ το stride αφήνεται στην προεπιλεγμένη τιμή 1. Το convolution layer ακολουθείται από μια συνάρτηση ενεργοποίησης ReLU, ενώ στη συνέχει ακολουθεί ένα pooling layer. Το pooling layer σαρώνει την εικόνα με «παράθυρα» πλευράς 3, κρατώντας μια τιμή που αντιπροσωπεύει τις τιμές που διαπερνά σε εκάστοτε βήμα της σάρωσης. Οι τιμές για το stride και το padding στο pooling layer είναι 2 και 1 αντίστοιχα. Στο τέλος του block εφαρμόζεται batch normalization, καθώς οι τιμές που προκύπτουν από το pooling, κανονικοποιούνται.

Μετά τα convolution blocks, όπως αναφέρθηκε, ακολουθεί ένα απλό MLP. Το δίκτυο αυτό αποτελείται από ένα επίπεδο 64 νευρώνων που ακολουθείται από μια συνάρτηση ενεργοποίησης ReLU. Εν συνεχεία εφαρμόζουμε την τεχνική dropout δίνοντας πιθανότητα 30% σε ένα νευρώνα να μηδενιστεί σε κάποιο forward pass. Εν τέλει υπάρχει το επίπεδο εξόδου που αποτελείται από 10 νευρώνες.



Σχήμα 2 Η αρχιτεκτονική των μοντέλων CNN.

Κατά τον σχεδιασμό της αρχιτεκτονικής CNN υπάρχει μια βασική λεπτομέρεια. Για να μπορέσει να λειτουργήσει ομαλά το μοντέλο, θα πρέπει οι τανυστές που προκύπτουν μετά το flatten layer, όπως βλέπουμε και στο Σχήμα 2, να έχουν κατάλληλη διάσταση ώστε να μπορεί να συνεχιστεί η ροή στο linear layer που ακολουθεί. Κατά τη δημιουργία ενός επιπέδου στην PyTorch, στα ορίσματα πρέπει να δοθούν οι διαστάσεις των τανυστών με τα οποία τροφοδοτούμε το επίπεδο. Επειδή λοιπόν δεν γνωρίζουμε τις διαστάσεις που θα προκύψουν μετά το flatten, μπορούμε να χρησιμοποιήσουμε μερικές εντολές `print` από

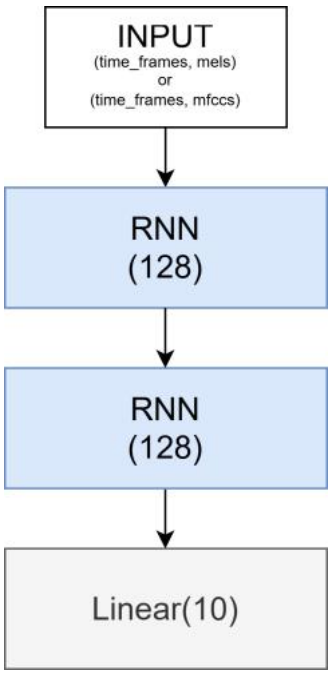
την Python. Αυτές τοποθετούνται κατάλληλα, ώστε να δούμε τις διαστάσεις των τανυστών μετά την επεξεργασία από κάθε convolution block αφού πραγματοποιηθεί ένα δοκιμαστικό forward pass. Από τις πληροφορίες που θα πάρουμε, μπορούμε να βγάλουμε συμπέρασμα για τις διαστάσεις που θα προκύψουν από τον τανυστή μετά το flatten layer.

Επιπροσθέτως, υπάρχει ακόμα ένα σημείο που πρέπει να δοθεί σημασία. Στο Σχήμα 2, βλέπουμε την αρχιτεκτονική του CNN και παρατηρούμε ότι ο τανυστής που δέχεται στην είσοδο, έχει διαστάσεις (channels, mels, time_frames). Τα μοντέλα CNN είναι σχεδιασμένα για να δέχονται εικόνες. Σε αυτή την περίπτωση, οι εικόνες αναπαρίστανται ως τανυστές με διαστάσεις (color_channels, height, width). Τα κανάλια (channels) αυτά, αφορούν σε καθένα από τα 3 βασικά χρώματα: κόκκινο, πράσινο, μπλε (RGB) και την ένταση του κάθε pixel σε αυτό το κανάλι. Τα MFCCs και τα Mel Spectrograms δεν έχουν τέτοια κανάλια. Αν κάναμε μια αντιπαραβολή, θα λέγαμε ότι αντιστοιχούν σε μονόχρωμες εικόνες, επομένως έχουν 1 κανάλι. Όταν λοιπόν προετοιμάσουμε τα δεδομένα για να τροφοδοτήσουμε ένα CNN μοντέλο, θα πρέπει να τροποποιήσουμε τις διαστάσεις, έτσι ώστε να είναι (color_channels, mels, time_frames) ή εναλλακτικά mfccs στη θέση των mels. Το "color_channels" θα τεθεί ίσο με 1. Η PyTorch μας δίνει τη δυνατότητα να κάνουμε όλη αυτή τη διαδικασία με τη μέθοδο `unsqueeze`.

3.2.3 Αρχιτεκτονική του μοντέλου RNN

Για την αρχιτεκτονική RNN θα δημιουργήσουμε 2 κλάσεις. Οι δύο κλάσεις θα έχουν τα ίδια επίπεδα, όμως η διαφορά τους θα είναι ότι στην μια περίπτωση θα χρησιμοποιήσουμε όλα τα hidden states που προκύπτουν από το δίκτυο, ενώ στην άλλη περίπτωση θα χρησιμοποιήσουμε μόνο το τελευταίο hidden state. Ο λόγος που το κάνουμε αυτό, είναι για να μελετήσουμε τη συμπεριφορά αυτών των 2 περιπτώσεων και εάν η περίπτωση που χρησιμοποιούμε πληροφορίες από όλα τα hidden states, θα βοηθήσει.

Η PyTorch όπως και στις περισσότερες αρχιτεκτονικές μας δίνει έτοιμα RNN layers. Μάλιστα έχουμε τη δυνατότητα, μέσω του ορίσματος "num_layers", να έχουμε συνεχόμενα RNN layers. Αυτό που είναι απαραίτητο κατά τη σχεδίαση, είναι να ορίσουμε ένα αρχικό hidden state το οποίο ουσιαστικά θα είναι ένας τανυστής με μηδενικά. Η διαδικασία αυτή ορίζεται από τις οδηγίες της PyTorch. Συνολικά λοιπόν, θα έχουμε 2 συνεχόμενα RNN layers, το αποτέλεσμα των οποίων θα τροφοδοτήσει ένα πλήρως συνδεδεμένο γραμμικό επίπεδο όπως αυτό ορίζεται στην κλάση `torch.nn.Linear` της PyTorch. Αυτό θα αποτελέσει και το επίπεδο εξόδου.



Σχήμα 3 Η αρχιτεκτονική των μοντέλων RNN

Το RNN είναι ένα αναδρομικό δίκτυο το οποίο δέχεται την πληροφορία σε "time steps". Συνεπώς, οι τανυστές που αποτελούν τα δεδομένα εισόδου του, θα πρέπει να του δίνουν την πληροφορία ανά «βήμα στο χρόνο». Όπως βλέπουμε λοιπόν και στο Σχήμα 3 που δείχνει την αρχιτεκτονική του RNN που χρησιμοποιούμε, τα δεδομένα εισόδου θα πρέπει να τροποποιηθούν ώστε να δίνουμε την πληροφορία ανά time_frame. Σε κάθε βήμα λοιπόν, το δίκτυο θα δέχεται την πληροφορία για τα mfccs ή τα mels του εκάστοτε time frame.

3.2.4 Αρχιτεκτονική των μοντέλων LSTM και GRU

Οι αρχιτεκτονικές LSTM και GRU είναι ουσιαστικά βελτιώσεις της αρχιτεκτονικής RNN η οποία συνήθως αντιμετωπίζει το πρόβλημα των vanishing gradients. Οι αρχιτεκτονική που σχεδιάσαμε για αυτά τα μοντέλα, είναι ουσιαστικά ίδια με αυτή του Σχήματος 3. Η μόνη διαφορά είναι πως αντί για τα RNN επίπεδα, έχουμε LSTM ή GRU επίπεδα. Η PyTorch προσφέρει έτοιμες κλάσεις και για αυτά.

Στην περίπτωση του LSTM, έχουμε μια ακόμη παράμετρο που πρέπει να ρυθμιστεί. Η αρχιτεκτονική αυτή, εκτός από τα hidden states, έχει ένα cell state, το οποίο λειτουργεί ως μνήμη. Κατά τον σχεδιασμό της κλάσης λοιπόν, εκτός από το αρχικό hidden state, πρέπει να φροντίσουμε να δημιουργήσουμε και ένα αρχικό cell state. Ουσιαστικά και σε αυτή την περίπτωση, όπως και στο αρχικό hidden state, έχουμε έναν τανυστή που αποτελείται από μηδενικά. Στη σχεδίαση του GRU, χρειάζεται απλά να προβλέψουμε ένα αρχικό hidden

state, καθώς cell state δεν υφίσταται. Τέλος, στην περίπτωση των μοντέλων LSTM και GRU, θα χρησιμοποιήσουμε πληροφορία μόνο από το τελευταίο hidden state.

3.2.5 Αρχιτεκτονική του μοντέλου Vision Transformer

Η αρχιτεκτονική του μοντέλου Transformer, είναι στην περίπτωσή μας, ουσιαστικά η αρχιτεκτονική ενός μοντέλου ViT. Αυτό είναι κάτι που προβλέπεται από τη βιβλιογραφία, συνεπώς στα πειράματά μας θα ακολουθήσουμε αυτή την οδό. Οι Dosovitskiy et al. (2020) στην παρουσίαση του ViT, περιλαμβάνουν λεπτομερές σχέδιο της αρχιτεκτονικής του. Παρατηρούμε ότι και στην αντίστοιχη βιβλιογραφία όπου το ViT χρησιμοποιείται για επεξεργασία ήχου, χρησιμοποιείται το ίδιο σχήμα και συνεπώς ίδια αρχιτεκτονική. Η όλη διαδικασία της επεξεργασίας των MFCCs ή των Mel Spectrograms παραπέμπει στη διαδικασία επεξεργασίας μιας εικόνας από το ViT και αποτελείται από τρία βασικά μέρη:

- Τον διαχωρισμό των απεικονίσεων σε patches και τη δημιουργία των embeddings.
- Την επεξεργασία των embeddings από τον Encoder του Transformer.
- Την επεξεργασία των αποτελεσμάτων από έναν ταξινομητή (classifier) που ουσιαστικά είναι ένα δίκτυο MLP.

Αρχικά λοιπόν, δημιουργούμε μια κλάση ονόματι `PatchEmbedding`. Η κλάση αυτή χρησιμοποιεί convolution layers για να δημιουργήσει patch embeddings, τα οποία αντιστοιχούν στον διαχωρισμό της απεικόνισης σε κομμάτια. Οι Khasgiwala και Tailor (2021) στη δημιουργία ενός ViT για χρήση με MFCCs, περιγράφουν πως διαχωρίζουν την απεικόνιση σε κομμάτια μεγέθους 12x12. Για να μπορέσει να διαχωριστεί η απεικόνιση, θα πρέπει να τροποποιηθεί το μέγεθός της. Η PyTorch έχει μέθοδο με την οποία μπορούμε να αναδιατάξουμε το μέγεθος των αρχικών μας τανυστών σε διαστάσεις 72x72. Με αυτόν τον τρόπο, παίρνουμε 36 τμήματα, τα οποία στη συνέχεια θα τοποθετηθούν διαδοχικά σε σειρά μέσω flattening ακολουθώντας την πρωτότυπη αρχιτεκτονική του ViT.

Η κλάση `PatchEmbedding` χρησιμοποιείται στην κλάση `ViT` που επίσης κατασκευάζουμε για τη δημιουργία των μοντέλων αυτής της αρχιτεκτονικής. Εκεί ακολουθούνται όλες οι προδιαγραφές του ViT. Η PyTorch μας δίνει έτοιμες κλάσεις για τον encoder του Transformer τις οποίες και χρησιμοποιούμε. Θεωρητικά, μπορεί κανείς να κατασκευάσει τον Encoder χρησιμοποιώντας άλλες κλάσεις της PyTorch, όμως με τη δική μας προσέγγιση, προσπαθούμε να έχουμε την καλύτερη δυνατή απόδοση και να αποφύγουμε πιθανά προβλήματα πολυπλοκότητας δημιουργώντας τον encoder από την αρχή. Τέλος,

χρησιμοποιώντας κλάσεις της PyTorch, μπορεί κανείς να προσθέσει και τον classifier στο τέλος του δικτύου.

4. Υλοποίηση μοντέλων και διεξαγωγή Πειραμάτων

Στο κεφάλαιο αυτό, θα περιγράψουμε την οργανωμένη, διαδοχική ροή βημάτων που ακολουθούμε για να χρησιμοποιήσουμε τα δεδομένα μας, έτσι ώστε να κατασκευάσουμε μοντέλα που θα προβλέπουν το μουσικό είδος ενός αρχείου. Η ροή αυτή που συχνά αναφέρεται και ως “pipeline”, είναι η ίδια σε όλα τα πειράματα. Με αυτόν τον τρόπο εξασφαλίζεται η συνοχή ενώ παράλληλα είναι ακριβέστερη η σύγκριση των μοντέλων και κατ’ επέκτασιν, των αρχιτεκτονικών. Επιπροσθέτως, θα αναλυθεί η στρατηγική των πειραμάτων, καθώς και τα ερευνητικά ερωτήματα που επιδιώκουμε να απαντήσουμε μέσω αυτών.

4.1 Διασφάλιση επαναληψιμότητας και χρήση GPU

Ένα από τα βασικά προτερήματα της PyTorch, είναι το γεγονός ότι μας δίνει τη δυνατότητα να εκτελέσουμε υπολογισμούς στην GPU. Η αρχιτεκτονική του επεξεργαστή της κάρτας γραφικών είναι τέτοια, ώστε να επιτρέπει την ταχύτερη εκτέλεση πράξεων συγκριτικά με τη CPU. Δεδομένου λοιπόν ότι η εκπαίδευση των μοντέλων DL σχετίζεται άμεσα με τις πράξεις τανυστών και γενικότερα τη γραμμική άλγεβρα, η χρήση μιας καλής κάρτας γραφικών μπορεί να επιταχύνει αρκετά τη διαδικασία.

Η PyTorch δίνει τη δυνατότητα χρήσης GPU που έχουν υποστήριξη CUDA. Επιπροσθέτως, αξίζει να αναφερθεί ότι μια καλύτερη κάρτα γραφικών, αποτελεί από μόνη της παράγοντα που θα επιταχύνει την όλη διαδικασία κατά πολύ. Ενδεικτικά, αναφέρουμε ότι η εκπαίδευση ενός μοντέλου AST για 15 epochs, χρησιμοποιώντας μια κάρτα γραφικών NVIDIA GeForce MX150 διήρκεσε σχεδόν 22 ώρες. Η ίδια διαδικασία, χρησιμοποιώντας κάρτα γραφικών NVIDIA GeForce RTX 4060, διήρκεσε περίπου 24 λεπτά. Η πρώτη GPU είναι κατασκευασμένη το 2017 και συναντάται κυρίως σε laptops, ενώ η δεύτερη είναι κατασκευασμένη το 2023. Η διαφορά στη διάρκεια είναι εντυπωσιακή και δείχνει το πόσο σημαντική μπορεί να είναι η χρήση μιας καλής GPU. Στα πειράματα που εκτελέσαμε, χρησιμοποιήσαμε την NVIDIA GeForce RTX 4060.

Μια ακόμα σημαντική παράμετρος κατά την εκτέλεση των πειραμάτων, είναι η διασφάλιση της επαναληψιμότητας. Η χρήση GPU δεν εξασφαλίζει τον ντετερμινισμό στα πειράματα. Αυτό σημαίνει ότι ο τρόπος που η PyTorch χρησιμοποιεί τους πόρους του συστήματος, δεν μας εξασφαλίζει τα ίδια αποτελέσματα κάθε φορά που θα εκτελέσουμε το ίδιο πείραμα. Είναι δυνατόν να υπάρχουν μερικές αποκλίσεις. Οι αποκλίσεις δεν είναι τέτοιες που να

αλλοιώνουν το αποτέλεσμα, όμως για λόγους συνοχής, πολλές φορές είναι θεμιτό να υπάρχει ντετερμινιστική συμπεριφορά. Η PyTorch έχει μεθόδους που μας επιτρέπουν να έχουμε τέτοια συμπεριφορά, κάτι που συμπεριλάβαμε στο pipeline.

Αξίζει να σημειωθεί όμως πως ακόμα και προβλέποντας αυτό το φαινόμενο, υπάρχουν περιπτώσεις που τίποτα δεν μας εγγυάται την απόλυτο ντετερμινισμό. Η εκτέλεση των πειραμάτων σε ένα σύστημα με διαφορετική GPU και τουτέστιν πιθανώς διαφορετική αρχιτεκτονική, σημαίνει ότι κάποιες πράξεις μπορούν να εκτελεστούν με διαφορετική σειρά και να προκύψουν ελαφρώς διαφορετικά αποτελέσματα. Επίσης, περίπλοκα μοντέλα όπως αυτό του ViT, εξαρτώνται από πολύ περισσότερες παραμέτρους και η πιθανότητα να μη διατηρηθεί η συνέπεια κατά την εκτέλεση των πράξεων, είναι μεγαλύτερη. Παρ' όλα αυτά οι διαφορές δεν είναι μεγάλες και η απόκλιση αυτή δεν επηρεάζει το αποτέλεσμα.

4.2 Μεθοδολογική οργάνωση πειραμάτων

Σκοπός των πειραμάτων είναι η μελέτη και η αξιολόγηση αρχιτεκτονικών DL όταν αυτές χρησιμοποιούνται για την κατηγοριοποίηση μουσικών κομματιών σε μουσικά είδη. Η απόδοση των μοντέλων μπορεί να επηρεαστεί από πληθώρα παραμέτρων. Ο αριθμός των διαφορετικών συνδυασμών αυτών των παραμέτρων είναι τεράστιος.

Για την έρευνά μας, μελετήσαμε την απόκριση των μοντέλων συγκρίνοντας κάποιους βασικούς παράγοντες και υπερπαραμέτρους της εκπαίδευσης των μοντέλων. Συγκεκριμένα, έλαβαν χώρα τρεις βασικές σειρές πειραμάτων. Σε αυτό το κεφάλαιο θα αναλύσουμε τις τρεις βασικές πειραματικές συγκρίσεις που ακολουθήθηκαν, καθώς και τα κριτήρια επιλογής τους.

4.2.1 Σύγκριση αναπαραστάσεων ήχου

Στην πρώτη σύγκριση πειραμάτων, συγκρίνουμε την ικανότητα γενίκευσης των μοντέλων όταν τα τροφοδοτούμε με διαφορετικές αναπαραστάσεις ήχου. Πιο συγκεκριμένα, ως δεδομένα εισόδου δίνουμε στα μοντέλα MFCCs και Mel Spectrograms. Η διαδικασία εξαγωγής αυτών των ακουστικών χαρακτηριστικών έχει ήδη περιγραφεί.

Για την καλύτερη δυνατή σύγκριση, όλες οι σχετικές υπερπαράμετροι έχουν παραμείνει ίδιες. Για παράδειγμα, στις δύο περιπτώσεις έχουμε επιλέξει τον ίδιο αριθμό Mel banks, ο οποίος είναι ίσος με 128. Επιπροσθέτως, οι υπερπαράμετροι που θέτουμε στην FFT είναι ίδιες και αντιστοιχούν στην «περίπτωση 1» του Πίνακα 1. Τέλος, στις δύο περιπτώσεις χρησιμοποιούμε την ίδια τμηματοποίηση των αρχικών δειγμάτων του GTZAN.

Συγκεκριμένα, τμηματοποιούμε τα δείγματα σε 10 ισόχρονα μέρη, καταλήγοντας σε δείγματα διάρκειας 3ων δευτερολέπτων. Εν τέλει λοιπόν, τα μοντέλα θα εκπαιδευθούν με το ίδιο πλήθος δειγμάτων, ενώ παρατηρούμε πως μετά την FFT στις δύο περιπτώσεις προκύπτουν 130 time frames. Στην περίπτωση των MFCCs όπως αναφέρθηκε, έχουμε 13 cepstral coefficients, ενώ στην περίπτωση των Mel spectrograms έχουμε 128 mels.

Στόχος αυτής της σύγκρισης πειραμάτων είναι σαφώς η αποτίμηση της απόδοσης των μοντέλων για αυτές τις δύο βασικές αναπαραστάσεις ακουστικών χαρακτηριστικών. Μοντέλα των αρχιτεκτονικών MLP, CNN, RNN, LSTM, GRU, ViT, θα εκπαιδευτούν ξεχωριστά για κάθε μια από τις αναπαραστάσεις. Ο σκοπός, σαφώς, είναι να αξιολογηθεί η ικανότητα γενίκευσής τους.

4.2.2 Σύγκριση υπερπαραμέτρων της FFT

Στη δεύτερη σύγκριση, θα τροφοδοτήσουμε τα μοντέλα με δεδομένα που προέκυψαν δίνοντας διαφορετικές τιμές υπερπαραμέτρων για την FFT. Με αυτή τη διάκριση και τις τιμές που θα δώσουμε, στην πρώτη περίπτωση θα προκύψουν δεδομένα τα οποία προσφέρουν καλύτερη ανάλυση στη διάσταση της συχνότητας. Στη δεύτερη περίπτωση θα προκύψουν δεδομένα με καλύτερη ανάλυση στη διάσταση του χρόνου.

Περιπτώσεις	Sample rate	FFT size	Window size	Hop size
Περίπτωση 1	22050	4096	1024	512
Περίπτωση 2	16000	512	400	160

Πίνακας 1 Ρυθμίσεις υπερπαραμέτρων FFT για τη σύγκριση ανάλυσης ηχητικών δειγμάτων

Στις δύο περιπτώσεις των πειραμάτων για τη σύγκριση της επίδρασης της ανάλυσης του ήχου, χρησιμοποιούμε Mel Spectrograms. Στον Πίνακα 1, βλέπουμε τις διαφορετικές τιμές των υπερπαραμέτρων που χρησιμοποιήθηκαν. Κατά την μεταφόρτωση των τανυστών που προέκυψαν με τις ρυθμίσεις του Πίνακα 1, παρατηρούμε ότι στην Περίπτωση 1 προκύπτουν 130 time frames, ενώ στην περίπτωση 2 προκύπτουν 297 time frames. Δεδομένου ότι και στις 2 περιπτώσεις χρησιμοποιούμε ηχητικά δείγματα διάρκειας 3ων δευτερολέπτων, εύκολα φαίνεται πως η ανάλυση στο επίπεδο του χρόνου είναι καλύτερη στη 2η περίπτωση.

Αντίστοιχα, βλέπουμε από τον Πίνακα 1 ότι ο λόγος του sample rate και του window size είναι μικρότερος στην πρώτη περίπτωση. Σύμφωνα με τον Vaseghi (2007), ο λόγος αυτός είναι ενδεικτικός της ανάλυσης στο επίπεδο της συχνότητας και μας δίνει την ελάχιστη

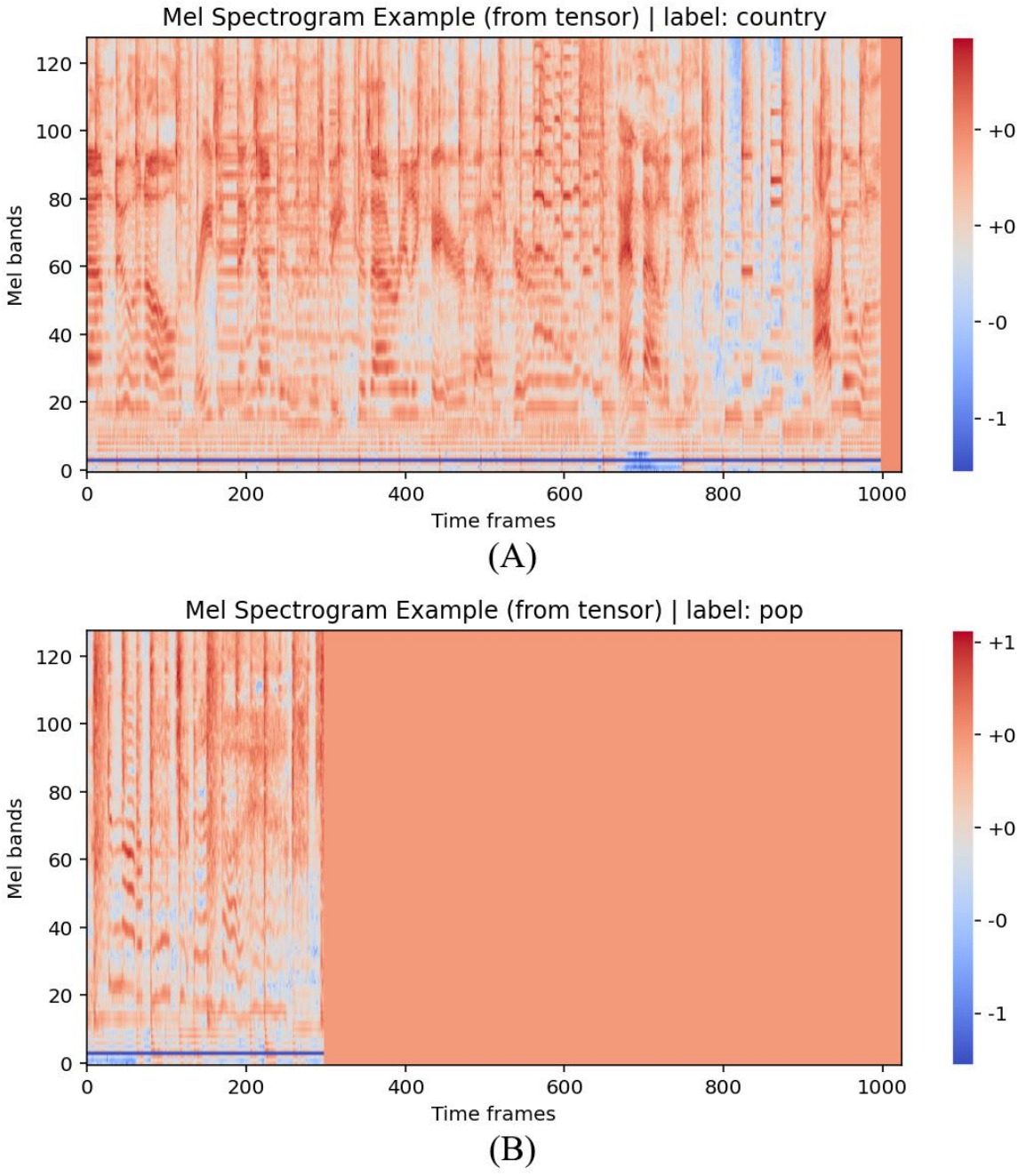
διαφορά συχνοτήτων (Δf) που μπορούν να διακριθούν σε ένα σήμα. Το FFT size θα επηρεάσει τον αριθμό των frequency bins, όμως η ικανότητα διάκρισης δυο κοντινών συχνοτήτων παραμένει ίδια.

Σκοπός μας με αυτή τη διάκριση πειραμάτων, είναι να δούμε εάν ο τρόπος ανάλυσης ενός ηχητικού σήματος είναι ικανός να επηρεάσει την απόδοση των μοντέλων. Οι τιμές που επιλέχθηκαν στον Πίνακα 1, είναι ενδεικτικές τιμές που συναντώνται στη βιβλιογραφία. Το σκεπτικό πίσω από τη σύγκρισή τους, είναι να εντοπίσουμε το κατά πόσο διαφορετικές τιμές είναι δυνατόν να επηρεάσουν την ικανότητα γενίκευσης, καθώς σε μεγάλο ποσοστό ερευνών, οι τιμές αυτές τίθενται εμπειρικά και η μελέτη γίνεται ανεξάρτητα από αυτές.

4.2.3 Σύγκριση διάρκειας ηχητικών αρχείων και ποσότητας δεδομένων

Όπως αναλύθηκε σε προηγούμενο υποκεφάλαιο, στη διαδικασία των πειραμάτων μας εκτελούμε τμηματοποίηση των ηχητικών δεδομένων. Αυτή είναι μια αρκετά συνήθης πρακτική η οποία όμως εξ ορισμού έρχεται με έναν συμβιβασμό. Όσο μεγαλύτερη είναι η τμηματοποίηση, τόσο αυξάνεται το πλήθος των δειγμάτων. Παράλληλα όμως, μειώνεται η διάρκεια του κάθε δείγματος. Αφενός, η εκπαίδευση με μεγαλύτερο πλήθος δειγμάτων αποτελεί έναν από τους σημαντικότερους παράγοντες που οδηγούν ένα μοντέλο στην ικανότητα να γενικεύει καλύτερα. Αφετέρου όμως, στην περίπτωσή της χρήσης μουσικών δειγμάτων, είναι πιθανόν μεγαλύτερης διάρκειας ηχητικά δείγματα να είναι ικανά να παρέχουν περισσότερη πληροφορία στα μοντέλα μας. Αυτό είναι κάτι φανερό ακόμα και διαισθητικά, καθώς και στην πραγματική ζωή η διάρκεια ακρόασης ενός μουσικού κομματιού μπορεί να επηρεάσει την κρίση του ακροατή όσον αφορά το συμπέρασμα για το μουσικό είδος που ακούει. Επιπροσθέτως, γνωρίζουμε πως στα αναδρομικά μοντέλα υπάρχει κάποιας μορφής μνήμη. Αυτό το γεγονός δυνητικά είναι πιθανό να τα οδηγεί να εκμεταλλευτούν χρονικές εξαρτήσεις και εξέλιξη της πληροφορίας στο χρόνο.

Η εν λόγω σύγκριση λοιπόν έχει ως σκοπό τη μελέτη απόδοσης των μοντέλων υπό διαφορετικές συνθήκες τμηματοποίησης. Παράλληλα όμως, έχει ακόμα έναν σκοπό. Το μοντέλο AST του Hugging Face δέχεται Mel Spectrograms με διαστάσεις 128 mels και 1024 time frames. Εάν το ηχητικό δείγμα είναι μικρό σε διάρκεια και τα time frames που θα προκύψουν από αυτό είναι λιγότερα, τότε τα εναπομείναντα time frames θα συμπληρωθούν με μηδενικές τιμές για τα mels. Αντίστοιχα, αν το ηχητικό δείγμα, παράγει περισσότερα time frames, κρατούνται τα πρώτα 1024.



Σχήμα 4 Mel Spectrograms που χρησιμοποιούνται με το AST όταν έχουμε segmentation σε 3 μέρη (A) και όταν έχουμε segmentation σε 10 μέρη (B)

Τα ηχητικά δείγματα των 3ων δευτερολέπτων που χρησιμοποιούμε στις υπόλοιπες σειρές πειραμάτων, παράγουν μια απεικόνιση όπως στο (B) μέρος του Σχήματος 4. Μια τέτοια απεικόνιση που στο μεγαλύτερο μέρος της έχει padding, είναι πιθανόν να μην αποτελεί δεδομένο εισόδου που θα οδηγήσει σε αξιοπρεπή απόδοση. Κατόπιν δοκιμών, βλέπουμε ότι εάν τμηματοποιήσουμε τα αρχικά δείγματα του GTZAN σε 3 μέρη, παίρνουμε μια απεικόνιση όπως αυτή του (A) μέρους του Σχήματος 4. Αυτή αποτελεί καλύπτει αρκετά

περισσότερο το διάγραμμα καθώς με αυτήν έχουμε 997 time frames. Εκπαιδεύοντας λοιπόν το μοντέλο AST στις δυο αυτές διαφορετικές συνθήκες, μπορούμε να δούμε αν θα επηρεαστεί η απόδοση του AST.

Εν κατακλείδι, στη σύγκριση αυτή, χρησιμοποιούμε τις ίδιες υπερπαραμέτρους για την FFT, έτσι ώστε να εξάγουμε Mel Spectrograms από ηχητικά αρχεία των 3ων δευτερολέπτων και των 10 δευτερολέπτων. Στην πρώτη περίπτωση τμηματοποιούμε τα αρχικά δείγματα σε 10 μέρη, ενώ στη δεύτερη τμηματοποιούμε σε 3 μέρη για να μελετήσουμε αυτές τις δύο διαφορετικές συνθήκες τμηματοποίησης.

4.2.4 Οργάνωση πειραμάτων

Έχοντας περιγράψει τις συγκρίσεις που επιθυμούμε να πραγματοποιήσουμε, θα περιγράψουμε τις σειρές πειραμάτων που πραγματοποιήθηκαν. Μέσω της μελέτης των αποτελεσμάτων από τα πειράματα αυτά, επιδιώκουμε να καταλήξουμε σε πορίσματα όσον αφορά τις τρείς συγκρίσεις που θέσαμε.

Θα πραγματοποιήσουμε τέσσερεις σειρές πειραμάτων. Σε όλες τις σειρές πειραμάτων που πραγματοποιούμε, εξετάζουμε τις αρχιτεκτονικές MLP, CNN, RNN, LSTM, GRU και ViT. Στις σειρές 3 και 4, επιπροσθέτως εξετάζουμε το μοντέλο AST.

