Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

3η Εργαστηρική Άσκηση

Ανγνώριση Προτύπων

Αλεξανδρόπουλος Σταμάτης

03117060
el17060@mail.ntua.com

Γκότση Πολυτίμη Άννα

03117201
el17201@mail.ntua.com

Περιεχόμενα

Εισαγωγή

Στόχος αυτής της εργαστηριακής άσκησης είναι η αναγνώριση του είδους και η εξαγωγή συναισθημάτων από φασματογραφήματα μουσικών κομματιών. Σε αυτή την εργασία θα χρησιμοποιήσουμε 2 σύνολα δεδομένων:

- Το Free Music Archive (FMA), το οποίο περιέχει 3834 δείγματα χωρισμένα σε 20 κλάσεις (είδη μουσικής)
- Τη βάση δεδομένων multitask music, που περιέχει 1497 δείγματα με labels για τις τιμές συναισθηματικών διαστάσεων όπως Valence, Energy, Danceability

Γενικά, τα δείγματα είναι φασματογραφήματα τα οποία έχουν εξαχθεί από clips 30 δευτερολέπτων από διαφορετικά τραγούδια. Στην άσκηση αυτή θα ασχοληθούμε με την ανάλυση φασματογραφημάτων με τη χρήση βαθιών αρχιτεκτονικών με τη συνελικτικά δίκτυα (CNN) και αναδρομικά νευρωνικά δίκτυα (RNN). Τέλος, σαν προγραμματιστικό περιβάλλον χρησιμοποιούμε το Kaggle.

Βήμα 0: Εξοικείωση με Kaggle kernels

Αρχικά ανοίγουμε ένα (private) Kaggle kernel στην σελίδα του Multitask Affective Music Classification 2022 και φορτώνουμε τα δεδομένα όπως περιγράφεται στο notebook. Στην συνέχεια τρέχουμε την εντολή os.listdir("../input/patreco3-multitask-affective-music/data/") και παρατηρούμε ότι έχουμε 4 φακέλους:

- multitask_dataset_beat
- fma_genre_spectrograms
- fma_genre_spectrograms_beat
- multitask_dataset

Βήμα 1: Εξοικείωση με φασματογραφήματα στην κλίμακα mel

Για τα πρώτα βήματα της εργαστηριακής άσκησης χρησιμοποιούμε το FMA dataset. Αυτό είναι μια βάση δεδομένων από ελεύθερα δείγματα (clips) μουσικής με επισημειώσεις ως προς το είδος της μουσικής.

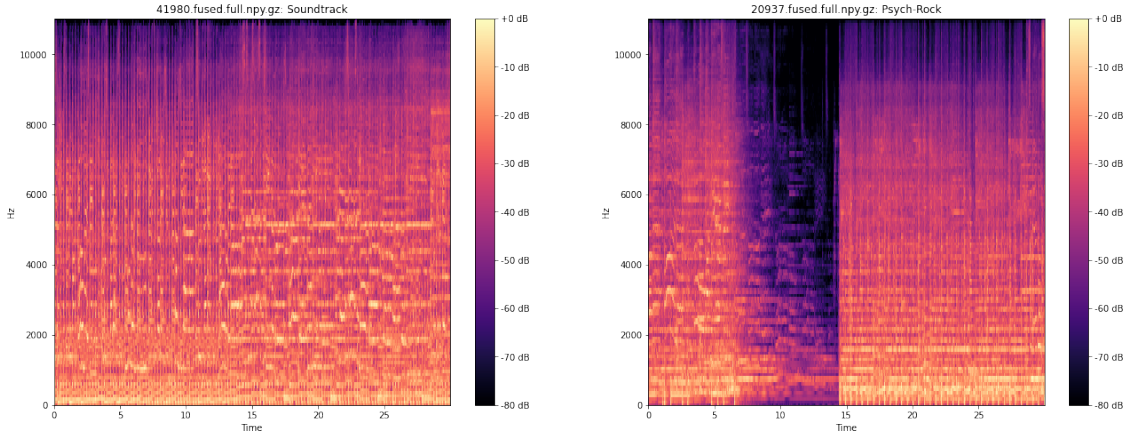
α) Αρχικά επιλέγουμε δύο τυχαίες γραμμές με διαφορετικά labels από το σύνολο δεδομένων εκπαίδευσης "../input/data/data/ fma_genre_spectrograms/train_labels.txt". Το αποτέλεσμα είναι το εξής:

Labels chosen: Soundtrack Psych-Rock
Spec files chosen: 41980.fused.full.npy.gz 20937.fused.full.npy.gz

β) Στην συνέχεια διαβάζουμε τα αρχεία. Αυτά, όπως φαίνεται παραπάνω, είναι σε μορφή .npy μορφή και αποτελούν concatenated mel spectrograms και chroma frequencies και προκύπτουν με την εφαρμογή ενός παραθύρου στον ήχο που έχουμε. Με βάση τις οδηγίες που μας δόθηκαν, λαμβάνουμε λοιπόν τα φασματογράφημα σε κλίμακα mel για να μπορέσουμε να παρατηρήσουμε το φάσμα των συχνοτήτων με την πάροδο του χρόνου. Για κάθενα ένα από τα δύο mel spectrograms που λαμβάνουμε οι διαστάσεις τους είναι αντίστοιχα:

(128, 1291)
(128, 1291)

γ) Με τη βοήθεια της συνάρτησης librosa.display.specshow απεικονίζουμε τα δεδομένα μας:



Παρατηρούμε ότι διαφορετικά είδη μουσικής έχουν και αρκετές διαφορές στα φασματογραφήματα. Συγκεκριμένα στην περίπτωσή μας, τα δύο δείγματα που μελετήσαμε ανήκουν στις κατηγορίες Soundtrack και Psych-Rock. Το κομμμάτι που ανήκει στα Soundtrack φαίνεται να συγκεντρώνει την ενέργεια του γύρω από συγκεκριμένες ζώνες συχνοτήτων. Αυτό οφείλεται στο ότι γενικά τα περισσότερα soundtracks έχουν ομαλές μεταβάσεις από το ένα μέτρο στο άλλο και περιοδικούς ηχούς. Αντίθετα το κομμάτι Psych-Rock στο σύνολο του έχει πιο ¨άναρχη' μορφή και αρκέτα spikes, πράγμα που οφείλεται στα όργανα που την απαρτίζουν. Παρ΄ όλ΄ αυτά τα τμήματα του κομματιού που έχουν υψηλή ένταση (πιο κόκκινα) εμφανίζουν μια περιοδικότητα.

Βήμα 2: Συγχρονισμός φασματογραφημάτων στο ρυθμό της μουσικής (beat-synced spectrograms)

α) Οι διαστάσεις των φασματογραφημάτων του βήματος 1 είναι:

Shape of spectrogram for Soundtrack: (128, 1291)
Shape of spectrogram for Psych-Rock: (128, 1291)

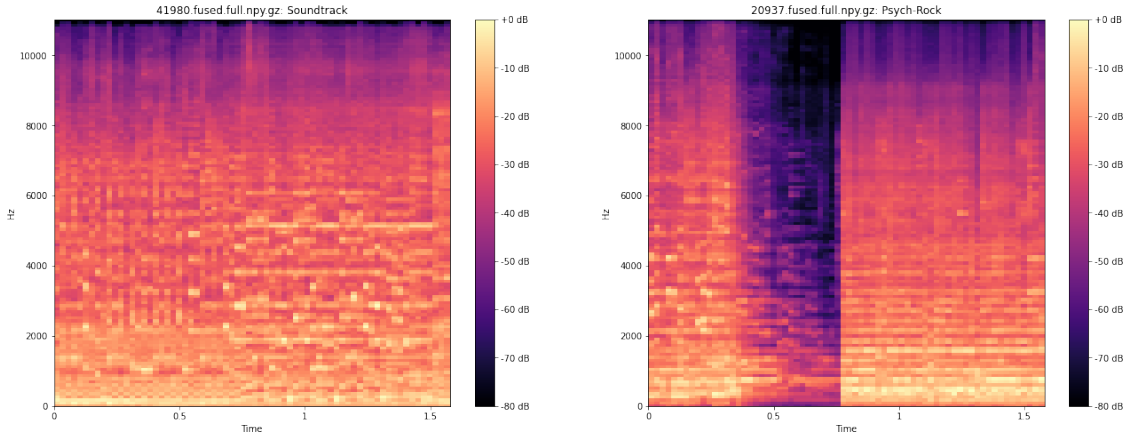
Παρατηρήσουμε ότι στα δύο μας είδη έχουμε τα ίδια μήκη για τα δείγματα. Βέβαια, θα μπορούσαμε να είχαμε και διαφορετικά μήκη αν διαλέγαμε άλλα δείγματα. Τα LSTM, όπως γνωρίζουμε, βασίζονται σε τένσορες που είναι ίδου μήκους. Συνεπώς, αν τα φασματογραμήματα μας δεν είχαν το ίδιο μήκος θα αναγκαζόμασταν να λύσουμε αυτό το πρόβλημα είτε μέσω zero-padding στο μέγιστο μήκος είτε κόβωντας πληροφορία από τα κομμάτια που υπερβαίνουν το ελάχιστο μήκος.