Σειρά Πειραμάτων	Δεδομένα Εισόδου	Ρυθμίσεις FFT	Διάρκεια Δειγμάτων
1η	MFCCs	«Περίπτωση 1»	3 sec
2η	Mel Spectrograms	«Περίπτωση 1»	3 sec
3η	Mel Spectrograms	«Περίπτωση 2»	3 sec
4η	Mel Spectrograms	«Περίπτωση 2»	10 sec

Πίνακας 2 Σειρές πειραμάτων προς εκτέλεση

Στον Πίνακα 2, βλέπουμε τις λεπτομέρειες κάθε σειράς πειραμάτων. Στη στήλη «Ρυθμίσεις FFT», οι περιπτώσεις αντιστοιχούν στις τιμές του Πίνακα 1. Ουσιαστικά, η «Περίπτωση 2» αντιστοιχεί στις ρυθμίσεις FFT που υφίστανται στο AST.

Συγκρίνοντας λοιπόν τα αποτελέσματα της 1ης και της 2ης σειράς, μελετάμε τη συμπεριφορά των διαφορετικών για διαφορετικά δεδομένα εισόδου. Συγκρίνοντας τη 2η με την 3η σειρά, μελετάμε την επίδραση των διαφορετικών ρυθμίσεων FFT, όταν αυτές

προσφέρουν καλύτερη ανάλυση συχνότητας ή χρόνου. Συγκρίνοντας την 3η με την 4η σειρά, ερευνούμε την επίδραση διαφορετικού segmentation στα αρχικά δείγματα.

4.3 Προετοιμασία πειραμάτων

Στο κεφάλαιο αυτό, περιγράφουμε την επεξεργασία των αποθηκευμένων δεδομένων ούτως ώστε να τα χρησιμοποιήσουμε στα πειράματα. Παράλληλα, θα αναλύσουμε τη διαδικασία εποπτείας των δεδομένων καθώς και τον τρόπο υλοποίησης των μοντέλων χρησιμοποιώντας τις αρχιτεκτονικές που έχουν ήδη περιγραφεί.

4.3.1 Ανάκτηση δεδομένων

Σε προηγούμενο κεφάλαιο, αναφέρθηκε ο τρόπος εξαγωγής των ακουστικών χαρακτηριστικών και η αποθήκευσή τους. Η φόρτωση αυτών των δεδομένων στο περιβάλλον του Jupyter είναι ένα από τα πρώτα βήματα του pipeline. Πιο συγκεκριμένα, αρχικά κατασκευάζουμε δύο εντολές, την `load_data` και την `load_classes`. Οι εντολές αυτές μας επιτρέπουν να επεξεργαστούμε τα αρχεία JSON ή .pt και να μεταφορτώσουμε τα δεδομένα τους σε λίστες. Συγκεκριμένα, αποθηκεύουμε μια λίστα με τα ακουστικά χαρακτηριστικά σε μορφή τανυστή, μια με τα labels και μια με τα ονόματα κλάσεων. Τα labels είναι επίσης σε μορφή τανυστή για ευκολότερη επεξεργασία.

Μετά τη μεταμόρφωση, σε ορισμένες περιπτώσεις, είναι απαραίτητη η επεξεργασία των τανυστών για χρήση με συγκεκριμένες αρχιτεκτονικές. Ορισμένες από αυτές τις οδηγίες, αναφέρθηκαν και στο κεφάλαιο της περιγραφής των αρχιτεκτονικών. Τα ακουστικά χαρακτηριστικά, αποθηκεύονται με την εξής μορφή: (batch_size, time_frames, mels) εάν έχουμε Mel Spectrograms και (batch_size, time_frames, mfccs) εάν έχουμε MFCCs. Το batch_size είναι το πλήθος των δειγμάτων. Κατόπιν, για κάθε mel band ή cepstral coefficient (mfcc) έχουμε την ένταση για κάθε time frame.

Στην περίπτωση του MLP, δεν είναι αναγκαία κάποια επεξεργασία, καθώς οι 2 τελευταίες διαστάσεις του τανυστή θα υποστούν flattening. Για τυπικούς λόγους, αυτό που κάνουμε είναι να αναστρέψουμε την διάσταση με τα mfccs/mels με τη διάσταση των time frames. Τυπικά, σε μια απεικόνιση, αυτή θα ήταν η δοθείσα μορφή, επομένως κάνουμε αυτή την τροποποίηση χρησιμοποιώντας την μέθοδο `permute` της PyTorch, ώστε να έχουμε τη μορφή (batch_size, mfccs, time_frames) και (batch_size, mels, time_frames).

Στην περίπτωση του CNN, όπως αναφέρθηκε, το μοντέλο αρχικά περιμένει να δεχθεί τον αριθμό των χρωματικών καναλιών, καθώς η αρχιτεκτονική αυτή ενδείκνυται για χρήση με

εικόνες. Χρησιμοποιώντας τις μεθόδους `permute` και `unsqueeze` της PyTorch, είναι εφικτό να φέρουμε τους τανυστές στη μορφή (batch_size, 1, mfccs, time_frames) και (batch_size, 1, mels, time_frames) για κάθε ακουστικό χαρακτηριστικό αντίστοιχα. Το «1» αντιστοιχεί στο 1 χρωματικό κανάλι που θα είχε η απεικόνιση κατ' αντιστοιχία με μια εικόνα.

Για την περίπτωση των μοντέλων RNN, LSTM και GRU ακολουθείται η ίδια επεξεργασία των δεδομένων. Όπως αναφέρθηκε, τα μοντέλα αυτά δέχονται την πληροφορία σε time steps. Επομένως, περιμένουν να δεχθούν την πληροφορία ανά time frame, που είναι ουσιαστικά το «βήμα στο χρόνο» στην περίπτωση των δικών μας δεδομένων. Δεδομένου λοιπόν ότι οι τανυστές αποθηκεύονται με αυτή τη μορφή εξ αρχής, δε χρειάζεται κάποια περαιτέρω επεξεργασία.

Για την περίπτωση του μοντέλου ViT, χρειάζονται ουσιαστικά δύο βήματα επεξεργασίας. Αρχικά, δεδομένου ότι οι απεικονίσεις θα υποστούν τμηματοποίηση μέσω convolutional layers, θα χρειαστεί να προστεθεί μια διάσταση στους τανυστές, όπως ακριβώς και στην περίπτωση του CNN. Κατόπιν, θα πρέπει να τροποποιήσουμε τις απεικονίσεις σε ένα κατάλληλο μέγεθος. Σύμφωνα με τους Khasgiwala και Tailor (2021) και το πρότυπό τους για τις ρυθμίσεις του ViT, τροποποιούμε το μέγεθος των απεικονίσεων σε 72x72. Για το σκοπό αυτό, χρησιμοποιούμε την κλάση `torch.nn.functional` της PyTorch και τη μέθοδο `interpolate`. Σύμφωνα με τους Khasgiwala και Tailor (2021), η απεικόνιση με την οποία τροφοδοτούμε το μοντέλο δεν χρειάζεται κάποια περαιτέρω αλλαγή, επομένως οι τανυστές μας έχουν διαστάσεις (batch_size, 1, time_frames, mfccs) και (batch_size, 1, time_frames, mels).

Τέλος, για την περίπτωση του AST δεν χρειάζεται κάποια περαιτέρω επεξεργασία. Η κλάση `ASTFeatureExtractor` που χρησιμοποιήθηκε για τα δεδομένα με τα οποία τροφοδοτούμε το μοντέλο AST, εξάγει τα δεδομένα έτσι ώστε να είναι έτοιμα για χρήση. Το μοντέλο AST δέχεται μόνο Mel Spectrograms και μάλιστα αυτά ως τανυστές, πρέπει να έχουν διαστάσεις (batch_size, time_frames, mels) όπου τα time frames πρέπει να είναι αυστηρά ίσα με 1024 και τα Mels 128.

4.3.2 Διαχωρισμός σε train set, test set και validation set

Το επόμενο βήμα στο pipeline πριν την εκπαίδευση, είναι η δημιουργία των "dataloaders". Οι dataloaders στην PyTorch είναι εργαλεία με τα οποία μπορούμε να οργανώσουμε τα δεδομένα μας. Τα εργαλεία αυτά είναι διαμορφωμένα έτσι ώστε να τροφοδοτούν τα

μοντέλα μας με συγκεκριμένες δέσμες δεδομένων (batches). Παράλληλα, τα μοντέλα δέχονται τα δεδομένα με τέτοια διάταξη, ώστε να επιτυγχάνεται η αποδοτικότερη και η καλύτερη δυνατή γενίκευση. Η PyTorch έχει ξεχωριστή κλάση για τους dataloaders, επιτρέποντάς μας να οργανώσουμε τα δεδομένα μας κατάλληλα.

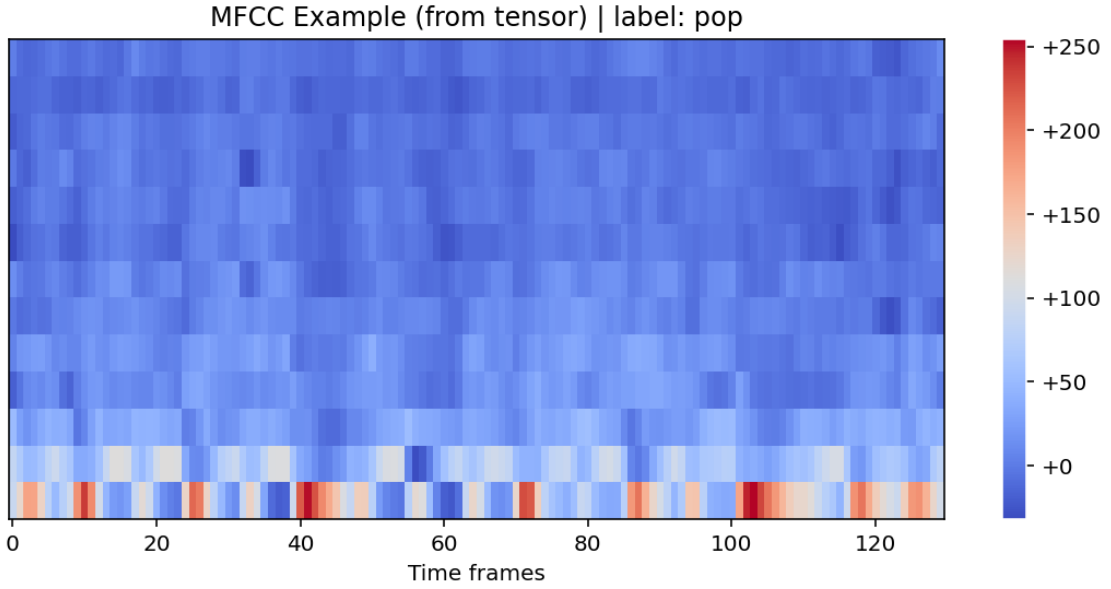
Βασική επιλογή κατά την εκπαίδευση του μοντέλου και την δημιουργία των dataloaders είναι η επιλογή του μεγέθους κάθε δέσμης (batch). Το σύνολο των δεδομένων, καθώς οργανώνεται στους dataloaders, χωρίζεται σε batches που αποτελούν υποσύνολα του όλου συνόλου. Το μοντέλο εκτίθεται σε αυτά τα σύνολα σε βήματα (training steps). Σε κάθε βήμα, μέσω της μεθόδου του backpropagation, ενημερώνονται τα weights και biases του δικτύου. Όταν το μοντέλο εκτεθεί σε όλα τα batches, λέμε ότι έχει παρέλθει μια epoch. Εναλλακτικά, κατά το backpropagation, το μοντέλο μπορεί να εκτεθεί σε ολόκληρο το σύνολο δεδομένων απευθείας. Αυτή η μέθοδος όμως στις περισσότερες φορές οδηγεί σε πολύ πιο αργή εκπαίδευση, καθώς μεταξύ άλλων τα δεδομένα στα οποία εκτίθεται το μοντέλο μεταφορτώνονται στη RAM. Για μεγάλο σύνολο δεδομένο λοιπόν, η χωρική πολυπλοκότητα αυξάνεται. Στα δικά μας πειράματα, επιλέγουμε batch size ίσο με 32. Τέλος, αξίζει να αναφερθεί ότι στην PyTorch τα περισσότερα layers είναι κατασκευασμένα έτσι ώστε η πρώτη διάσταση ενός τανυστή να είναι το batch size.

Επιπροσθέτως, σε αυτό το βήμα γίνεται ο διαχωρισμός του συνόλου δεδομένων. Το 60% των δεδομένων θα αποτελέσουν το train set. Το 15% θα χρησιμοποιηθεί ως validation set και το 25% θα χρησιμοποιηθεί ως test set. Τα train set και validation set χρησιμοποιούνται στην εκπαίδευση. Το test set είναι ένα σύνολο στο οποίο το μοντέλο δε θα εκτεθεί καθόλου κατά την εκπαίδευση και θα χρησιμοποιηθεί μόνο στο τέλος για αξιολόγηση.

4.3.3 Οπτικοποίηση δεδομένων

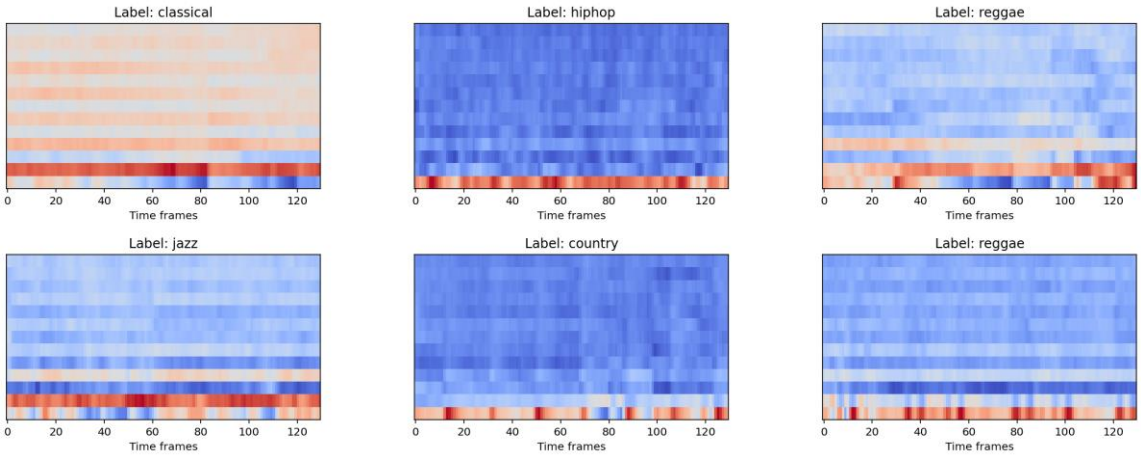
Βασικό κομμάτι της ροής ενεργειών, είναι η οπτικοποίηση των δεδομένων, ούτως ώστε να βεβαιωθούμε ότι οι απεικονίσεις μας είναι σωστές. Παράλληλα, η οπτικοποίηση των δεδομένων βοηθά στην καλύτερη κατανόηση και αξιολόγησή τους. Πριν τη δημιουργία και την εκπαίδευση των μοντέλων, λοιπόν, οπτικοποιούμε τα δεδομένα μας, ώστε να βεβαιωθούμε πως έχουν τη μορφή που περιμένουμε. Για τη διαδικασία αυτή, επιλέγουμε τυχαία δείγματα. Η οπτικοποίηση των δεδομένων γίνεται με χρήση της βιβλιοθήκης Matplotlib. Η Matplotlib είναι βιβλιοθήκη της Python για δημιουργία γραφημάτων. Η Matplotlib, δεν είναι κατασκευασμένη έτσι ώστε να επεξεργάζεται απευθείας τανυστές της PyTorch. Επομένως μετατρέπουμε τους τανυστές μας, σε τανυστές της NumPy,

χρησιμοποιώντας κατάλληλες μεθόδους. Η NumPy είναι βιβλιοθήκη της Python για αριθμητικούς υπολογισμούς και διαχείριση πινάκων, ενώ η Matplotlib μπορεί να διαχειριστεί τα δεδομένα της.



Σχήμα 5 Οπτικοποίηση παραδείγματος MFCC

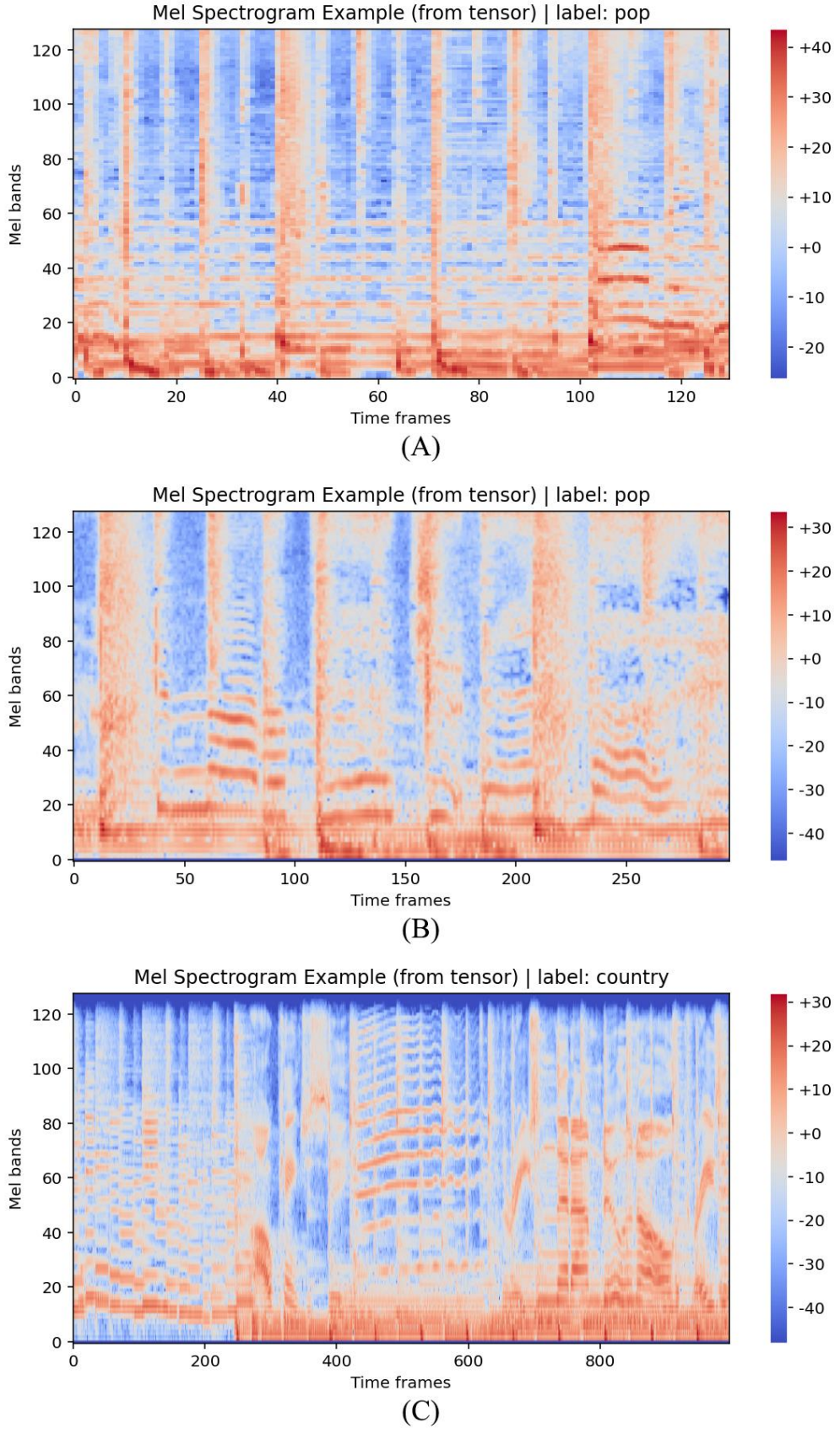
Τα δεδομένα μας είναι MFCCs και Mel Spectrograms. Ένα παράδειγμα MFCC για μουσικό δείγμα που ανήκει στην κλάση “pop”, βλέπουμε στο Σχήμα 5. Παρατηρούμε λοιπόν ότι η οπτικοποίηση αυτή, παραπέμπει στην απεικόνιση που γνωρίζουμε ότι έχουν τα MFCCs. Στο σχήμα, κοιτώντας τον παράλληλο άξονα διακρίνονται 13 παράλληλες σειρές. Αυτές αντιστοιχούν στα 13 cepstral coefficients που θέσαμε ως υπερπαράμετρο κατά την εξαγωγή τους. Παράλληλα, στον κάθετο άξονα βλέπουμε ότι έχουμε 130 time frames όπως περιμέναμε από τις διαστάσεις τανυστών που προκύπτουν. Γενικότερα, η απεικόνιση των MFCCs δεν βοηθά έναν άνθρωπο να καταλάβει πολλά διαισθητικά, βλέπουμε όμως ότι οι βασικές πληροφορίες υπάρχουν κυρίως στα πρώτα coefficients.



Σχήμα 6 Οπτικοποιήσεις MFCCs για διάφορα μουσικά είδη

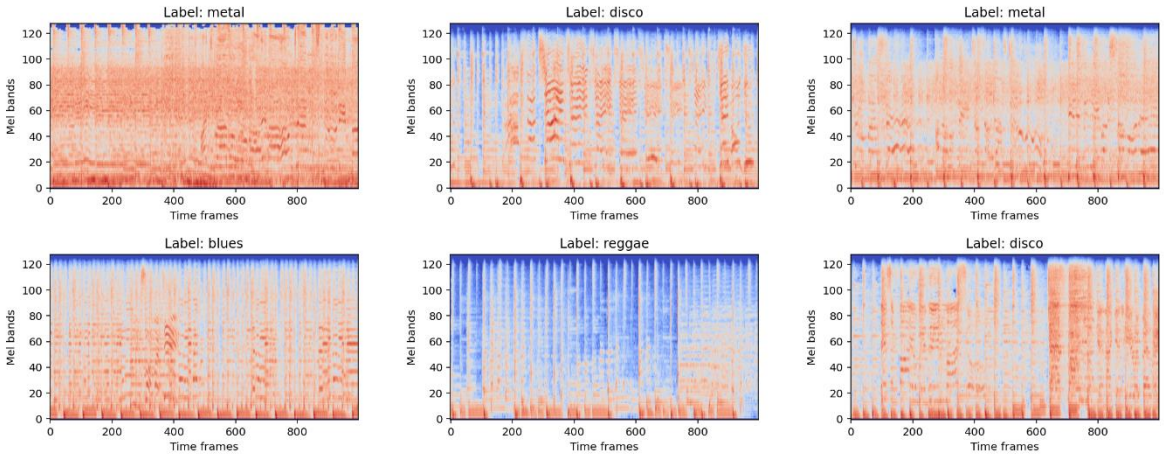
Η απεικόνιση των MFCCs μπορεί να μην προσφέρει πολλή πληροφορία διαισθητικά, όμως βλέποντας MFCCs από διαφορετικά μουσικά είδη είναι εφικτό να δούμε διαφορές. Στο Σχήμα 6 μπορούμε να δούμε τα MFCCs για διαφορετικά είδη. Εκεί διακρίνουμε πως στα περισσότερα παραδείγματα, φαίνεται πως το μεγαλύτερο μέρος της πληροφορίας που παίρνουμε παραμένει στα πρώτα coefficients, όμως υπάρχουν διαφοροποιήσεις μεταξύ τους. Αυτό είναι μια καλή ένδειξη πως μπορεί να υπάρχουν μοτίβα σε ίδια είδη. Χαρακτηριστικό είναι επίσης ότι η MFCC απεικόνιση για την κλασσική μουσική, διαφέρει αρκετά. Αυτό το γεγονός δεν προκαλεί έκπληξη, καθώς τα όργανα και συνεπώς οι χροιές που εμφανίζονται σε αυτό το είδος είναι αρκετά διαφορετικές συγκριτικά με τα υπόλοιπα είδη.

Αντίστοιχα, στην περίπτωση των Mel Spectrograms, οπτικοποιούμε τα δεδομένα μας για κάθε μια από τις διαφορετικές ρυθμίσεις υπερπαραμέτρων που έχουμε ανά σειρά πειραμάτων. Στο Σχήμα 7 βλέπουμε τρείς διαφορετικές απεικονίσεις. Το μέρος (A) του σχήματος 7 αντιστοιχεί στην «περίπτωση 1» του Πίνακα 1. Το μέρος (B) αντιστοιχεί στην «περίπτωση 2» του ίδιου πίνακα. Στο μέρος (C) έχουμε τις ρυθμίσεις της περίπτωσης 2, αλλά για ένα δείγμα 10 δευτερολέπτων, ενώ στα μέρη (A) και (B) το δείγμα είναι για 3 δευτερόλεπτα. Οι απεικονίσεις αυτές, προέκυψαν από το pipeline του αντίστοιχου πειράματος καθώς δοκιμάζουμε μοντέλα ίδιων αρχιτεκτονικών, δίνοντάς τους Mel Spectrograms που εξήχθησαν με διαφορετικές ρυθμίσεις.



Σχήμα 7 Οπτικοποιήσεις Mel Spectrograms για διαφορετικές ρυθμίσεις και διάρκειες ακουστικών δειγμάτων

Παρατηρούμε πως οι απεικονίσεις έχουν τη μορφή που περιμένουμε. Στον οριζόντιο άξονα έχουμε τις προβλεπόμενες τιμές για τα time frames και στον κάθετο άξονα διακρίνονται 128 mel bands.



Σχήμα 8 Απεικονίσεις Mel Spectrograms για ηχητικά δείγματα διαφορετικών ειδών, διάρκειας 10 δευτερολέπτων

Τα Mel Spectrograms αποτελούν μια απεικόνιση όπου αντίθετα με τα MFCCs, ένας άνθρωπος διαισθητικά μπορεί να βγάλει πολλά περισσότερα συμπεράσματα. Στο Σχήμα 8 βλέπουμε τα αντίστοιχα φασματογραφήματα για διαφορετικά είδη μουσικής. Μπορούμε για παράδειγμα να φανταστούμε πως οι διαδοχικές «κορυφές», μπορεί να είναι χτυπήματα ταμπούρου. Μέσω αυτών, μπορούμε να δούμε τα επαναλαμβανόμενα ρυθμικά μοτίβα κάθε παραδείγματος. Επίσης μπορούμε να παρατηρήσουμε ότι στα metal παραδείγματα υπάρχει έντονη συχνοτική κάλυψη, γεγονός που ενδεχομένως οφείλεται στην παρουσία παραμορφωμένης κιθάρας.

4.3.4 Υλοποίηση μοντέλων

Εφόσον έχει ολοκληρωθεί η μεταφόρτωση, η επεξεργασία και ο έλεγχος των δεδομένων μας, το επόμενο βήμα είναι η υλοποίηση των μοντέλων. Κάθε Jupyter notebook περιλαμβάνει και ένα μοντέλο το οποίο ελέγχεται για την απόδοσή του με διαφορετικά δεδομένα. Η μόνη εξαίρεση αποτελούν τα αναδρομικά δίκτυα, στα οποία ελέγχουμε όλα τα σχετικά μοντέλα στο ίδιο notebook, καθώς η αρχιτεκτονική τους είναι παρεμφερής.

Για την υλοποίηση κάθε μοντέλου, δημιουργούμε μια κλάση με την αρχιτεκτονική, όπως αυτή αναλύθηκε στο αντίστοιχο κεφάλαιο. Κατόπιν, δημιουργούμε ένα μοντέλο το οποίο ουσιαστικά αποτελεί ένα στιγμιότυπο της κλάσης. Πριν προχωρήσουμε στην εκπαίδευση

του μοντέλου, είναι απαραίτητο να κάνουμε μερικούς ελέγχους, ώστε να βεβαιωθούμε ότι το μοντέλο μας λειτουργεί σωστά. Αυτό γίνεται κάνοντας ένα δοκιμαστικό forward pass με ένα δείγμα από τα δεδομένα μας. Επιπροσθέτως, μπορούμε να κάνουμε το δοκιμαστικό forward pass χρησιμοποιώντας την εντολή `summary` από τη βιβλιοθήκη `torchinfo`. Η εν λόγω εντολή, εκτός του ότι μας επιβεβαιώνει ότι το μοντέλο λειτουργεί σωστά, επιστρέφει και πληροφορίες για τη δομή του μοντέλου καθώς και γενικά στοιχεία αυτού. Αναφορικά, μας δίνει μια εκτίμηση του μεγέθους που θα είχε το συγκεκριμένο μοντέλο αν το αποθηκεύαμε. Για παράδειγμα, ένα MLP μοντέλο που χρησιμοποιεί ηχητικά $3^{ων}$ δευτερολέπτων με τις ρυθμίσεις της «Περίπτωσης 1» του Πίνακα 1, θα είχε μέγεθος 34,67 MB. Ένα MLP μοντέλο που χρησιμοποιεί ηχητικά $3^{ων}$ δευτερολέπτων με τις ρυθμίσεις της «Περίπτωσης 2» του Πίνακα 1 θα είχε μέγεθος 78,45 MB. Ένα MLP μοντέλο που χρησιμοποιεί ηχητικά 10 δευτερολέπτων με τις ρυθμίσεις της «Περίπτωσης 2» του Πίνακα 1 θα είχε μέγεθος 261,95 MB. Ενώ ένα MLP Μοντέλο που χρησιμοποιεί MFCCs θα είχε μέγεθος μόλις 4.06 MB. Βλέπουμε λοιπόν πως το μέγεθος ενός μοντέλου ίδιας αρχιτεκτονικής, μπορεί να ποικίλει σημαντικά ανάλογα τα δεδομένα τα οποία δέχεται. Αυτός μπορεί να είναι ένας σημαντικός παράγοντας προτίμησης κάποιου μοντέλου, ακόμα κι αν υπάρχει κάποιο άλλο που αποδίδει λίγο καλύτερα. Ένα μικρότερο μοντέλο είναι πιθανώς πιο συμφέρουσα λύση εάν το αυτό πρόκειται να χρησιμοποιηθεί σε εφαρμογή για smartphones, όπου η μνήμη είναι μικρότερη.

Ειδικά για την περίπτωση του μοντέλου AST, η υλοποίηση γίνεται λίγο διαφορετικά. Η κάθε βιβλιοθήκη έχει προβλέψει μεθόδους για τη δημιουργία pre-trained μοντέλων. Η PyTorch έχει επίσης τα δικά της μοντέλα. Εμείς όμως χρησιμοποιούμε το AST από την πλατφόρμα του Hugging Face και την αντίστοιχη βιβλιοθήκη "Transformers", καθώς η PyTorch δεν περιλαμβάνει το AST. Η διαδικασία όμως είναι ίδια. Αρχικά δημιουργούμε ένα στιγμιότυπο του μοντέλου και κατόπιν του μεταφορτώνουμε τις τιμές των παραμέτρων του (weights, biases). Κάθε βιβλιοθήκη μπορεί να έχει διαφορετικές τιμές παραμέτρων, ανάλογα το πως εκπαιδεύτηκε κάθε μοντέλο.

Αφού δημιουργηθεί ένα pre-trained μοντέλο, ενδεχομένως θα χρειαστεί κάποια τροποποίηση της αρχιτεκτονικής του, ώστε αυτό να προσαρμοστεί στις ανάγκες της εργασίας του προγραμματιστή. Συγκεκριμένα στη δική μας περίπτωση, θα πρέπει να τροποποιήσουμε τον classifier που βρίσκεται στο τέλος του δικτύου, έτσι ώστε να ταξινομεί τα δεδομένα σε 10 κλάσεις. Κατόπιν, το μοντέλο θα πρέπει να επανεκπαιδευθεί στα

δεδομένα, αφού όμως «παγώσουμε» (freeze) τις τιμές των υπόλοιπων τιμών των παραμέτρων εκτός του classifier που αλλάξαμε. Έτσι λοιπόν, κατά το training, θα προσαρμοστούν μόνο τα weights και biases του classifier στο τέλος, ο οποίος όπως έχουμε αναφέρει είναι ουσιαστικά ένα MLP δίκτυο. Αυτή η διαδικασία είναι γνωστή ως transfer learning και είναι μια πολύ σημαντική τεχνική στο χώρο του DL. Αυτό ισχύει, γιατί μέσω αυτής δίνεται η δυνατότητα ακόμα και σε απλούς χρήστες να δημιουργήσουν και να εκμεταλλευθούν μοντέλα τα οποία δεν έχουν τους πόρους να δημιουργήσουν από την αρχή.

4.4 Εκπαίδευση μοντέλων

Στο κεφάλαιο αυτό θα παρουσιάσουμε τη διαδικασία εκπαίδευσης των μοντέλων, καθώς και τις τιμές τυχόν υπερπαραμέτρων που θέτονται σε αυτή. Τα μοντέλα εκπαιδεύονται χρησιμοποιώντας τις ίδιες συναρτήσεις για λόγους ομοιογένειας των πειραμάτων. Οι συναρτήσεις αυτές παρουσιάζονται αναλυτικά στο κάθε Jupyter notebook. Επίσης, κατά την εκπαίδευση χρησιμοποιούμε την κλάση `SummaryWriter` της βιβλιοθήκης TensorBoard. Το εργαλείο αυτό μας επιτρέπει να αποθηκεύουμε αρχεία καταγραφής (logs) των αποτελεσμάτων εκπαίδευσης, ώστε να κατασκευάζουμε κοινά διαγράμματα για την απόδοση των μοντέλων μας.

4.4.1 Συνάρτηση κόστους και αλγόριθμος βελτιστοποίησης

Απαραίτητο βήμα για την εκπαίδευση των μοντέλων είναι η επιλογή μιας συνάρτησης κόστους (loss function) και ενός αλγόριθμου βελτιστοποίησης (optimizer). Ως συνάρτηση κόστους επιλέγεται η cross entropy loss. Η επιλογή αυτή είναι πολύ συχνή για προβλήματα ταξινόμησης και συναντάται αρκετά στη βιβλιογραφία.

Η PyTorch έχει υλοποιήσει την cross entropy loss, με την κλάση `torch.nn.CrossEntropyLoss` την οποία και χρησιμοποιήσαμε για την εκπαίδευση των μοντέλων. Ένα σημείο που πρέπει να δοθεί βάση, είναι το ότι στην PyTorch, η cross entropy loss τροφοδοτείται με logits. Πολύ συχνά, στην υλοποίηση των μοντέλων ταξινόμησης, στο τέλος υφίσταται μια συνάρτηση ενεργοποίησης όπως η Softmax, ούτως ώστε το μοντέλο να εξάγει απευθείας τιμές πιθανοτήτων. Στα μοντέλα που εκπαιδεύονται με cross entropy loss, δεν περιλαμβάνουμε ένα τέτοιο επίπεδο. Αν θέλουμε να δούμε τις τιμές των πιθανοτήτων, κανονικοποιούμε το αποτέλεσμα αργότερα.

Ως αλγόριθμο βελτιστοποίησης επιλέξαμε τον Adam. Η επιλογή αυτή είναι επίσης μια αρκετά συχνή επιλογή στη βιβλιογραφία. Η PyTorch επίσης περιλαμβάνει υλοποίηση του

αλγορίθμου, με την κλάση `torch.optim.Adam`. Καθώς δημιουργούμε στιγμιότυπο αυτής της κλάσης, δίνουμε κάποιες τιμές για ορίσματα ώστε να ρυθμίσουμε κάποιες υπερπαραμέτρους της εκπαίδευσης. Συγκεκριμένα, ορίζουμε το learning rate και το weight decay.

Το learning rate είναι μια πολύ σημαντική παράμετρος για την εκπαίδευση του μοντέλου και κάθε αρχιτεκτονική μπορεί να αποδίδει καλύτερα για διαφορετικές τιμές της. Επιπροσθέτως υπάρχουν τεχνικές όπου το learning rate μεταβάλλεται κατά την εκπαίδευση. Για λόγους ομοιογένειας, εκπαιδεύσαμε όλα τα μοντέλα με ένα σταθερό learning rate. Η προσέγγιση αυτή επιλέχθηκε επίσης επειδή περισσότερο μας ενδιαφέρει να συγκρίνουμε την απόδοση των αρχιτεκτονικών και όχι τόσο να παράγουμε μοντέλο με τη μέγιστη απόδοση. Παρ' όλα αυτά επειδή κάθε αρχιτεκτονική είναι διαφορετική, δεν είχαμε την ίδια τιμή learning rate σε όλα και πειραματιστήκαμε με διάφορες τιμές, ώστε να καταλήξουμε σε κάποια που προσφέρει καλύτερη γενίκευση. Τα μοντέλα λοιπόν MLP, CNN και ViT, εκπαιδεύτηκαν με learning rate ίσο με 0,001. Στα αναδρομικά μοντέλα (RNN-LSTM-GRU) καθώς και το AST, η τιμή του learning rate τέθηκε ίση με 0,0001.

4.4.2 Υπολογισμός πιστότητας (accuracy)

Η διαδικασία της εκπαίδευσης υπερβαίνει την απλή εξαγωγή ενός τελικού μοντέλου. Απαιτείται συστηματική εποπτεία και ανάλυση των μεταβολών ανά epoch, προκειμένου να αξιολογηθεί η όλη διαδικασία και να προσδιοριστούν οι απαραίτητες βελτιώσεις στην εκπαίδευση του μοντέλου. Για τον λόγο αυτό, χρησιμοποιούμε κάποιες μετρικές και παρακολουθούμε την πορεία των τιμών τους, καθώς η εκπαίδευση εκτελείται βηματικά.

Η συνάρτηση κόστους θα μας δώσει τις τιμές του κόστους (loss) για κάθε epoch και μπορούμε να παρατηρούμε την εξέλιξη τις τιμής του καθώς η εκπαίδευση περατώνεται. Παράλληλα όμως, είναι συνήθης τακτική να παρακολουθείται και η εξέλιξη της τιμής της πιστότητας ανά epoch. Κατά τη διαδικασία της εκπαίδευσης λοιπόν, συμπεριλαμβάνουμε τον υπολογισμό της.

Συγκεκριμένα, αυτό που κάνουμε είναι να χρησιμοποιούμε τον τύπο (26) για να υπολογίσουμε την πιστότητα σε κάθε epoch. Η διαδικασία αυτή γίνεται για το train set αλλά και για το validation set. Εν τέλει λοιπόν, παίρνουμε μια τιμή για την accuracy ανά epoch για το train set αλλά και μια αντίστοιχη τιμή για το validation set, έτσι ώστε συγκρίνοντας

την εξέλιξη των τιμών αυτών, να βγάλουμε συμπέρασμα για το αν παρουσιάζεται overfitting από ένα σημείο και μετά.

4.4.3 Συναρτήσεις για την εκπαίδευση των μοντέλων

Η εκπαίδευση των μοντέλων πραγματοποιείται μέσω τριών συναρτήσεων που ορίζουμε. Οι συναρτήσεις αυτές είναι η `train_step`, η `test_step` και η `train`. Η τελευταία χρησιμοποιεί τις δύο πρώτες, οι οποίες ουσιαστικά περιλαμβάνουν τη διαδικασία της εκπαίδευσης.

Η διαδικασία της εκπαίδευσης των μοντέλων στην PyTorch γίνεται με την ακολουθία συγκεκριμένων βημάτων:

- Forward pass: Το μοντέλο τροφοδοτείται με όλα τα δεδομένα μια φορά και εξάγει αποτελέσματα.
- Υπολογισμός του loss: Τα αποτελέσματα του forward pass συγκρίνονται με τα πραγματικά αποτελέσματα και υπολογίζεται η απώλεια (loss).
- Μηδενισμός των κλίσεων (gradients): Οι αποθηκευμένες τιμές των κλίσεων που έχει ο αλγόριθμος βελτιστοποίησης, μηδενίζονται ώστε να είναι έτοιμος για το επόμενο βήμα της εκπαίδευσης.
- Εκτέλεση backpropagation: Υπολογίζονται οι μερικές παράγωγοι των παραμέτρων του μοντέλου, καθώς εκτελείται η διαδικασία του backpropagation.
- Gradient descent: Ενημερώνονται οι τιμές των παραμέτρων του μοντέλου, βάσει των υπολογισμών του προηγούμενου βήματος.

Τα παραπάνω βήματα υλοποιούνται χρησιμοποιώντας κατάλληλες μεθόδους της PyTorch και συμπεριλαμβάνονται στην συνάρτηση `train_step`. Παράλληλα, στη συγκεκριμένη συνάρτηση, υπολογίζεται η accuracy του training για ένα συγκεκριμένο βήμα. Η `train_step` τροφοδοτείται με τα δεδομένα του train set. Η συνάρτηση επιστρέφει την τιμή του loss και του accuracy μίας epoch σε μορφή πλειάδας (tuple). Η `test_step` αντίστοιχα, επιστρέφει το αντίστοιχο loss και accuracy στην ίδια δομή δεδομένων. Απλά σε αυτή την περίπτωση, οι μετρικές υπολογίζονται πάνω στο validation set. Οι δύο συναρτήσεις αυτές εκτελούν τη διαδικασία για κάθε batch του dataloader που τους δίνουμε, εκτελώντας ουσιαστικά τα “steps” της μιας epoch.

Η συνάρτηση `train` αναλαμβάνει να εκτελέσει τις άλλες δύο συναρτήσεις για όσες φορές της ορίσουμε, σύμφωνα με τον αριθμό των epochs. Επιπλέον, εμφανίζει στην οθόνη τα

αποτελέσματα του loss και του accuracy για το train set και το validation set της κάθε epoch. Με αυτόν τον τρόπο, είναι εφικτό να δούμε την πορεία των τιμών και να βγάλουμε συμπεράσματα για την εκπαίδευση. Τέλος, η συνάρτηση επιστρέφει ένα λεξικό με τις τιμές του loss και του accuracy για το train set και το validation set.

Η συνάρτηση `train` είναι ρυθμισμένη έτσι ώστε να χρησιμοποιεί την κλάση `SummaryWriter` της βιβλιοθήκης TensorBoard. Το TensorBoard είναι ένα εργαλείο οπτικοποίησης και επίβλεψης της εκπαίδευσης μοντέλων DL. Αποθηκεύοντας τα αποτελέσματα του training σε logs που χρησιμοποιούνται με αυτό, είναι εφικτό να μεταφορτώσουμε τα δεδομένα αυτά στην πλατφόρμα του TensorBoard και να εξάγουμε γραφήματα που παρουσιάζουν την εξέλιξη του loss και του accuracy για τα μοντέλα μας. Το σημαντικό είναι όμως ότι μπορούμε να έχουμε τις γραφικές παραστάσεις πάνω από ενός μοντέλου, στο ίδιο γράφημα. Το TensorBoard, δυναμικά μας δίνει την επιλογή να επιλέξουμε ποια γραφική παράσταση θα έχουμε στο ίδιο διάγραμμα ώστε να συγκρίνουμε τα μοντέλα. Η `train` έχει ως παράμετρο ένα αντικείμενο της κλάσης `SummaryWriter` και είναι ρυθμισμένη έτσι ώστε αν της δώσουμε ένα τέτοιο αντικείμενο, να κρατήσει logs.

Τέλος, αναφέρουμε ότι στα κελιά που εκτελούμε την εκπαίδευση με τη συνάρτηση `train`, χρησιμοποιούμε την κλάση `timer` από τη βιβλιοθήκη timeit, για να χρονομετρήσουμε τη διάρκεια της εκπαίδευσης. Η εν λόγω βιβλιοθήκη της Python είναι εργαλείο για ακριβή χρονομέτρηση τμημάτων κώδικα.

4.5 Μέθοδοι και τεχνικές Regularization

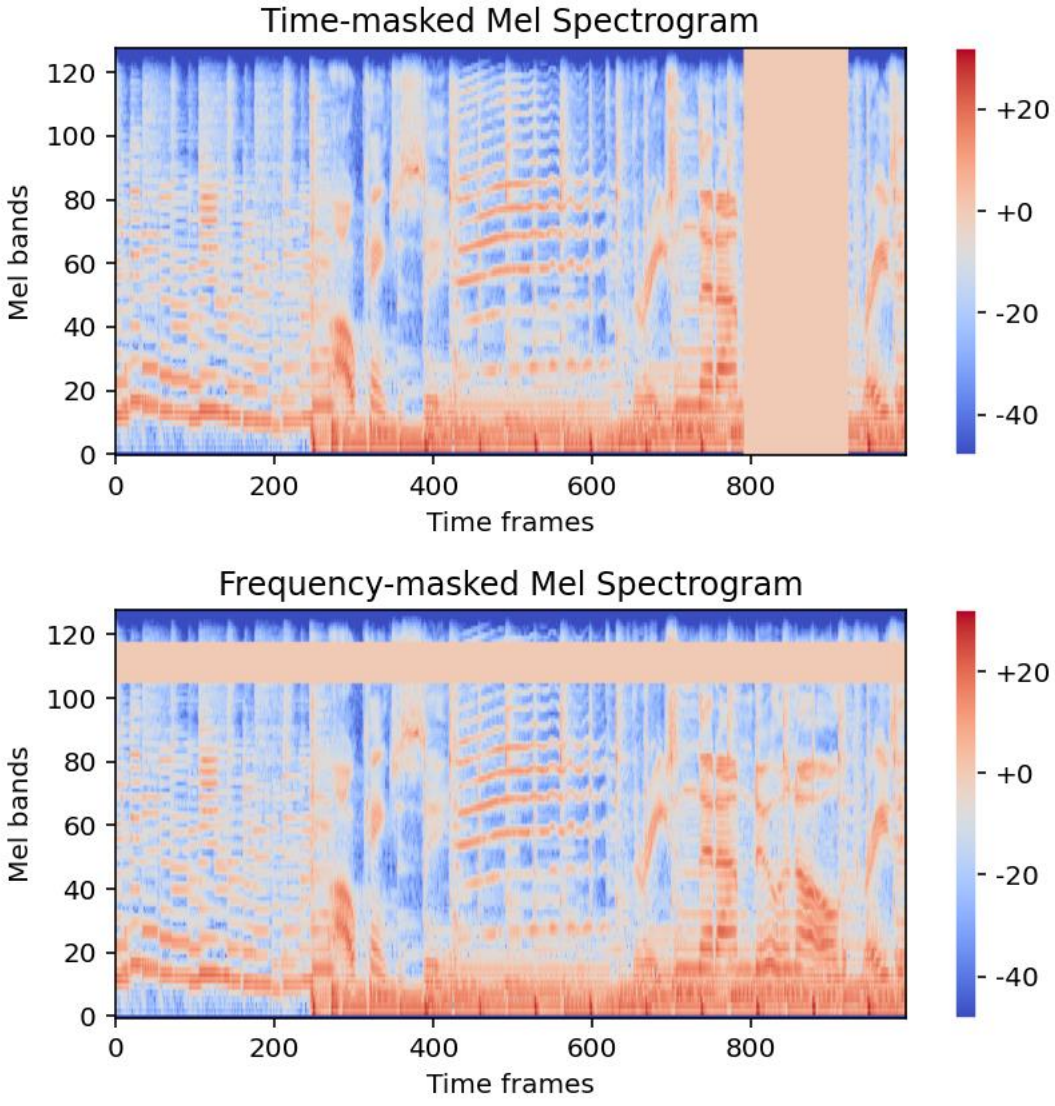
Μια από τις κύριες προκλήσεις κατά τη διαδικασία της εκπαίδευσης ενός μοντέλου, είναι η παρουσία του overfitting. Στο κεφάλαιο αυτό, παρουσιάζουμε τις τεχνικές που χρησιμοποιούμε για την αντιμετώπιση του φαινομένου αυτού. Επειδή ο σκοπός μας είναι η μελέτη και η σύγκριση της ικανότητας των μοντέλων διαφορετικής αρχιτεκτονικής να γενικεύουν, θα χρησιμοποιήσουμε συγκεκριμένες μεθόδους regularization έτσι ώστε να μην επηρεαστεί η έρευνά μας. Αυτό γίνεται επειδή κάθε μέθοδος μπορεί να επιδρά με διαφορετικό τρόπο στο κάθε μοντέλο.

4.5.1 Data augmentation

Η τεχνική του data augmentation είναι από τις πιο βασικές τεχνικές regularization. Στην επεξεργασία ήχου υπάρχουν διάφοροι τρόποι να εξάγουμε augmented data. Ενδεικτικά, μπορούμε να αλλάξουμε την τονικότητα, να μεταβάλλουμε το tempo ή απλά να

προσθέσουμε θόρυβο στα δείγματα. Δεδομένου ότι εμείς έχουμε ήδη εξάγει τα MFCCs και τα Mel Spectrograms, επιλέγουμε μια τεχνική που μπορεί να εφαρμοστεί σε αυτά. Η τεχνική αυτή, είναι η μέθοδος του time/frequency masking. Οι Gong et al. (2021) κατά την εκπαίδευση του AST, επίσης χρησιμοποιούν τη μέθοδο αυτή.

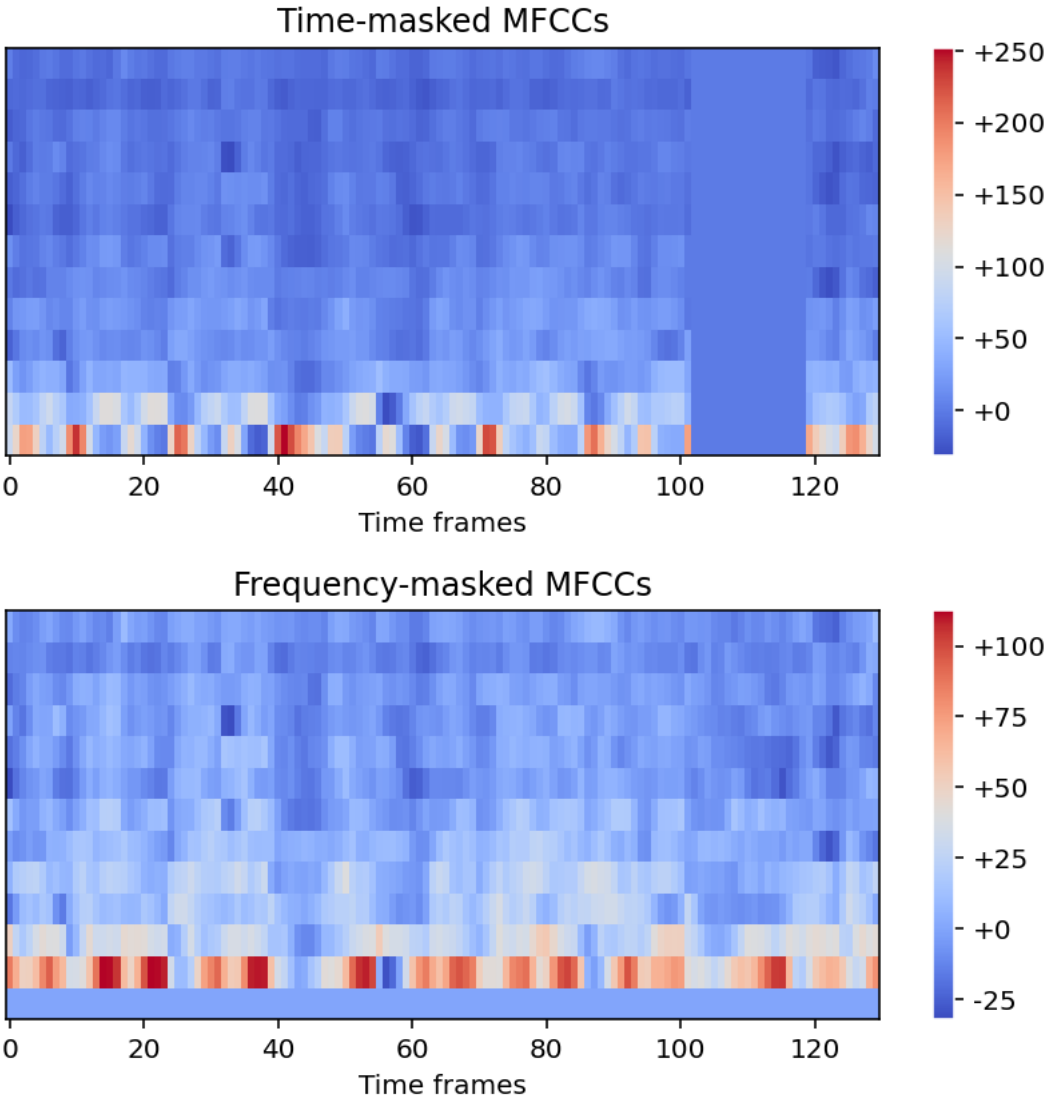
Η PyTorch περιλαμβάνει τις κλάσεις `TimeMasking` και `FrequencyMasking`. Με τα εν λόγω εργαλεία καταφέρνουμε να «κρύψουμε» κάποιο ποσοστό του κάθετου και του οριζόντιου άξονα αντίστοιχα.



Σχήμα 9 Mel Spectrograms στα οποία έχει εφαρμοστεί Time masking και Frequency masking αντίστοιχα

Για να βεβαιωθούμε ότι τα δεδομένα μας τροποποιούνται σωστά, εκτελούμε την διαδικασία οπτικοποίησης που περιεγράφηκε σε προηγούμενο κεφάλαιο, ώστε να απεικονίσουμε τα

δεδομένα μας. Στο Σχήμα 9 βλέπουμε την εφαρμογή των Time masking και Frequency masking σε ένα Mel Spectrogram. Για καλύτερη γενίκευση, η PyTorch έχει ρυθμίσει τις κλάσεις αυτές έτσι ώστε κάθε φορά να εφαρμόζουν το masking σε μια τυχαία περιοχή του κάθε άξονα. Επίσης το ποσοστό που καλύπτουν είναι τυχαίο, έχοντας όμως μια μέγιστη τιμή που δίνεται από το χρήστη. Εμείς, για τα Mel Spectrograms ως μέγιστη τιμή δίνουμε το 15% του κάθετου άξονα στην `TimeMasking` και το 12% του οριζόντιου άξονα στη `FrequencyMasking`. Στην περίπτωση του ViT όπου το μέγεθος των διαγραμμάτων μετατρέπεται σε 72x72, κάνουμε masking το πολύ σε 10 τιμές του κάθε άξονα. Τέλος, στην περίπτωση του AST με δείγματα των 3^{ων} δευτερολέπτων, χρησιμοποιούμε μόνο Frequency Masking. Όπως μπορούμε να διακρίνουμε στο Σχήμα 4(B), το μεγαλύτερο μέρος του κάθετου άξονα είναι ούτως ή άλλως σαν να έχει υποστεί masking, επομένως αυτό κρίθηκε άσκοπο.



Σχήμα 10 MFCCs στα οποία έχει εφαρμοστεί Time masking και Frequency masking αντίστοιχα

Η αντίστοιχη διαδικασία ακολουθείται και στα MFCCs με μέγιστη τιμή το 15% του κάθετου άξονα στην `TimeMasking` και το 12% του οριζόντιου άξονα στη `FrequencyMasking`. Στο Σχήμα 10 βλέπουμε πως αυτό εφαρμόζεται στα MFCCs με τη `FrequencyMastking` σε αυτή την περίπτωση να έχει καλύψει το πρώτο cepstral coefficient. Επίσης στην περίπτωση του ViT, εφαρμόζουμε masking το πολύ σε 10 τιμές του κάθε άξονα.

Για την εφαρμογή του data augmentation, τροποποιούμε την συνάρτηση `train_step` και την συνάρτηση `train`, έτσι ώστε να περιλαμβάνουν τις τροποποιήσεις που αναφέραμε. Εν τέλει ορίζουμε δύο νέες συναρτήσεις. Η εκπαίδευση με τις νέες συναρτήσεις `train_step_with_augmentation` και `train_with_augmentation`, θα περιλαμβάνει το Time masking και Frequency masking. Στην πειραματική διαδικασία, εκπαιδεύουμε ένα

μοντέλο χωρίς data augmentation και ένα μοντέλο με data augmentation, ούτως ώστε να εξετάσουμε αν η τεχνική αυτή ωφέλησε το μοντέλο.

4.5.2 Weight Decay

Το weight decay, είναι ακόμα μια τεχνική κανονικοποίησης (regularization) που χρησιμοποιείται για τον έλεγχο του overfitting. Όπως αναφέρθηκε και σε προηγούμενο υποκεφάλαιο, η τιμή του τίθεται ως όρισμα στο στιγμιότυπο του αλγόριθμου βελτιστοποίησης Adam που επιλέξαμε. Στην περίπτωση του weight decay που χρησιμοποιείται με τον αλγόριθμο Adam, το weight decay υλοποιείται ως L2 regularization. Δοκιμάζοντας διάφορες τιμές για αυτό, συμπεράναμε ότι τα καλύτερα αποτελέσματα εμφανίζονταν όταν είχαμε ίδια τιμή για learning rate και weight decay. Επομένως όπως και στο learning rate, η τιμή του τέθηκε ίση με 0,001 για τα μοντέλα MLP, CNN, ViT και ίση με 0,0001 για τα μοντέλα RNN, LSTM, GRU και AST.

4.5.5 Early Stopping

Η μέθοδος του early stopping είναι ένας από τους πιο απλούς και συνάμα αποδοτικούς τρόπους να αποφευχθεί το overfitting. Η διαδικασία που ακολουθούμε είναι η εξής. Αφού εκπαιδεύσουμε ένα μοντέλο μιας συγκεκριμένης αρχιτεκτονικής χωρίς data augmentation και με data augmentation, συγκρίνουμε την απόδοσή τους. Διακρίνοντας το μοντέλο που γενικεύει καλύτερα, προσπαθούμε να δούμε εάν έχει εμφανιστεί overfitting. Εφόσον εντοπίσουμε κάτι τέτοιο, διακρίνουμε την epoch που αυτό αρχίζει να εμφανίζεται.

Για τη διάκριση της epoch που το overfitting εμφανίζεται, υπάρχουν διάφορες προσεγγίσεις. Αρχικά, μια πρώτη εικόνα, είναι η διάκριση των γραφικών παραστάσεων του train loss/test loss ή του train accuracy/test accuracy. Το σημείο που οι γραφικές παραστάσεις αρχίζουν να αποκλίνουν και να μην ακολουθούν παρόμοια πορεία είναι μια ένδειξη overfitting. Κατόπιν, ένδειξη overfitting είναι και η συμπεριφορά της γραφικής παράστασης του train loss ή του test loss από μόνες τους. Διακρίνουμε λοιπόν το σημείο που οι παραστάσεις αυτές σταματάνε να βελτιώνονται. Εμείς στηριζόμαστε περισσότερο στην επίδοση του loss και λιγότερο στην accuracy. Είναι πιθανόν να παρατηρήσουμε το loss να αυξάνεται, αλλά την accuracy να παραμένει σταθερή. Επίσης, μια συνήθης τακτική είναι αυτή της “patience”, όπου επιλέγεται ένας αριθμός epochs για τον οποίο επιτρέπεται στο μοντέλο να παρουσιάσει σταθερές τιμές που δε βελτιώνονται, πριν επιλεγεί η epoch όπου η εκπαίδευση θα σταματήσει. Στα πειράματά μας, από τη στιγμή που εκπαιδεύουμε νέο μοντέλο εξ αρχής, διακρίνουμε την epoch που θα γίνει το early stopping με το που εμφανιστεί στασιμότητα.

4.6 Αξιολόγηση μοντέλων

Στο κεφάλαιο αυτό, θα αναλυθούν οι μέθοδοι και οι μετρικές που χρησιμοποιούνται για την αξιολόγηση των μοντέλων μετά την εκπαίδευσή τους. Η αξιολόγηση είναι το τελευταίο βήμα του pipeline κάθε πειράματος και για λόγους ομοιογένειας, χρησιμοποιούμε τις ίδιες μεθόδους σε όλα τα μοντέλα.

4.6.1 Διαγράμματα Loss και Accuracy

Όπως αναφέρθηκε και σε προηγούμενα κεφάλαια, οι συναρτήσεις που χρησιμοποιούνται για την εκπαίδευση των μοντέλων επιστρέφουν λεξικά της Python τα οποία περιέχουν τα αποτελέσματα της εκπαίδευσης. Τα αποτελέσματα αυτά, είναι ουσιαστικά οι τιμές για το loss και το accuracy του train set και του test set. Μετά την εκπαίδευση του κάθε μοντέλου, χρησιμοποιούμε αυτά τα δεδομένα για να απεικονίσουμε τις γραφικές παραστάσεις loss-epoch και accuracy-epoch.

Για την απεικόνιση της γραφικής παράστασης, ορίζουμε μια συνάρτηση ονόματι `plot_loss_curves`. Η συνάρτηση αυτή, χρησιμοποιεί την βιβλιοθήκη Matplotlib ώστε να εμφανίσει δύο απεικονίσεις. Το διάγραμμα loss-epoch και το διάγραμμα accuracy-epoch, έχοντας δύο γραφικές παραστάσεις σε κάθε διάγραμμα. Τη γραφική παράσταση που αντιστοιχεί στο train set και τη γραφική παράσταση που αντιστοιχεί στο test set. Η `plot_loss_curves` δέχεται ως όρισμα το λεξικό που επιστρέφει η `train` ή η `train_with_augmentation` και τη χρησιμοποιούμε μετά την εκπαίδευση κάθε μοντέλου.

Τα εν λόγω διαγράμματα μπορούν να χρησιμοποιηθούν ως μετρική για να συγκρίνουμε την απόδοση μοντέλων, όμως η `plot_loss_curves` παρουσιάζει τα αποτελέσματα ενός μοντέλου. Για τη σύγκριση μοντέλων, χρησιμοποιούμε την πλατφόρμα του TensorBoard, όπου έχουμε μεταφορτώσει τα logs της εκπαίδευσης κάθε μοντέλου.

4.6.2 Αξιολόγηση στο test set

Μια μετρική που θα χρησιμοποιήσουμε αρκετά για την επίδοση των μοντέλων είναι η αξιολόγηση στο test set. Για τη διαδικασία αυτή δεν απαιτείται να ορίσουμε κάποιας νέα συνάρτηση, καθώς ήδη έχουμε τα εργαλεία που χρειαζόμαστε. Συγκεκριμένα, η συνάρτηση `test_step`, μπορεί να χρησιμοποιηθεί, για να πάρουμε την τιμή του loss και του accuracy, εφαρμόζοντάς την στο test set. Το εν λόγω σύνολο, αποτελείται από δεδομένα στα οποία το μοντέλο δεν έχει εκτεθεί καθόλου κατά την εκπαίδευσή του. Πραγματοποιούμε την

αξιολόγηση αυτή στο καλύτερο μοντέλο που έχει προκύψει από την κάθε αρχιτεκτονική. Είτε αυτό είναι μέσω data augmentation ή μέσω early stopping.

4.6.3 Συνάρτηση πρόβλεψης για νέα δεδομένα

Για περαιτέρω αξιολόγηση, ορίζουμε μια συνάρτηση που μπορεί να χρησιμοποιηθεί για πρόβλεψη. Ονομάζουμε αυτή τη συνάρτηση `predict` και μπορούμε να της δώσουμε τυχαία δείγματα από το test set. Η `predict`, αφού εκτιμήσει το αποτέλεσμα, το εμφανίζει στην οθόνη.

4.6.4 Confusion matrix

Οι πίνακες σύγχυσης, αποτελούν ένα πολύ σύνηθες εργαλείο για την αξιολόγηση μοντέλων ταξινόμησης. Για τη δημιουργία του confusion matrix, χρησιμοποιούμε δύο βιβλιοθήκες: την mlxtend και την torchmetrics. Η mlxtend είναι βιβλιοθήκη της Python που παρέχει βοηθητικά εργαλεία για ML και Data Science και θα τη χρησιμοποιήσουμε για να σχεδιάσουμε τον πίνακα. Η torchmetrics είναι επίσης βιβλιοθήκη της Python και παρέχει έτοιμες υλοποιήσεις μετρικών αξιολόγησης για μοντέλα, ειδικά για χρήση με PyTorch.

Πρακτικά, τα βήματα που ακολουθούμε είναι δύο. Αρχικά, επιλέγουμε το μοντέλο που θέλουμε να αξιολογήσουμε και πραγματοποιούμε προβλέψεις με αυτό. Οι προβλέψεις αυτές γίνονται αξιοποιώντας τα δεδομένα του test set και αποθηκεύονται σε μια λίστα. Κατόπιν, χρησιμοποιούμε την κλάση `ConfusionMatrix` της torchmetrics, δίνοντάς της τα αποτελέσματά μας, καθώς και τις πραγματικές τιμές. Το εργαλείο αυτό θα δημιουργήσει τον πίνακα σύγχυσης, τον οποίο θα κατασκευάσουμε χρησιμοποιώντας τη συνάρτηση `plot_confusion_matrix` από τη βιβλιοθήκη mlxtend.

Οι πίνακες σύγχυσης μπορούν να μας βοηθήσουν ιδιαίτερα στη μελέτη μας. Χρησιμοποιώντας τους, είναι εφικτό να δούμε συγκεκριμένα σε ποια μουσικά είδη το εκάστοτε μοντέλο αποδίδει καλύτερα. Δημιουργούμε έναν πίνακα για κάθε ένα από τα καλύτερα μοντέλα κάθε αρχιτεκτονικής, για κάθε σειρά πειραμάτων.

4.7 Ροή εργασιών κάθε πειράματος

Έχοντας παρουσιάσει όλα τα βήματα του pipeline, συνοψίζουμε τη σειρά των βημάτων κάθε πειράματος:

- Φόρτωση απαραίτητων βιβλιοθηκών, ορισμός ντετερμινιστικής συμπεριφοράς και ρύθμιση χρήσης GPU.

- Μεταφόρτωση δεδομένων.
- Διαχωρισμός δεδομένων σε train set, validation set και test set και δημιουργία αντίστοιχων dataloaders για το καθένα.
- Οπτικοποίηση δεδομένων.
- Κατασκευή κλάσης με την αρχιτεκτονική που μελετάται.
- Υλοποίηση μοντέλου και εκπαίδευσή του.
- Απεικόνιση διαγραμμάτων με τα αποτελέσματα της εκπαίδευσης.
- Υλοποίηση νέου μοντέλου και εκπαίδευση με data augmentation.
- Σύγκριση διαγραμμάτων με τα αποτελέσματα κάθε εκπαίδευσης.
- Υλοποίηση τελικού μοντέλου και εκπαίδευσή του με early stopping.
- Αξιολόγηση τελικού μοντέλου υπολογίζοντας τα loss και accuracy στο test set.
- Χρήση του τελικού μοντέλου με την συνάρτηση `predict`.
- Κατασκευή confusion matrix.

Το παραπάνω pipeline ακολουθείται σε κάθε πείραμα και παρουσιάζεται αναλυτικά σε καθένα από τα Jupyter notebooks που εργαζόμαστε. Σε ορισμένες περιπτώσεις υπάρχει προσαρμογή του pipeline, αναλόγως τις συνθήκες. Για παράδειγμα, στην περίπτωση των αναδρομικών δικτύων, κατασκευάζουμε κλάσεις για τις αρχιτεκτονικές RNN, LSTM, GRU και αξιολογούμε τον αντίστοιχο αριθμό μοντέλων.

5. Αποτελέσματα

Στο κεφάλαιο αυτό παρουσιάζουμε τα αποτελέσματα των πειραμάτων. Η παρουσίαση των αποτελεσμάτων οργανώνεται ως εξής. Αρχικά παρουσιάζουμε τις καμπύλες εκπαίδευσης του καλύτερου μοντέλου κάθε αρχιτεκτονικής ταξινομημένες σύμφωνα με τις τέσσερεις διαφορετικές σειρές πειραμάτων. Ομοίως παρουσιάζονται και οι αντίστοιχοι πίνακες σύγχυσης. Στην παρουσίαση περιλαμβάνονται και κάποια σχόλια για τα αποτελέσματα της εκπαίδευσης με χρήση των τεχνικών Regularization. Κατόπιν ακολουθεί μια συγκριτική παρουσίαση των αποτελεσμάτων. Εκεί περιλαμβάνονται και οι υπόλοιπες μετρικές που αναφέρθηκαν ότι έχουν υπολογισθεί.