β) Ένας τρόπος μείωσης των χρονικών βημάτων ειτυγχάνεται με τον συγχρονισμό τον φασματογραφημάτων τους πάνω στο ρυθμό. Για το σκοπό αυτό, λαμβάνουμε τη διάμεσο median ανάμεσα στα σημεία που χτυπάει το beat της μουσικής. Αυτή η διαδικασία έχει γίνει ήδη στο εργαστήριο και τα αντίστοιχα αρχεία βρίσκονται στο φάκελο '../input/patreco3-multitask-affective-music/data/fma_genre_spectrograms_beat'. Ακολουθώντας την ίδια διαδικασία με το βήμα 1 παρατηρούμε ότι πλέον, για κάθενα ένα από τα δύο αυτά αρχεία το σχήμα του mel spectrogram είναι αντίστοιχα:

(128, 68)
(128, 68)

Βλέπουμε ότι υπάρχει μια τεράστια μείωση τη διάστασης των χρονικών βημάτων, περίπου 95%. Στην περίπτωση μας, έτυχε τα δύο είδη να έχουν ίδιο αριθμό χρονικών βημάτων. Αυτό πιθανώς οφείλεται στην περιοδικότητα που έχουν τα δύο κομμάτια. Στην γενική περίπτωση, όμως, δεν ισχύει πάντα κάτι τέτοιο. Μπορούμε να υποθέσουμε, είναι ότι τα υπόλοιπα τραγούδια του ίδιου είδους θα έχουν παρεμφερή χρονικά βήματα. Επιπλέον, από τη στιγμή που ο ρυθμός αποτελεί έναν από τους πιο σημαντικούς παράγοντες για την ποιότητα των χαρακτηριστικών μας, αναμένουμε πως δε θα χάσουμε πολλή πληροφορία με τη χρήση αυτού του συνόλου δεδομένων κατά την ταξινόμηση.

Τα φασματογραφήματα είναι τα εξής:



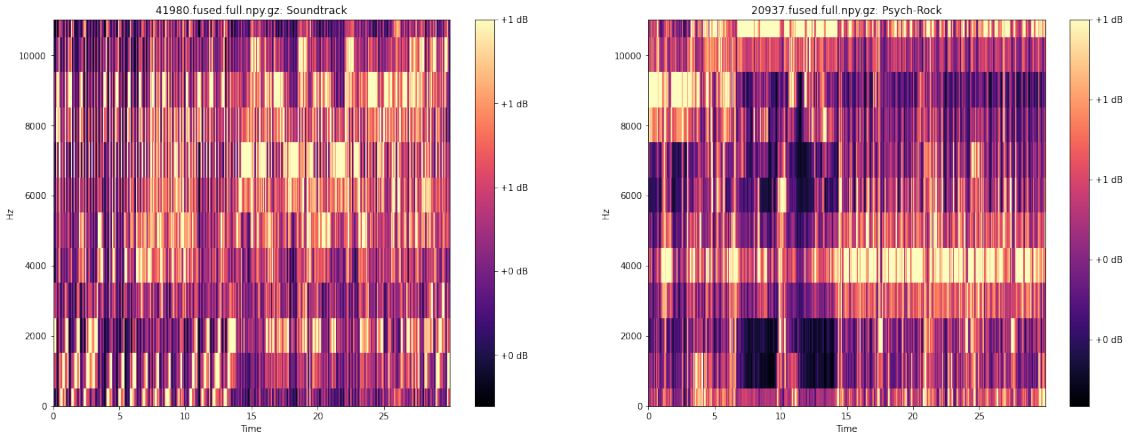
Απο την παραπάνω εικόνα βλέπουμε, ότι η διάσταση του χρόνου μειώθηκε και για τα δύο φασματογραφήματα, αφού πλέον φαίνεται ότι δεν παρέχουν τόσο καλή ανάλυση για τα σήματά μας. Παρ΄ όλ΄ αυτά, διατηρούν παρόμοια μορφή με πριν αφού η συγκέντρωση ενέργειας είναι σε παρόμοιες περιοχές και σημεία.

Βήμα 3: Εξοικείωση με χρωμογραφήματα

Τα χρωμογραφήματα απεικονίζουν την ενέργεια του σήματος μουσικής για τις ζώνες συχνοτήτων που αντιστοιχούν στις νότες μιας μουσικής οκτάβας. Είναι ένα ιδιαίτερα χρήσιμο εργαλείο στην ανάλυση μουσικών σημάτων, επειδή παρουσιάζει αρκετά καλά τα αρμονικά και μελωδικά χαρακτηριστικά του κομματιού και είναι εύρωστο στις αλλαγές του ηχοχρώματος και του οργάνου. Η διαδικασία εξαγωγής ηχοχρώματος έχει γίνει ήδη στο εργαστήριο. Ακολουθώντας την ίδια διαδικασία με το βήμα 1 παρατηρούμε ότι πλέον, για κάθενα ένα από τα δύο αυτά αρχεία το σχήμα του chromagram είναι αντίστοιχα:

Shape of chromagram for Soundtrack: (12, 1291)
Shape of chromagram for Psych-Rock: (12, 1291)

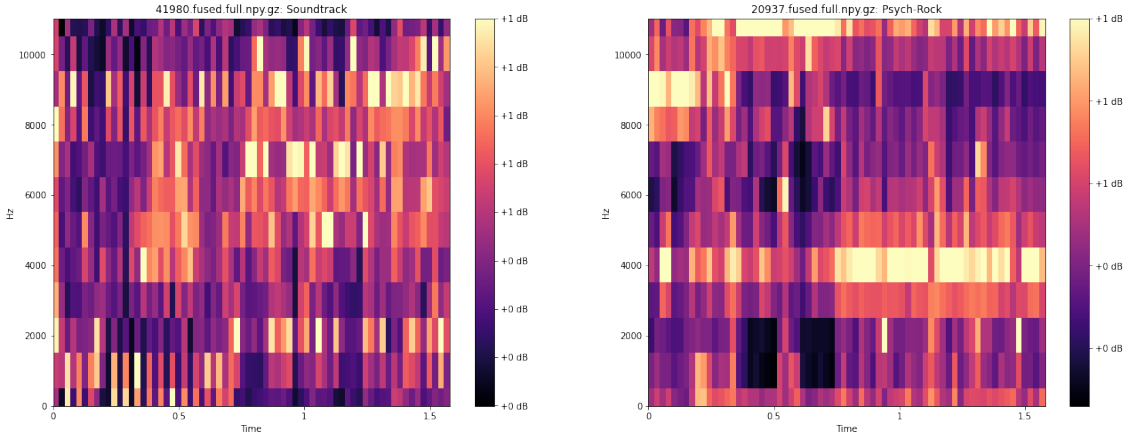
Τα χρωμογραφήματα είναι τα εξής:



Γενικά, ισχύουν παρόμοια συμπεράσματα με πριν. Αξίζει να παρατηρήσουμε πόσο δραστικά αλλάζουν οι συχνότητες των νοτών που χρησιμοποιούνται αναλόγως με το πόσο ομοιόμορφο είναι το χρωματογράφημά μας.
Στην συνέχεια, χρησιμοποιούμε την μέθοδο συγχρονισμού του βήματος 2 που στηρίζεται στο median ανάμεσα στα σημεία που χτυπάει το beat της μουσικής για να μειώσουμε την διαστατικότητα. Για κάθενα ένα από τα δύο αυτά αρχεία το σχήμα του chromagram είναι αντίστοιχα:

Shape of beat-synced chromagram for Soundtrack: (12, 68)
Shape of beat-synced chromagram for Psych-Rock: (12, 68)

Τα χρωμογραφήματα είναι τα εξής:



Όπως και προηγουμένως, η ανάλυση των χρωματογραφημάτων μας χειροτερεύει με τη χρήση λιγότερων χαρακτηριστικών αλλά διατηρείται η συγκέντρωση της ενέργειας σε περιοχές στοιχείων. Επομένως διατηρείται η σημαντική πληροφορία.

Βήμα 4: Φόρτωση και ανάλυση δεδομένων

Στο βήμα αυτό χρησιμοποιήσαμε τον έτοιμο κώδικα όπου παρέχεται έτοιμη μια υλοποίηση ενός PyTorch Dataset, η οποία διαβάζει τα δεδομένα και μας επιστρέφει τα δείγματα.

α) Από τον δοθέντα κώδικα αξίζει να μελετήσουμε ορισμένες συναρτήσεις και να σχολιάσουμε τη λειτουργία τους.