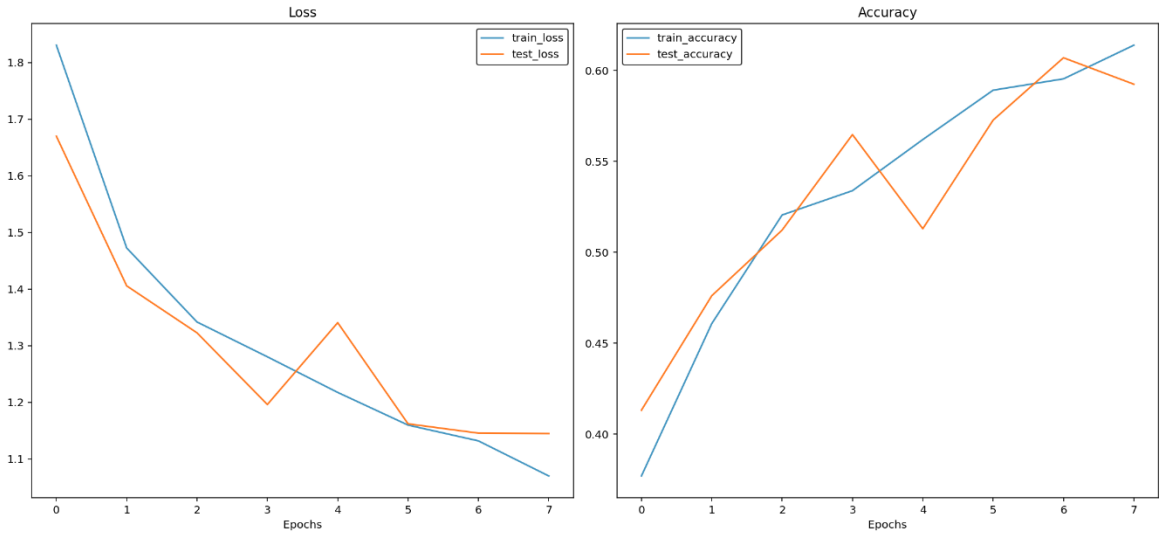
Επιπροσθέτως, κρίνεται σκόπιμο να σχολιάσουμε τη διάρκεια της εκπαίδευσης. Στις περισσότερες περιπτώσεις λοιπόν η όλη διαδικασία διήρκεσε μερικά λεπτά, ενώ στις αρχιτεκτονικές Transformer αρκετά περισσότερο. Στις περιπτώσεις αυτές, αναφέρουμε τη διάρκεια εκπαίδευσης.

5.1 Εκπαίδευση με MFCCs

Στο εν λόγω υποκεφάλαιο, παρουσιάζουμε την πρώτη σειρά πειραμάτων, όπως αυτή περιγράφεται στον Πίνακα 2. Για κάθε αρχιτεκτονική, παρατίθεται και ο αντίστοιχος σχολιασμός της όλης διαδικασίας της εκπαίδευσης.

5.1.1 Καμπύλες εκπαίδευσης

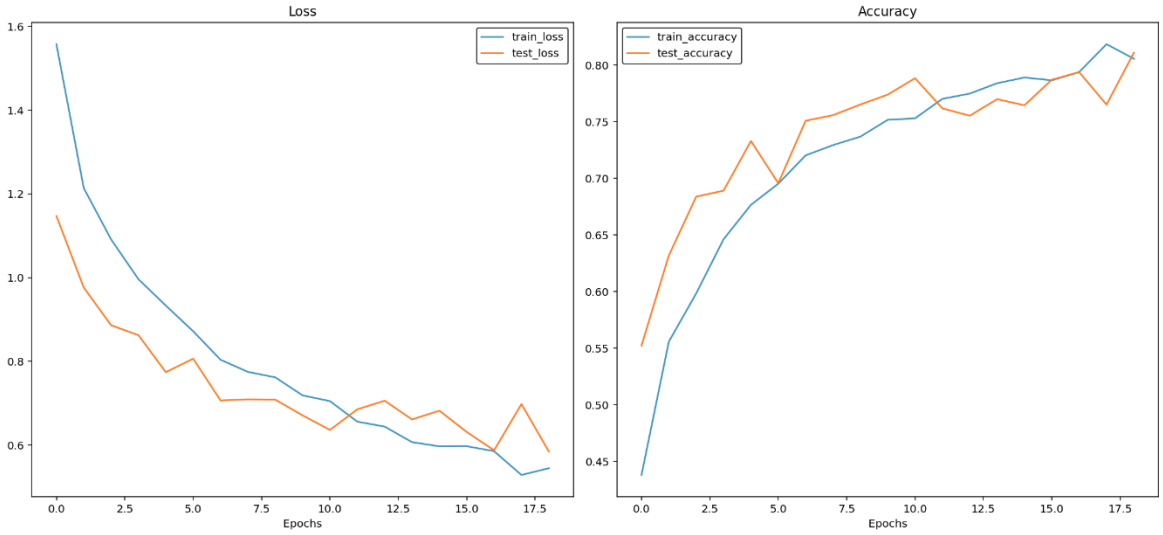
Αρχικά εκπαιδεύσαμε ένα μοντέλο αρχιτεκτονικής MLP. Το εν λόγω μοντέλο στην αρχική μας εκπαίδευση που έγινε για 30 epochs, παρουσίασε overfitting ήδη από την 5η epoch. Η χρήση data augmentation παράτεινε το φαινόμενο το οποίο στην ακόλουθη εκπαίδευση, εμφανίστηκε εντονότερα μετά την 10η epoch.



Σχήμα 11 Καμπύλες εκπαίδευσης μοντέλου MLP με MFCCs

Κοιτώντας τα αποτελέσματα κάθε epoch λεπτομερώς, επιλέξαμε να εκπαιδεύσουμε το εν λόγω μοντέλο για 7 epochs. Τα αποτελέσματα της εκπαίδευσης φαίνονται στο Σχήμα 11.

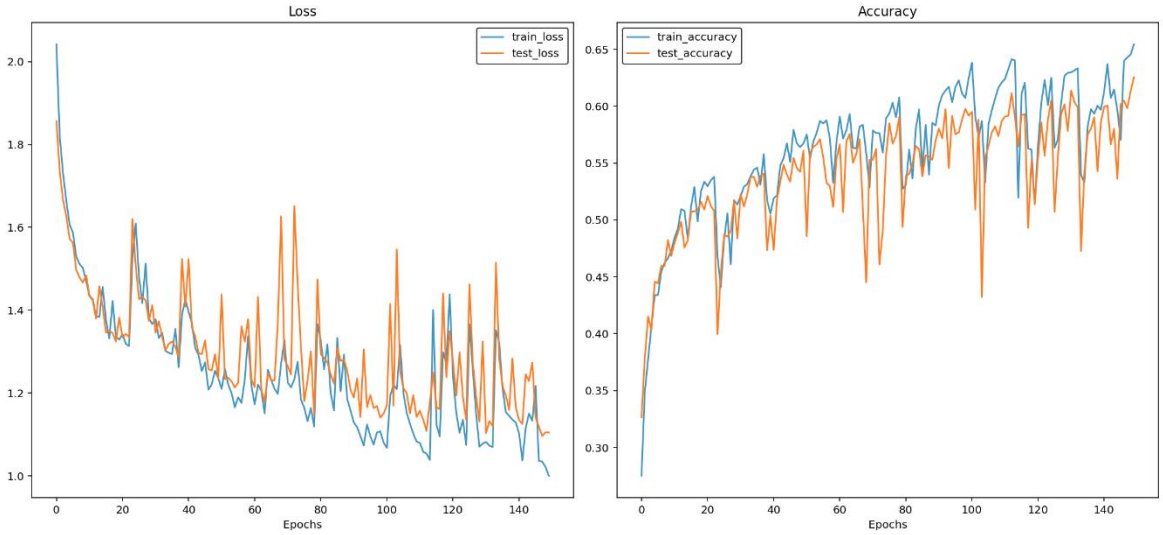
Εν συνεχεία εκπαιδεύσαμε το μοντέλο αρχιτεκτονικής CNN. Το εν λόγω μοντέλο εκπαιδεύτηκε αρχικά για 50 epochs και άρχισε να παρουσιάζει overfitting λίγο μετά τη 10η. Η χρήση data augmentation παράτεινε το overfitting και επέτρεψε το μοντέλο να φτάσει υψηλότερες τιμές accuracy. Εν τέλει έγινε εκπαίδευση για 19 epochs.



Σχήμα 12 Καμπύλες εκπαίδευσης μοντέλου CNN με MFCCs

Παρατηρούμε ότι το μοντέλο φθάνει το 80% accuracy. Τα αποτελέσματα φαίνονται στο Σχήμα 12. Η βελτίωση συγκριτικά με το μοντέλο MLP είναι εμφανής.

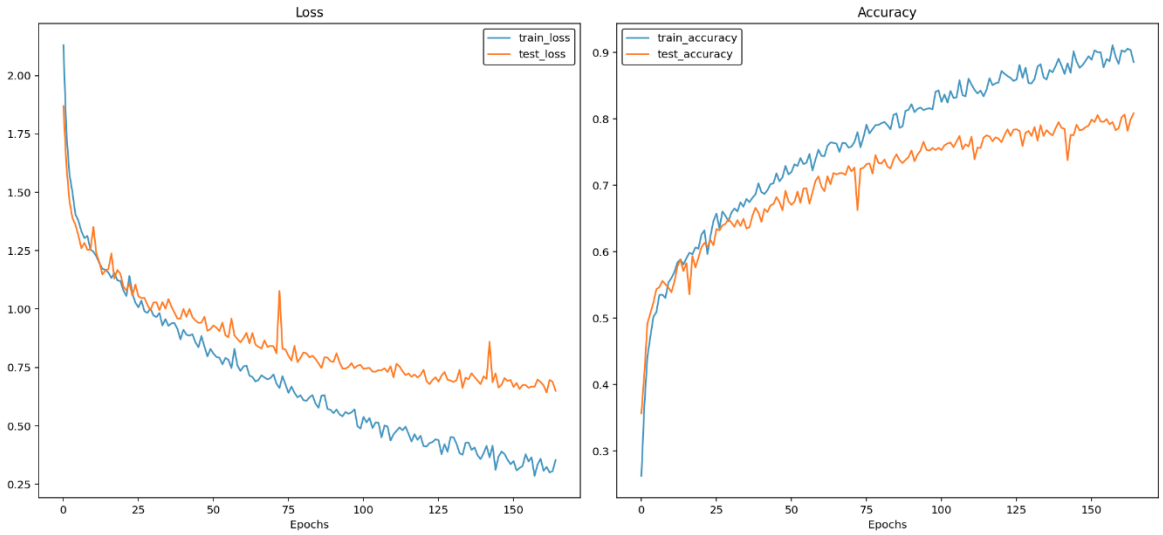
Εν συνεχεία εκπαιδεύσαμε δύο μοντέλα RNN. Το μοντέλο που χρησιμοποιεί όλα τα hidden states εμφάνισε overfitting πολύ νωρίς, ήδη από την 5η epoch, με το loss να αυξάνεται έκτοτε. Επομένως επικεντρωθήκαμε στο μοντέλο που χρησιμοποιεί μόνο το τελευταίο hidden state, καθώς εκείνο παρουσίασε καλύτερα αποτελέσματα. Το μοντέλο αυτό ευνοήθηκε από το data augmentation, καθώς έτσι έφτασε ψηλότερες τιμές accuracy, ενώ η τεχνική του επέτρεψε να εκπαιδευθεί για περισσότερες epochs.



Σχήμα 13 Καμπύλες εκπαίδευσης μοντέλου RNN με MFCCs

Το μοντέλο RNN εκπαιδεύτηκε για 150 epochs. Τα αποτελέσματα της εκπαίδευσης φαίνονται στο Σχήμα 13. Οι συνεχόμενες «κορυφές» που φαίνονται στο σχήμα συναντώνται συχνά στην εκπαίδευση των RNN. Οι Wu et al. (2018) στην έρευνά τους πάνω στη χρήση RNNs για ταξινόμηση μουσικών ειδών, συναντούν το ίδιο φαινόμενο. Οι Chang et al. (2024) επίσης παρουσιάζουν το ίδιο φαινόμενο. Οι κορυφές αυτές οφείλονται στα vanishing και exploding gradients, τα οποία το RNN εξ ορισμού δεν είναι σε θέση να διαχειριστεί.

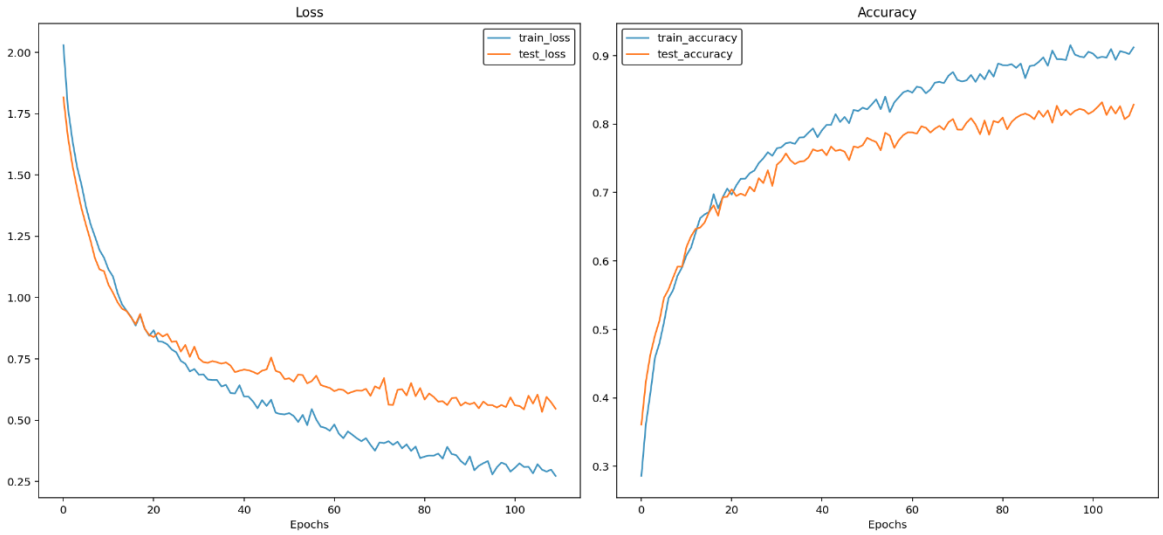
Εν συνεχεία, εκπαιδεύσαμε μοντέλα με αρχιτεκτονική LSTM. Εκπαιδεύοντας το πρώτο μοντέλο χωρίς data augmentation για 100 epochs και το δεύτερο με data augmentation για 300 epochs, παρατηρήσαμε ότι η τεχνική βελτιώνει την απόδοση του μοντέλου. Αυξήσαμε τον αριθμό των epochs αρκετά, καθώς παρατηρήσαμε πως η τεχνική ωφέλησε αρκετά το μοντέλο και θέλαμε να δούμε τη συμπεριφορά για μεγάλη τιμή epochs.



Σχήμα 14 Καμπύλες εκπαίδευσης μοντέλου LSTM με MFCCs

Εν τέλει εκπαιδεύσαμε το μοντέλο για 165 epochs. Τα αποτελέσματα της εκπαίδευσης φαίνονται στο Σχήμα 14. Οι καμπύλες παρουσιάζουν επίσης «κορυφές», όμως παρατηρούμε πως το φαινόμενο αυτό έχει μετριαστεί κατά πολύ συγκριτικά με την περίπτωση του RNN.

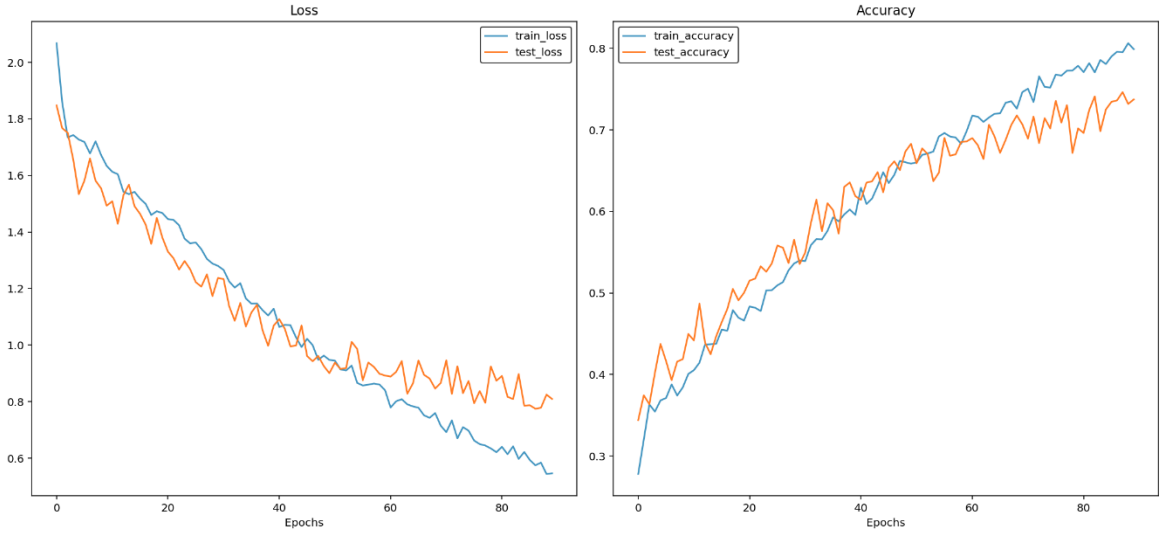
Στη συνέχεια εκπαιδεύσαμε μοντέλο GRU για 100 epochs και παρατηρήσαμε overfitting σχετικά νωρίς, κατά την 25η epoch. Το data augmentation βοήθησε σημαντικά το μοντέλο μας το οποίο εκπαιδεύτηκε για 150 epochs με το overfitting να εμφανίζεται αυτή τη φορά αρκετά μετά την 100η epoch.



Σχήμα 15 Καμπύλες εκπαίδευσης μοντέλου GRU με MFCCs

Με προσεκτική μελέτη της εξέλιξης των τιμών ανά epoch, επιλέξαμε να κάνουμε early stopping στην 110η epoch, Τα αποτελέσματα της εκπαίδευσης φαίνονται στο Σχήμα 15.

Τέλος, εκπαιδεύσαμε το μοντέλο ViT για 150 epochs, παρατηρώντας overfitting λίγο μετά την 60η epoch. Το data augmentation παράτεινε το φαινόμενο και οδήγησε σε πιο ομαλές καμπύλες.

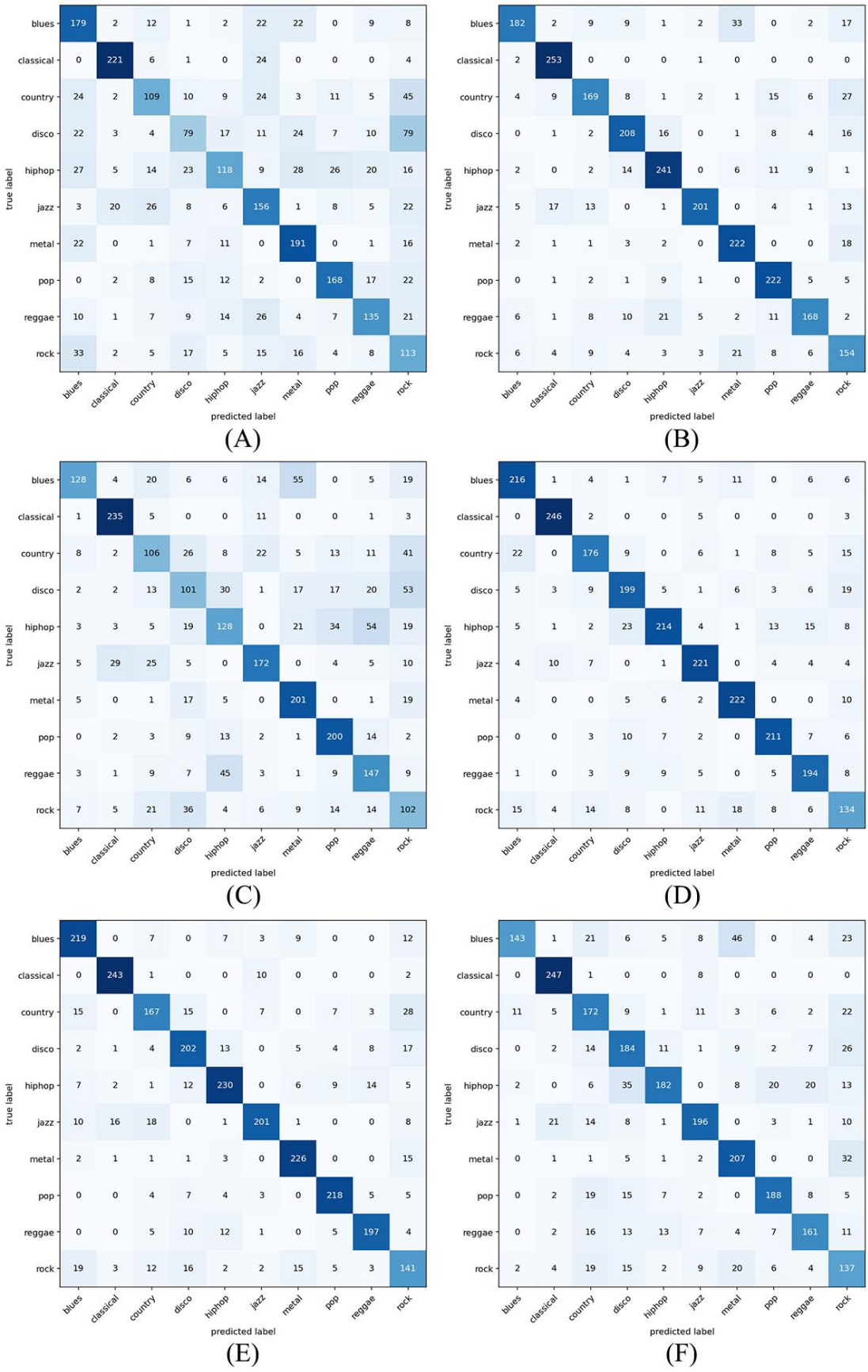


Σχήμα 16 Καμπύλες εκπαίδευσης μοντέλου ViT με MFCCs

Το early stopping επιλέχθηκε να γίνει στην 90η epoch. Τα αποτελέσματα της εκπαίδευσης φαίνονται στο Σχήμα 16, ενώ το μοντέλο εκπαιδεύτηκε για 36 λεπτά.

5.1.2 Πίνακες σύγχυσης

Στη συνέχεια παρουσιάζονται οι πίνακες σύγχυσης για κάθε ένα από τα μοντέλα που εκπαιδεύτηκαν με MFCCs ως δεδομένα εισόδου. Στους πίνακες, μπορούμε να διακρίνουμε την επίδοση κάθε μοντέλου για κάθε είδος ξεχωριστά. Βλέπουμε λοιπόν πως η επίδοση του κάθε μοντέλου, μπορεί να είναι καλύτερη σε συγκεκριμένα είδη συγκριτικά με άλλα. Στο Σχήμα 17 βλέπουμε τους αντίστοιχους πίνακες σύγχυσης. Το σχήμα 17(A) αντιστοιχεί στην αρχιτεκτονική MLP, το 17(B) στη CNN, το 17(C) στην RNN, το 17(D) στην LSTM, το 17(E) στη GRU και τέλος το 17(F) στην ViT.



Σχήμα 17 Πίνακες σύγχυσης των μοντέλων που εκπαιδεύτηκαν με MFCCs

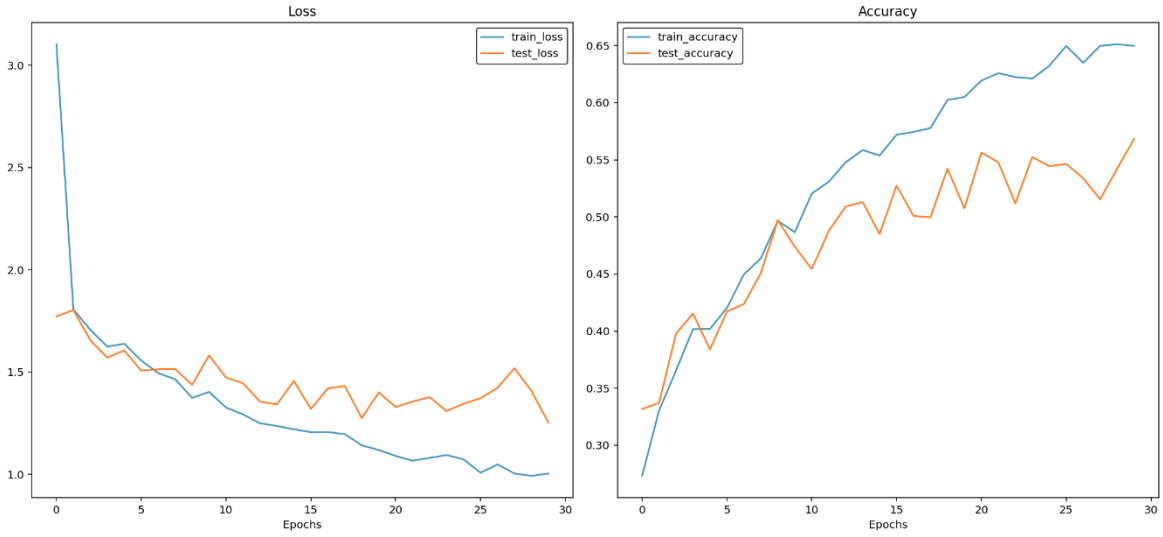
Η επίδοση των μοντέλων στους πίνακες σύγχυσης φαίνεται ότι ακολουθεί ανάλογο μοτίβο με αυτό ων καμπυλών εκπαίδευσης. Σημειώνουμε όμως ότι οι δύο μετρικές έχουν υπολογισθεί σε διαφορετικά set.

5.2 Εκπαίδευση με Mel Spectrograms

Στο συγκεκριμένο υποκεφάλαιο παρουσιάζουμε τη δεύτερη σειρά πειραμάτων, όπως αυτή περιγράφεται στον Πίνακα 2. Για κάθε αρχιτεκτονική, αντίστοιχα παρατίθεται και ο κατάλληλος σχολιασμός.

5.2.1 Καμπύλες εκπαίδευσης

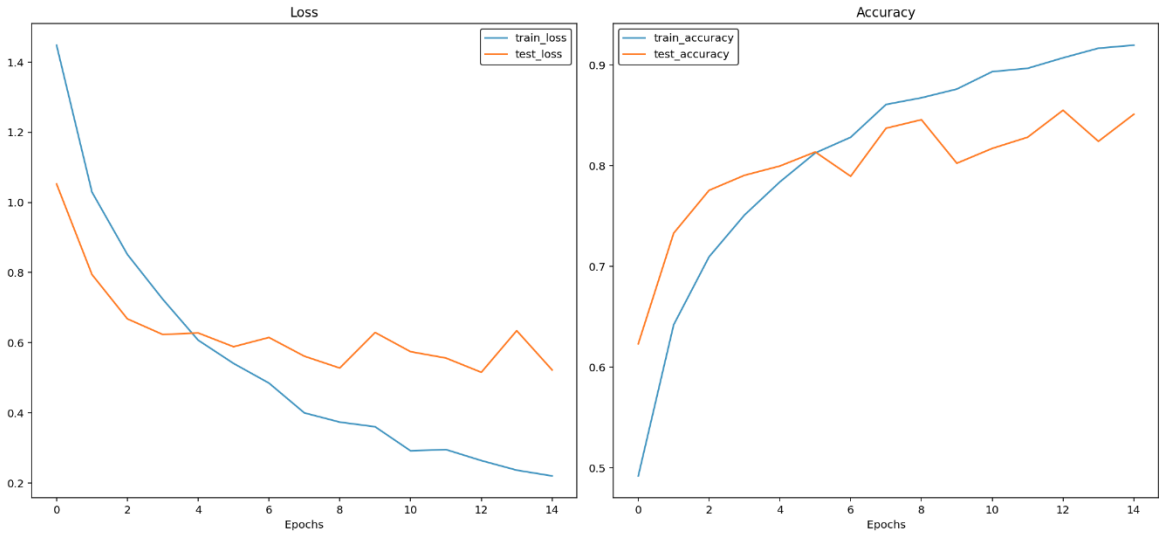
Ακολουθώντας την ίδια σειρά παρουσίασης των αρχιτεκτονικών, στην τρέχουσα σειρά ξεκινήσαμε επίσης με την εκπαίδευση μοντέλων MLP. Το μοντέλο εκπαιδεύτηκε για 50 epochs και κατά την 30η epoch παρατηρήθηκε overfitting. Η χρήση data augmentation ευνόησε την εκπαίδευση, καθώς το μοντέλο αύξησε την επίδοσή του σχεδόν κατά 10%, ενώ το overfitting άρχισε να παρατηρείται πέραν της 30ης epoch.



Σχήμα 18 Καμπύλες εκπαίδευσης μοντέλου MLP με Mel Spectrograms

Παρατηρούμε λοιπόν πολύ διαφορετική συμπεριφορά από αυτή των MFCCs, καθώς εδώ μπορούμε να εκπαιδεύσουμε το μοντέλο για αρκετά περισσότερες epochs, ενώ οι επιδόσεις είναι ψηλότερες. Στο Σχήμα 18 φαίνονται οι αντίστοιχες καμπύλες εκπαίδευσης του τελευταίου μοντέλου που εκπαιδεύτηκε με data augmentation και early stopping στις 30 epochs.

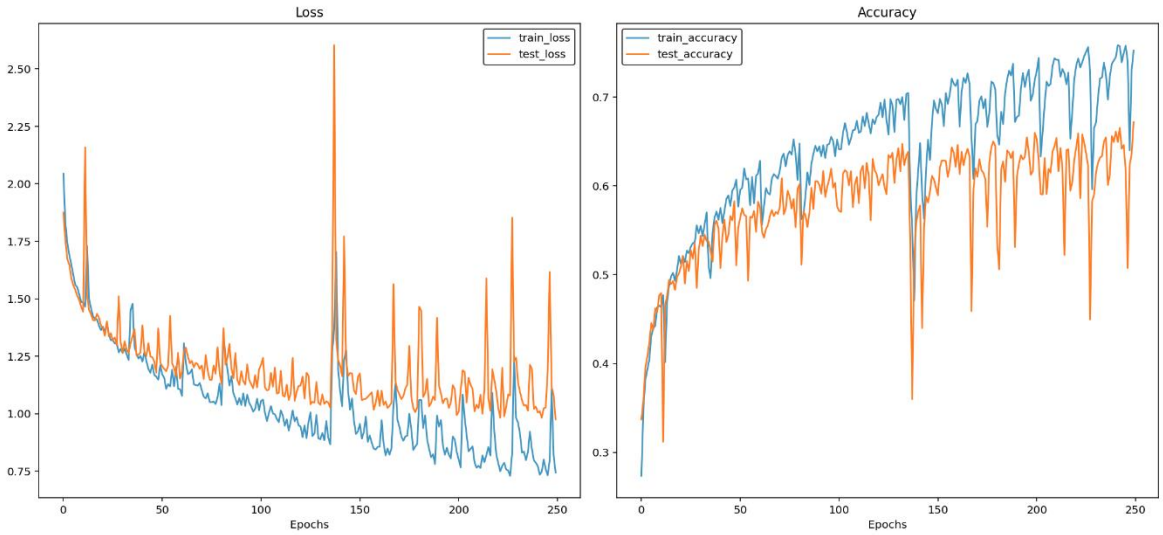
Στη συνέχεια εκπαιδεύσαμε ένα μοντέλο CNN αρχικά για 30 epochs με το overfitting να εμφανίζεται νωρίς, στις 5 epochs. Η χρήση data augmentation βελτίωσε τις επιδόσεις, ενώ παράτεινε το overfitting για περίπου 5 epochs ακόμα.



Σχήμα 19 Καμπύλες εκπαίδευσης μοντέλου CNN με Mel Spectrograms

Για το τελικό μοντέλο χρησιμοποιήσαμε data augmentation και early stopping στις 15 epochs. Τα αποτελέσματα της εκπαίδευσης φαίνονται στο σχήμα 19.

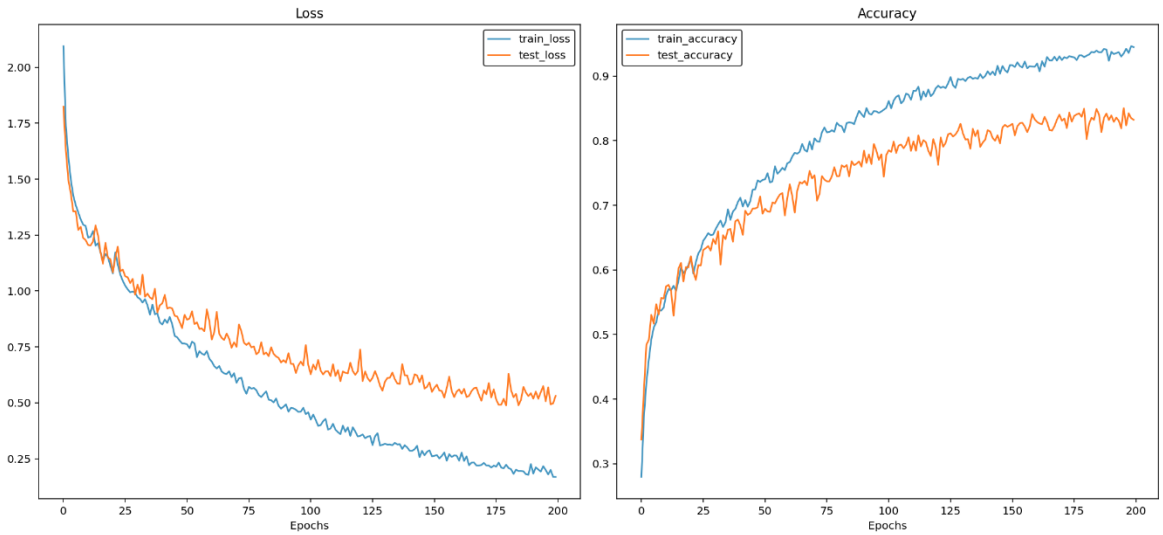
Στη συνέχεια εκπαιδεύσαμε δύο μοντέλα RNN. Το πρώτο χρησιμοποιεί μόνο το τελευταίο hidden state, ενώ το δεύτερο χρησιμοποιεί όλα τα hidden states. Για μια ακόμη φορά το δεύτερο μοντέλο παρουσίασε απότομο overfitting στις πρώτες κιόλας epochs, επομένως συνεχίσαμε τα πειράματα με την πρώτη αρχιτεκτονική. Το φαινόμενο αυτό γενικά είναι συχνό στη βιβλιογραφία και προτιμάται η χρήση RNNs με το τελευταίο hidden state. Ειδικά σε περιπτώσεις που το μέγεθος του dataset δεν είναι τόσο μεγάλο. Το πρώτο μοντέλο λοιπόν εκπαιδεύτηκε για 300 epochs χωρίς data augmentation και κατόπιν για 300 epochs με data augmentation. Στη δεύτερη περίπτωση, η απόκλιση μεταξύ του train loss και validation loss ήταν πολύ μικρότερη, γεγονός που δείχνει ότι η τεχνική βοήθησε.



Σχήμα 20 Καμπύλες εκπαίδευσης μοντέλου RNN με Mel Spectrograms

Εν τέλει εκπαιδεύσαμε το τελικό μοντέλο για 250 epochs. Τα αποτελέσματα της εκπαίδευσης φαίνονται στο Σχήμα 20. Όπως και στην περίπτωση των MFCCs, έτσι κι εδώ παρατηρούμε τις χαρακτηριστικές «κορυφές» που οφείλονται στα vanishing/exploding gradients.

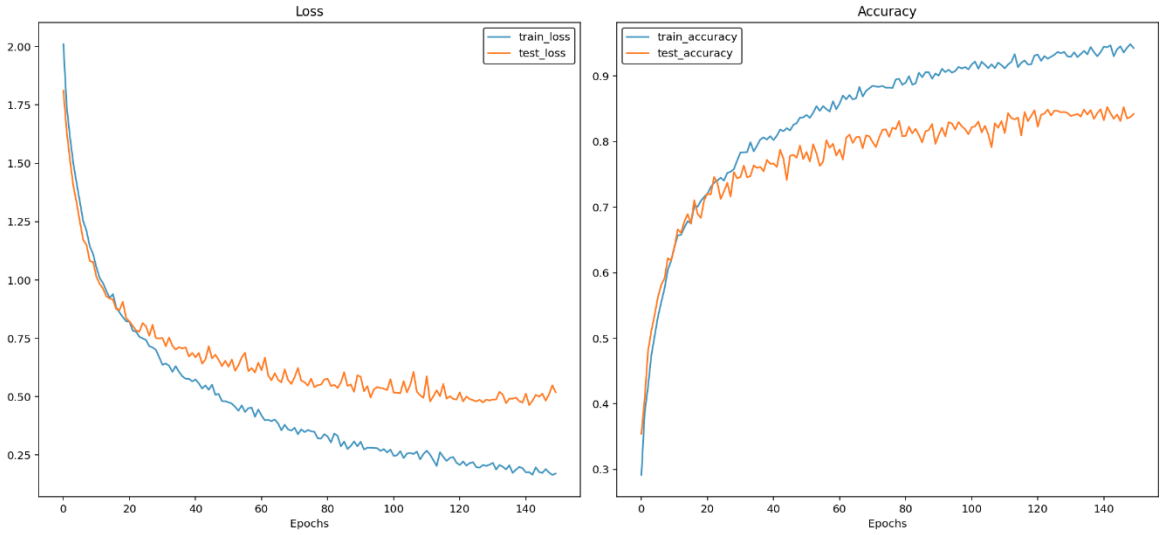
Στη συνέχεια εκπαιδεύσαμε ένα μοντέλο αρχιτεκτονικής LSTM για 100 epochs χωρίς data augmentation. Κατόπιν προχωρήσαμε στην εκπαίδευση άλλου μοντέλου για 300 epochs με data augmentation. Συγκρίνοντας τα αποτελέσματα των δύο διαδικασιών, είδαμε πως και εδώ η τεχνική βοήθησε. Για το τελικό μοντέλο κάναμε early stopping στις 200 epochs, κρίνοντας από τα αποτελέσματα της προηγούμενης εκπαίδευσης.



Σχήμα 21 Καμπύλες εκπαίδευσης μοντέλου LSTM με Mel Spectrograms

Αντίστοιχα και σε αυτή την περίπτωση, διαπιστώνουμε πως το LSTM ως βελτίωση του RNN, αντιμετωπίζει το πρόβλημα των vanishing/exploding gradients. Τα αποτελέσματα της εκπαίδευσης του τελικού μοντέλου LSTM, φαίνονται στο Σχήμα 21.

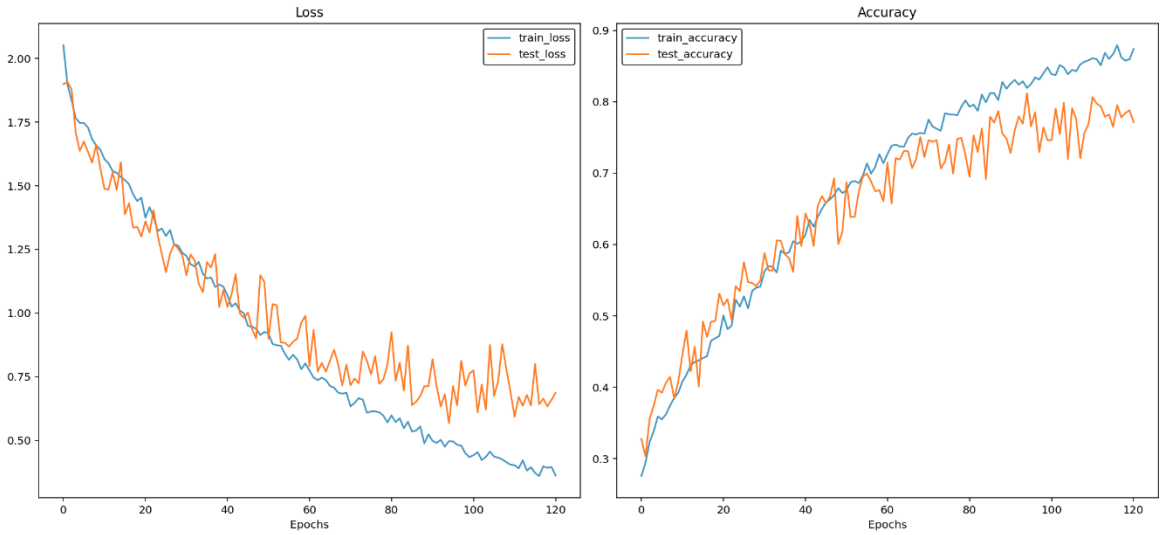
Το τελευταίο αναδρομικό δίκτυο αυτής της σειράς πειραμάτων, είναι το GRU. Αρχικά εκπαιδεύσαμε ένα μοντέλο της αρχιτεκτονικής αυτής για 100 epochs, παρατηρώντας overfitting αρκετά νωρίς, στις 20 epochs. Η χρήση data augmentation βοήθησε και εδώ, αφού εκπαιδεύσαμε ένα δεύτερο μοντέλο για 300 epochs και είδαμε μεγάλη διαφορά. Το overfitting σε αυτή την περίπτωση άρχισε να παρατηρείται πιο έντονα μετά τη 200η epoch, ενώ οι επιδόσεις ήταν εμφανώς καλύτερες.



Σχήμα 22 Καμπύλες εκπαίδευσης μοντέλου GRU με Mel Spectrograms

Λεπτομερέστερη επισκόπηση της εκπαίδευσης του δεύτερου μοντέλου, μας οδήγησε στο συμπέρασμα να κάνουμε early stopping στις 150 epochs. Οι επιδόσεις του μοντέλου στην εκπαίδευση, φαίνονται στο Σχήμα 22.

Τελευταίο μοντέλο και σε αυτή τη σειρά πειραμάτων είναι το ViT. Η εκπαίδευση του πρώτου μοντέλου αυτής της αρχιτεκτονικής έγινε για 150 epochs και παρατηρήθηκε overfitting ήδη μετά τις 30 epochs. Η χρήση data augmentation βελτίωσε τις επιδόσεις σχεδόν κατά 10%, ενώ η εκπαίδευση του δεύτερου μοντέλου, έγινε επίσης για 150 epochs.

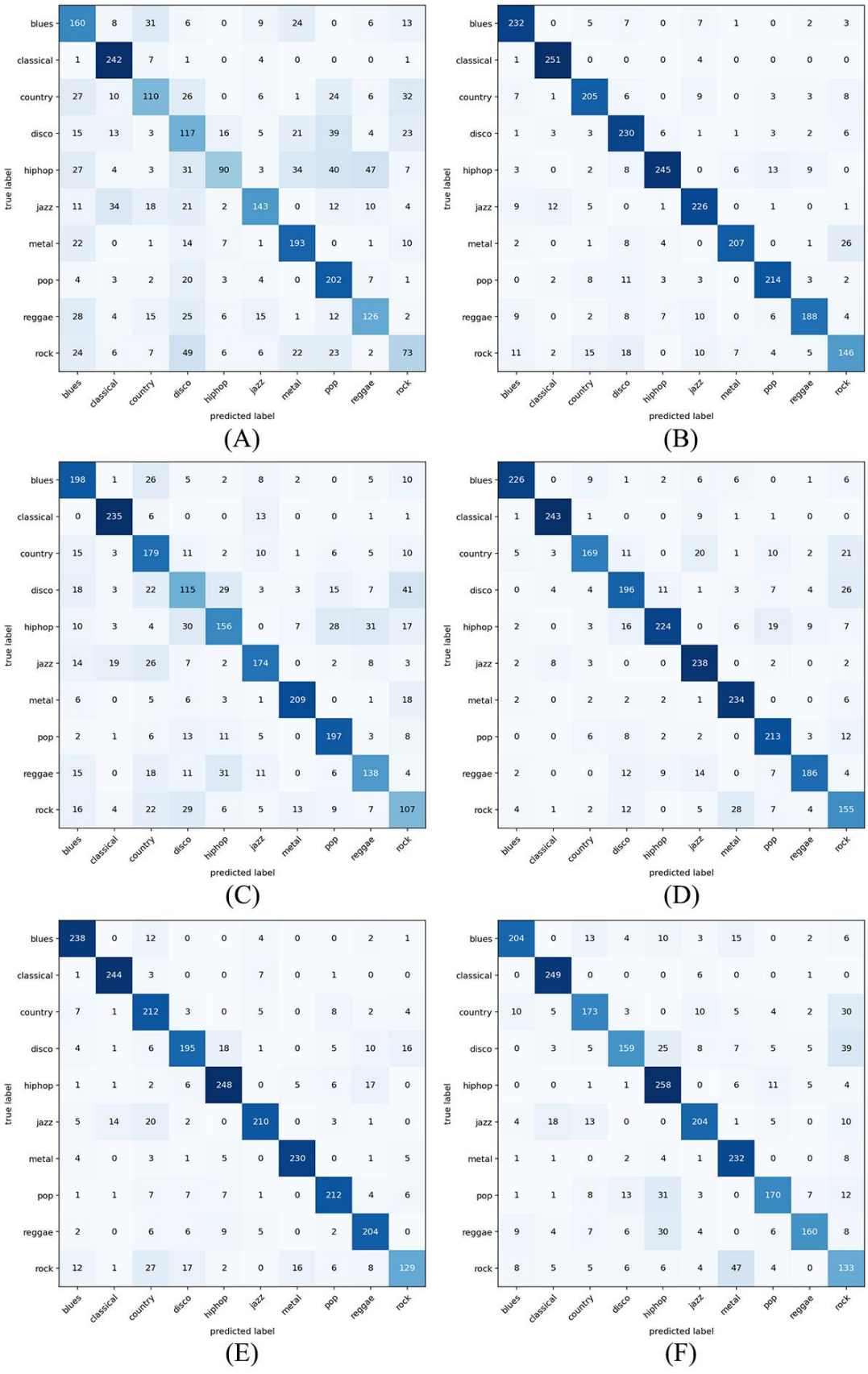


Σχήμα 23 Καμπύλες εκπαίδευσης μοντέλου ViT με Mel Spectrograms

Για την εκπαίδευση του τελικού μοντέλου αυτής της αρχιτεκτονικής, κάναμε early stopping στις 121 epochs. Οι επιδόσεις του μοντέλου αυτού στην εκπαίδευση, φαίνονται στο Σχήμα 23, ενώ η διάρκεια της εκπαίδευσης ήταν σχεδόν 50 λεπτά.

5.2.2 Πίνακες σύγχυσης

Στη συνέχεια παρουσιάζουμε τους πίνακες σύγχυσης για την εκπαίδευση με Mel Spectrograms της 2ης σειράς πειραμάτων. Στο Σχήμα 18 βλέπουμε τους αντίστοιχους πίνακες σύγχυσης. Το σχήμα 24(A) αντιστοιχεί στην αρχιτεκτονική MLP, το 24(B) στη CNN, το 24(C) στην RNN, το 24(D) στην LSTM, το 24(E) στη GRU και τέλος το 24(F) στην ViT. Παρατηρούμε ομοίως να ακολουθείται αντίστοιχο μοτίβο επιδόσεων με αυτό που φαίνεται στις καμπύλες εκπαίδευσης των μοντέλων.



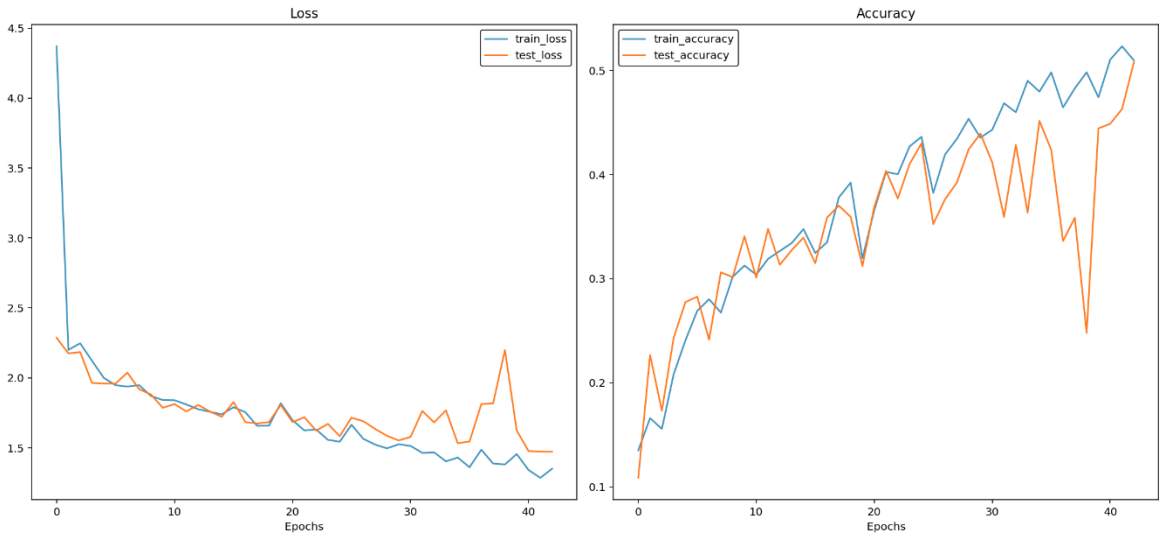
Σχήμα 24 Πίνακες σύγχυσης των μοντέλων που εκπαιδεύτηκαν με Mel Spectrograms στη 2η σειρά πειραμάτων

5.3 Εκπαίδευση με πρότυπο AST

Στην επόμενη σειρά πειραμάτων χρησιμοποιούμε Mel Spectrograms, αλλά αυτή την φορά η ανάλυση στο επίπεδο του χρόνου είναι καλύτερη. Παρουσιάζουμε τα αποτελέσματα των καμπυλών εκπαίδευσης και των πινάκων σύγχυσης για την 3η σειρά πειραμάτων του Πίνακα 2, όπως αυτή έχει περιγραφεί σε προηγούμενα υποκεφάλαια.

5.3.1 Καμπύλες εκπαίδευσης

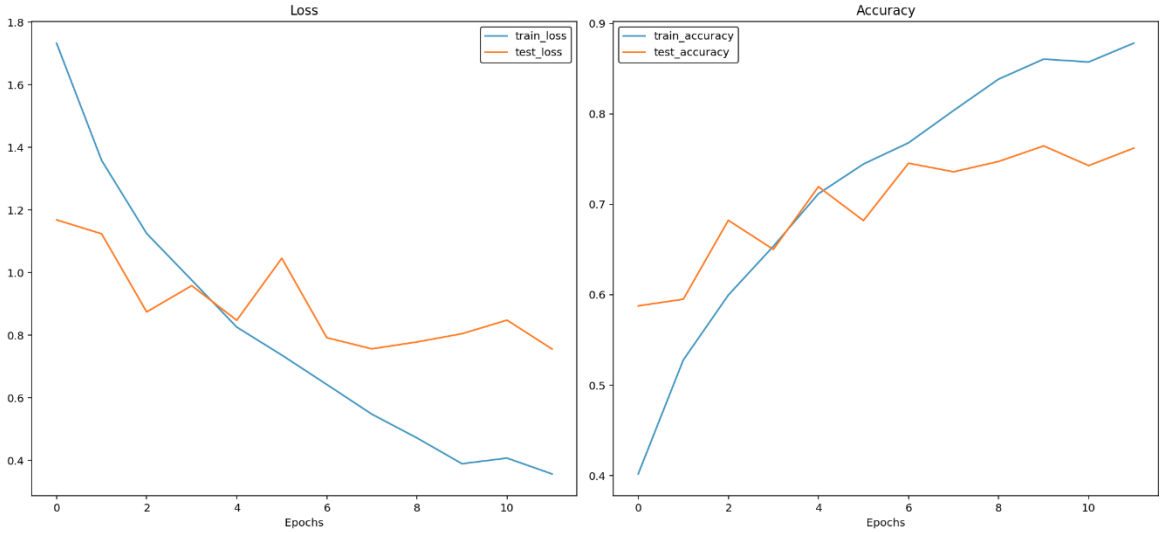
Το πρώτο μοντέλο που παρουσιάζουμε είναι το μοντέλο αρχιτεκτονικής MLP. Το μοντέλο αυτό αρχικά εκπαιδεύτηκε για 50 epochs χωρίς data augmentation και οι καμπύλες του train loss και validation loss άρχισαν να αποκλίνουν κατά την 20η epoch. Το data augmentation παράτεινε το overfitting για περίπου 15 epochs, όπως φάνηκε στο 2ο μοντέλο το οποίο εκπαιδεύσαμε για 100 epochs. Ενδιαφέρον παρουσίασε το γεγονός ότι και εδώ άρχισαν να φαίνονται οι «κορυφές» που μέχρι τώρα παρατηρούσαμε στα RNN μοντέλα και αποτελούν ένδειξη vanishing/exploding gradients.



Σχήμα 25 Καμπύλες εκπαίδευσης μοντέλου MLP με πρότυπο AST

Το τελευταίο μοντέλο αυτής της αρχιτεκτονικής, εκπαιδεύτηκε για 43 epochs ώστε να αποφύγουμε τα επαναλαμβανόμενα vanishing/exploding gradients, αλλά και το overfitting. Η μεταβολή των τιμών loss και accuracy φαίνεται στο Σχήμα 25.

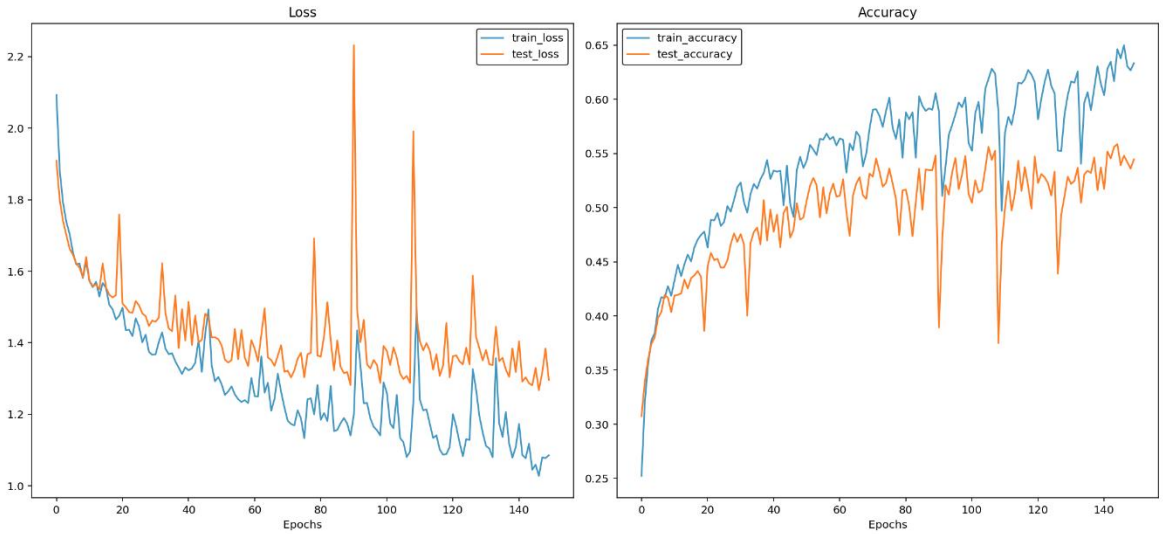
Στη συνέχεια εκπαιδεύουμε το πρώτο μοντέλο της αρχιτεκτονικής CNN για αυτή τη σειρά πειραμάτων. Η εκπαίδευση γίνεται για 100 epochs, αλλά ήδη από τις πρώτες 10 παρατηρείται πολύ μεγάλη απόκλιση μεταξύ τιμών του train loss και validation loss.



Σχήμα 26 Καμπύλες εκπαίδευσης μοντέλου CNN με πρότυπο AST

Η χρήση data augmentation βελτιώνει πολύ λίγο την κατάσταση, ενώ τελικά επιλέγεται early stopping στις 12 epochs. Τα αποτελέσματα της εκπαίδευσης του τελευταίου μοντέλου CNN φαίνονται στο Σχήμα 26.

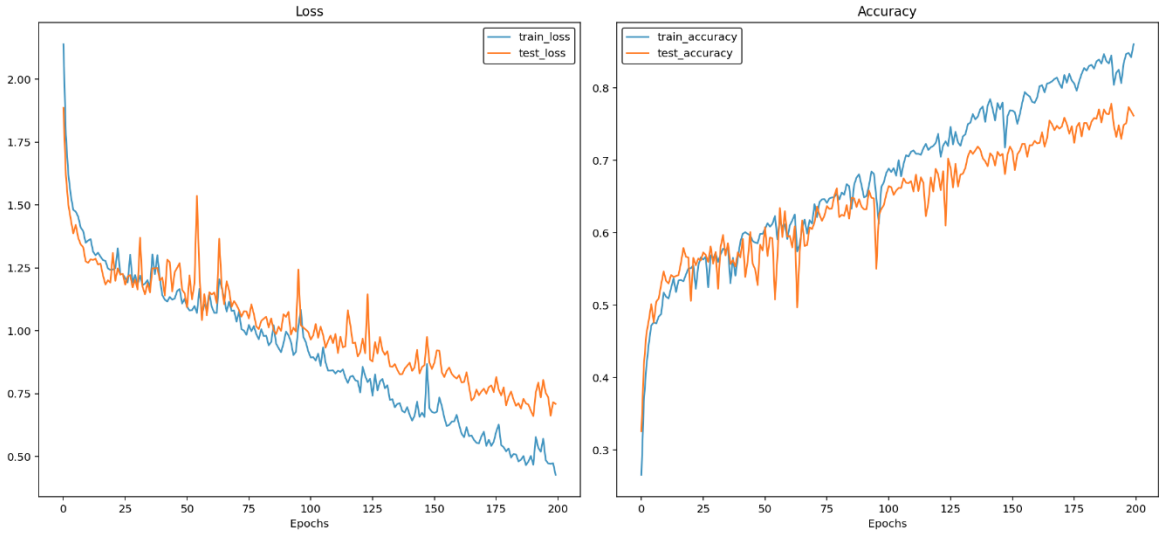
Στη συνέχεια ακολουθεί η εκπαίδευση μοντέλου αρχιτεκτονικής RNN. Για μια ακόμα φορά προτιμούμε το μοντέλο που χρησιμοποιεί το τελευταίο hidden state για τους ίδιους λόγους με τις προηγούμενες σειρές πειραμάτων. Αρχικά εκπαιδεύουμε το μοντέλο για 300 epochs παρατηρώντας ότι εμφανίζεται overfitting κατά την 70η epoch. Το data augmentation μειώνει τη διαφορά των τιμών μεταξύ train set και validation set και εν τέλει κάνουμε early stopping στις 150 epochs.



Σχήμα 27 Καμπύλες εκπαίδευσης μοντέλου RNN με πρότυπο AST

Η συμπεριφορά του μοντέλου φαίνεται να είναι εφάμιλλη των προηγούμενων περιπτώσεων. Τα αποτελέσματα της εκπαίδευσης του RNN φαίνονται στο Σχήμα 27.

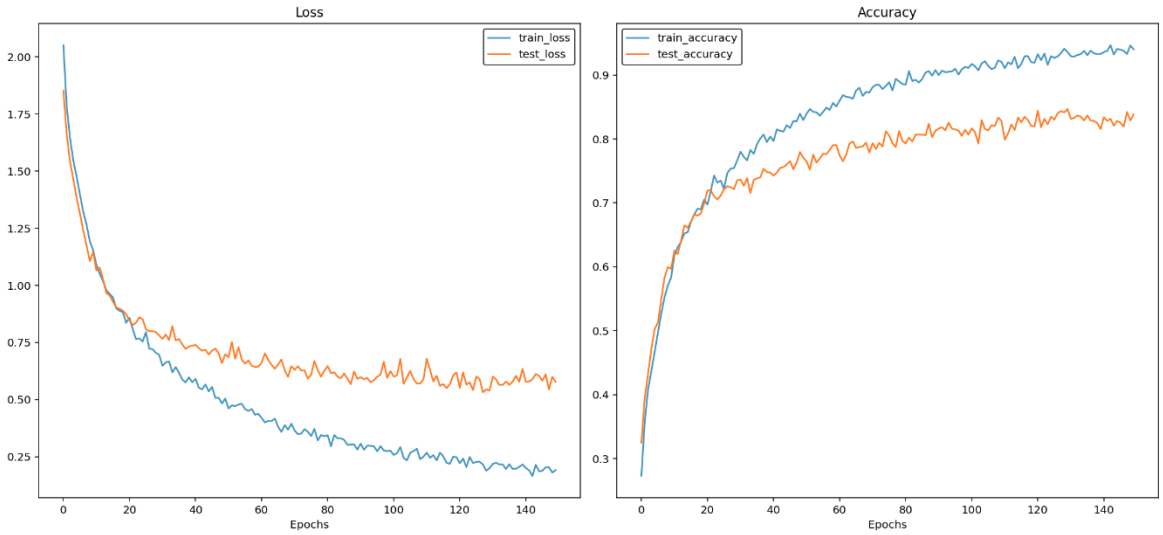
Το μοντέλο LSTM αρχικά εκπαιδεύεται για 100 epochs με τις τιμές του train loss να αποκλίνουν από αυτές του validation loss ξεκινώντας λίγο πριν την 40η epoch. Η εκπαίδευση με data augmentation διαρκεί για 500 epochs, καθώς η τεχνική φαίνεται ότι έφερε πολύ καλύτερη σύγκλιση μεταξύ των τιμών.



Σχήμα 28 Καμπύλες εκπαίδευσης μοντέλου LSTM με πρότυπο AST

Το τελικό μοντέλο LSTM εκπαιδεύεται για 200 epochs ξεπερνώντας το 70% πιστότητας. Τα αποτελέσματα διακρίνονται στο Σχήμα 28.

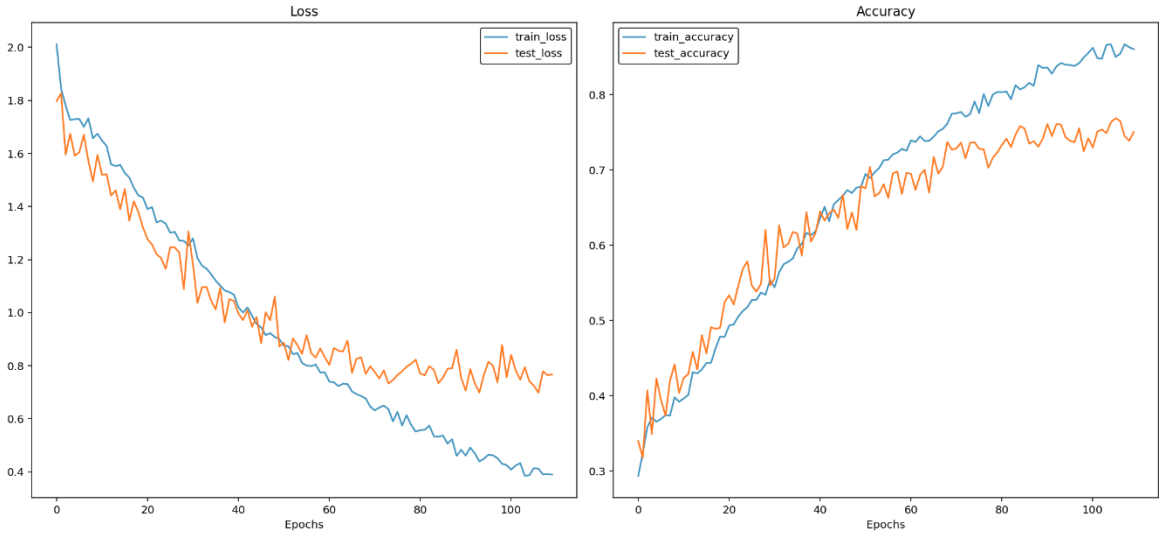
Το μοντέλο GRU που ακολουθεί, αρχικά εκπαιδεύεται για 100 epochs, όμως και σε αυτή την περίπτωση παρατηρείται απότομα overfitting σχετικά νωρίς, κατά την 20η epoch. Εδώ, οι τιμές του validation loss φαίνεται να αυξάνονται σχετικά αρκετά συγκριτικά με άλλα μοντέλα. Το data augmentation βοηθά σημαντικά σε αυτή τη συμπεριφορά, καθώς εκπαιδεύουμε ένα δεύτερο μοντέλο για 300 epochs με το validation loss να παραμένει σχετικά στάσιμο από ένα σημείο και μετά, αλλά να μην αυξάνεται.



Σχήμα 29 Καμπύλες εκπαίδευσης μοντέλου GRU με πρότυπο AST

Για το τελευταίο μοντέλο GRU, κάνουμε early stopping μετά από 150 epochs. Το μοντέλο όπως φαίνεται και στο Σχήμα 29 ξεπερνά το 80% accuracy στο validation set.

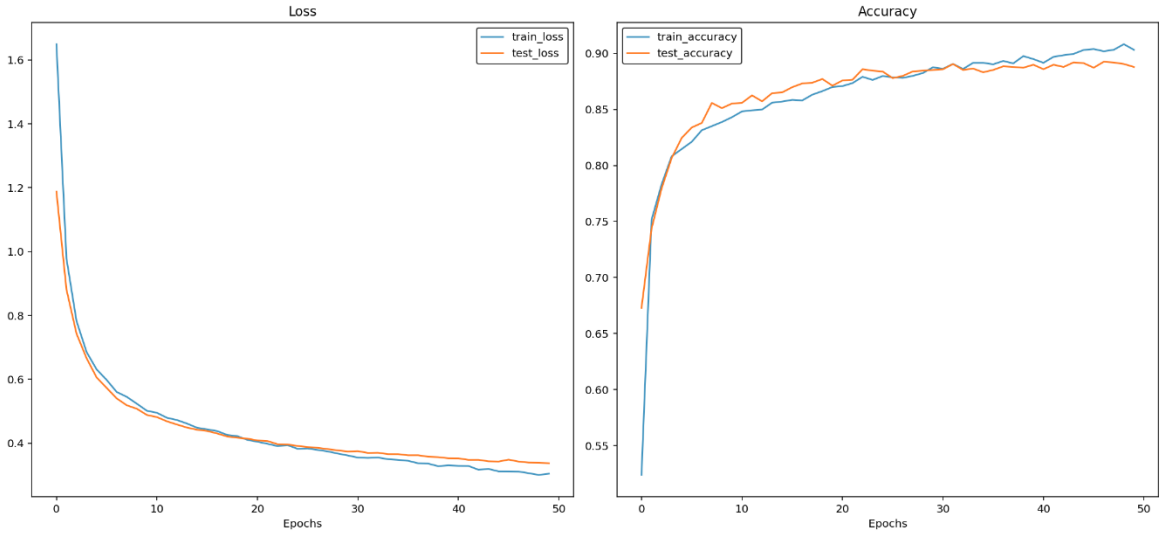
Το μοντέλο ViT αρχικά εκπαιδεύεται για 150 epochs με τα Mel Spectrograms του πρότυπου AST. Μετά την 20η epoch παρατηρείται overfitting το οποίο όμως διορθώνεται με χρήση data augmentation. Η δεύτερη εκπαίδευση λοιπόν διαρκεί 250 epochs, όμως λίγο μετά την 60η epoch η τιμή του validation loss φαίνεται να σταθεροποιείται.