Σημειώνουμε ότι πραγματοποιούμε ορισμένες αλλαγές στον δοσμένο από το εργαστήριο κώδικα. Συγκεκριμένα, προσθέτουμε δύο συναρτήσεις αντίστοιχες της read_spectrogram, τις read_mel_spectrogram και read_chromagram, των οποίων η λειτουργία θα εξηγηθεί στη συνέχεια. Επιπλέον, αλλάζουμε την διαμόρφωση των τίτλων των αρχείων που διαβάζονται από τα δεδομένα μας, καθώς παρατηρούμε πως υπάρχουν διαφορές σε σχέση με τους σημιουργούμενους από τον κώδικα τίτλους και τους τίτλους στα δεδομένα μας.

Οι συναρτήσεις που περιέχονται είναι οι εξής:

- torch_train_val_split(): Με τη συνάρτηση αυτή πραγματοποιείται η διάσπαση των δεδομένων εκπαίδευσης σε δεδομένα εκπαίδευσης (train) και επικύρωσης (validation). Ουσιαστικά, παίρνει αρχικά το σύνολο των δεδομένων εκπαίδευσης και δημιουργεί loaders με μεγέθη που θέλουμε διαχωρίζοντας τα indices των αρχείων μας αφού πρώτα τα έχουμε ανακατέψει (shuffle (default=True)). Εάν η shuffle δεν λάβει την default τιμή True τότε τα δεδομένα που αποδίδονται στο validation set είναι πάντα τα τελευταία δεδομένα του συνόλου εκπαίδευσης. Τελικά αυτά τα ανακατεμένα indices χρησιμοποιούνται για να φορτωθούν στα αρχεία στους τυπικούς dataloader του torch. Αυτοί χρησιμοποιούνται για να μπορούμε κατά την εκπαίδευση να παίρνουμε στοιχεία μεγέθους batch_size που έχουμε ορίσει. Όσον αφορά την μεταβλητή seed (default=None), αυτή εάν λάβει οποιαδήποτε πραγματικό αριθμό φροντίζει κάθε εκτέλεση συνάρτησης που περιλαμβάνει τυχαιότητα να δίνει τα ίδια αποτελέσματα κάθε φορά που την τρέχουμε με ίδιες αρχικές συνθήκες του συστήματος.Η μεταβλητή αυτή είναι χρήσιμο να αρχικοποιείται σε κάποια πραγματική τιμή όσο κατασκευάζουμε το μοντέλο διότι η ύπαρξη τυχαιότητας (π.χ. στο split των δεδομένων) αποτελεί παράγοντα που επηρεάζει την απόδοση ενός μοντέλου. Όταν όμως εκτελούμε πραγματική εκπαίδευση ενός μοντέλου και αξιολόγηση του θέλουμε να λαμβάνουμε αποτελέσματα σε πραγματικά τυχαίες αρχικές συνθήκες προκεμένου να αποφεύγουμε την εισαγωγή bias στα μοντέλα μας και συνεπώς λανθασμένα αποτελέσματα.
- read_mel_spectrogram(): γυρίζει τα MFCC (128) χαρακτηριστικά του κάθε αρχείου.
- read_chromagram():γυρίζει τα 12 χαρακτηριστικά του χρώματος του κάθε αρχείου.
- read_spectrogram(): γυρίζει τα συνολικά 140 χαρακτηριστικά κάθε αρχείου.
- LabelTransformer(): η κλάση αυτή μετατρέπει τις κλάσεις μας από Στρινγς σε λίστα ακεραίων. Αντιστοιχεί, με άλλα λόγια κάθε κλάση με έναν ξεχωριστό αριθμό.
- PaddingTransform(): η κλάση αυτή είναι χρήσιμη για τα μοντέλα LSTM που θα εκπαιδεύσουμε στο επόμενο βήμα. Γενικά συμβάλλει στο να έχουν όλα τα στοιχεία μας το ίδιο μήκος κάνοντας zero-padding. Ο κυριότερος λόγος που το χρειαζόμαστε είναι για να να επιταχύνουμε την εκπαίδευση των μοντέλων μας, αφού πλέον μπορούμε να τα εκπαιδεύουμε σε batches.
- SpectrogramDataset(): Θα εξηγήσουμε τη χρήση του SpectrogramDataset έστω για τα train δεδομένα. Αρχικά αναλύουμε το αρχείο train_labels.txt ανά γραμμή μέσω της μεθόδου get_files_labels(). Κάθε γραμμή του αρχείου αυτού έχει τη μορφή X.fused.full.npy.gz, Y, όπου X το αποθηκευμένο αρχείο και Y το label. Για κάθε label κάνουμε το mapping με τη χρήση του class_mapping. Έτσι έχουμε files, labels: num_train_samples. Στη συνέχεια επεξεργαζόμαστε τα id από τη λίστα με τα αρχεία και φορτώνουμε τα κατάλληλα κάθε φορά χαρακτηριστικά με τη χρήση της np.load μέσω της read_spectrogram(), read_mel_spectrogram(), read_chromagram() αντίσοτιχα. Τέλος, καλούμε τις κλάσεις PaddingTransform(), LabelTransformer() για να πραγματοποιηθεί η απαραίτητη επεξεργασία στο Dataset.
- __getitem__(): η συγκεκριμένη συνάρτηση είναι υπεύθυνη να επιστρέφει για ένα δεδομένο εκπαίδευσης τα εξής μεγέθη:
  1. Την zero padded ακολουθία των δεδομένων εισόδου η οποία μάλιστα είναι μιας διάστασης
  2. Τα labels των δεδομένων εισόδου

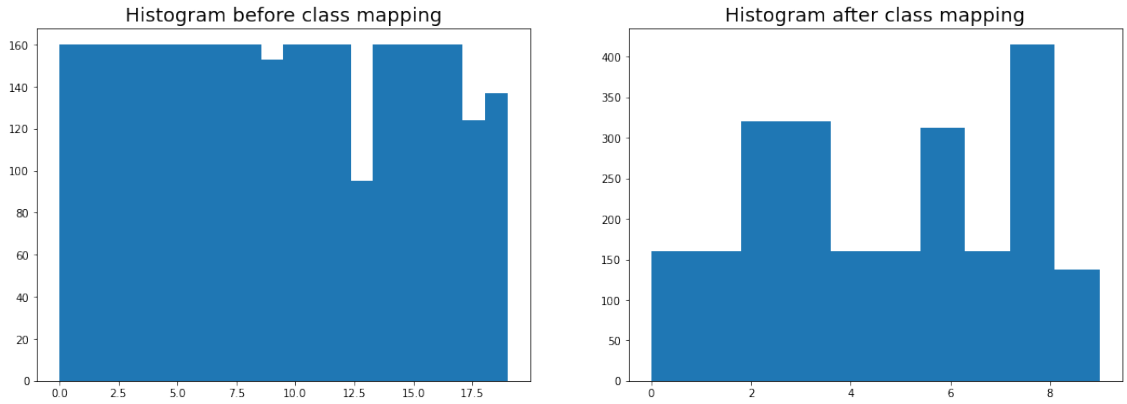
3. Το ελάχιστο ανάμεσα στο μέγεθος της ακολουθίας των δεδομένων εισόδου και του μέγιστο μήκους ακολουθίας που έχουμε επιλέξει. Αυτό είναι απαραίτητο ώστε κατά την εκπαίδευση του LSTM να μην λαμβάνονται υπ΄ όψιν στην ανανέωση των βαρών τα βήματα τα οποία περιλαμβάνουν 0 που οποία υπάρχουν λόγω του zero-padding

- __len__(): η συνάρτηση αυτή επιστρέφει το μήκος των labels.

β) Η αναγκαιότητα της συγχώνευσης των κλάσεων και «διαγραφή» αυτών που έχουν λίγα στοιχεία ταυτίζεται με την καλύτερη ταξινόμηση των στοιχείων. Γενικά αν έχουμε πολύ λίγα στοιχεία σε μια κλάση αλλά πολλές κλάσεις θα είναι δύσκολο να μάθει το μοντέλο μας και άρα να κάνει καλή ταξινόμηση. Γι' αυτό και είναι πιο επιθυμητό να μην έχουμε αυτές τις κλάσεις αλλά και λιγότερες συνολικά κλάσεις. Η συχώνευση που έχει γίνει είναι η εξής:

Category	Subcategory
Rock	Rock,Psych-Rock, Post-Rock
Folk	Folk, Psych-Folk
Metal	Metal, Punk
Trip-Hop	Trip-Hop
Pop	Pop
Electronic	Electronic, Chiptune
Blues	Blues
Jazz	Jazz
Classical	Classical

γ) Τα ιστογράμματα πριν και μετά τις συγχωνεύσεις είναι:



Από τα παραπάνω βλέπουμε τόσο πλεονεκτήματα όσο και μειονεκτήματα της συγχώνευσης. Αρχικά παρατηρούμε ότι ορισμένες κλάσεις πλέον έχουν σημαντικά παραπάνω δεδομένα εκπαίδευσης από άλλες. Παρ΄όλάυτά αυτές είναι πιο εύκολο να διαχωριστούν αυτές σε σχέση με τις πιο συγκεκριμένες που είχαμε προηγουμένως, αφού αρκετές έχουν πολύ παρόμοια χαρακτηριστικά και άρα είναι καλύτερο να τις εντάξουμε σε μια πιο γενική κατηγορία.