Σχήμα 30 Καμπύλες εκπαίδευσης μοντέλου ViT με πρότυπο AST

Εν τέλει πραγματοποιούμε early stopping στην 110η epoch όπου το μοντέλο φτάνει τιμή 75% accuracy. Τα αποτελέσματα, φαίνονται στο Σχήμα 30. Το μοντέλο εκπαιδεύθηκε για περίπου 45 λεπτά.

Σε αυτή τη σειρά πειραμάτων, δεδομένου ότι έχουμε δεδομένα εισόδου με πρότυπο AST, έχουμε την ευκαιρία να εκπαιδεύσουμε πρώτη φορά μοντέλο AST στα πειράματά μας. Αρχικά λοιπόν εκπαιδεύουμε το μοντέλο για 30 epochs και το μοντέλο φαίνεται ότι αγγίζει τιμές accuracy που φθάνουν το 89% στο validation set, ενώ δεν παρατηρείται overfitting. Οι τιμές όμως παραμένουν σχετικά στάσιμες. Κατόπιν εκπαιδεύουμε το μοντέλο με data augmentation για να ελέγξουμε αν θα βελτιωθεί η απόδοση μετά από 50 epochs επίσης δεν παρατηρείται overfitting ενώ οι τιμές δε βελτιώνονται ιδιαίτερα.

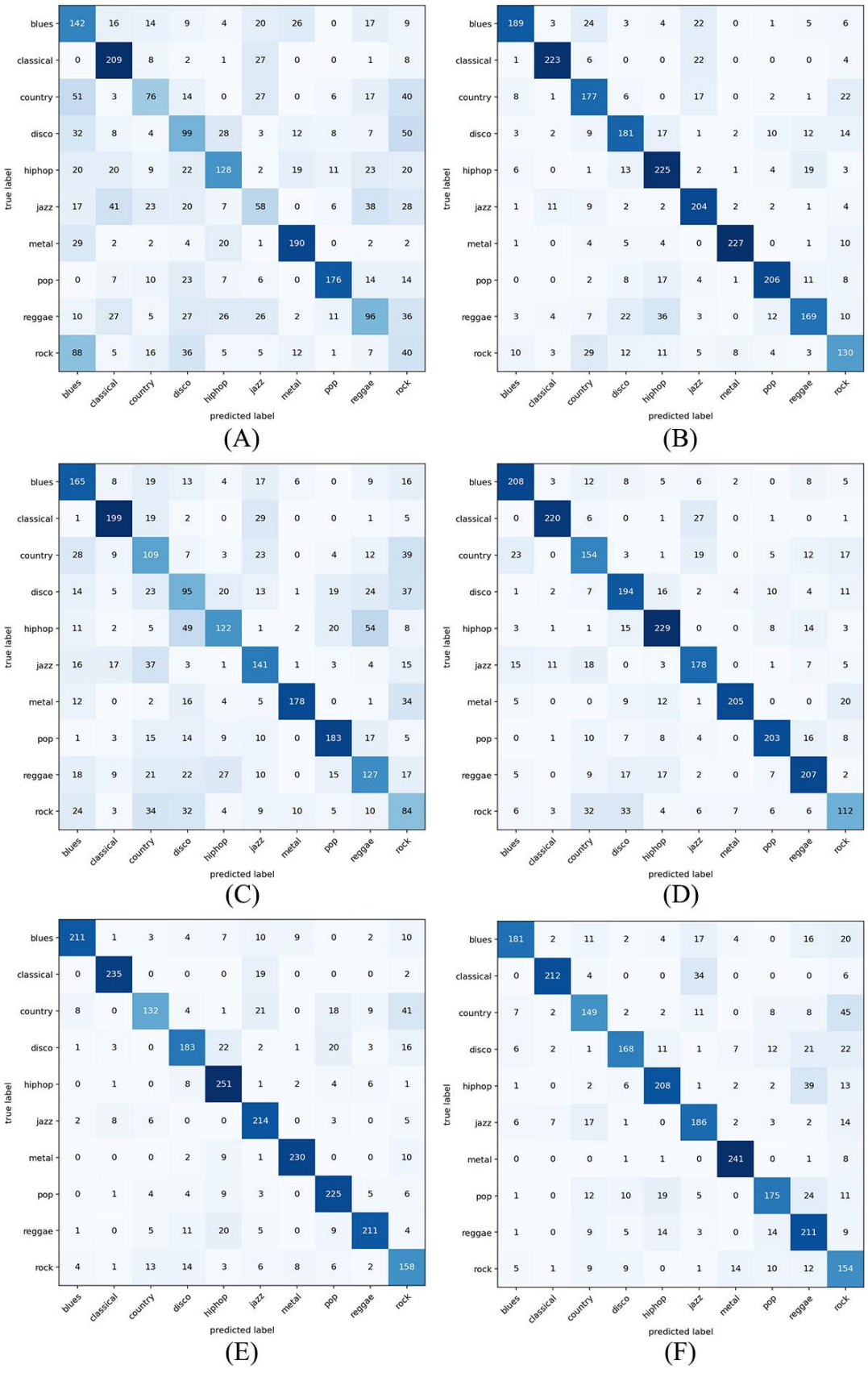


Σχήμα 31 Καμπύλες εκπαίδευσης μοντέλου AST για δείγματα διάρκειας 3ων δευτερολέπτων

Δεδομένου ότι δεν παρατηρείται overfitting αλλά οι τιμές που το μοντέλο φθάνει είναι ελαφρώς καλύτερες στην εκπαίδευση με data augmentation, δεν εκπαιδεύουμε άλλο μοντέλο με early stopping. Κρατάμε το 2ο μοντέλο, τα αποτελέσματα της εκπαίδευσης του οποίου φαίνονται στο Σχήμα 31. Η εκπαίδευση του εν λόγω μοντέλου διήρκεσε σχεδόν 4 ώρες και 25 λεπτά.

5.3.2 Πίνακες σύγχυσης

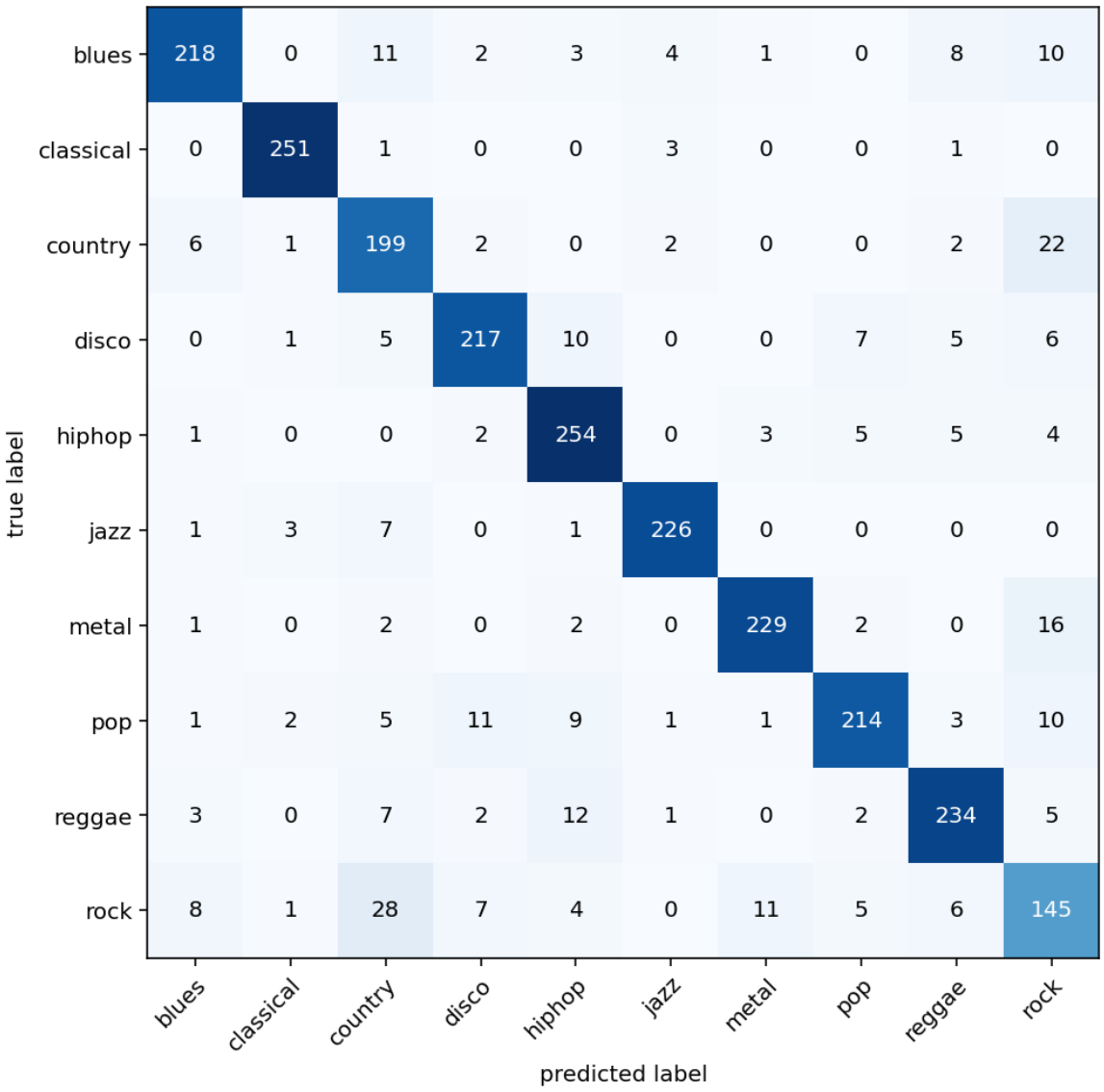
Εφόσον παρουσιάσαμε τα αποτελέσματα της εκπαίδευσης, παρουσιάζουμε τους πίνακες σύγχυσης για τις 6 πρώτες αρχιτεκτονικές των οποίων την απόδοση εκτιμήσαμε στο test set μέσω αυτής της μετρικής. Το σχήμα 32(A) αντιστοιχεί στην αρχιτεκτονική MLP, το 32(B) στη CNN, το 32(C) στην RNN, το 32(D) στην LSTM, το 32(E) στη GRU και τέλος το 32(F) στην ViT.



Σχήμα 32 Πίνακες σύγχυσης των μοντέλων που εκπαιδεύτηκαν με Mel Spectrograms στην 3η σειρά πειραμάτων

Στο Σχήμα 32(A) παρατηρούμε για πρώτη φορά κάποιο μοντέλο να μην εκτιμά σωστά την πλειοψηφία των δειγμάτων κάποιου είδους. Συγκεκριμένα, το μοντέλο MLP φαίνεται ότι συγχέει τα δείγματα rock με αυτά των blues.

Στη συνέχεια θα παρουσιάσουμε το confusion matrix του μοντέλου AST που φαίνεται πως έχει αποδώσει καλύτερα από κάθε άλλο μοντέλο έως τώρα.



Σχήμα 33 Πίνακας σύγχυσης του μοντέλου AST που εκπαιδεύεται με δείγματα 3ων δευτερολέπτων

Παρατηρούμε πως το μοντέλο AST αποδίδει εξαιρετικά συγκριτικά με άλλα μοντέλα σε άλλες περιπτώσεις. Στο Σχήμα 33 φαίνεται πως το μοντέλο συγχέει κυρίως τα rock δείγματα με αυτά της country. Παράλληλα, σε είδη όπως η κλασσική, κάνει λάθος εκτίμηση μόνο 4

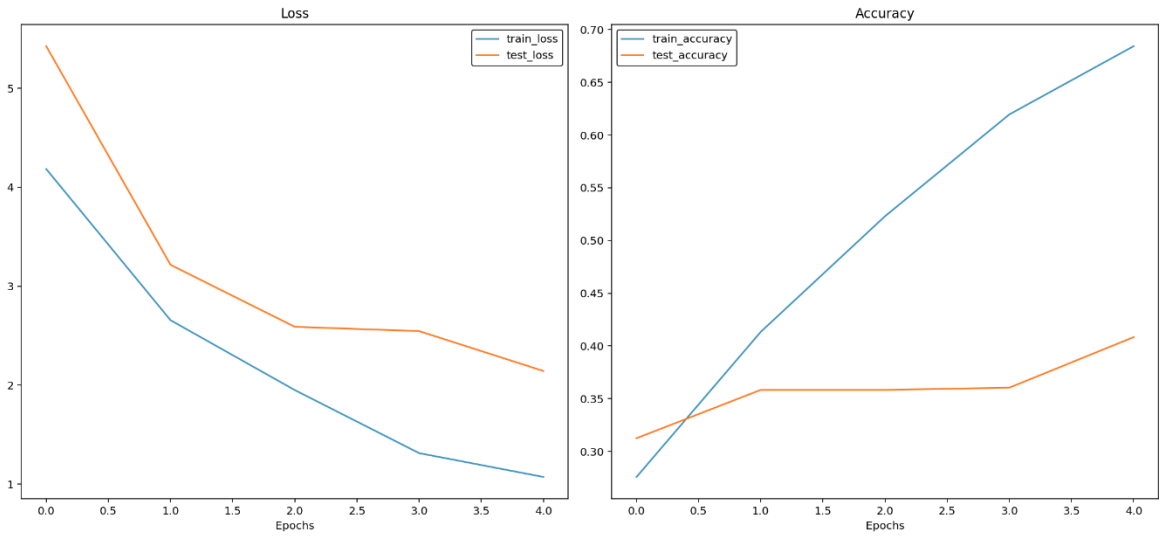
φορές. Παρόμοια επίδοση στα είδη βλέπουμε στους περισσότερους πίνακες σύγχυσης έως τώρα. Συνολικά όμως, το AST φαίνεται πως αποδίδει πολύ καλύτερα.

5.4 Εκπαίδευση με πρότυπο AST μικρότερης τμηματοποίησης

Σε αυτό το κεφάλαιο παρουσιάζουμε τα αποτελέσματα της 4ης σειράς πειραμάτων όπως αυτή παρουσιάζεται στον Πίνακα 2. Ουσιαστικά έχουμε και πάλι Mel Spectrograms στο πρότυπο του AST, όμως αυτή τη φορά έχουμε δείγματα 10 δευτερολέπτων. Σημαντική σημείωση είναι πως σε αυτή την περίπτωση έχουμε dataloaders με μικρότερο πλήθος δεδομένων. Επομένως ο αντίστοιχος αριθμός δειγμάτων στους πίνακες σύγχυσης θα είναι μικρότερος.

5.4.1 Καμπύλες εκπαίδευσης

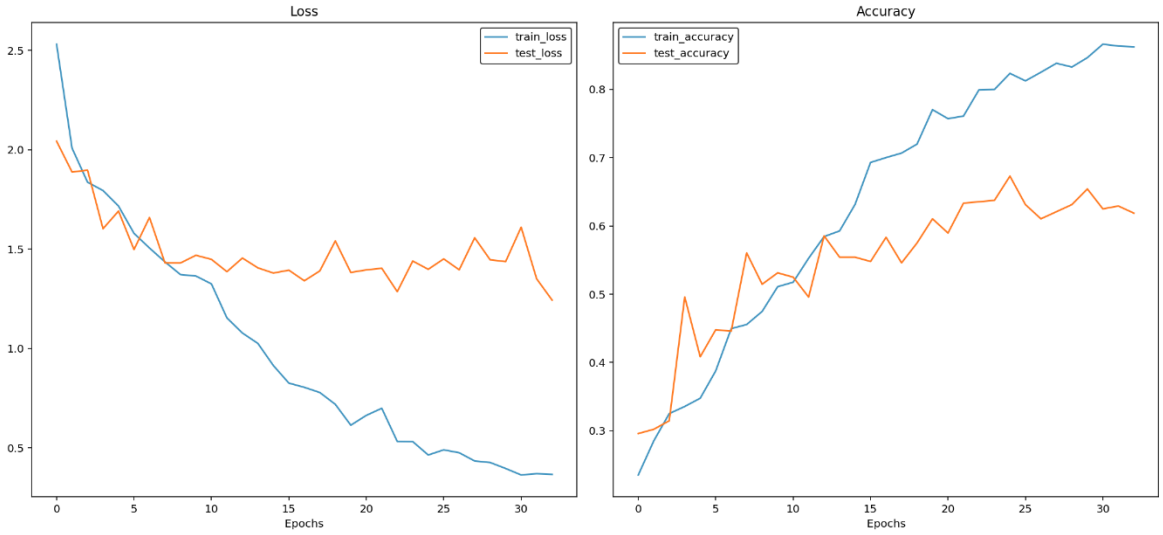
Στην τελευταία αυτή σειρά πειραμάτων ακολουθούμε την ίδια σειρά παρουσίασης των αρχιτεκτονικών. Αρχικά λοιπόν εκπαιδεύουμε ένα μοντέλο MLP για 50 epochs, όμως από πολύ νωρίς παρατηρείται overfitting και απότομη αύξηση των τιμών του validation loss. Ταυτόχρονα, το μοντέλο φαίνεται πως από πολύ νωρίς «προσαρμόζεται» στις τιμές του train set, καθώς όπως αναφέραμε το πλήθος δεδομένων στην περίπτωση αυτή είναι μικρότερο. Η χρήση data augmentation βελτιώνει ελαφρώς την εικόνα, όμως στο τελικό μοντέλο χρειάζεται να κάνουμε early stopping κατά την 5η epoch, καθώς έκτοτε το overfitting παραμένει έντονο.



Σχήμα 34 Καμπύλες εκπαίδευσης μοντέλου MLP με πρότυπο AST για μικρότερη τμηματοποίηση

Το μοντέλο αυτό δεν φαίνεται ιδιαίτερα αποδοτικό. Αυτό διακρίνεται και από το Σχήμα 34 με τα αποτελέσματα της εκπαίδευσής του.

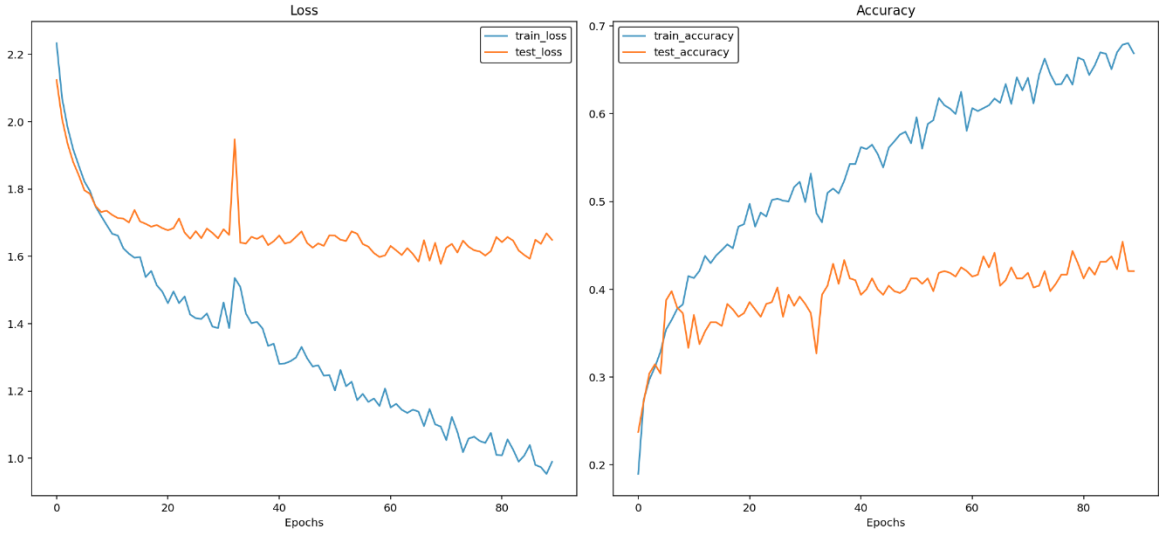
Κατόπιν, συνεχίζουμε με την αρχιτεκτονική CNN όπου αρχικά εκπαιδεύουμε ένα μοντέλο για 100 epochs. Η αρχιτεκτονική αυτή, φαίνεται να ανταποκρίνεται πολύ καλύτερα από την MLP όπως ήταν αναμενόμενο. Εδώ παρατηρήσαμε το overfitting να ξεκινά κατά τη 15η epoch, κάτι που βελτιώθηκε ελαφρώς με χρήση data augmentation. Σε αυτή την περίπτωση εκπαίδευσης, το φαινόμενο δείχνει να ξεκινά λίγο μετά την 20η epoch.



Σχήμα 35 Καμπύλες εκπαίδευσης μοντέλου CNN με πρότυπο AST για μικρότερη τμηματοποίηση

Εν τέλει επιλέξαμε να κάνουμε early stopping στην 33η epoch. Τα αποτελέσματα της εκπαίδευσης διακρίνονται στο Σχήμα 35.

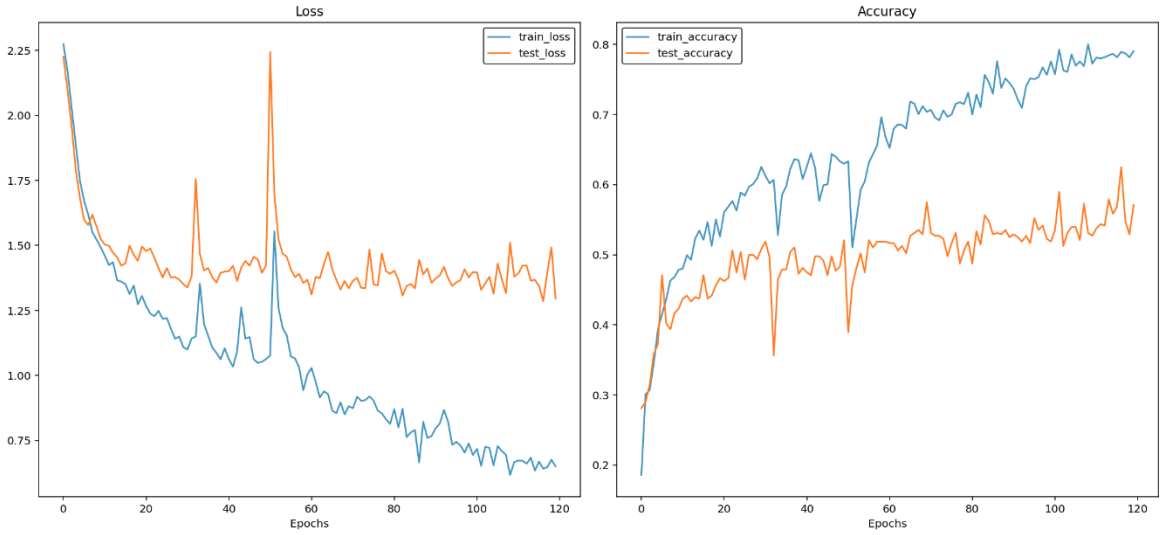
Στη συνέχεια παρουσιάζουμε το μοντέλο αρχιτεκτονικής RNN, όπου για τους ίδιους λόγους με τις προηγούμενες σειρές πειραμάτων επιλέξαμε την αρχιτεκτονική που χρησιμοποιεί μόνο το τελευταίο hidden state. Επίσης, εδώ έχουμε overfitting στην 25η epoch, δηλαδή αρκετά νωρίτερα σχετικά με άλλες φορές. Στη συνέχεια, οι τιμές του validation loss συνεχώς αυξάνονται με μεγαλύτερο ρυθμό από πριν. Το data augmentation φαίνεται ότι καθυστερεί την εμφάνιση του overfitting, ενώ σταματά την αύξηση των τιμών του validation loss, κρατώντας το σταθερό από ένα σημείο και μετά.



Σχήμα 36 Καμπύλες εκπαίδευσης μοντέλου RNN με πρότυπο AST για μικρότερη τμηματοποίηση

Εν τέλει κάνουμε early stopping στην 120η epoch. Σε αυτή την περίπτωση είναι χαρακτηριστικό, όπως φαίνεται και στο Σχήμα 36, ότι δεν παρατηρούνται έντονα vanishing/exploding gradients όπως στις άλλες περιπτώσεις. Φαίνεται πως η μεγαλύτερη διάρκεια των δειγμάτων συνετέλεσε σε αυτό.

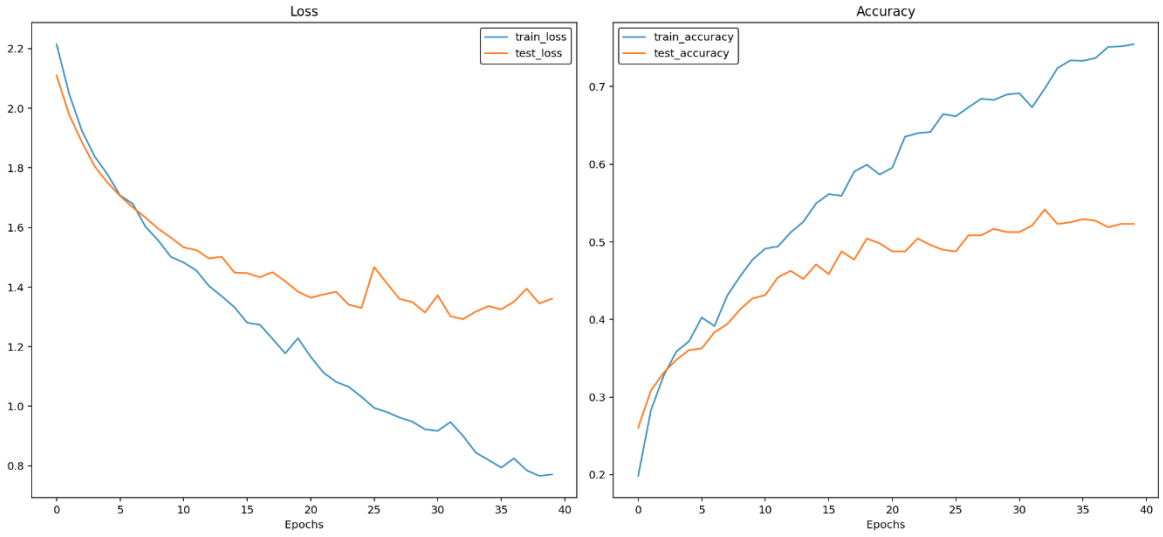
Στη συνέχεια εκπαιδεύσαμε ένα μοντέλο αρχιτεκτονικής LSTM. Σε αυτή την περίπτωση, το αρχικό μοντέλο παρουσιάζει overfitting μετά την 20η epoch. Η χρήση data augmentation εδώ φαίνεται να βοηθά, όμως παρουσιάζονται μερικές «κορυφές» όπως αυτές που συναντώνται στα μοντέλα RNN.



Σχήμα 37 Καμπύλες εκπαίδευσης μοντέλου LSTM με πρότυπο AST για μικρότερη τμηματοποίηση

Το τελικό μοντέλο αρχιτεκτονικής LSTM εκπαιδεύεται για 120 epochs και τα αποτελέσματα της εκπαίδευσης φαίνονται στο Σχήμα 37.

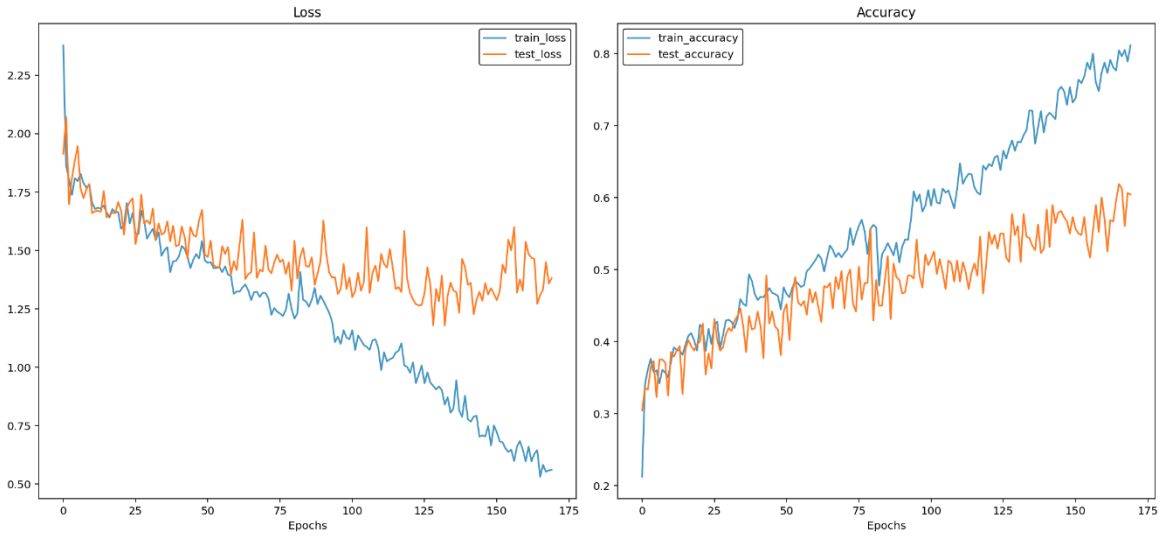
Το τελευταίο μοντέλο αναδρομικής αρχιτεκτονικής είναι το GRU και αρχικά εκπαιδεύεται για 100 epochs. Παρατηρούμε απότομη αύξηση των τιμών του validation loss μετά την 20η epoch, ενώ το μοντέλο προσαρμόζεται τελείως στο train set από την 70η epoch και έπειτα. Το data augmentation βοηθά σχετικά να ομαλοποιηθεί η συμπεριφορά αυτή.



Σχήμα 38 Καμπύλες εκπαίδευσης μοντέλου GRU με πρότυπο AST για μικρότερη τμηματοποίηση

Τελικά επιλέγεται early stopping στην 40η epoch, σχετικά νωρίς για τα δεδομένα αυτής της αρχιτεκτονικής κρίνοντας από προηγούμενα μοντέλα. Τα αποτελέσματα φαίνονται στο Σχήμα 38.

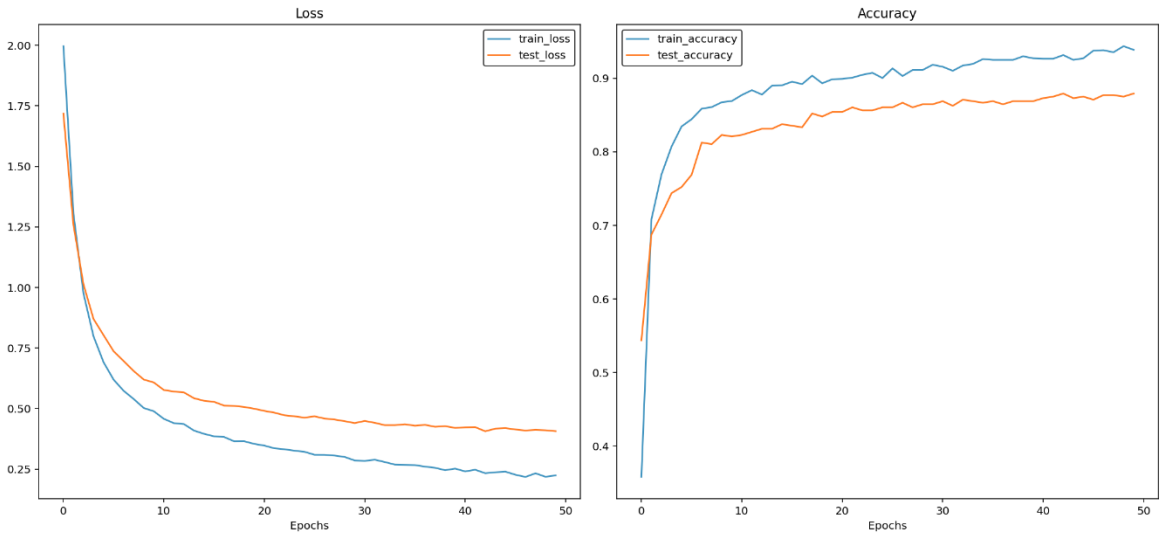
Το μοντέλο ViT σε αυτή τη σειρά πειραμάτων, αρχικά εκπαιδεύεται για 150 epochs. Παρατηρείται overfitting λίγο πριν την 20η epoch, ενώ και σε αυτή την περίπτωση στην εν λόγω σειρά πειραμάτων, η τιμή του validation loss εν συνεχεία αυξάνεται. Το πρόβλημα βελτιώνεται με χρήση data augmentation και σε αυτή την περίπτωση.



Σχήμα 39 Καμπύλες εκπαίδευσης μοντέλου ViT με πρότυπο AST για μικρότερη τμηματοποίηση

Αρχικά εκπαιδεύουμε το μοντέλο με data augmentation για 250 epochs. Μετά από μελέτη των τιμών του loss και του accuracy, επιλέγουμε early stopping στις 170 epochs. Τα αποτελέσματα της εκπαίδευσης του τελευταίου μοντέλου ViT, φαίνονται στο Σχήμα 39. Η εν λόγω εκπαίδευση, διαρκεί 22 λεπτά. Ο χρόνος είναι περίπου ο μισός από την αντίστοιχη εκπαίδευση με τμηματοποίηση σε 10 μέρη.

Στην εν λόγω σειρά πειραμάτων, επίσης έχουμε τη δυνατότητα να πειραματιστούμε πάνω στο μοντέλο AST. Αρχικά εκπαιδεύουμε το μοντέλο για 30 epochs. Ομοίως, όπως και στην προηγούμενη σειρά πειραμάτων, δεν παρατηρείται έντονο overfitting. Δοκιμάζουμε όμως και με data augmentation για να δούμε μήπως υπάρξει βελτίωση στις τιμές.

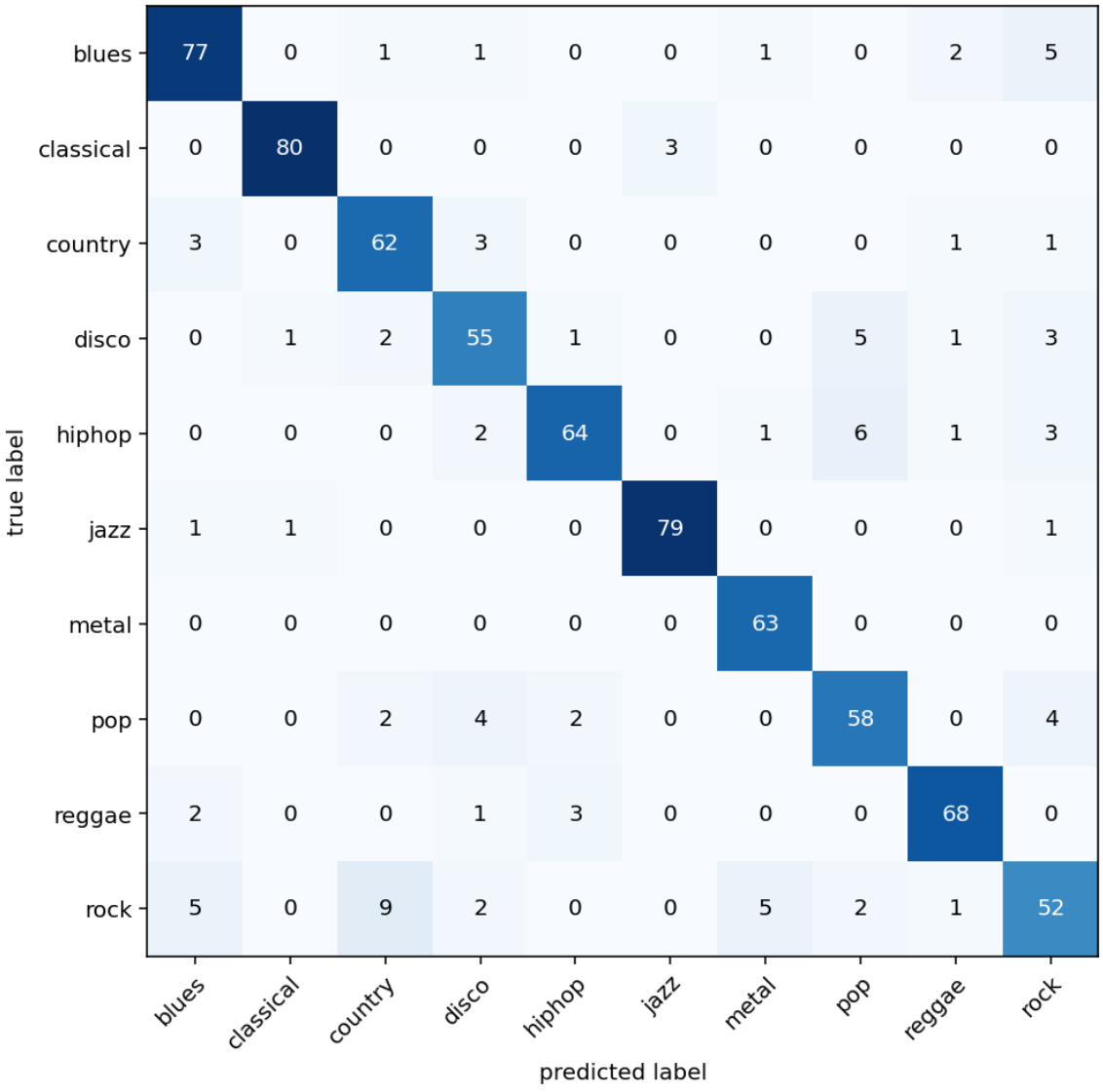


Σχήμα 40 Καμπύλες εκπαίδευσης μοντέλου AST για δείγματα διάρκειας 10 δευτερολέπτων

Η βελτίωση, όπως και στην εκπαίδευση του AST στην προηγούμενη σειρά πειραμάτων, παραμένει της τάξης του 1%. Στο Σχήμα 40 όμως, φαίνεται πως η απόκλιση μεταξύ train set και validation set είναι ελαφρώς μεγαλύτερη συγκριτικά με την προηγούμενη σειρά πειραμάτων. Το φαινόμενο αυτό, είναι πιθανόν να οφείλεται στον μικρότερο αριθμό δειγμάτων του train set. Η εκπαίδευση διήρκεσε σχεδόν 1 ώρα και 20 λεπτά, ενώ και σε αυτή την περίπτωση εκπαίδευσης του AST κρίθηκε ότι δε χρειάζεται early stopping.

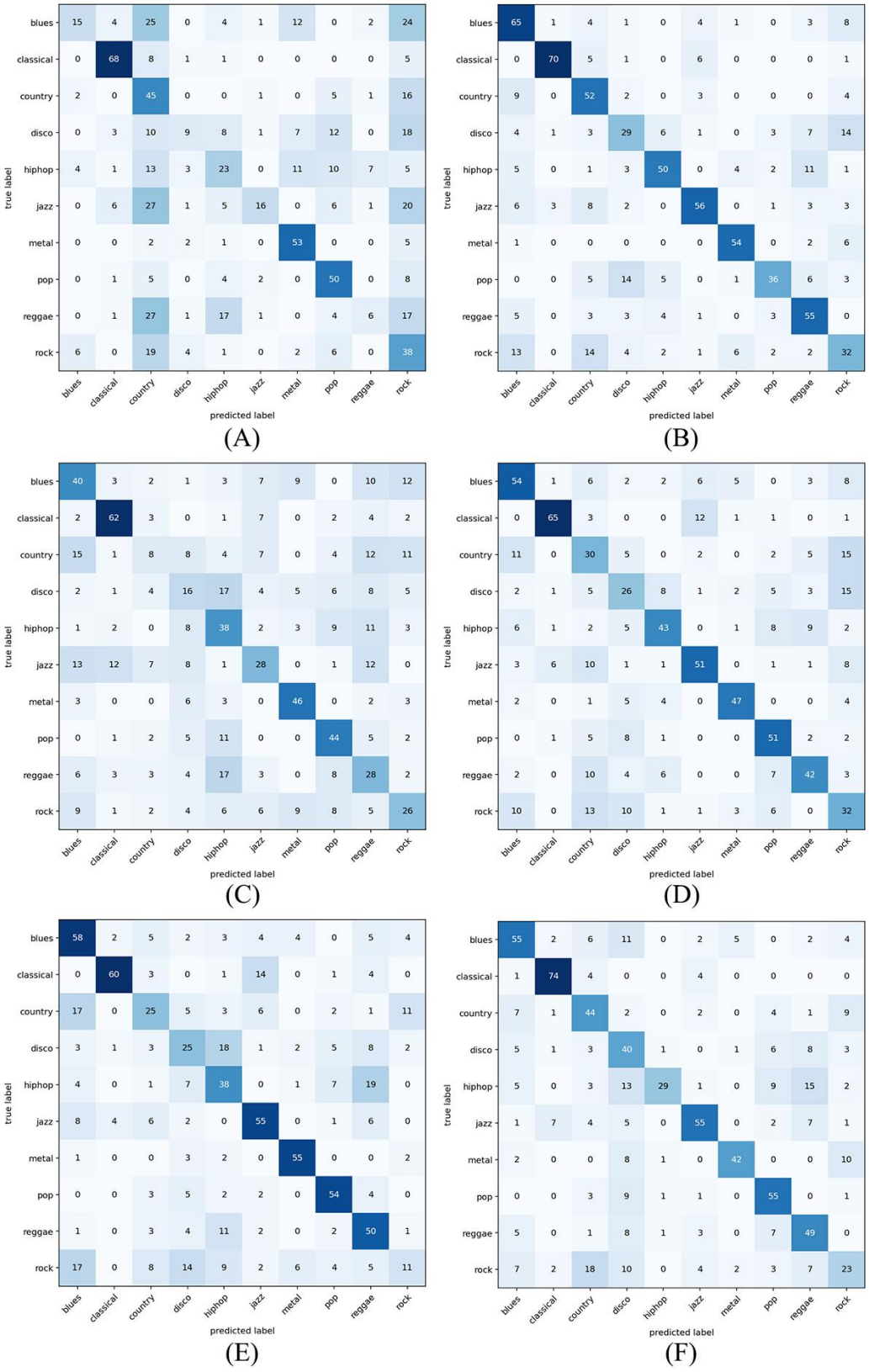
5.4.2 Πίνακες σύγχυσης

Στη συνέχεια παρουσιάζουμε τους πίνακες σύγχυσης για την τελευταία σειρά πειραμάτων. Αρχικά ξεκινάμε με το μοντέλο AST, το οποίο φαίνεται πως διατήρησε τις υψηλές επιδόσεις του αν και τα Mel Spectrograms έχουν τη μορφή του Σχήματος 4(B).



Σχήμα 41 Πίνακας σύγχυσης του μοντέλου AST που εκπαιδεύεται με δείγματα 10 δευτερολέπτων

Στη συνέχεια παρουσιάζουμε τους πίνακες σύγχυσης των υπόλοιπων αρχιτεκτονικών.



Σχήμα 42 Πίνακες σύγχυσης των μοντέλων που εκπαιδεύτηκαν με Mel Spectrograms στην 4η σειρά πειραμάτων

Στο Σχήμα 42 είναι διακριτό πως τα μοντέλα αυτής της σειράς πειραμάτων αποδίδουν αρκετά υποδεέστερα συγκριτικά με προηγούμενες σειρές. Είναι χαρακτηριστικό όμως ότι αυτό δεν συνέβη με το μοντέλο AST.

5.5 Σύγκριση απόδοσης μοντέλων

Στο κεφάλαιο αυτό θα παραθέσουμε συγκριτικά αποτελέσματα. Αρχικά θα παρουσιάσουμε τα αποτελέσματα της εκπαίδευσης για όλα τα μοντέλα κάθε σειράς πειραμάτων στο ίδιο διάγραμμα και κατόπιν θα παραθέσουμε τον υπολογισμό της απόδοσης στο test set για κάθε σειρά πειραμάτων.

5.5.1 Σύγκριση καμπυλών εκπαίδευσης

Για την παράθεση των αποτελεσμάτων της εκπαίδευσης σε ένα διάγραμμα, χρησιμοποιούμε το TensorBoard. Όπως αναφέρθηκε σε προηγούμενα κεφάλαια, χρησιμοποιούμε αυτό το εργαλείο για να αποθηκεύσουμε logs της εκπαίδευσης κάθε μοντέλου. Αποθηκεύσαμε logs από την εκπαίδευση του τελικού μοντέλου κάθε αρχιτεκτονικής και παρουσιάζουμε τα συγκριτικά αποτελέσματα.

Σε κάθε διάγραμμα, παρουσιάζεται η εξέλιξη της πιστότητας ανά epoch όπως αυτή υπολογίστηκε στο validation set. Στο Σχήμα 43 διακρίνονται τα στοιχεία αυτά για την πρώτη σειρά πειραμάτων.

Σχήμα 43 Αποτελέσματα εκπαίδευσης μοντέλων στην 1η σειρά πειραμάτων

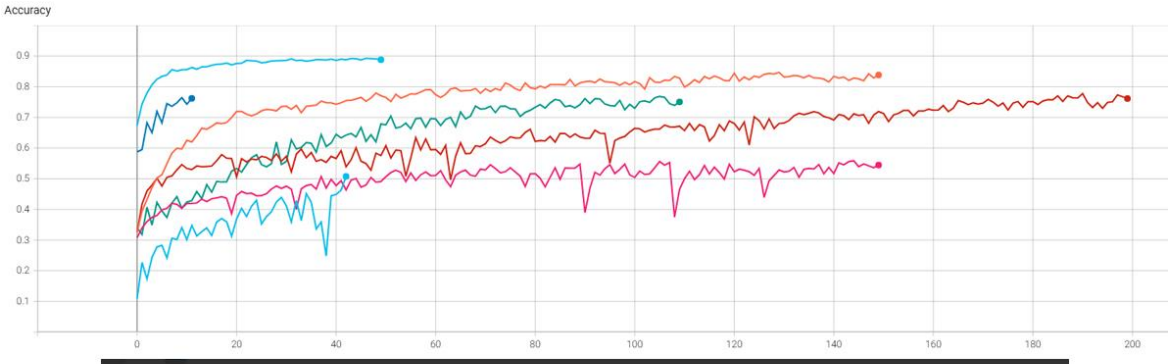
Το TensorBoard μας δίνει τη δυνατότητα να έχουμε πληροφορίες για την τιμή της πιστότητας, τον αριθμό των epochs (αναφέρονται “steps” στο διάγραμμα), καθώς και τη διάρκεια εκπαίδευσης κάθε μοντέλου. Βλέπουμε λοιπόν πως στην πρώτη σειρά

πειραμάτων, η αρχιτεκτονική CNN και η αρχιτεκτονική GRU έχουν σχεδόν την ίδια απόδοση, με την πρώτη να χρειάστηκε πολύ λιγότερες epochs για να εκπαιδευθεί.

Name	Smoothed	Value	Step	Time	Relative
09-02-2026\melspec\10_segments\CNNClassifierV0\Accuracy_test_acc	0.8512	0.8512	14	Mon Feb 9, 23:59:52	28s
09-02-2026\melspec\10_segments\MLPClassifierV0\Accuracy_test_acc	0.5682	0.5682	29	Mon Feb 9, 23:12:25	40s
10-02-2026\melspec\10_segments\GRUClassifierV0\Accuracy_test_acc	0.8422	0.8422	149	Tue Feb 10, 02:15:55	1m 53s
10-02-2026\melspec\10_segments\LSTMClassifierV0\Accuracy_test_acc	0.8324	0.8324	199	Tue Feb 10, 02:14:01	3m 31s
10-02-2026\melspec\10_segments\RNNClassifierV0\Accuracy_test_acc	0.6716	0.6716	249	Tue Feb 10, 02:10:27	3m 42s
10-02-2026\melspec\10_segments\ViT\Accuracy_test_acc	0.7714	0.7714	120	Tue Feb 10, 20:10:34	48m 23s

Σχήμα 44 Αποτελέσματα εκπαίδευσης μοντέλων στην 2η σειρά πειραμάτων

Αντίστοιχα, στη δεύτερη σειρά πειραμάτων βλέπουμε αντίστοιχη συμπεριφορά, με την LSTM αρχιτεκτονική να βρίσκεται επίσης πολύ κοντά στις CNN και GRU. Τα αποτελέσματα της δεύτερης σειράς διακρίνονται στο Σχήμα 44. Η αρχιτεκτονική CNN φαίνεται πως και πάλι χρειάστηκε πολύ λιγότερες epochs για να φτάσει την τελική της επίδοση.



Name	Smoothed	Value	Step	Time	Relative
11-02-2026\melspec\10_segments\CNNClassifierV0\transformer_options\Accuracy_test_acc	0.7621	0.7621	11	Wed Feb 11, 12:11:52	1m 1s
11-02-2026\melspec\10_segments\GRUClassifierV0\transformer_options\Accuracy_test_acc	0.8382	0.8382	149	Wed Feb 11, 13:42:27	4m 7s
11-02-2026\melspec\10_segments\LSTMClassifierV0\transformer_options\Accuracy_test_acc	0.7615	0.7615	199	Wed Feb 11, 13:38:16	7m 9s
11-02-2026\melspec\10_segments\MLPClassifierV0\transformer_options\Accuracy_test_acc	0.5076	0.5076	42	Wed Feb 11, 11:06:32	1m 47s
11-02-2026\melspec\10_segments\RNNClassifierV0\transformer_options\Accuracy_test_acc	0.5445	0.5445	149	Wed Feb 11, 13:31:03	2m 45s
11-02-2026\melspec\10_segments\ViT\transformer_options\Accuracy_test_acc	0.7504	0.7504	109	Wed Feb 11, 18:12:22	43m 56s
12-02-2026\melspec\10_segments\AST\transformer_options\Accuracy_test_acc	0.8879	0.8879	49	Thu Feb 12, 06:21:02	4h 19m 19s

Σχήμα 45 Αποτελέσματα εκπαίδευσης μοντέλων στην 3η σειρά πειραμάτων

Όπως διακρίνουμε στο Σχήμα 45, στην τρίτη σειρά πειραμάτων συμπεριλαμβάνεται το AST μοντέλο το οποίο σχεδόν φθάνει το 90% accuracy. Σε αυτήν την περίπτωση η αρχιτεκτονική

GRU έχει ξεκάθαρο προβάδισμα συγκριτικά με την CNN, η οποία έχει σχεδόν ίδια επίδοση με την LSTM και τη ViT. Το ότι τα μοντέλα AST και ViT ανήκουν στην ίδια αρχιτεκτονική αλλά το πρώτο αποδίδει πολύ καλύτερα, δεν είναι κάτι που προκαλεί έκπληξη. Στο AST οι τιμές των περισσότερων παραμέτρων είναι ήδη ρυθμισμένες. Έχουν προκύψει από εκπαίδευση με πολύ μεγάλο πλήθος δεδομένων και σε καλύτερες συνθήκες. Όπως αναφέρουν και οι Gong et al. (2021) που παρουσίασαν το AST, το μοντέλο βασίζεται στο ViT, όμως χρειάζεται να εκπαιδευθεί με μεγάλο πλήθος δεδομένων.

Name	Smoothed	Value	Step	Time	Relative
10-02-2026\melspec\3_segments\CNNClassifierV0\transformer_options\Accuracy_test_acc	0.6187	0.6187	32	Tue Feb 10, 15:29:30	2m 48s
10-02-2026\melspec\3_segments\LSTMClassifierV0\transformer_options\Accuracy_test_acc	0.5708	0.5708	119	Tue Feb 10, 17:10:59	3m 15s
10-02-2026\melspec\3_segments\MLPClassifierV0\transformer_options\Accuracy_test_acc	0.4083	0.4083	4	Tue Feb 10, 15:00:29	8s
10-02-2026\melspec\3_segments\RNNClassifierV0\transformer_options\Accuracy_test_acc	0.4208	0.4208	89	Tue Feb 10, 17:07:42	1m 1s
10-02-2026\melspec\3_segments\ViT\transformer_options\Accuracy_test_acc	0.6042	0.6042	169	Wed Feb 11, 00:00:04	22m 4s
11-02-2026\melspec\3_segments\AST\transformer_options\Accuracy_test_acc	0.8792	0.8792	49	Wed Feb 11, 03:03:42	1h 17m 57s
11-02-2026\melspec\3_segments\GRUClassifierV0\transformer_options\Accuracy_test_acc	0.5229	0.5229	39	Wed Feb 11, 12:40:18	37s

Σχήμα 46 Αποτελέσματα εκπαίδευσης μοντέλων στην 4η σειρά πειραμάτων

Στην τελευταία σειρά πειραμάτων οι επιδόσεις είναι αρκετά χαμηλότερα. Όπως βλέπουμε στο Σχήμα 46, η αρχιτεκτονική CNN φαίνεται να ξεπερνά τις αναδρομικές αρχιτεκτονικές. Βρίσκεται όμως αρκετά χαμηλότερα από την AST, η οποία είναι ξανά λίγο χαμηλότερα από την τιμή του 90%. Ενδιαφέρον παρουσιάζει επίσης το ότι όπως βλέπουμε στο Σχήμα 46 και στο Σχήμα 45, όταν χρησιμοποιούμε δεδομένα με το πρότυπο του AST, η αρχιτεκτονική RNN οριακά ξεπερνά την αρκετά απλούστερη αρχιτεκτονική του MLP.

5.5.2 Σύγκριση απόδοσης στο test set

Όπως αναφέρθηκε σε προηγούμενο κεφάλαιο, μετά την εκπαίδευση του τελικού μοντέλου κάθε αρχιτεκτονικής, αξιολογούμε την απόδοσή του στο test set. Τα μοντέλα δεν έχουν εκτεθεί στα δεδομένα του test set σε κανένα σημείο της εκπαίδευσης, επομένως η τιμή της πιστότητάς τους όταν επιχειρούν να ταξινομήσουν τα δεδομένα του, αποτελεί μια «αμερόληπτη» αξιολόγηση.

Παραθέτουμε τα αποτελέσματα του υπολογισμού της πιστότητας, οργανωμένα σε πίνακες. Κάθε πίνακας αντιστοιχεί σε μια σειρά σύγκρισης πειραμάτων. Με τον τρόπο αυτό, κάποια αποτελέσματα θα επαναλαμβάνονται σύμφωνα με όσα έχουμε αναφέρει σε προηγούμενα υποκεφάλαια. Όμως με αυτόν τον τρόπο είναι ευκολότερο να αξιολογηθούν οι τρείς συνθήκες που προσπαθούμε να συγκρίνουμε: Η εκπαίδευση με διαφορετικά ακουστικά χαρακτηριστικά, με διαφορετική ανάλυση συχνότητας/χρόνου και με διαφορετική τμηματοποίηση.

Αρχιτεκτονική Μοντέλου	Mel Spectrograms	MFCCs
MLP	58%	59%
CNN	86%	81%
RNN	68%	61%
LSTM	83%	82%
GRU	85%	82%
ViT	78%	73%

Πίνακας 3 Ποσοστά Accuracy αρχιτεκτονικών για διαφορετικά ακουστικά χαρακτηριστικά

Τα αποτελέσματα της πιστότητας για την πρώτη σύγκριση συνθηκών, φαίνονται στον Πίνακα 3. Διαπιστώνουμε πως τα Mel Spectrograms συνολικά αποτέλεσαν λίγο καλύτερα δεδομένα εισόδου. Η διαφορά της τιμής για την πιστότητα, ανάλογα την αρχιτεκτονική, κυμαίνεται στο 1-7%.

Στον Πίνακα 4 διακρίνουμε πως η καλύτερη ανάλυση συχνότητας, συνολικά προσφέρει καλύτερη απόδοση. Η καλύτερη ανάλυση χρόνου όμως, βοήθησε τις αρχιτεκτονικές GRU και LSTM να ξεπεράσουν και να φτάσουν αντίστοιχα την αρχιτεκτονική CNN. Η εκπαίδευση με δεδομένα τα οποία έχουν καλύτερη ανάλυση στο επίπεδο του χρόνου γενικότερα δεν ευνοεί τη συνολική εικόνα των μοντέλων.

Αρχιτεκτονική Μοντέλου	Καλύτερη ανάλυση συχνότητας	Καλύτερη ανάλυση χρόνου
MLP	58%	49%
CNN	86%	77%
RNN	68%	56%
LSTM	83%	76%
GRU	85%	82%
ViT	78%	75%

Πίνακας 4 Ποσοστά Accuracy αρχιτεκτονικών για διαφορετικές ρυθμίσεις FFT

Το γεγονός ότι οι αρχιτεκτονικές LSTM και GRU ανταγωνίζονται την CNN όταν τροφοδοτούνται με δεδομένα που έχουν καλύτερη ανάλυση στο επίπεδο του χρόνου, δε σημαίνει ότι αποδίδουν συνολικά καλύτερα. Παρατηρούμε ότι αυτές οι αρχιτεκτονικές πετυχαίνουν υψηλότερες τιμές accuracy όταν τους δίνονται δείγματα με καλύτερη ανάλυση στο επίπεδο της συχνότητας.

Αρχιτεκτονική Μοντέλου	3 part segmentation	10 part segmentation
MLP	42%	49%
CNN	66%	77%
RNN	45%	56%
LSTM	59%	76%
GRU	57%	82%
ViT	62%	75%
AST	88%	88%

Πίνακας 5 Ποσοστά Accuracy αρχιτεκτονικών για διαφορετικές περιπτώσεις τμηματοποίησης

Αντίστοιχα, στον Πίνακα 5 βλέπουμε πως η μεγαλύτερη τμηματοποίηση που προσφέρει περισσότερα δείγματα βοηθά, ακόμα κι αν αυτά είναι μικρότερα σε διάρκεια. Ο παράγοντας αυτός δε φαίνεται να επηρεάζει το μοντέλο AST, το οποίο συνολικά απέδωσε σχεδόν το ίδιο στις δύο περιπτώσεις κρίνοντας από τον Πίνακα 5 και το Σχήμα 33.

6. Συμπεράσματα

Στο παρόν κεφάλαιο συνοψίζονται τα βασικά πορίσματα της έρευνας, όπως αυτά διαμορφώθηκαν μέσα από την παρατήρηση των αποτελεσμάτων. Η ανάλυση συνδυάζει τα πειραματικά ευρήματα με τη βιβλιογραφική τεκμηρίωση των αρχιτεκτονικών. Παράλληλα, λαμβάνονται υπόψιν οι ιδιαιτερότητες των ακουστικών χαρακτηριστικών που χρησιμοποιήθηκαν για την εκπαίδευση των μοντέλων

6.1 Επίδραση της αναπαράστασης ακουστικών χαρακτηριστικών

Παρατηρούμε πως συνολικά τα Mel Spectrograms συνετέλεσαν στην καλύτερη απόδοση των μοντέλων συγκριτικά με τα MFCCs. Αυτό είναι κάτι που γενικότερα επιβεβαιώνεται από τη βιβλιογραφία και την έρευνα, καθώς τα μοντέλα που συνδυάζουν αρχιτεκτονική CNN και Mel Spectrograms φαίνεται πως αποτελούν μια αξιόπιστη λύση για την ταξινόμηση μουσικών κομματιών σε είδη.

Τα MFCCs εμπεριέχουν πληροφορία που σχετίζεται με την εξέλιξη της χροιάς του ήχου στο χρόνο (των επονομαζόμενων formants). Τα Mel Spectrograms όμως περιλαμβάνουν περισσότερη πληροφορία. Πιο συγκεκριμένα, περιλαμβάνουν την εξέλιξη συχνοτήτων στο χρόνο καθώς και την αποτύπωση των αρμονικών μοτίβων, αποθηκευμένη σύμφωνα με την ανθρώπινη αντίληψη. Όλα αυτά τα στοιχεία σχετίζονται με τα μουσικά είδη και αυτός είναι ο λόγος που φαίνεται ότι αποδίδουν.

Τα MFCCs φαίνεται ότι αποδίδουν καλύτερα σε εργασίες όπως η αναγνώριση φωνής. Θα μπορούσαν επίσης να λειτουργήσουν σε πιο απλά μοντέλα ή όταν έχουμε λιγότερη πληροφορία. Διαπιστώσαμε για παράδειγμα, ότι στην περίπτωση του MLP που είναι η απλούστερη αρχιτεκτονική, τα MFCCs απέδωσαν καλύτερα.

Τέλος, είναι σημαντικό να αναφερθεί, ότι αρκετά μουσικά είδη όπως blues, rock και metal χρησιμοποιούν όργανα με παρόμοιες χροιές. Τα έντονα τύμπανα, η παραμορφωμένη κιθάρα και το ηλεκτρικό μπάσο εμφανίζονται σε όλα αυτά τα ήδη. Αυτό είναι ένα αρκετά σημαντικό στοιχείο καθώς, όπως αναφέρθηκε, τα MFCCs επικεντρώνονται ακριβώς σε αυτή την πληροφορία.

6.2 Επίδραση των παραμέτρων της FFT

Συγκρίνοντας τις επιδόσεις των μοντέλων όταν εκτίθενται σε δεδομένα που προσφέρουν καλύτερη ανάλυση στο πεδίο της συχνότητας, βλέπουμε ότι υπάρχει καλύτερη απόδοση.

Φαίνεται πως η περισσότερη πληροφορία για τις συχνότητες που εμφανίζονται, συγκριτικά με τη χρονική τους εξέλιξη, βοηθά περισσότερο. Περαιτέρω παρατήρηση της απόδοσης κάθε μοντέλου όμως, οδηγεί σε μερικά ακόμα ενδιαφέροντα συμπεράσματα.

Η καλύτερη χρονική ανάλυση οδηγεί ένα GRU μοντέλο να αποδώσει καλύτερα από ένα CNN. Παράλληλα, το LSTM μοντέλο ανταγωνίζεται το CNN αποδίδοντας σχεδόν το ίδιο. Η αρχιτεκτονική CNN αποτελεί ένα εργαλείο που σχετίζεται με το ζήτημα του computer vision. Όταν λοιπόν τη χρησιμοποιούμε σε απεικονίσεις ακουστικών χαρακτηριστικών, είναι σαν να τα «φωτογραφίζουμε» ως στιγμιότυπα. Οι αναδρομικές αρχιτεκτονικές με τη σειρά τους, εξ ορισμού επωφελούνται από τη χρονική συνέπεια της πληροφορίας. Τα δεδομένα με τα οποία τροφοδοτούμε τις αρχιτεκτονικές στην περίπτωση της καλύτερης χρονικής ανάλυσης έχουν 297 time frames συγκριτικά με τα 130 time frames της άλλης περίπτωσης, για δείγματα ίδιας διάρκειας. Η ανάλυση αυτή βοηθά αρκετά τα αναδρομικά δίκτυα GRU και LSTM. Τα RNN δε φαίνεται ότι μπορούν να ακολουθήσουν αυτή τη συμπεριφορά, λόγω των vanishing και exploding gradients.

Παρατηρώντας τους πίνακες σύγχυσης των GRU μοντέλων που εκπαιδεύτηκαν με δεδομένα των δύο διαφορετικών ρυθμίσεων FFT, παρατηρούμε κάτι ενδιαφέρον. Το GRU μοντέλο που εκπαιδεύτηκε με καλύτερη ανάλυση στο επίπεδο της συχνότητας έφτασε υψηλότερες τιμές accuracy, συγκριτικά με το αντίστοιχο που εκπαιδεύτηκε με καλύτερη ανάλυση στο επίπεδο του χρόνου. Παρ όλα αυτά, το δεύτερο φαίνεται ότι απέδωσε καλύτερα σε μουσικά είδη όπως “hip hop”, “reggae” και “pop”. Στα είδη αυτά, τα ρυθμικά στοιχεία συνιστούν σημαντικό χαρακτηριστικό. Ο ρυθμός όμως, είναι μια έννοια άρρηκτα συνδεδεμένη με το χρόνο.

6.3 Επίδραση της τμηματοποίησης

Παρατηρούμε πως συνολικά το μεγαλύτερο πλήθος δεδομένων οδηγεί σε καλύτερη απόδοση των μοντέλων. Η μεγαλύτερη διάρκεια των δειγμάτων δε βοηθά τόσο πολύ όσο βοηθά το πλήθος των δεδομένων. Συνεπώς η τμηματοποίηση των δειγμάτων ωφελεί ως τακτική. Αξίζει όμως να αναφερθούν κάποιες περαιτέρω σημαντικές λεπτομέρειες που παρατηρούμε.

Στην περίπτωση των δειγμάτων 10 δευτερολέπτων, παρατηρούμε ότι το μοντέλο LSTM αποδίδει καλύτερα από το GRU. Αυτό είναι μια καλή ένδειξη πως το cell state της

αρχιτεκτονικής αυτής βοηθά το μοντέλο να διαχειριστεί καλύτερα τη μεγαλύτερη διάρκεια του δείγματος.

Επίσης, παρατηρούμε πως στην περίπτωση του AST, το πλήθος των δειγμάτων και η διάρκειά τους δεν επηρεάζει την απόδοση. Πρόκειται για ένα pre-trained μοντέλο, επομένως τα περισσότερα weights και biases είναι ήδη ρυθμισμένα. Η εκπαίδευση του μοντέλου επηρεάζει τα weights και biases μόνο του MLP μέρους του και όχι του encoder. Το τελευταίο όμως παραμένει βασικό κομμάτι της αρχιτεκτονικής.

6.4 Συνολική αποτίμηση αρχιτεκτονικών

Τα σχόλια που ήδη αναφέρθηκαν αφορούν μόνο την επιρροή της εκάστοτε συνθήκης στην απόδοση των μοντέλων. Πέραν των σχολίων αυτών, μπορούμε να αναφέρουμε κάποιες παρατηρήσεις σχετικά με τη γενική συμπεριφορά των μοντέλων.

Η αρχιτεκτονική MLP είναι η απλούστερη που συμπεριλάβαμε στη μελέτη μας. Δεν περιμένουμε να αποδώσει καλά, όμως δεδομένης της απλότητάς τα μοντέλα αυτής της αρχιτεκτονικής παρουσιάζουν μια σχετικά ικανοποιητική γενίκευση. Η τιμή της accuracy στο σύνολο των πειραμάτων βρίσκεται κοντά στο 50%. Αυτή η απόδοση δεν είναι πολύ κακή αν αναλογιστούμε ότι μια τυχαία επιλογή του μουσικού είδους στην έρευνά μας, έχει 10% πιθανότητα να είναι σωστή.

Η αρχιτεκτονική CNN φαίνεται πως είναι η πιο αξιόπιστη αρχιτεκτονική. Δείχνει να πετυχαίνει υψηλά επίπεδα απόδοσης σε όλες τις διαφορετικές συνθήκες και δεν είναι τυχαίο που μέχρι σήμερα αποτελεί συνήθη επιλογή για ταξινόμηση μουσικών ειδών. Υπό συνθήκες την ξεπερνούν σε απόδοση οι αναδρομικές αρχιτεκτονικές. Πιο συγκεκριμένα, αυτό συμβαίνει όταν οι δεύτερες μπορούν να εκμεταλλευθούν καλύτερα τη χρονική πληροφορία. Τα μοντέλα AST επίσης ξεπέρασαν κατά πολύ την αρχιτεκτονική CNN, όμως εδώ υπάρχει μια ενδιαφέρουσα σημείωση. Ένα μοντέλο αρχιτεκτονικής CNN είναι πολύ πιθανότερο να καταλαμβάνει λιγότερο χώρο στη μνήμη από ένα αντίστοιχο AST. Σε πρακτικές εφαρμογές, αυτό μπορεί να είναι σημαντικό, καθώς η διαφορά στην απόδοση είναι πιθανόν να αντισταθμίζεται από το γεγονός ότι μια εφαρμογή που θα χρησιμοποιούσε CNN μοντέλο θα ήταν μικρότερη σε μέγεθος. Αυτό μπορεί να αποτελεί καταλυτικό παράγοντα για εφαρμογές σε smartphones, όπου η μνήμη είναι πολύ μικρότερη.