Βήμα 5: Αναγνώριση μουσικού είδους με ΛΣΤΜ

Στο βήμα αυτό ενεργοποιήσαμε τη gpu για τη γρηγορότερη εκτέλεση των πειραμάτων μας. Παρατηρούμε ότι μπορούμε να χρησιμοποιήσουμε το cuda.

```python
# setting device on GPU if available, else CPU
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
print('Using device:', device)
```

```
Using device: cuda
```

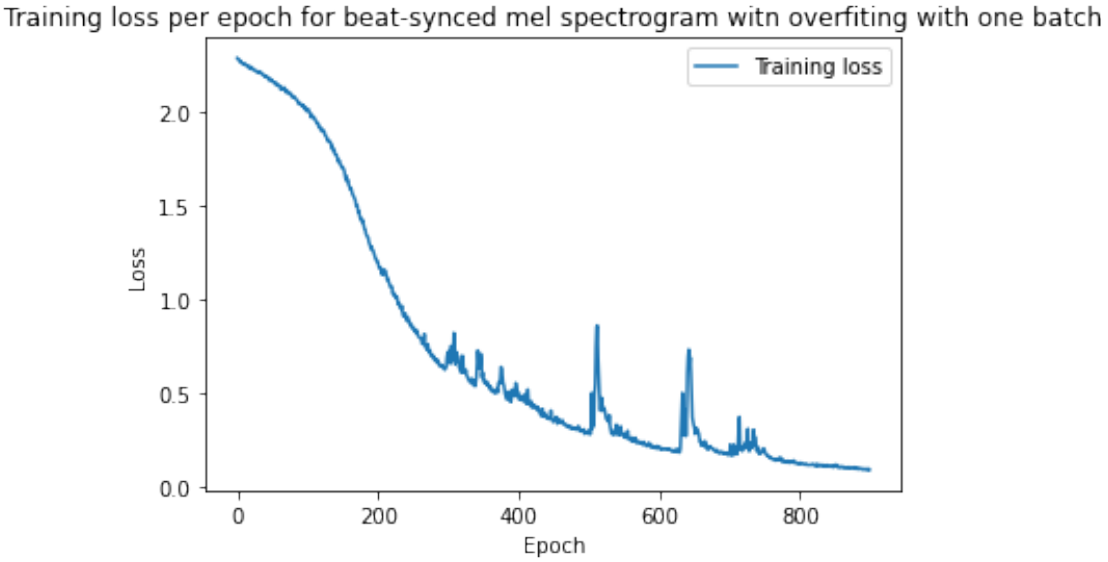
α) Βασιζόμενοι στον κώδικα της προηγούμενης εργαστηριακής άσκησης, τον τροποποιούμε για να δημιουργήσουμε ένα LSTM το οποίο είναι επιθυμητό για το σύνολο δεδομένων που έχουμε αλλά ταυτόχρονα μπορεί να χρησιμοποιηθεί τόσο με cpu όσο και gpu.

β) Προκειμένου να επιταχύνουμε τη διαδικασία ανάπτυξης και αποσφαλμάτωσης των μοντέλων, προσθέτουμε στην συνάρτηση train_model() που εκπαιδεύει το μοντέλο μας μια boolean παράμετρο overfit_batch. Όταν είναι False το δίκτυο εκπαιδεύεται κανονικά, αλλιώς πραγματοποιείται υπερεκπαίδευση του δικτύου σε ένα μικρό σύνολο από batches. Στο παράδειγμα μας χρησιμοποιούμε 1 batch και 900 εποχές. Το αποτέλεσμα που παίρνουμε είναι το εξής:

Epoch 0: Mean training loss per epoch: 2.291266441345215
Epoch 1: Mean training loss per epoch: 2.284403085708618

Epoch 2: Mean training loss per epoch: 2.2835097312927246
Epoch 897: Mean training loss per epoch: 0.09038498997688293
Epoch 898: Mean training loss per epoch: 0.09370958060026169
Epoch 899: Mean training loss per epoch: 0.08848758041858673

Έχουμε την ακόλουθη γραφική παράσταση:



Παρατηρούμε ότι το σφάλμα εκπαίδευσης τείνει στο μηδέν. Επίσης, παρακάτω φαίνονται μερικές χρήσιμες μετρικές, αξιολογώντας σε test δεδομένα:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	40
1	0.00	0.00	0.00	40
2	0.24	0.41	0.31	80
3	0.24	0.30	0.26	80
4	0.14	0.03	0.04	40
5	0.05	0.05	0.05	40
6	0.19	0.19	0.19	78
7	0.02	0.03	0.02	40
8	0.28	0.29	0.29	103
9	0.09	0.06	0.07	34
accuracy			0.19	575
macro avg	0.12	0.14	0.12	575
weighted avg	0.16	0.19	0.17	575

Διαπιστώνουμε πως αν και είχαμε πολύ μικρό loss στο τέλος της εκπαίδευσης, η επίδοση του ταξιοξομητή σε test δεδομένα δεν είναι καθόλου καλή, ενώ διαφέρει ανά κλάση. Αυτό είναι λογικό, αφού το LSTM που εκπαιδεύσαμε σε ένα μόνο batch είναι overfited στα δεδομένα εκπαίδευσης, με αποτέλεσμα να είναι biased προς συγκεκριμένες κλάσεις, να μην μπορεί να ξεχωρίσει άλλες οι οποίες ενδεχομένως δεν υπήρχαν καν ή υπήρχαν ελάχιστα στα δεδομένα εκπαίδευσης κοκ.

γ) Στο βήμα αυτό εκπαιδεύουμε ένα LSTM δίκτυο, το οποίο δέχεται ως είσοδο τα φασματογραφήματα του συνόλου εκπαίδευσης και προβλέπει τις διαφορετικές κλάσεις του συνόλου δεδομένων. Αξίζει να αναφέρομυε ότι σε όλα τα μοντέλα χρησιμοποιήσαμε τις εξής παραμέτρους για το μοντέλο:

- rnn_dim = 64
- input_dim = 128 length of mel spectogram
- output_dim = 10 number of classes
- num_layers = 2
- num_epochs = 60
- learning_rate = 1e-4
- bidirectional = True
- dropout = 0.4
- weight_decay = 1e-4
- early_stopping = True

Θέτοντας τον αριθμό των εποχών ίσο με 60 και χρησιμοποιώντας early stopping, έχουμε τα εξής αποτελέσματα:

Epoch 0: Mean training loss per epoch: 2.2791072360930906
Epoch 0: Mean validation loss per epoch: 2.252644733949141

Epoch 1: Mean training loss per epoch: 2.2418303412775837
Epoch 1: Mean validation loss per epoch: 2.2211387807672676

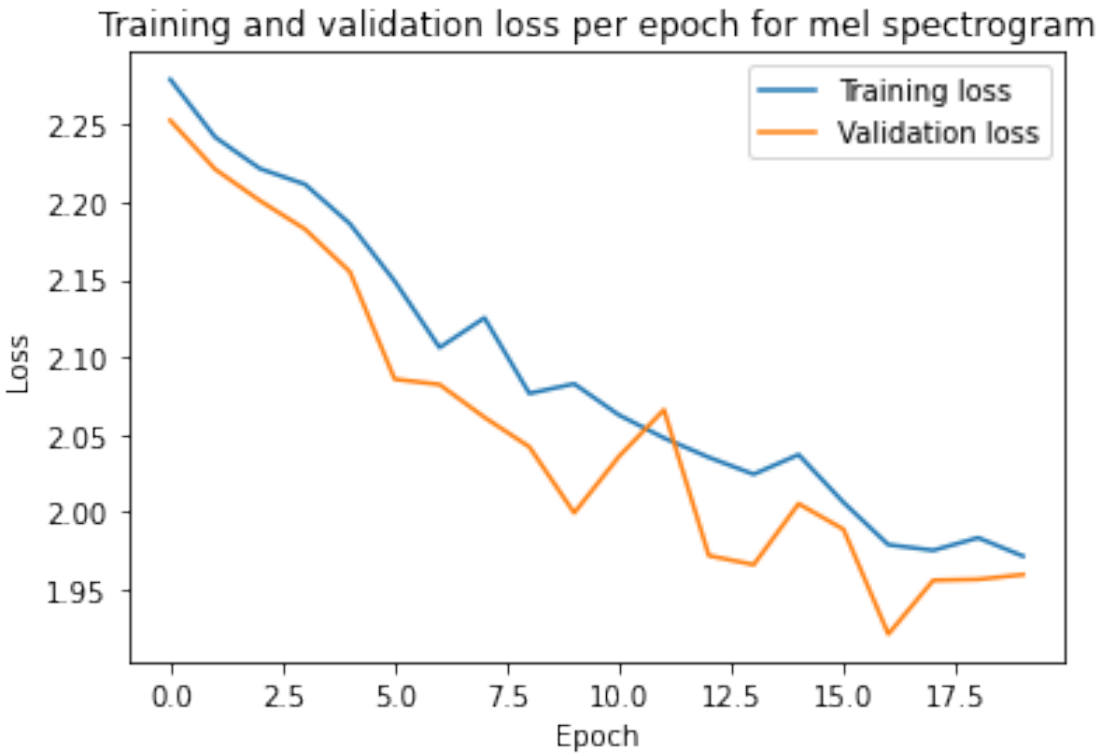
⋮

Epoch 19: Mean training loss per epoch: 1.9713482952887011
Epoch 19: Mean validation loss per epoch: 1.9593764110045

Number of times validation loss has not decreased: 3

Early stopping...

Το διάγραμμα που προκύπτει είναι το εξής:



δ) Στο βήμα αυτό εκπαιδεύουμε ένα LSTM δίκτυο, το οποίο δέχεται ως είσοδο τα beat-synced spectrograms του συνόλου εκπαίδευσης και προβλέπει τις διαφορετικές κλάσεις του συνόλου δεδομένων. Θέτοντας τον αριθμό των εποχών ίσο με 60 και early stopping, έχουμε τα εξής αποτελέσματα:

Epoch 0: Mean training loss per epoch: 2.2838137111356183
Epoch 0: Mean validation loss per epoch: 2.2560584111647173

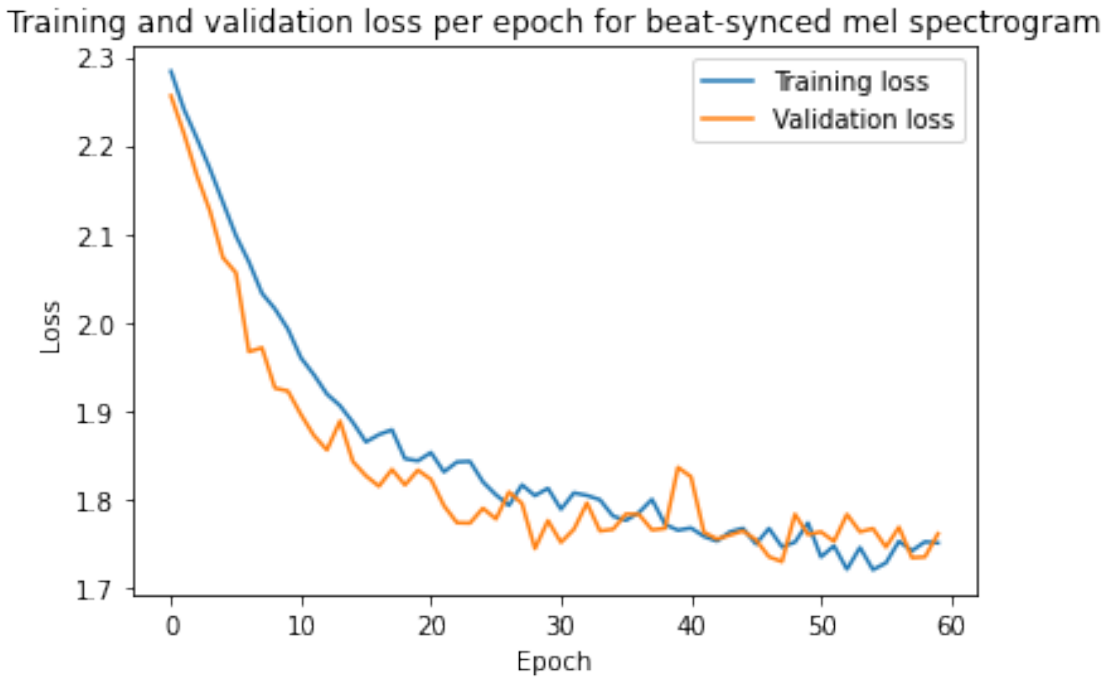
Epoch 1: Mean training loss per epoch: 2.240862569501323
Epoch 1: Mean validation loss per epoch: 2.21335675499656

⋮

Epoch 59: Mean training loss per epoch: 1.7503420922064012
Epoch 59: Mean validation loss per epoch: 1.7607851787046953

Number of times validation loss has not decreased: 2

Το διάγραμμα που προκύπτει είναι το εξής:



ε) Στο βήμα αυτό εκπαιδεύουμε ένα LSTM δίκτυο, το οποίο δέχεται ως είσοδο τα χρωμογραφήματα του συνόλου εκπαίδευσης και προβλέπει τις διαφορετικές κλάσεις του συνόλου δεδομένων. Θέτοντας τον αριθμό των εποχών ίσο με 60 και early stopping, έχουμε τα εξής αποτελέσματα:

Epoch 0: Mean training loss per epoch: 2.3099416417460286
Epoch 0: Mean validation loss per epoch: 2.3021902821280738

Epoch 1: Mean training loss per epoch: 2.285534808712621
Epoch 1: Mean validation loss per epoch: 2.264400460503318

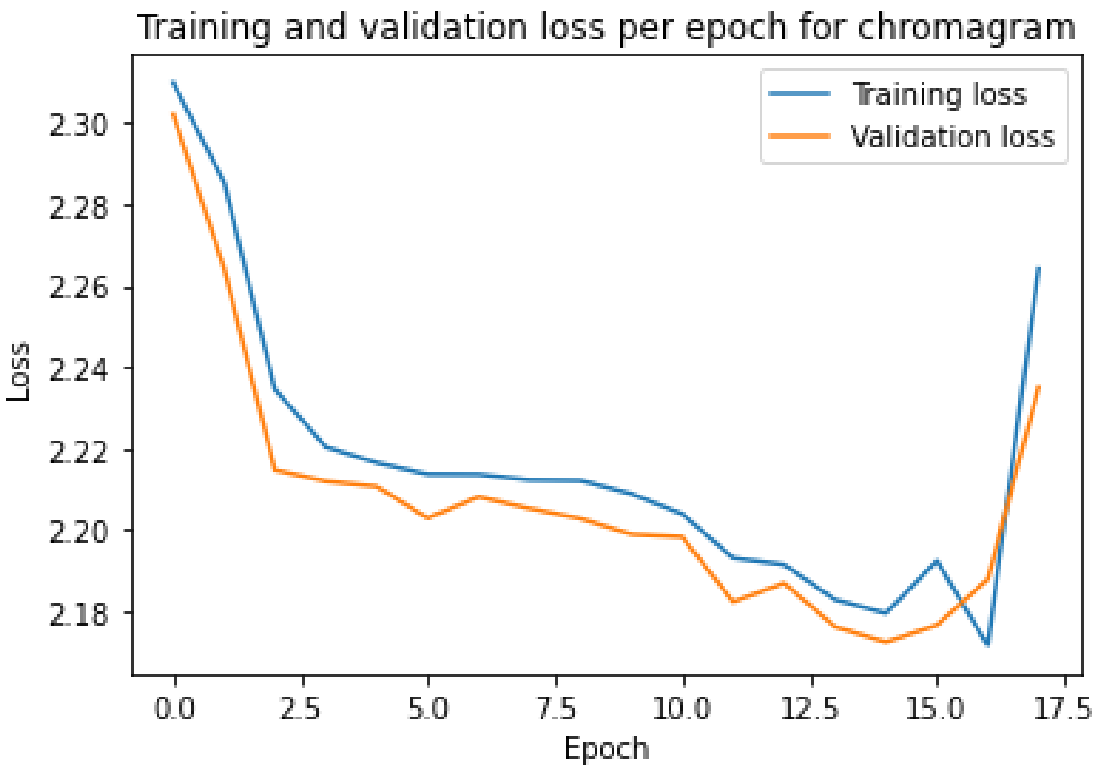
⋮

Epoch 17: Mean training loss per epoch: 2.264242626005603
Epoch 17: Mean validation loss per epoch: 2.235049074346369

Number of times validation loss has not decreased: 3

Early stopping...

Το διάγραμμα που προκύπτει είναι το εξής:



Για τα beat-synced chromograms έχουμε:

Epoch 0: Mean training loss per epoch: 2.3152841060392317
Epoch 0: Mean validation loss per epoch: 2.301451639695601

Epoch 1: Mean training loss per epoch: 2.2812094188505605
Epoch 1: Mean validation loss per epoch: 2.2488734938881616

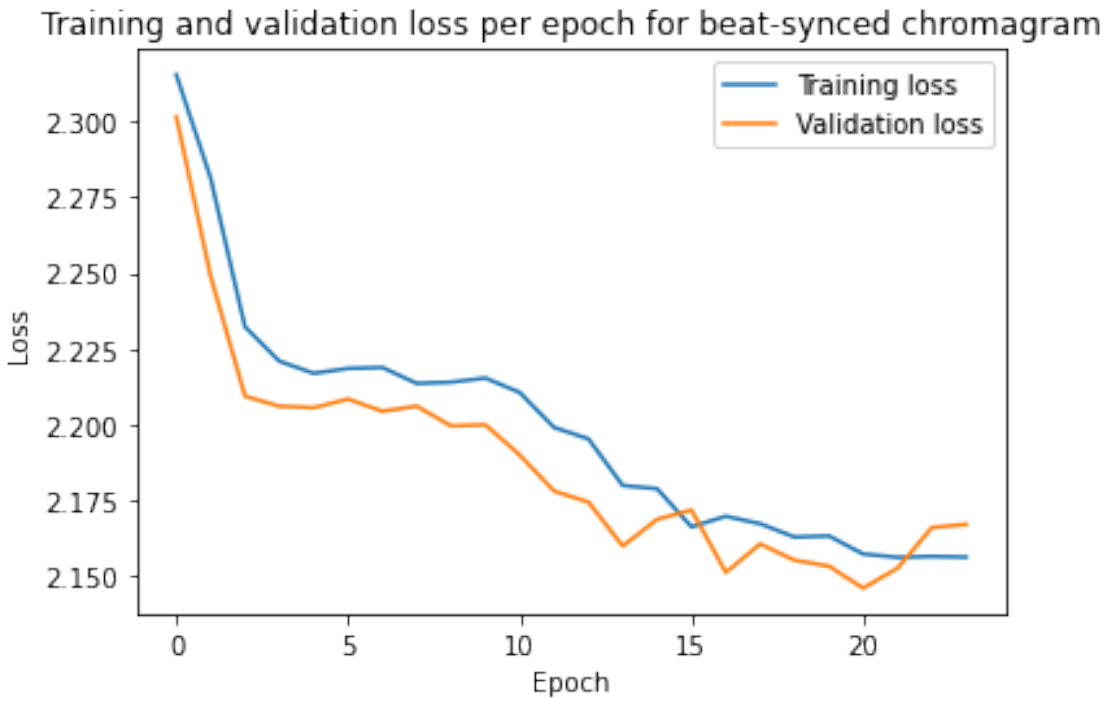
⋮

Epoch 23: Mean training loss per epoch: 2.1563590745772085
Epoch 23: Mean validation loss per epoch: 2.1671628301793877

Number of times validation loss has not decreased: 3

Early stopping...

Το διάγραμμα που προκύπτει είναι το εξής:



ζ) Στο βήμα αυτό εκπαιδεύουμε ένα LSTM δίκτυο, το οποίο δέχεται ως είσοδο τα ενωμένα χρωμογραφήματα και φασματογραφήματα και προβλέπει τις διαφορετικές κλάσεις του συνόλου δεδομένων. Θέτοντας τον αριθμό των εποχών ίσο με 60 και early stopping, έχουμε τα εξής αποτελέσματα για τα χρωμογραφήματα:

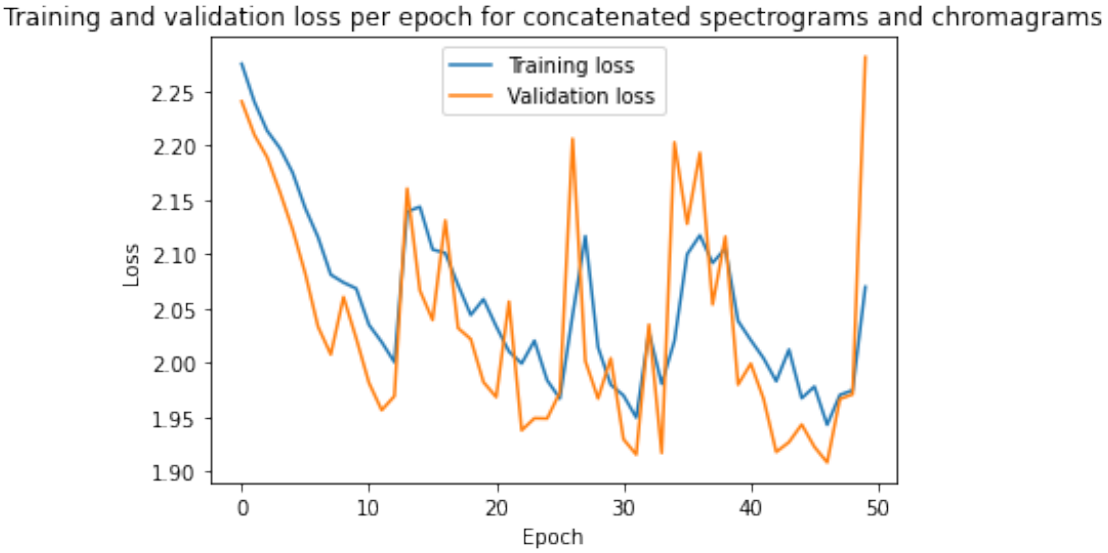
Epoch 0: Mean training loss per epoch: 2.275066867951424
Epoch 0: Mean validation loss per epoch: 2.240444226698442

Epoch 1: Mean training loss per epoch: 2.2399498070439985
Epoch 1: Mean validation loss per epoch: 2.2096242037686435
⋮
Epoch 49: Mean training loss per epoch: 2.0697778674864
Epoch 49: Mean validation loss per epoch: 2.2812235788865523

Number of times validation loss has not decreased: 3

Early stopping...

Το διάγραμμα που προκύπτει είναι το εξής:



Για τα beat-synced concatanated spectrograms-chromagrams έχουμε:

Epoch 0: Mean training loss per epoch: 2.2864481556800103
Epoch 0: Mean validation loss per epoch: 2.2626854939894243

Epoch 1: Mean training loss per epoch: 2.249795094613106
Epoch 1: Mean validation loss per epoch: 2.224394061348655

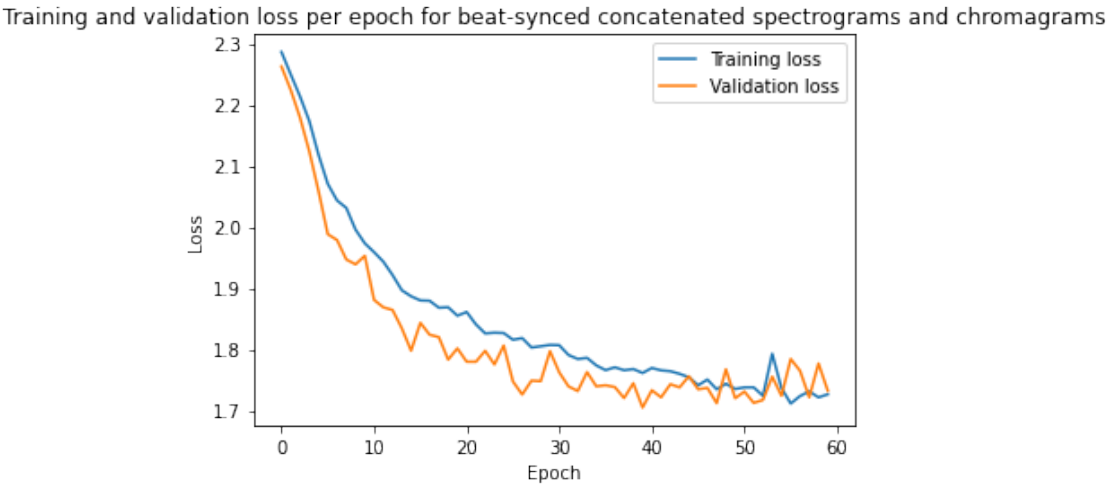
⋮

Epoch 58: Mean training loss per epoch: 1.7216004498543278
Epoch 58: Mean validation loss per epoch: 1.7769918333400379

Number of times validation loss has not decreased: 1

Epoch 59: Mean training loss per epoch: 1.726576114854505
Epoch 59: Mean validation loss per epoch: 1.7323578921231357

Το διάγραμμα που προκύπτει είναι το εξής:



Συμπεράσματα

Όπως μπορούμε να δούμε, τα chromagram και ο συνδυασμός που τα περιέχει έχουν σχετικά χειρότερα αποτελέσματα. Αυτό μπορούμε να υποθέσουμε ότι οφείλεται στο ότι αρκετά από τα είδη μπορεί να έχουν παρεμφερείς χρωματικές αλληλουχίες και άρα να μην μπορούν να διαχωριστούν τόσο καλά. Από την άλλη όμως, το beat-synced διαφέρει αρκετά στις περισσότερες κατηγορίες μουσικής και για το λόγο αυτό μπορούμε να δούμε ότι έχει τα καλύτερα αποτελέσματα κατά την ταξινόμηση.

Βήμα 6: Αξιολόγηση των μοντέλων

Στο ερώτημα αυτό πραγματοποιούμε αξιολόγηση των μοντέλων που εκπαιδεύσαμε προηγουμένως. Παρουσιάζουμε τις μετρικές accuracy, presicion, recall, F1-score. Γενικά για τις μετρικές αναφέρουμε τα εξής:

- Το accuracy αποτελεί το ποσοστό των σωστών προβλέψεων σε σχέση με το σύνολο των δεδομένων για τα οποία κάνουμε προβλέψεις.
- Το precision για μία κλάση ορίζεται ως το πλήθος των φορών που προβλέψαμε ότι ένα δείγμα ανήκει στην κλάση και ανήκει πράγματι σε αυτήν προς το άθροισμα του πλήθους των φορών που προβλέψαμε ότι ένα δείγμα ανήκει στην κλάση και πράγματι ανήκει σε αυτήν και του πλήθους του φορών που προβλέψαμε ότι ένα δείγμα ανήκει στην κλάση αλλά δεν ανήκει σε αυτή. Αποτελεί έτσι μία ένδειξη του πλήθους των φορών που ταξινομήσαμε ένα δείγμα στην συγκεκριμένη κλάση ενώ δεν θα έπρεπε.
- Το recall για μία κλάση ορίζεται ως το πλήθος των φορών που προβλέψαμε ότι ένα δείγμα ανήκει στην κλάση και ανήκει πράγματι σε αυτήν προς το άθροισμα του πλήθους των φορών που προβλέψαμε ότι ένα δείγμα ανήκει στην κλάση και πράγματι ανήκει σε αυτήν και του πλήθους του φορών που ένα δείγμα ανήκει στην κλάση αλλά εμείς το ταξινομήσαμε ως δείγμα μίας άλλης κλάσης. Αποτελεί έτσι μία ένδειξη του πόσα από τα αντικείμενα της συγκεκριμένης κλάσης ταξινομήσαμε σωστά.
- Το F1-score αποτελεί έναν μέσο μεταξύ των μετρικών precision και recall, διατηρώντας μια ισορροπία μεταξύ τους. Υπολογίζεται από την σχέση:
  F1=2*(Precision×Recall)/(Precision+Recall)

Τις μετρικές precision, recall, F1-score τις υπολογίζουμε γενικά ανά κλάση. Αν επιθυμούμε να έχουμε μία ενιαία τιμή για όλες τις κλάσης μπορούμε να υπολογίσουμε το macro, micro ή weighted average των τιμών των διαφορετικών κλάσεων. Όσον αφορά την διαφορά μεταξύ των macro και micro average αξίζει να αναφέρουμε ότι το macro δίνει το ίδιο βάρος σε όλες τις κλάσεις κατά τον υπολογισμό του μέσου όρου, ενώ το micro δίνει το ίδιο βάρος σε όλα τα δείγματα. Αυτό σημαίνει στην πράξη πως το macro υπολογίζει την τιμή της μετρικής ξεχωριστά για κάθε κλάση και ύστερα λαμβάνει την μέση τιμή, ενώ το micro λαμβάνει υπ΄ όψιν το πλήθος δειγμάτων κλαθε κλάσης, συναθροίζοντας την συνεισφορά όλων των κλάσεων για τον υπολιγσμό του μέσου. Η διαφορά στον τρόπου υπολογισμού του μέσου από τα macro και micro average είναι ιδιαίτερα σημαντική στην περίπτωση μη ισορροπημένων δεδομένων, στην οποία κατά τον υπολογισμό του μέσου η ανισορροπία θα λαμβάνεται υπ΄ όψιν με τον micro αλλά όχι με τον macro μέσο όρο. Κάτι τέτοιο μπορεί να οδηγήσει σε αποκλίσεις μεταξύ των δύο μέσων: Στην περίπτωση που στα δεδομένα μας παρουσιάζεται μεγάλη ανισσοροπία, αν για παράδειγμα για μία κλάση με λίγα δεδομένα έχουμε πολύ κακές επιδόσεις και για μία άλλη με πολλά δεδομένα πολύ καλές, το macro θα δώσει το ίδιο βάρος και στις δύο κλάσεις, παρουσιάζοντας χαμηλότερες τιμές από το micro, το οποίο θα λάβει υπ΄ όψιν του το πλήθος δειγμάτων της κάθε κλάσης, δίνοντας μεγαλύτερη αξία επομένως στην κλάση με πολλά δείγματα. Το πιο από τα macro, micro θα επιλέξουμε τελικά ως πιο κατάλληλο εξαρτάται από το πρόβλημά μας: Αν μας ενδιαφέρει η επίδοση του μοντέλου μας στο σύνολο των δεδομένων, τότε το micro είναι πιο κατάλληλο. Αν όμως οι κλάσεις-μειοψηφίες είναι ιδιαίτερα σημαντικές και μας ενδιαφέρει το μοντέλο μας να έχει καλή επίδοση σε αυτές, τότε θέλουμε να τους δώσουμε ίδιο βάρος με τις κλάσεις-πλειοψηφίες και άρα θα πρέπει να επιλέξουμε το macro average.

Σχολιάζοντας τα accuracy και F1-score, αξίζει να αναφέρουμε ότι αυτά μπορεί να διαφέρουν αρκετά για ένα μοντέλο, γι΄ αυτό και συχνά παρατηρούμε και τα δύο. Συγκεκριμένα, οι δύο μετρικές μπορεί να διαφέρουν σημαντικά στην περίπτωση μη ισσοροπημένων δεδομένων, αφού του accuracy δεν λαμβάνει υπ΄ όψιν την ανισορροπία αυτή: Αν για παράδειγμα έχουμε μία κλάση με πάρα πολλά δεδομένα και περισσότερα ταξινομούνται σωστά και έχουμε μία κλάση με πολύ λίγα δεδομένα αλλά σε αυτή κανένα από αυτά δεν ταξινομείται σωστά, κάτι τέτοιο δεν θα φανεί στο accuracy, το οποίο θα είναι υψηλό, αφού στο σύνολο των δεδομένων τα περισσότερα ταξιμομούνται σωστά. Αντίθετα, το F1-score, το οποίο λαμβάνει υπ΄ όψιν τις επιδόσεις ανά κλάση θα είναι πιο χαμηλό, δίνοντας μία ένδειξη πως κάτι δεν είναι καλό στην λειτουργία του ταξινομητή μας. Κάτι τέτοιο μπορεί να είναι ιδιαίτερα σημαντικό. Για παράδειγμα, αν εκπαιδεύαμε ένα μοντέλο το οποίο ταξινομεί ασθενείς σε καρκινοπαθείς και μη, οι περισσότεροι ασθενείς στατιστικά θα είναι υγιείς. Στην περίπτωση αυτή, ακόμα και αν το μοντέλο μας δεν ανίχνευε σχεδόν ποτέ την ύπαρξη καρκίνου επιτυχώς, η ακρίβειά του θα εμφανιζόταν υψηλή, παραπλανώντας μας στο να το θεωρήσουμε αξιόπιστο για την διάγνωση ενός ασθενούς.

Τέλος, σχολιάζοντας την διαφορά μεταξύ precision και recall μπορούμε να αναφέρουμε πως το precision είναι κατάλληλο για προβλήματα που μας ενδιαφέρει η ελαχιστοποίηση των false positive ενώ το recall για προβλήματα που μας ενδιαφέρει η ελαχιστοποίηση των false negative. Για παράδειγμα, σε ένα σύστημα ασφαλείας το οποίο ταξινομεί δικτυακή κίνηση ως κακόβουλη ή μη χρησιμοποιώντας κάποιο μοντέλο, δεν μας πειράζει τόσο να ταξινομούμε κάποιες κινήσεις ως απειλές και ας μην είναι τελικά. Αυτό που είναι σημαντικό είναι όταν υπάρξει όντως κακόβουλη κίνηση να ταξινομηθεί αυτή κατάλληλα, ώστε να αποτρέψουμε τον κίνδυνο για το σύστημά μας. Έτσι, στην περίπτωση αυτή, το recall είναι πιο κατάλληλη μετρική. Αντίθετα στην περίπτωση που το κόστος ενός ψευδός θετικού είναι πολύ πιο ψηλού από το κόστος ενός ψευδώς αρνητικού, το precision είναι πιο κατάλληλο. Ένα τέτοιο παράδειγμα θα αποτελούσε ένα μοντέλο που βοηθά έναν επιχειρηματία να επιλέξει το κατάλληλο κρασί για το εστιατόριό του ή τον κατάλληλο υπάλληλο για μια δουλειά. Στην περίπτωση αυτή δεν θα τον πείραζε τόσο να απορρίψει πολλές καλές επιλογές μέχρι τελικά να βρει μία καλή επιλογή, αρκεί η επιλογή που θα κάνει το σύστημα να είναι πράγματι καλή. Αναλογιζόμενοι τα παραπάνω προβλήματα μπορούμε να διαπιστώσουμε την αξία των μετρικών precision και recall. Σε περιπτώσεις δηλαδή που μας ενδιαφέρει συγκεκριμένα η ελαχιστοποίηση των false positive ή η ελαχιστοποίηση των false negative, όπως αυτές που αναφέρθηκαν παραπάνω, τα F1-score και accuracy δεν θα ήταν κατάλληλες επιλογές, καθώς δεν θα έδιναν μία αρκετά ακριβή εικόνα για την επίδοση του μοντέλου μας στο κομμάτι που μας ενδιαφέρει ακριβώς και θα μπορούσαν να οδηγήσουν σε παραπλανητικά συμπεράσματα για το αν το μοντέλο μας είναι αρκτετά καλό για το πρόβλημά μας.

α) Παρουσιάζουμε το accuracy των διαφορετικών μοντέλων που εκπαιδεύσαμε:

data type	accuracy
Mel spectrograms	0.29
Beat synced mel spectrograms	0.34
Chromagrams	0.18
Beat synced chromagrams	0.19
Fused mel spectrograms and chromagrams	0.17
Beat synced fused mel spectrograms and chromagrams	0.38

Διαπιστώνουμε πως οι βέλτιστη ακρίβεια επιτυγχάνεται για τον συνδυασμό beat synced spectrograms chromagrams, ενώ αρκετά κοντινή είναι και η ακρίβεια στην περίπτωση που χρησιμοποιούμε μόνο beat synced spectrograms. Αντίθετα, παρατηρούμε ότι με χρήση μόνο chromagrams έχουμε την χειτότερη ακρίβεια. Έτσι, συμπαιρένουμε πως τα chromagrms προσφέρουν μεν κάποια βελτίωση της ακρίβειας αν χρησιμοποιηθούν προσθετικά στα spectrograms, από μόνα τους όμως δεν είναι κατάλληλα για επιτυχή ταξινόμηση. Τέλος, συγκρίνοντας τα beat synced με τα απλά spectrograms, διαπιστώνουμε ότι έχουμε αύξηση της ακρίβειας με χρήση των πρώτων. Αυτό είναι λογικό αν αναλογιστούμε ότι

τα beat synced διατηρούν την βασική πληροφορία, μειώνοντας όμως σημαντικά την διαστατικότητα, γεγονός που επιδρά θετικά, επιτρέποντας επιτυχέστερη εκπαίδευση του νευρωνικού.

β) Παρουσιάζουμε το precision, recall, F1-score ανά κλάση:

Mel-spectrograms:

class	precision	recall	F1score
0	0.00	0.00	0.00
1	0.39	0.42	0.40
2	0.34	0.57	0.43
3	0.27	0.29	0.28
4	0.00	0.00	0.00
5	0.00	0.00	0.00
6	0.28	0.17	0.21
7	0.00	0.00	0.00
8	0.26	0.68	0.38
9	0.00	0.00	0.00

Beat synced mel-spectrograms:

class	precision	recall	F1score
0	0.00	0.00	0.00
1	0.30	0.70	0.42
2	0.30	0.69	0.42
3	0.31	0.53	0.39
4	0.25	0.07	0.12
5	0.19	0.12	0.15
6	0.58	0.42	0.49
7	0.00	0.00	0.00
8	0.41	0.28	0.34
9	0.00	0.00	0.00

Chromagrams:

class	precision	recall	F1score
0	0.00	0.00	0.00
1	0.00	0.00	0.00
2	0.00	0.00	0.00
3	0.27	0.04	0.07
4	0.00	0.00	0.00
5	0.00	0.00	0.00
6	0.00	0.00	0.00
7	0.00	0.00	0.00
8	0.18	0.97	0.30
9	0.00	0.00	0.00

Beat synced Chromagrams:

class	precision	recall	F1score
0	0.00	0.00	0.00
1	0.00	0.00	0.00
2	0.00	0.00	0.00
3	0.21	0.41	0.28
4	0.00	0.00	0.00
5	0.00	0.00	0.00
6	0.19	0.71	0.30
7	0.00	0.00	0.00
8	0.16	0.20	0.18
9	0.00	0.00	0.00

Mel-spectrogram + Chromagrams:

class	precision	recall	F1score
0	0.00	0.00	0.00
1	0.18	0.68	0.28
2	0.16	0.85	0.27
3	0.10	0.01	0.02
4	0.00	0.00	0.00
5	0.00	0.00	0.00
6	0.00	0.00	0.00
7	0.00	0.00	0.00
8	0.00	0.00	0.00
9	0.00	0.00	0.00

Beat synced Mel-spectrogram + Chromagrams:

class	precision	recall	F1score
0	0.00	0.00	0.00
1	0.60	0.45	0.51
2	0.34	0.71	0.46
3	0.36	0.53	0.43
4	0.21	0.25	0.23
5	0.00	0.00	0.00
6	0.51	0.59	0.55
7	0.00	0.00	0.00
8	0.37	0.44	0.40
9	0.00	0.00	0.00

Από τα παραπάνω διαπιστώνουμε αρχικά πως οι τιμές precision και recall διαφέρουν αρκετά σε πολλές περιπτώσεις, ενώ το F1-score, όπως θα αναμέναμε από το γεγονός πως αποτελεί εξισσορόπιση των δύο βρίσκεται πάντα κάπου ανάμεσα. Συγκρίνοντας τώρα τις επιδόσεις ανά κλάση, μπορούμε να διαπιστώσουμε πως αυτές διαφέρουν σημαντικά για διαφορετικές κλάσεις, ενώ έχουν μία τάση να είναι ψηλότερες για κλάσεις με περισσότερα δείγματα (π.χ. για την κλάση 8 έχουμε αρκετά καλές επιδόσεις, ενώ για την 0 οι επιδόσεις είναι παντού μηδενικές).

γ) Παρουσιάζουμε το macro-averaged precision, recall, F1-score:

Data type	Precision (macro)	Recall (macro)	F1-score (macro)
Mel spectrograms	0.12	0.14	0.12
Beat synced mel spectrograms	0.15	0.21	0.17
Chromagrams	0.23	0.28	0.23
Beat synced chromagrams	0.05	0.10	0.04
Fused mel spectrograms and chromagrams	0.06	0.13	0.08
Beat synced fused mel spectrograms and chromagrams	0.04	0.15	0.06

Παρατηρώντας τον παραπάνω πίνακα διαπιστώνουμε ότι η εικόνα που λαμβάνουμε διαφέρει αρκετά από αυτή που μας έδωσε το accuracy. Συγκεκριμένα, οι βέλτιστες επισόσεις επιτυγχάνονται τώρα με τη χρήση των απλών chromagrams, ενώ ακολουθούν οι επιδόσεις με χρήση beate synced mel spectrograms και ύστερα απλών mel spectrograms. Η χρήση από κοινού mel spectrograms και chromagrams έχει τώρα τις χειρότερες τιμές μετρικών. Η μεγάλη διαφοροποίηση σε σχέση με το accuracy μπορεί να εξηγηθεί πρώτον από την ανισσοροπία μεταξύ των δειγμάτων των διαφορετικών κλάσεων: Καθώς το macro average δίνει ίση αξία σε όλες τις κλάσεις, το βάρος που δίνεται σε κλάσεις μειοψηφίας είναι πολύ σημαντικό σε σχέση με το πλήθος των δειγμάτων τους. Δεύτερον, δεν πρέπει να ξεχνάμε την διαφορά στο τι δείχνει η ακρίβεια σε σχέση με τα precision, recall, F1-score, η οποία εξηγήθηκε νωρίτερα, και οδηγεί προφανώς και σε διαφορετικές τιμές στις μετρικές.

δ) Στην περίπτωση που κάθε δείγμα αντιστοιχεί μόνο σε μία κλάση, προκύπτει πως τα micro precision, recall και F1-score είναι όλα ίδια και ίσα με το accuracy. Επομένως, ο πίνακας και τα συμπεράσματα ταυτίζονται με αυτά του ερωτήματος (α) του παρόντος βήματος.

Έχοντας αναλύσει τις διάφορες μετρικές και εξάγει πλήθος συμπερασμάτων καλούμαστε να σχολιάσουμε ποια θα ήταν η κατάλληλη μετρική για το πρόβλημά μας. Καθώς έχουμε ένα αρκετά imbalanced σύνολο δεδομένων, σίγουρα δεν θα προτιμήσουμε το accuracy, το οποίο δεν λαμβάνει υπ΄ όψιν την ανισορροπία, ούτε και τα micro average των precision, recall, F1-score, τα οποία ταυτίζονται με την ακρίβεια. Επιπλέον, δεν έχουμε κάποιο λόγο να ενδιαφερόμαστε περισσότερο για τα ψευδώς θετικά ή ψευδώς αρνητικά κάθε κλάσης, άρα δεν προτιμάμε ένα εκ των precision, recall. Έτσι, θα επιλέξουμε μία ισσοροπία μεταξύ των δύο, και άρα ως κατάλληλη μετρική θα προτιμήσουμε το F1-score (ανά κλάση, ή για το σύνολο των κλάσεων το macro average).