Η απλή αναδρομική αρχιτεκτονική του RNN δε φαίνεται να αποδίδει τόσο καλά για το επίπεδό της. Πετυχαίνει αξιοσημείωτες τιμές, αλλά γενικά φαίνεται ότι υποφέρει αρκετά

από vanishing/exploding gradients, κάτι που προβλέπεται και από τη θεωρία. Οι βελτιώσεις αυτής της αρχιτεκτονικής φαίνεται ότι διορθώνουν αυτό το πρόβλημα καθώς τα μοντέλα αρχιτεκτονικής LSTM φαίνεται ότι αποδίδουν καλύτερα. Η εν λόγω αρχιτεκτονική φαίνεται ότι μπορεί υπό συνθήκες να ξεπεράσει τη CNN και ακόμα και τη GRU. Παράλληλα, γενικότερα αγγίζει ικανοποιητικές τιμές πιστότητας. Όσον αφορά την αρχιτεκτονική GRU, στην περίπτωση που αυτή μπορεί να εκμεταλλευθεί την χρονική πληροφορία, παρατηρούμε ότι είναι μια εξαιρετική επιλογή που ξεπερνά τα μοντέλα της CNN. Παράλληλα, επίσης αγγίζει ικανοποιητικές τιμές πιστότητας.

Όσον αφορά την αρχιτεκτονική του ViT, τα μοντέλα φαίνεται ότι αποδίδουν καλά, φτάνοντας τιμές που ακολουθούν μεν τη συμπεριφορά των LSTM/GRU, αλλά πετυχαίνουν λίγο χαμηλότερη απόδοση. Αυτό οφείλεται στο γεγονός ότι μοντέλα που εκμεταλλεύονται αυτή την τεχνολογία, χρειάζονται πολύ μεγάλη ποσότητα δεδομένων για να εκπαιδευθούν. Παράλληλα, όπως αναφέρεται και στη βιβλιογραφία, η αποδοτική εκπαίδευση τέτοιων μοντέλων απαιτεί χρόνο και χρήμα, κάτι μη προσβάσιμο για έναν μέσο χρήστη.

Το μοντέλο AST φαίνεται πως αποδίδει καλύτερα από κάθε άλλο. Πρόκειται για την τελευταία λέξη της τεχνολογίας συγκριτικά με τις υπόλοιπες αρχιτεκτονικές. Δεδομένου ότι έχουμε ένα pre-trained μοντέλο, βλέπουμε ότι η αρχιτεκτονική Transformer μπορεί να αποτελέσει την καλύτερη λύση, ενώ η ποσότητα δεδομένων δε φαίνεται να επηρεάζει ιδιαίτερα την απόδοσή της. Παράλληλα, βλέπουμε τη δύναμη της τεχνικής του transfer learning. Καθώς μέσω αυτής ένας μέσος χρήστης, μπορεί να έχει πρόσβαση σε ένα αποδοτικό μοντέλο, χωρίς να κατασκευάζει και να εκπαιδεύει μοντέλα εξ αρχής.

Τέλος, αναφέρουμε κάποιες περαιτέρω σημειώσεις που μπορούμε να θέσουμε παρατηρώντας τους πίνακες σύγχυσης. Παρατηρούμε λοιπόν, πως τα μοντέλα κυρίως παρουσιάζουν σφάλματα μεταξύ ειδών που είναι παρεμφερή. Όπως για παράδειγμα η “rock”, η “blues” και η “country”. Τα είδη αυτά, αποτελούν παραδείγματα που και ένας άνθρωπος είναι πιθανόν να ταξινομούσε εσφαλμένα. Στην περίπτωση της κλασσικής μουσικής, παρατηρούμε τα λιγότερα σφάλματα, πράγμα που επίσης συμβαδίζει με την ανθρώπινη παρατήρηση. Στην περίπτωση που υπήρχαν σφάλματα ταυτοποίησης της κλάσης “classical” συχνά η σύγχυση γινόταν με τη “jazz”. Αυτό είναι επίσης λογικό, καθώς συγκριτικά με άλλα ήδη η κλασσική μοιράζεται τα περισσότερα χαρακτηριστικά με τη Jazz. Γενικότερα, πολλές φορές σε προβλήματα ταξινόμησης, δεν είναι εφικτό να φτάσουμε το τέλειο αποτέλεσμα, καθώς αρκετές φορές ούτε οι ίδιοι οι άνθρωποι θα ταξινομούσαν τα

ίδια δεδομένα ομοίως. Αυτό σίγουρα ισχύει για την ταξινόμηση μουσικών ειδών. Με αυτά τα δεδομένα λοιπόν, θα λέγαμε πως οι τιμές πιστότητας που πετυχαίνουν τα μοντέλα μας είναι αρκετά ικανοποιητικές.

Βιβλιογραφία

Ακολουθούν οι βιβλιογραφικές αναφορές (πηγές) της Εργασίας.

Aigrain, P. (1999). New Applications of Content Processing of Music. *Journal of New Music Research*, *28*(4), 271–280. https://doi.org/10.1076/0929-8215(199912)28:04;1-o;ft271

Backstrom, T. (2019). Overlap-add Windows with Maximum Energy Concentration for Speech and Audio Processing. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 491–495. https://doi.org/10.1109/ICASSP.2019.8683577

Bainbridge, D., Cunningham, S. J., & Downie, J. S. (2003). Analysis of queries to a Wizard-of-Oz MIR system: Challenging assumptions about what people really want. *4th International Conference on Music Information Retrieval (ISMIR 2003)*. http://ismir2003.ismir.net/papers/Bainbridge.pdf

Benois-Pineau, J., & Zemmari, A. (2021). Multi-faceted Deep Learning: Models and Data. *Multi-Faceted Deep Learning: Models and Data*, 1–316. https://doi.org/10.1007/978-3-030-74478-6/COVER

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Bogert, B. P., Healy, M. J., & Tukey, J. W. (1963). The Quefrency Analysis of Time Series for Echoes: Cepstrum, Pseudo-Autocovariance, Cross-Cepstrum, and Saphe Cracking. *Proceedings of the Symposium on Time Series Analysis*, 209–243.

Božić, M., & Horvat, M. (2024). *A Survey of Deep Learning Audio Generation Methods*. http://arxiv.org/abs/2406.00146

Brackett, D. (2016). *Categorizing Sound*. University of California Press. https://doi.org/10.1525/california/9780520248717.001.0001

Briggs, W. L. ., & Henson, V. Emden. (1995). *The DFT : an owner's manual for the discrete Fourier transform*. Society for Industrial and Applied Mathematics.

Cauchy, A.-L. (1847). Méthode générale pour la résolution de systèmes d'équations simultanées. *Comptes Rendus de l'Academie Des Science*, *25*, 536–538.

Chakroborty, S., Roy, A., & Saha, G. (2008). Improved Closed Set Text-Independent Speaker Identification by Combining MFCC with Evidence from Flipped Filter Banks. *International Journal of Electronics and Communication Engineering*, *2*(11), 2554–2561. https://doi.org/doi.org/10.5281/zenodo.1330573

Chang, X., Zhang, X., Zhang, H., & Ran, Y. (2024). *Music Emotion Prediction Using Recurrent Neural Networks*. http://arxiv.org/abs/2405.06747

Chełkowska-Zacharewicz, M., & Paliga, M. (2020). Music emotions and associations in film music listening: The example of leitmotifs from the Lord of the Rings movies. *Annals of Psychology*, *22*(2), 151–175. https://doi.org/10.18290/RPSYCH.2019.22.2-4

Chelladurai, K., & Sujatha, N. (2023). A Survey on Different Algorithms Used in Deep Learning Process. *E3S Web of Conferences*, *387*, 05008. https://doi.org/10.1051/e3sconf/202338705008

Choi, K., Fazekas, G., & Sandler, M. (2016). Automatic tagging using deep convolutional neural networks. *Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR 2016*, 805–811. https://arxiv.org/pdf/1606.00298

Christensen, M. G. (2019). Introduction to Audio Processing. In *Introduction to Audio Processing*. Springer International Publishing. https://doi.org/10.1007/978-3-030-11781-8

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. https://arxiv.org/pdf/1412.3555

Clevert, D. A., Unterthiner, T., & Hochreiter, S. (2015). Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*. https://arxiv.org/pdf/1511.07289

Cooley, J. W., & Tukey, J. W. (1965). An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, *19*(90), 297–301. https://doi.org/10.1090/S0025-5718-1965-0178586-1

Davies, S. (2010). Emotions Expressed and Aroused by Music. In *Handbook of Music and Emotion: Theory, Research, Applications* (pp. 15–43). Oxford University PressOxford. https://doi.org/10.1093/acprof:oso/9780199230143.003.0002

Davis, S., & Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *28*(4), 357–366. https://doi.org/10.1109/TASSP.1980.1163420

Deliège, I. (2001). Prototype Effects in Music Listening: An Empirical Approach to the Notion of Imprint. *Music Perception*, *18*(3), 371–407. https://doi.org/10.1525/mp.2001.18.3.371

Deshpande, O., Solanki, K., Suribhatla, S. P., Zaveri, S., & Ghodasara, L. (2021). *Simulating the DFT Algorithm for Audio Processing*.

Donahue, C., Li, B., & Prabhavalkar, R. (2018). *Exploring Speech Enhancement with Generative Adversarial Networks for Robust Speech Recognition*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *ICLR 2021 - 9th International Conference on Learning Representations*. https://arxiv.org/pdf/2010.11929

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, *12*(61), 2121–2159. http://jmlr.org/papers/v12/duchi11a.html

Elharrouss, O., Mahmood, Y., Bechqito, Y., Serhani, M. A., Badidi, E., Riffi, J., & Tairi, H. (2025). *Task-based Loss Functions in Computer Vision: A Comprehensive Review*. https://arxiv.org/pdf/2504.04242

Eronen, A. (2001). Comparison of features for musical instrument recognition. *Proceedings of the 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (Cat. No.01TH8575)*, 19–22. https://doi.org/10.1109/ASPAA.2001.969532

Eronen, A. (2007). Chorus detection with combined use of mfcc and chroma features and image processing filters. *International Conference on Digital Audio Effects*, 229–236.

Eronen, A., Peltonen, V. T., Tuomi, J. T., Klapuri, A. P., Fagerlund, S., Sorsa, T., Lorho, G., & Huopaniemi, J. (2006). Audio-based context recognition. *IEEE Transactions on Audio, Speech and Language Processing*, *14*(1), 321–329. https://doi.org/10.1109/TSA.2005.854103

Fabbri, F. (1981). A Theory of Musical Genres: Two Applications. In Horn David & Tagg Phillip (Eds.), *Popular Music Perspectives* (pp. 52–81). International Association for the Study of Popular Music.

Fastl, H. ., & Zwicker, Eberhard. (2007). *Psychoacoustics : facts and models* (3rd ed.). Springer.

Foote, J. (1997). A Similarity Measure for Automatic Audio Classification. *Proceedings of the 14th National Conference on Artificial Intelligence, Providence, RI*.

Foote, J. (1999). An overview of audio information retrieval. *Multimedia Systems*, *7*(1), 2–10. https://doi.org/10.1007/s005300050106

Géron, A. (2023). *Hands-on machine learning with Scikit-Learn, Keras and TensorFlow : concepts, tools, and techniques to build intelligent systems* (3rd ed.). O'Reilly Media, Inc.

Gjerdingen, R. O., & Perrott, D. (2008). Scanning the dial: The rapid recognition of music genres. *Journal of New Music Research*, *37*(2), 93–100. https://doi.org/10.1080/09298210802479268

Gong, Y., Chung, Y. A., & Glass, J. (2021). AST: Audio Spectrogram Transformer. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, *1*, 56–60. https://doi.org/10.21437/Interspeech.2021-698

Goodfellow, Ian., Bengio, Yoshua., & Courville, Aaron. (2017). *Deep learning*. The MIT Press.

Gouyon, F., & Dixon, S. (2004). Dance music classification: A tempo-based approach. *5th International Conference on Music Information Retrieval (ISMIR 2004))*.

Hansen, C. H. (2018). *Foundations of Vibroacoustics* (1st Edition). CRC Press. https://doi.org/10.1201/b22303

Haykin, S. (2009). *Neural networks and learning machines* (3rd ed.). Prentice Hall/Pearson.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *2015 IEEE International Conference on Computer Vision (ICCV)*, 1026–1034. https://doi.org/10.1109/ICCV.2015.123

Heffner, H. E., & Heffner, R. S. (2007). Hearing ranges of laboratory animals. *Journal of the American Association for Laboratory Animal Science : JAALAS*, *46*(1), 20–22.

Hendrycks, D., & Gimpel, K. (2016). *Gaussian Error Linear Units (GELUs)*. https://arxiv.org/pdf/1606.08415

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/NECO.1997.9.8.1735

Huang, C.-Z. A., Vaswani, A., Uszkoreit, J., Shazeer, N., Simon, I., Hawthorne, C., Dai, A. M., Hoffman, M. D., Dinculescu, M., & Eck, D. (2018). *Music Transformer*. https://arxiv.org/pdf/1809.04281

Huang, Xuedong., Acero, Alejandro., & Hon, H.-Wuen. (2001). *Spoken language processing : a guide to theory, algorithm, and system development* (1st ed.). Prentice Hall PTR.

Hubbard, B. Burke. (2010). *The world according to wavelets : the story of a mathematical technique in the making* (2nd ed.). CRC Press, Taylor & Francis Group.

Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *32nd International Conference on Machine Learning, ICML 2015*, *1*, 448–456. https://arxiv.org/pdf/1502.03167

Jackson, L. L., Heffner, R. S., & Heffner, H. E. (1999). Free-field audiogram of the Japanese macaque (Macaca fuscata). *The Journal of the Acoustical Society of America*, *106*(5), 3017–3023. https://doi.org/10.1121/1.428121

Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, *31*(3), 685–695. https://doi.org/10.1007/s12525-021-00475-2

Jin, X., Xu, C., Feng, J., Wei, Y., Xiong, J., & Yan, S. (2016). Deep Learning with S-Shaped Rectified Linear Activation Units. *Proceedings of the AAAI Conference on Artificial Intelligence*, *30*(1), 1737–1743. https://doi.org/10.1609/AAAI.V30I1.10287

Khasgiwala, Y., & Tailor, J. (2021). Vision Transformer for Music Genre Classification using Mel-frequency Cepstrum Coefficient. *2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON)*, 1–5. https://doi.org/10.1109/GUCON50781.2021.9573568

Kim, H., Moreau, N., & Sikora, T. (2005). *MPEG-7 Audio and Beyond*. Wiley. https://doi.org/10.1002/0470093366

Kim, Y. E., Schmidt, E. M., Migneco, R., Morton, B. G., Richardson, P., Scott, J., Speck, J. A., & Turnbull, D. (2010). State of the Art Report: Music Emotion Recognition: A State of the Art Review. *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, 255–266. https://doi.org/https://doi.org/10.5281/zenodo.1417945

Kingma, D. P., & Ba, J. L. (2014). Adam: A Method for Stochastic Optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. https://arxiv.org/pdf/1412.6980

Klapuri, A. (1999). Sound onset detection by applying psychoacoustic knowledge. *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*, 3089–3092 vol.6. https://doi.org/10.1109/ICASSP.1999.757494

Knees, P., & Schedl, M. (2016). *Music Similarity and Retrieval* (1st ed., Vol. 36). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-49722-7

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/NATURE14539;SUBJMETA

Lee, J. H., & Downie, S. (2004). Survey Of Music Information Needs, Uses, And Seeking Behaviours: Preliminary Findings. *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*. https://doi.org/https://doi.org/10.5281/zenodo.1417637

Lenssen, N., & Needell, D. (2013). *On the Mathematics of Music: From Chords to Fourier Analysis*. https://arxiv.org/pdf/1306.2859

Lerch, A. (2021). *Audio Content Analysis*.

Lerch, A. (2022). *An Introduction to Audio Content Analysis* (first). Wiley. https://doi.org/10.1002/9781119890980

Leskovec, J., Rajaraman, A., & Ullman, J. D. (2020). *Mining of Massive Datasets* (3rd ed.). Cambridge University Press. https://doi.org/10.1017/9781108684163

Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2022). A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(12), 6999–7019. https://doi.org/10.1109/TNNLS.2021.3084827

Lippens, S., Martens, J. P., & De Mulder, T. (2004). A comparison of human and automatic musical genre classification. *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, *4*, iv-233-iv–236. https://doi.org/10.1109/ICASSP.2004.1326806

Liu, H., Liu, X., Kong, Q., Wang, W., & Plumbley, M. D. (2022). Learning Temporal Resolution in Spectrogram for Audio Classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, *38*(12), 13873–13881. https://doi.org/10.1609/aaai.v38i12.29294

Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013). Rectifier Nonlinearities Improve Neural Network Acoustic Models. *Proceedings of the 30th International Conference on Machine Learning*, *28*(3). https://ai.stanford.edu/~amaas/papers/relu_hybrid_icml2013_final.pdf

Mandic, D. P. ., & Chambers, J. A. . (2001). *Recurrent neural networks for prediction : learning algorithms, architectures, and stability*. John Wiley.

Martin, K. D., Scheirer, E. D., & Vercoe, B. L. (1998). Musical content analysis through models of audition. *ACM Multimedia Workshop on Content Processing of Music for Multimedia Applications*, *12*.

Masuyama, Y., Ueno, N., & Ono, N. (2025). Mel-Spectrogram Inversion via Alternating Direction Method of Multipliers. *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. https://doi.org/10.1109/ICASSP49660.2025.10887761

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115–133. https://doi.org/10.1007/BF02478259/METRICS

McFee, B. (2023). *Digital Signals Theory* (1st ed.). Chapman and Hall/CRC. https://doi.org/10.1201/9781003264859

Mckay, C., & Fujinaga, I. (2005). Automatic music classification and the importance of instrument identification Introduction to automated music classification. *Proceedings of the Conference on Interdisciplinary Musicology*, 1–10.

Mckay, C., & Fujinaga, I. (2006). Musical genre classification: Is it worth pursuing and how can it be improved? *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006)*, 101–106. https://doi.org/https://doi.org/10.5281/zenodo.1417417

Melchior, V. (2019). High-Resolution Audio: A History and Perspective. *Journal of the Audio Engineering Society*, *67*(5), 246–257. https://doi.org/10.17743/jaes.2018.0056

Mitra, S. K. (1998). *Digital signal processing : a computer-based approach*. McGraw-Hill.

Mitra, S. K., & Kaiser, J. F. (1993). *Handbook for Digital Signal Processing* (1st ed.). Wiley-Interscience.

Moore, B. (2013). *An Introduction to the Psychology of Hearing* (6th ed.). Brill.

Müller, M. (2021). Fundamentals of music processing: Using Python and Jupyter notebooks. In *Fundamentals of Music Processing: Using Python and Jupyter Notebooks*. Springer. https://doi.org/10.1007/978-3-030-69808-9

Müller, M., Ellis, D. P. W., Klapuri, A., & Richard, G. (2011). Signal processing for music analysis. *IEEE Journal on Selected Topics in Signal Processing*, *5*(6), 1088–1110. https://doi.org/10.1109/JSTSP.2011.2112333

Namgyal, T., Hepburn, A., Santos-Rodriguez, R., Laparra, V., & Malo, J. (2024). *The Effect of Perceptual Metrics on Music Representation Learning for Genre Classification*. http://arxiv.org/abs/2409.17069

Neumayer, R., & Rauber, A. (2007). Integration of Text and Audio Features for Genre Classification in Music Information Retrieval. In G. Amati, G. Romano, & C. Carpineto (Eds.), *Advances in Information Retrieval* (pp. 724–727). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-71496-5_78

Noll, A. M. (1967). Cepstrum Pitch Determination. *The Journal of the Acoustical Society of America*, *41*(2), 293–309. https://doi.org/10.1121/1.1910339

North, A. C., & Hargreaves, D. J. (1997). Liking for Musical Styles. *Musicae Scientiae*, *1*(1), 109–128. https://doi.org/10.1177/102986499700100107

Oppenheim, A. V., & Schafer, R. W. (2004). Dsp history - From frequency to quefrency: a history of the cepstrum. *IEEE Signal Processing Magazine*, *21*(5), 95–106. https://doi.org/10.1109/MSP.2004.1328092

Oppenheim, A., Willsky, A., & Nawab, S. (1996). *Signals and Systems* (2nd ed.). Pearson.

Oppenheim, A. V. (1965). *Superposition in a class of nonlinear systems*.

O'Shaughnessy, Douglas. (2000). *Speech communications : human and machine* (2nd ed.). IEEE Press.

Pachet, F., & Cazaly, D. (2000). A Taxonomy of Musical Genres. *Proceedings of the 6th International Conference on Computer-Assisted Information Retrieval (RIAO 2000)*, 1238–1245.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., Devito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 8026–8037. https://doi.org/10.5555/3454287.3455008

Perumal, T., Mustapha, N., Mohamed, R., & Shiri, F. M. (2024). A Comprehensive Overview and Comparative Analysis on Deep Learning Models. *Journal on Artificial Intelligence*, *6*(1), 301–360. https://doi.org/10.32604/jai.2024.054314

Polson, N., & Sokolov, V. (2023). *Deep Learning: A Tutorial*. https://arxiv.org/pdf/2310.06251

Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., Shyu, M. L., Chen, S. C., & Iyengar, S. S. (2019). A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys*, *51*(5), 92. https://doi.org/10.1145/3234150;PAGE:STRING:ARTICLE/CHAPTER

Proakis, J. G. ., & Manolakis, D. G. . (2013). *Digital signal processing* (4th ed.). Pearson.

Purves, D., Paydarfar, J. A., & Andrews, T. J. (1996). The wagon wheel illusion in movies and reality. *Proceedings of the National Academy of Sciences*, *93*(8), 3693–3697. https://doi.org/10.1073/pnas.93.8.3693

Purwins, H., Li, B., Virtanen, T., Schluter, J., Chang, S.-Y., & Sainath, T. (2019). Deep Learning for Audio Signal Processing. *IEEE Journal of Selected Topics in Signal Processing*, *13*(2), 206–219. https://doi.org/10.1109/JSTSP.2019.2908700

Rabiner, L. R. ., & Schafer, R. W. . (1978). *Digital processing of speech signals*. Prentice-Hall.

Rabiner, L. R. ., & Schafer, R. W. . (2011). *Theory and applications of digital speech processing* (1st ed.). Pearson.

Raina, R., Madhavan, A., & Ng, A. Y. (2009). Large-scale deep unsupervised learning using graphics processors. *ACM International Conference Proceeding Series*, *382*. https://doi.org/10.1145/1553374.1553486;PAGE:STRING:ARTICLE/CHAPTER

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386–408.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536. https://doi.org/10.1038/323533A0;KWRD

Rumsey, F., & McCormick, T. (2021). *Sound and Recording* (8th ed.). Routledge. https://doi.org/10.4324/9781003092919

Sahidullah, M., & Saha, G. (2012). Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition. *Speech Communication*, *54*(4), 543–565. https://doi.org/10.1016/j.specom.2011.11.004

Sarker, I. H. (2021). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Computer Science 2021 2:6*, *2*(6), 420-. https://doi.org/10.1007/S42979-021-00815-1

Schlüter, J., & Böck, S. (2014). Improved musical onset detection with Convolutional Neural Networks. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 6979–6983. https://doi.org/10.1109/ICASSP.2014.6854953

Schuller, B., Hagel, C., Schuller, D., & Rigoll, G. (2010). “Mister D.j., Cheer me Up!”: Musical and textual features for automatic mood classification. *Journal of New Music Research*, *39*(1), 13–34. https://doi.org/10.1080/09298210903430475

Sethares, W. (2005). *Tuning, Timbre, Spectrum, Scale* (2nd ed.). Springer London. https://doi.org/10.1007/b138848

Shao, X., Xu, C., & Kankanhalli, M. S. (2003). Applying neural network on the content-based audio classification. *ICICS-PCM 2003 - Proceedings of the 2003 Joint Conference of the 4th International Conference on Information, Communications and Signal Processing and 4th Pacific-Rim Conference on Multimedia*, *3*, 1821–1825. https://doi.org/10.1109/ICICS.2003.1292781

Shinde, P. P., & Shah, S. (2018). A Review of Machine Learning and Deep Learning Applications. *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, *3*, 1–6. https://doi.org/10.1109/ICCUBEA.2018.8697857

Shuvaev, S., Giaffar, H., & Koulakov, A. A. (2017). *Representations of Sound in Deep Learning of Audio Features from Music*. https://arxiv.org/pdf/1712.02898

Sigtia, S., & Dixon, S. (2014). Improved music feature learning with deep neural networks. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6959–6963. https://doi.org/10.1109/ICASSP.2014.6854949

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. https://arxiv.org/pdf/1409.1556

Smith, Steven. (2013). *Digital Signal Processing: a Practical Guide for Engineers and Scientists* (1st ed.). Newnes.

Soori, M., Arezoo, B., & Dastres, R. (2023). Artificial intelligence, machine learning and deep learning in advanced robotics, a review. *Cognitive Robotics*, *3*, 54–70. https://doi.org/10.1016/j.cogr.2023.04.001

Srivastava, N., Hinton, G., Krizhevsky, A., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958.

Steiglitz, K. (1996). *A Digital Signal Processing Primer Whit Applications to Digital Audio and Computer Music* (1st ed.). Benjamin/Cummings.

Stevens, S. S., Volkmann, J., & Newman, E. B. (1937). A Scale for the Measurement of the Psychological Magnitude Pitch. *The Journal of the Acoustical Society of America*, *8*(3), 185–190. https://doi.org/10.1121/1.1915893

Strang, G., & Nguyen, T. (1996). *Wavelets and Filter Banks* (2nd ed.). Wellesley-Cambridge Press.

Sujatha, C. (2023). *Vibration, Acoustics and Strain Measurement* (1st ed.). Springer International Publishing. https://doi.org/10.1007/978-3-031-03968-3

Toyoshima, I., Okada, Y., Ishimaru, M., Uchiyama, R., & Tada, M. (2023). Multi-Input Speech Emotion Recognition Model Using Mel Spectrogram and GeMAPS. *Sensors*, *23*(3), 1743. https://doi.org/10.3390/s23031743

Tsalera, E., Papadakis, A., & Samarakou, M. (2021). Comparison of Pre-Trained CNNs for Audio Classification Using Transfer Learning. *Journal of Sensor and Actuator Networks 2021, Vol. 10, Page 72*, *10*(4), 72. https://doi.org/10.3390/JSAN10040072

Tzanetakis, G., & Cook, P. (2002). Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, *10*(5), 293–302. https://doi.org/10.1109/TSA.2002.800560

Tzanetakis, G., Essl, G., & Cook, P. (2001). Automatic Musical Genre Classification Of Audio Signals. *2nd International Symposium on Music Information Retrieval (ISMIR)*, 205–210. https://ismir2001.ismir.net/pdf/tzanetakis.pdf

Vaseghi, S. V. (2007). *Multimedia Signal Processing*. Wiley. https://doi.org/10.1002/9780470066508

Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). *Attention Is All You Need*. 1. https://arxiv.org/pdf/1706.03762

Wold, E., Blum, T., Keislar, D., & Wheaten, J. (1996). Content-based classification, search, and retrieval of audio. *IEEE Multimedia*, *3*(3), 27–36. https://doi.org/10.1109/93.556537

Wu, W., Han, F., Song, G., & Wang, Z. (2018). Music Genre Classification Using Independent Recurrent Neural Network. *Proceedings 2018 Chinese Automation Congress, CAC 2018*, 192–195. https://doi.org/10.1109/CAC.2018.8623623

Xiao, T., & Zhu, J. (2023). *Introduction to Transformers: an NLP Perspective*. https://arxiv.org/pdf/2311.17633

Xu, B., Wang, N., Kong, H., Chen, T., & Li, M. (2015). *Empirical Evaluation of Rectified Activations in Convolutional Network*. https://arxiv.org/pdf/1505.00853

Xu, W. (2024). Music genre classification using deep learning: a comparative analysis of CNNs and RNNs. *Applied Mathematics and Nonlinear Sciences*, *9*(1). https://doi.org/10.2478/amns-2024-3309

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X. (Andrew), Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., & Woodland, P. (2006). *The HTK Book (for HTK Version 3.4)*. Cambridge University Engineering Department.

Zain, S. B. (2024). The Physics of Sound and Music: A complete course text (Textbook). In *The Physics of Sound and Music: A complete course text (Textbook)* (Vol. 1). Institute of Physics Publishing. https://doi.org/10.1088/978-0-7503-5212-3

Zaman, K., Sah, M., Direkoglu, C., & Unoki, M. (2023). A Survey of Audio Classification Using Deep Learning. *IEEE Access*, *11*, 106620–106649. https://doi.org/10.1109/ACCESS.2023.3318015

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2024). *Dive into deep learning*. Cambridge University Press.

Zhang, G., Wang, C., Xu, B., & Grosse, R. (2018). Three Mechanisms of Weight Decay Regularization. *7th International Conference on Learning Representations, ICLR 2019*. https://arxiv.org/pdf/1810.12281

Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*(4). https://doi.org/10.1002/WIDM.1253

Παράρτημα Α: Παρουσίαση των αρχείων του κώδικα

Τα αποτελέσματα των πειραμάτων που εκτελέστηκαν, μαζί με τον αντίστοιχο κώδικα, είναι διαθέσιμα στο αποθετήριο του GitHub και συγκεκριμένα στη διεύθυνση: https://github.com/parisppdo/pytorch-genre-classifiers. Το αποθετήριο περιέχει το σύνολο των Jupyter notebooks που περιέχουν τον κώδικα που εκτελέστηκε για την πραγματοποίηση των πειραμάτων, καθώς και κατάλληλα σχόλια. Ακολουθεί μια περιγραφή της οργάνωσης των αρχείων:

Φάκελος "MFCC Classifiers"

Στον συγκεκριμένο φάκελο βρίσκονται τα Jupyter notebooks των πειραμάτων με τα MFCCs ως δεδομένα εισόδου. Επίσης, βρίσκεται το αρχείο "preprocess.py" το οποίο περιέχει τον κώδικα που αποθηκεύει τα δεδομένα σε αρχείο JSON. Εάν τρέξουμε το αρχείο αυτό, θα δημιουργηθεί ένα αρχείο JSON εντός του φακέλου. Μπορούμε να μεταφορτώσουμε τα δεδομένα αυτά στο κάθε Jupyter notebook.

Φάκελος "Mel Spectrograms"

Στον συγκεκριμένο φάκελο βρίσκονται τα Jupyter notebooks των πειραμάτων με τα Mel Spectrograms ως δεδομένα εισόδου. Ο φάκελος αυτός αντιστοιχεί στα Mel Spectrograms με καλύτερη ανάλυση στο επίπεδο της συχνότητας. Η δομή είναι ακριβώς ίδια με τον φάκελο "MFCC Classifiers".

Φάκελος "AST-Tuned Mel Spectrogram Classifiers"

Στον συγκεκριμένο φάκελο εμπεριέχονται τα πειράματα τα οποία λαμβάνουν χώρα με δεδομένα εισόδου από το πρότυπο AST. Εντός του φακέλου υπάρχουν 2 φάκελοι ονόματι "10 segment division" και "3 segment division". Καθένας αντιστοιχεί στη διαφορετική τμηματοποίηση των ηχητικών δειγμάτων, όπως ορίζει το όνομά του.

Καθένας από τους εν λόγω φακέλους έχει τα αντίστοιχα Jupyter notebooks για τα πειράματα της κάθε αρχιτεκτονικής. Εντός του κάθε φακέλου, υπάρχουν 3 Python scripts:

- **preprocess_AST_extractor_to_pt.py**: Το εν λόγω Python script αποθηκεύει σε μορφή .pt τα δεδομένα που πρόκειται να χρησιμοποιηθούν με το μοντέλο AST καθώς για την εξαγωγή των δεδομένων χρησιμοποιεί την αντίστοιχη κλάση του μοντέλου από το Hugging Face.

- **preprocess_with_AST_options.py**: Το συγκεκριμένο Python script αποθηκεύει σε μορφή JSON τα δεδομένα που πρόκειται να χρησιμοποιηθούν με τα μοντέλα εκτός του AST. Θα εξάγει Mel Spectrograms τα οποία έχουν τις ίδιες ρυθμίσεις με αυτές του AST.
- **preprocess_original.py**: Το συγκεκριμένο Python script αποθηκεύει σε μορφή JSON τα δεδομένα που πρόκειται να χρησιμοποιηθούν με τα μοντέλα εκτός του AST. Θα εξάγει Mel Spectrograms τα οποία έχουν τις ίδιες ρυθμίσεις τα αρχεία του φακέλου “Mel Spectrograms”. Χρησιμοποιείται για πιθανή σύγκριση.

Λοιπά σχόλια

Περισσότερες λεπτομέρειες για τη δομή και τα αρχεία που βρίσκονται στο αποθετήριο, βρίσκονται στο αρχείο README.md που είναι επίσης ανεβασμένο.

Υπεύθυνη Δήλωση Συγγραφέα:

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν.1599/1986, η παρούσα εργασία αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης.