

Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Ακαδημαϊκό Έτος: 2020-2021



Αναγνώριση Προτύπων

3η Εργαστηριακή Άσκηση

Θέμα: Αναγνώριση Είδους και Εξαγωγή Συναισθήματος από Μουσική

Τζε Χριστίνα-Ουρανία | 03116079
Ψαρουδάκης Ανδρέας | 03116001

15 Φεβρουαρίου 2021

Εισαγωγή

Σκοπός της άσκησης είναι η αναγνώριση του είδους και η εξαγωγή συναισθηματικών διαστάσεων από φασματογραφήματα μουσικών κομματιών. Για το σκοπό αυτό δουλεύουμε με τα ακόλουθα δύο σύνολα δεδομένων:

- Το Free Music Archive genre με 3834 δείγματα χωρισμένα σε 20 κλάσεις (είδη μουσικής)
- Τη βάση δεδομένων (dataset) multitask music με 1487 δείγματα με επισημειώσεις (labels) για τις τιμές συναισθηματικών διαστάσεων όπως valence, energy και danceability.

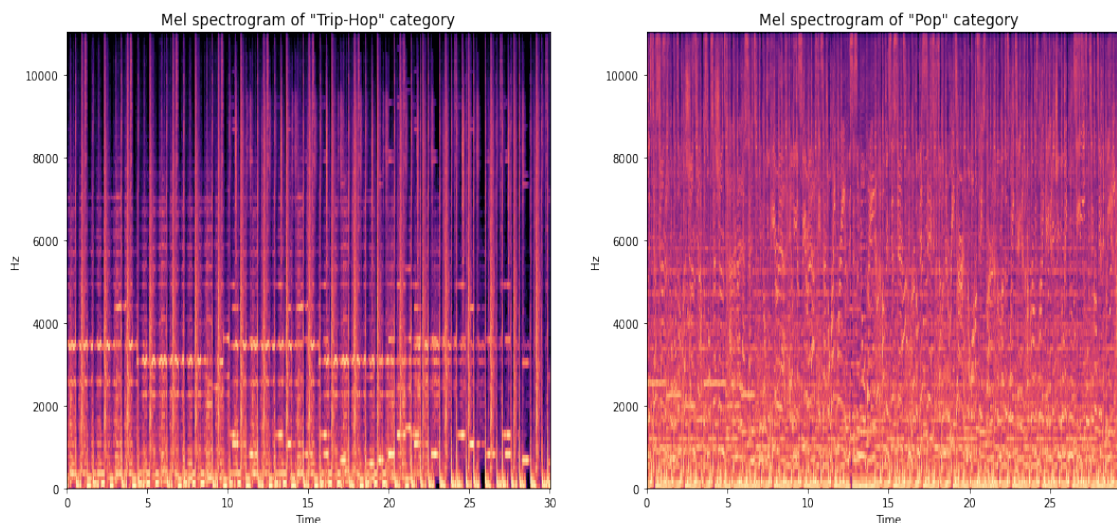
Βήμα 1: Εξοικείωση με φασματογραφήματα στην κλίμακα mel

Στο πρώτο μέρος της παρούσας εργαστηριακής άσκησης ασχολούμαστε με ένα υποσύνολο του Free Music Archive (FMA) dataset για την αναγνώριση είδους μουσικής με βάση το φασματογράφημα (spectrogram). Αυτό είναι μία οπτική αναπαράσταση του συχνοτικού περιεχομένου ενός σήματος, όπου η εξαγόμενη εικόνα αναπαριστά τις διαφορετικές ζώνες συχνοτήτων ως προς το χρόνο.

Το FMA είναι μία βάση δεδομένων από ελεύθερα δείγματα clips μουσικής με επισημειώσεις ως προς το είδος της μουσικής. Τα φασματογραφήματα και οι επισημειώσεις τους βρίσκονται στο φάκελο `fma_genre_spectrogram`.

- (α) Ανοίγουμε το αρχείο `fma_genre_spectrograms/train_labels.txt` και διαλέγουμε αρχικά με χρήση της `np.random.choice` μία τυχαία γραμμή (εξαιρουμένης της πρώτης η οποία περιέχει τα ονόματα των στηλών). Για την γραμμή αυτή αποθηκεύουμε σε δύο μεταβλητές, **first_id** και **first_label**, το μοναδικό προσδιοριστικό και την επισημείωση του κομματιού μουσικής αντίστοιχα. Στη συνέχεια, επαναλαμβάνουμε την ίδια διαδικασία μέχρις ότου να επιλέξουμε τυχαία μία δεύτερη γραμμή για την οποία όμως η επισημείωση είναι διαφορετική της πρώτης.
- (β) Έχοντας πλέον καταλήξει στους δύο τυχαίους δείκτες γραμμών, **first_index** και **second_index**, διαβάζουμε τα αντίστοιχα αρχεία τα οποία βρίσκονται στο φάκελο `fma_genre_spectrograms/train`. Για διευκόλυνση αποθηκεύουμε σε μία μεταβλητή **path** το μονοπάτι μέχρι και τον φάκελο `fma_genre_spectrograms`. Αν `first_id` είναι το ID του πρώτου κομματιού, τότε για το διάβασμα του αρχείου που του αντιστοιχεί χρησιμοποιούμε την δοθείσα συνάρτηση **read_fused_spectrogram** η οποία καλεί την **np.load** με όρισμα το μονοπάτι που καταλήγει σε αυτό. Συγκεκριμένα αυτό προκύπτει από την συνένωση της μεταβλητής `path` με την `first_id` και την κατάληξη **fused.full.npy**. Ομοίως διαβάζουμε και το δεύτερο αρχείο αλλάζοντας μόνο το `first_id` σε `second_id`. Τα δύο αρχεία αποθηκεύονται στους πίνακες **spec1** και **spec2**, διαστάσεων (mel + chroma frequencies, timesteps). Τα φασματογραφήματα σε κλίμακα mel προκύπτουν ως οι πρώτες 128 γραμμές των πινάκων αυτών.

(γ) Απεικονίζουμε τώρα τα φασματογραφήματα για τα διαφορετικά labels μέσω της συνάρτησης `librosa.display.specshow`.



Σχήμα 1: Απεικόνιση φασματογραφημάτων

Παρατηρούμε πως τα δύο φασματογραφήματα διαφέρουν αρκετά μεταξύ τους τόσο στην υφή όσο και στο συχνοτικό τους περιεχόμενο. Το γεγονός αυτό είναι αναμενόμενο καθώς ανάλογα με το είδος στο οποίο ανήκει ένα μουσικό κομμάτι διαφοροποιούνται μεταξύ άλλων η ένταση του ήχου, η συχνότητα και η ταχύτητα του ρυθμού.

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

(α) Τυπώνουμε τις διαστάσεις των φασματογραφημάτων του προηγούμενου Βήματος.

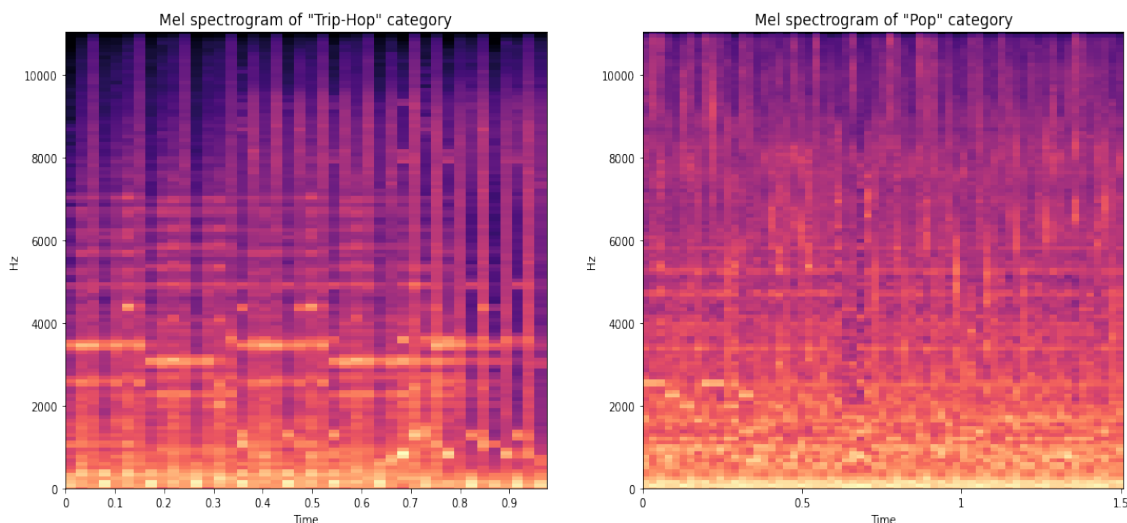
Προφανώς ο αριθμός των γραμμών είναι σταθερός και ίσος με **128**. Ο αριθμός των στηλών συμπίπτει με τα χρονικά βήματα και ισούται με **1293** για το πρώτο τυχαίο δείγμα και **1291** για το δεύτερο.

Γενικά, τα LSTMs είναι ιδιαίτερα αποδοτικά σε περιπτώσεις όπου τα δείγματα εισόδου έχουν χρονική εξάρτηση, κάτι το οποίο ισχύει για δεδομένα μουσικής. Συνεπώς, για την δική μας εφαρμογή θα μπορούσαν να θεωρηθούν κατά μία άποψη αποτελεσματικά. Ωστόσο, τρέχοντας πολλές φορές όλα τα παραπάνω βήματα (οπότε λόγω τυχειότητας αλλάζουν συνεχώς οι γραμμές που επιλέγονται) διαπιστώνουμε πως τα χρονικά βήματα των φασματογραφημάτων κυμαίνονται πάντα από 1291 μέχρι και 1293. Το γεγονός αυτό μειώνει την αποτελεσματικότητα των LSTMs και αυξάνει σημαντικά τον χρόνο εκπαίδευσης καθώς το μήκος των ακολουθιών εισόδου είναι σε κάθε περίπτωση αρκετά μεγάλο.

(β) Ένας τρόπος να μειώσουμε τα χρονικά βήματα είναι να συγχρονίσουμε τα φασματογραφήματα πάνω στο ρυθμό. Για το λόγο αυτό παίρνουμε τη διάμεσο (median)

ανάμεσα στα σημεία που χτυπάει το beat της μουσικής. Επαναλαμβάνουμε την διαδικασία του Βήματος 1 για τα beat-synced spectrograms. Η μόνη διαφορά σε σχέση με πριν είναι ότι η μεταβλητή **path** περιέχει τώρα το μονοπάτι που καταλήγει στον φάκελο `fma_genre_spectrograms_beat` αντί του `fma_genre_spectrograms` που ήταν προηγουμένως.

Στο ακόλουθο Σχήμα φαίνεται η γραφική απεικόνιση των δύο φασματογραφημάτων που προκύπτουν.



Σχήμα 2: Απεικόνιση beat-synced φασματογραφημάτων

Παρατηρούμε πως στην περίπτωση των beat-synced spectrograms, η ανάλυση είναι χειρότερη. Ωστόσο, δεν παρατηρείται σχεδόν καμία άλλη διαφορά στο συχνοτικό περιεχόμενο του σήματος από όπου συμπεραίνουμε πως διατηρείται το μεγαλύτερο ποσοστό της πληροφορίας του.

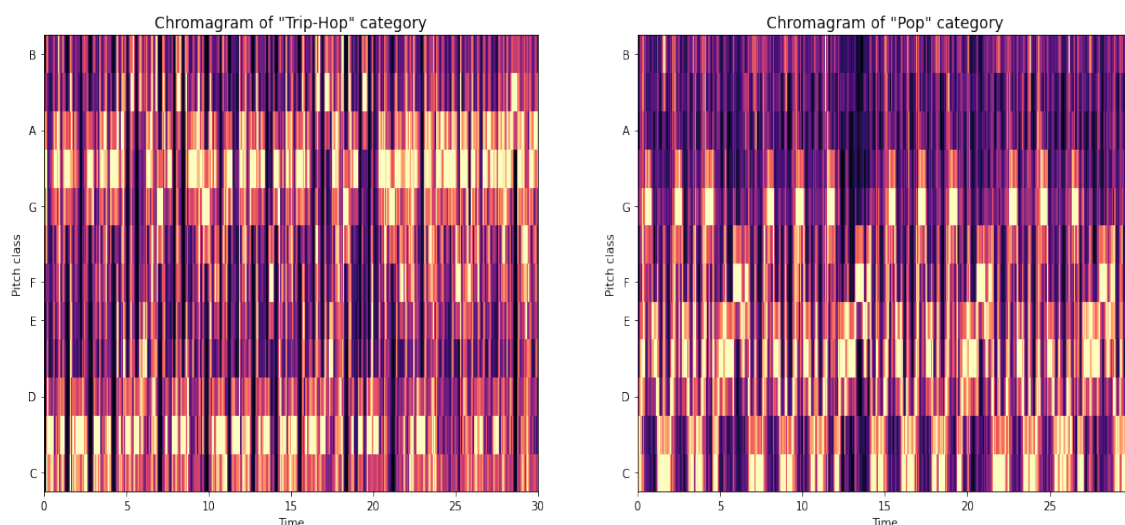
Τυπώνοντας επιπλέον τις διαστάσεις των δύο beat-synced φασματογραφημάτων διαπιστώνουμε πως μειώθηκε αρκετά ο αριθμός των χρονικών βημάτων. Συγκεκριμένα, για το πρώτο δείγμα ο αριθμός των time steps μειώθηκε από **1293** σε **42**, ενώ για το δεύτερο από **1291** σε **65**. Συνεπώς, με βάση τα παραπάνω, αναμένουμε πως η μείωση των χρονικών βημάτων μέσω του συγχρονισμού των φασματογραφημάτων πάνω στο ρυθμό θα αυξήσει την αποδοτικότητα των LSTMs και θα μειώσει αισθητά τον χρόνο εκπαίδευσής τους.

Βήμα 3: Εξοικείωση με χρωμογραφήματα

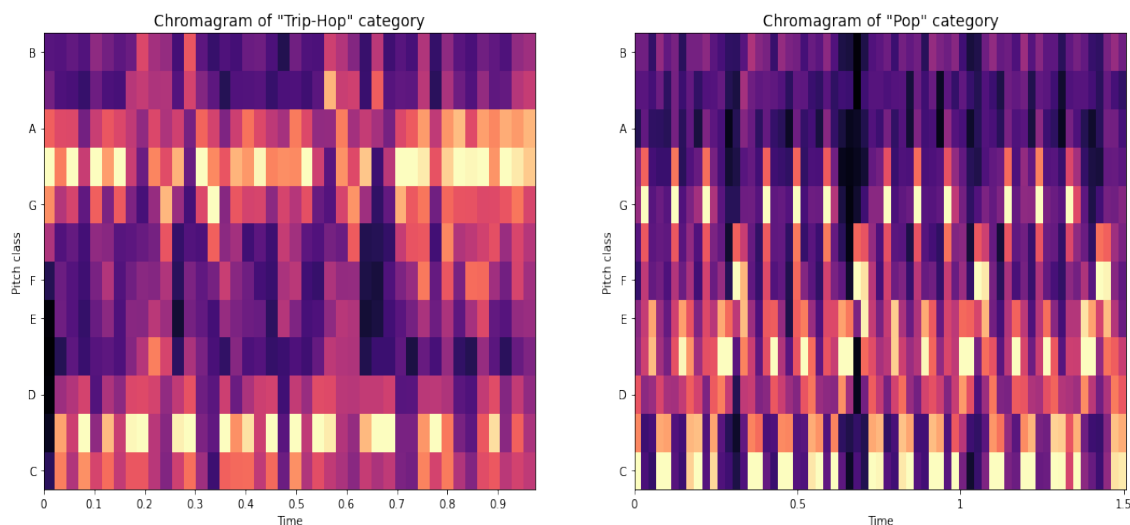
Τα χρωμογραφήματα (chromagrams) σχετίζονται με δώδεκα διαφορετικές νότες (ημιτόνια) C, C#, D, D#, E, F, F#, G, G#, A, A#, B και μπορούν να χρησιμοποιηθούν ως εργαλείο για την ανάλυση της μουσικής αναφορικά με τα αρμονικά και μελωδικά χαρακτηριστικά της ενώ επίσης είναι και αρκετά εύρωστα και στην αναγνώριση των αλλαγών του ηχοχρώματος και των οργάνων.

Επαναλαμβάνουμε τα υποερωτήματα από τα Βήματα 1 και 2 για τα χρωμογραφήματα των ανίστοιχων αρχείων. Αυτά προκύπτουν ως οι 12 τελευταίες γραμμές των `spec1` και `spec2` καθώς οι πρώτες 128, όπως είδαμε προηγουμένως, αφορούν τα φασματογραφήματα. Για την εξαγωγή τους χρησιμοποιούμε την έτοιμη συνάρτηση `read_chromagram`, ενώ αυτά αποθηκεύονται στις μεταβλητές `chroma1` και `chroma2` για το πρώτο και δεύτερο δείγμα αντίστοιχα.

Στο ακόλουθο Σχήμα φαίνονται τα chromagrams χωρίς και με beat-sync.



Σχήμα 3: Απεικόνιση χρωμογραφημάτων



Σχήμα 4: Απεικόνιση beat-synced χρωμογραφημάτων

Παρατηρούμε πάλι πως το είδος της μουσικής επηρεάζει σημαντικά την μορφή του χρωμογραφήματος. Επιπλέον, οι εικόνες των Σχημάτων 3 και 4 είναι αναμφίβολα πιο θορυβώδεις σε σχέση με αυτές των 1 και 2.

Στη συνέχεια τυπώνουμε τις διαστάσεις τους. Προφανώς, οι αριθμοί των χρονικών βημάτων για κάθε δείγμα είναι ίδιοι με αυτούς του Βήματος 2α (αφού για λόγους σύγκρισης δεν θεωρήσαμε καινούργια δείγματα). Αντίστοιχα, ο αριθμός των γραμμών είναι σταθερός για κάθε διαφορετικό δεδομένο και ίσος με 12. Υπενθυμίζουμε πως το πρώτο δείγμα έχει **1923** χρονικά βήματα ενώ το δεύτερο έχει **1921**, ενώ οι αντίστοιχοι αριθμοί στην περίπτωση συγχρονισμού πάνω στο ρυθμό (beat-sync) μειώνονται σε **42** και **65** αντίστοιχα.

Βήμα 4: Φόρτωση και ανάλυση δεδομένων

(α) Χρησιμοποιούμε την έτοιμη υλοποίηση ενός PyTorch Dataset που μας δίνεται από [εδώ](#). Σε αυτήν ορίζονται οι ακόλουθες κλάσεις και συναρτήσεις:

- 1) Η κλάση **SpectrogramDataset** επεκτείνει την Dataset του PyTorch και δέχεται τα εξής ορίσματα:
 - **path**: το μονοπάτι που καταλήγει στον φάκελο που βρίσκονται τα αρχεία. Ανάλογα με το αν θέλουμε να δημιουργήσουμε ένα dataset για τα non-beat-synced δεδομένα (mel spectrograms και chromagrams) ή για τα beat-synced, ο φάκελος στον οποίο καταλήγει είναι ο fma_genre_spectrograms ή ο fma_genre_spectrograms_beat αντίστοιχα.
 - **class_mapping**: πρόκειται για ένα λεξικό το οποίο αν δίνεται σαν όρισμα (καθώς η default τιμή της παραμέτρου αυτής είναι None) συγχωνεύονται κλάσεις που μοιάζουν μεταξύ τους και αφαιρούνται εκείνες που αντιπροσωπεύονται από πολύ λίγα δείγματα.
 - **train**: είναι μία boolean μεταβλητή η οποία καθορίζει ποιά από τα δύο αρχεία (train_labels.txt και test_labels.txt) που βρίσκονται μέσα στον φάκελο που υποδεικνύει η μεταβλητή path θα διαβαστεί. Αν η τιμή της παραμέτρου train είναι True τότε διαβάζεται το train_labels.txt, αλλιώς το test_labels.txt.
 - **max_length**: πρόκειται για μία μεταβλητή η οποία χρησιμοποιείται για το zero-padding των δεδομένων μέσω της κλάσης PaddingTransform. Αν η τιμή της είναι αρνητική, τότε όλα τα δείγματα αποκτούν μέγεθος όσο το μήκος (length) του μεγαλύτερου μουσικού κομματιού. Ο μετασχηματισμός αυτός είναι απαραίτητος για την εκπαίδευση ενός LSTM δικτύου, το οποίο δέχεται δεδομένα μόνο του ίδιου μήκους.
 - **read_spec_fn**: καθορίζει ποιά συνάρτηση εκ των read_fused_spectrogram, read_mel_spectrogram και read_chromagram θα χρησιμοποιηθεί για το διάβασμα των δεδομένων.
- 2) Η συνάρτηση **get_files_labels** η οποία δέχεται τα ακόλουθα ορίσματα:
 - **txt**: είναι το μονοπάτι το οποίο καταλήγει στο αρχείο train_labels.txt αν train=True, αλλιώς στο test_labels.txt.
 - **class_mapping**: αν δίνεται τότε πραγματοποιούνται όσα αναφέρθηκαν στην ομώνυμη παράμετρο της SpectrogramDataset

Επιστρέφει δύο λίστες, **files** και **labels**, οι οποίες περιέχουν αντίστοιχα τα ονόματα των αρχείων που βρίσκονται εντός του .txt (της μορφής id.fused.full.npy για ένα μοναδικό id) και τις επισημειώσεις τους. Η συνάρτηση αυτή καλείται εντός της κλάσης SpectrogramDataset και για κάθε αρχείο της επιστρεφόμενης λίστας files διαβάζονται με χρήση της read_spec_fn τα αντίστοιχα χαρακτηριστικά τα οποία προστίθενται στη λίστα **feats**. Η διάσταση των χαρακτηριστικών (128 για τα φασματογραφήματα, 12 για τα χρωμογραφήματα και 140 για τα fused) αποθηκεύεται στη μεταβλητή **feat_dim**. Στη συνέχεια, μέσω των κλάσεων PaddingTransform και LabelTransformer που αναλύονται παρακάτω, πραγματοποιούνται αντιστοίχως το zero_padding των δεδομένων και ο μετασχηματισμός των labels.

3) Η κλάση **PaddingTransform** με ορίσματα:

- **max_length**: πρόκειται για την ομώνυμη παράμετρο της SpectrogramDataset
- **padding_value**: είναι η τιμή με την οποία γίνονται padded τα δεδομένα. Αν δεν δίνεται σαν όρισμα, τότε από προεπιλογή ισούται με μηδέν.

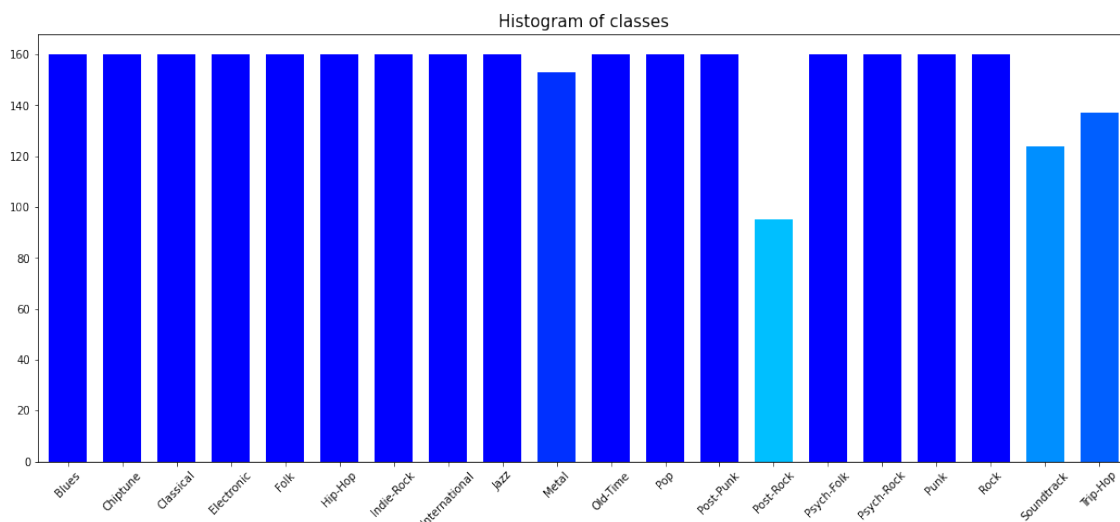
4) Η κλάση **LabelTransformer** η οποία μετασχηματίζει της επισημειώσεις των δεδομένων (strings με τα είδη μουσικής) σε ακέραιους αριθμούς από μηδέν μέχρι n, όπου n ο αριθμός των κλάσεων.

5) Η συνάρτηση **torch_train_val_split**. Πραγματοποιεί τον χωρισμό του dataset σε δεδομένα εκπαίδευσης και δοκιμής. Αρχικά, παράγεται τυχαία ένα σύνολο από indices, από τα οποία τα πρώτα val_split το πλήθος (εξαρτάται από το μέγεθος του validation set που επιθυμούμε) χρησιμοποιούνται για να λάβουμε τα αντίστοιχα στοιχεία του dataset ως το validation set. Το training set προκύπτει με βάση τους υπόλοιπους δείκτες. Στη συνέχεια τα indices δίνονται σαν όρισμα στην **SubsetRandomSampler** για την δημιουργία των samplers που χρειάζονται για τον **Dataloader**. Ο τελευταίος οργανώνει τα αντίστοιχα δεδομένα του dataset σε batches μεγέθους batch_train για τα training data και batch_eval για τα validation.

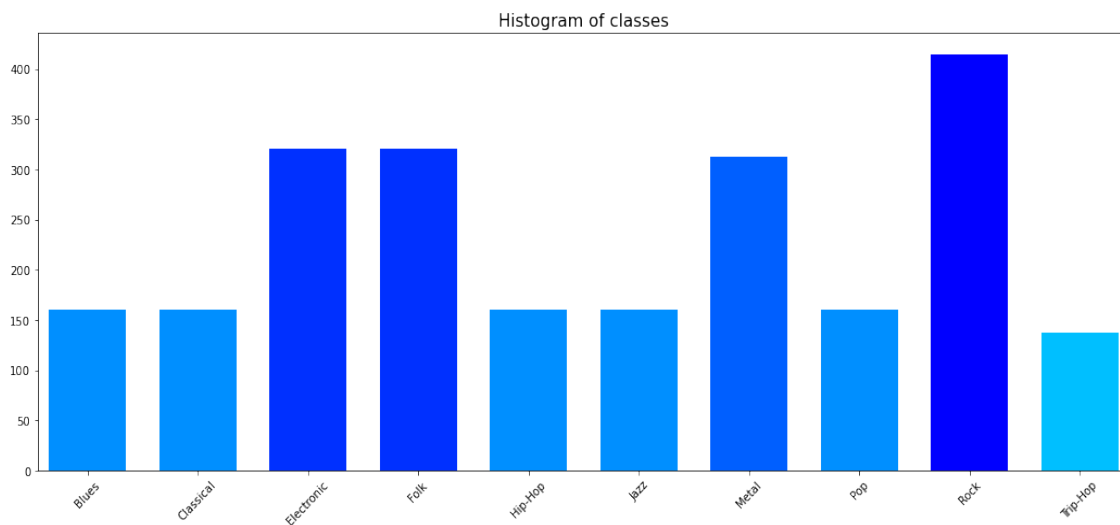
(β) Όπως έχουμε ήδη αναφέρει, ένα από τα ορίσματα της κλάσης SpectrogramDataset που ενδεχομένως να δίνεται είναι το λεξικό **class_mapping**. Μέσω αυτού πραγματοποιείται η συγχώνευση κλάσεων που μοιάζουν μεταξύ τους καθώς και η αφαίρεση ορισμένων που αντιπροσωπεύονται από πολύ λίγα δείγματα. Χαρακτηριστικό παράδειγμα της πρώτης περίπτωσης είναι οι κατηγορίες μουσικής 'Psych-Rock' και 'Post-Rock' οι οποίες αντιστοιχίζονται και οι δύο στη 'Rock'. Αν δεν πραγματοποιούνταν αντιστοιχίσεις σαν και αυτές θα ήταν πολύ δύσκολο για έναν οποιονδήποτε ταξινομητή να διακρίνει επιτυχώς μεταξύ τους τόσο συναφή μουσικά είδη. Εξίσου δύσκολο θα ήταν να αναγνωρίσει τα είδη εκείνα τα οποία αντιπροσωπεύονται από πολύ λίγα δείγματα. Συνεπώς, μέσω του τεχνάσματος αυτού διευκολύνουμε, όσο αυτό είναι δυνατό, το πρόβλημα του ταξινομητή που πρόκειται να κατασκευάσουμε.

(γ) Σχεδιάζουμε τώρα δύο ιστογράμματα τα οποία δείχνουν πόσα δείγματα αντιστοιχούν σε

κάθε κλάση, ένα πριν την διαδικασία του Βήματος 4β και ένα μετά. Αρχικά μέσω της `get_files_labels` λαμβάνουμε όλες τις διαθέσιμες επισημειώσεις. Στη συνέχεια, χρησιμοποιούμε την συνάρτηση `np.unique` προκειμένου να κρατήσουμε τα μοναδικά labels και τον αριθμό των δειγμάτων σε καθένα από αυτά (θέτοντας `True` την παράμετρο `return_counts`). Για τον σχεδιασμό των ιστογραμμάτων ορίζουμε την συνάρτηση `plot_histogram`.



Σχήμα 5: Ιστόγραμμα κλάσεων χωρίς class_mapping



Σχήμα 6: Ιστόγραμμα κλάσεων με class_mapping

Παρατηρούμε πως μετά την διαδικασία του class_mapping, ο αριθμός των κλάσεων μειώθηκε από 20 που ήταν αρχικά σε 10.

Βήμα 5: Αναγνώριση μουσικού είδους με LSTM Network

Αξιοποιώντας τον κώδικα που υλοποιήσαμε στην 2η εργαστηριακή άσκηση, εκπαιδεύουμε ένα LSTM δίκτυο για κάθε ένα από τα σύνολα εκπαίδευσης που φαίνεται στον ακόλουθο πίνακα. Για κάθε μοντέλο αναγράφονται οι αντίστοιχες παράμετροι που χρησιμοποιούνται.

	LSTM parameters			
Dataset	Input dim	Hidden layer dim	Number of hidden layers	Ouput dim
Mel spectrograms	128	256	2	10
Beat-synced mel spectrograms	128	256	2	10
Chromagrams	12	256	2	10
Beat-synced chromagrams	12	256	2	10
Beat-synced fused spectrograms	140	256	2	10

Βήμα 6: Αξιολόγηση των μοντέλων

Χρησιμοποιούμε διάφορες μετρικές απόδοσης για την αξιολόγηση των μοντέλων μας. Για τον ορισμό τους προσδιορίζουμε αρχικά τις έννοιες: **True positives** (T_p), **False positives** (F_p), **True negatives** (T_n) και **False negatives** (F_n) για μια δεδομένη κλάση C_k .

- **True positives** (T_p): Το σύνολο των δειγμάτων που ταξινομήθηκαν στην κλάση C_k και πράγματι άνηκαν στην κλάση C_k .
- **False positives** (F_p): Το σύνολο των δειγμάτων που ταξινομήθηκαν στην κλάση C_k ενώ δεν άνηκαν σε αυτή.
- **True negatives** (T_n): Το σύνολο των δειγμάτων που δεν ταξινομήθηκαν στην κλάση C_k και πράγματι δεν άνηκαν στην κλάση C_k .
- **False negatives** (F_n): Το σύνολο των δειγμάτων που δεν ταξινομήθηκαν στην κλάση C_k ενώ άνηκαν σε αυτή.

Τα παραπάνω μεγέθη μπορούν να γίνουν καλύτερα κατανοητά μέσω της απεικόνισής τους σε έναν Πίνακα Σύγχησης (Confusion Matrix) για μια δεδομένη κλάση:

[illegible]

Σχήμα 7: Πίνακας σύγκρισης που απεικονίζει τις έννοιες T_p , F_p , T_n και F_n για μια δεδομένη κλάση

Τώρα μπορούμε να ορίσουμε τις μετρικές αξιολόγησης που θα χρησιμοποιήσουμε:

- Πιστότητα - Accuracy

Αποτελεί την πιο συνηθισμένη μετρική αξιολόγησης και εκφράζει το σύνολο των ορθών προβλέψεων στο σύνολο όλων των δειγμάτων του dataset. Για ισορροπημένα σύνολα δεδομένων αποτελεί μια αρκετά αξιόπιστη ένδειξη της απόδοσης του μοντέλου. Ωστόσο, για imbalanced datasets μπορεί να οδηγήσει σε λανθασμένα συμπεράσματα, καθώς στην περίπτωση αυτή οι υψηλές τιμές του accuracy δεν αποτελούν ένδειξη καλής προβλεπτικής ικανότητας για τις κλάσεις με μικρό αριθμό δειγμάτων. Αυτό αποτελεί πρόβλημα, καθώς σε αρκετές περιπτώσεις οι κατηγορίες που συγκεντρώνουν το μεγαλύτερο ενδιαφέρον είναι αυτές που περιλαμβάνουν και τη μεινότητα των δεδομένων (π.χ. ανίχνευση ανωμαλιών).

- Ακρίβεια - Precision - (P)

Ορίζεται ως ο λόγος των True positives (T_p) προς τον αριθμό των True positives (T_p) συν τον αριθμό των False positives (F_p). Εκφράζει δηλαδή το λόγο των δειγμάτων που ορθώς ταξινομήθηκαν σε μια κλάση προς το σύνολο όλων των δειγμάτων που προβλέφθηκε ότι ανήκουν στην κλάση αυτή. Είναι ιδιαίτερα χρήσιμη μετρική όταν το κόστος των False positives είναι υψηλό (π.χ. email spam detection).

$$P = \frac{T_p}{T_p + F_p}$$

- Ανάκληση - Recall - (R)

Ορίζεται ως ο λόγος των True positives (T_p) ως προς τον αριθμό των True positives (T_p) συν τον αριθμό των False negatives (F_n). Εκφράζει δηλαδή το λόγο των δειγμάτων που

ορθώς ταξινομήθηκαν σε μια κλάση προς το σύνολο όλων των δειγμάτων που ανήκουν στην κλάση αυτή. Είναι ιδιαίτερα σημαντική μετρική όταν το κόστος των False negatives είναι υψηλό (π.χ. διάγνωση καρκίνου).

$$R = \frac{T_p}{T_p + F_n}$$

Ιδανικά θέλουμε και υψηλή ακρίβεια και υψηλή ανάκληση, ωστόσο μεταξύ της ακρίβειας και της ανάκλησης υπάρχει γενικά trade-off. Για παράδειγμα, στην οριακή περίπτωση ενός δυαδικού ταξινομητή που επιστρέφει σταθερά μόνο τη θετική κλάση, το recall είναι 1 αλλά το precision λαμβάνει τη μικρότερη δυνατή τιμή του. Γενικά, κατεβάζοντας το κατώφλι της απόφασης του ταξινομητή, αυξάνουμε την ανάκληση και μειώνουμε την ακρίβεια και αντιστρόφως.

- **F1—score**

Ορίζεται ως ο αρμονικός μέσος της ακρίβειας (Precision) και της ανάκλησης (Recall). Χρησιμοποιείται όταν υπάρχει μια άνιση κατανομή των κλάσεων και θέλουμε να πετύχουμε μια ισορροπία μεταξύ (Precision) και (Recall). Είναι ιδιαίτερα χρήσιμη μετρική όταν το κόστος των False Negatives και False Positives είναι υψηλό (σε αντίθεση με το accuracy που χρησιμοποιείται όταν το κόστος των True positives και True Negatives είναι μεγάλο). Ωστόσο, όπως αναφέραμε, σε αρκετά προβλήματα, το precision ενδιαφέρει περισσότερο από ότι το recall και αντίστροφα. Στην περίπτωση αυτή, είναι προφανές πως η χρήση του F1-score δεν εξυπηρετεί, καθώς σταθμίζει ισόποσα την ακρίβεια και την ανάκληση.

$$F1 = 2 \cdot \frac{P \cdot R}{P + R} = \frac{2 \cdot T_p}{2 \cdot T_p + F_p + F_n}$$

Όταν έχουμε ένα **multi-classification** πρόβλημα να επιλύσουμε, θέλουμε με κάποιο τρόπο να συνδυάσουμε τις τιμές μια μετρικής για κάθε κλάση (π.χ. recall), έτσι ώστε να λάβουμε μια τελική τιμή που αντιπροσωπεύει τη συνολική ικανότητα κατηγοριοποίησης του μοντέλου μας. Υπάρχουν διάφοροι τρόποι να πραγματοποιηθεί αυτό. Ο απλούστερος από αυτούς είναι να λάβουμε τον αριθμητικό μέσο των τιμών της μετρικής κάθε κλάσης. Η τεχνική αυτή ονομάζεται **macro-average**. Αντίστοιχα, μια άλλη μέθοδος είναι η **micro-average**, κατά την οποία όλες οι κλάσεις εξετάζονται μαζί. Συγκεκριμένα, για οποιαδήποτε εκ των μετρικών precision, recall και F1-score, η τεχνική **micro-average** δίνει την ίδια τιμή, η οποία συμπίπτει με αυτή του accuracy. Είναι δηλαδή:

$$\text{Precision_micro-avg} = \text{Recall_micro-avg} = \text{F1_micro-avg} = \text{Accuracy}$$

Σε **μη-ισορροπημένα datasets**, όπου οι δύο μετρικές ενδέχεται να έχουν μεγάλη απόκλιση, προτιμάται η χρήση της τεχνικής **micro-average**. Αυτό οφείλεται στο γεγονός ότι εξετάζει

συγκεντρωτικά τη συνεισφορά όλων των κλάσεων για την εξαγωγή του average metric, σε αντίθεση με τη **macro-average**, η οποία υπολογίζει τη μετρική κάθε κλάσης ανεξάρτητα και έπειτα λαμβάνει το μέσο όρο τους.

Στους ακόλουθους πίνακες αναγράφονται οι μετρικές **Precision**, **Recall** και **F1-score** για κάθε κλάση, καθώς επίσης και τα αντίστοιχα **micro** και **macro averaged** metrics. Τέλος υπάρχει και μια στήλη **Support** όπου αναγράφεται το πλήθος των δειγμάτων κάθε κλάσης (καθώς και ο συνολικός αριθμός από samples). Κάθε πίνακας αφορά την αξιολόγηση στο test set ενός εκ των πέντε μοντέλων που εκπαιδεύσαμε στο Βήμα 5.

Mel spectrograms test dataset				
Class	Precision	Recall	F1-score	Support
0	0.12	0.03	0.04	40
1	0.42	0.65	0.51	40
2	0.41	0.56	0.47	80
3	0.40	0.36	0.38	80
4	0.34	0.40	0.37	40
5	0.15	0.10	0.12	40
6	0.47	0.71	0.57	78
7	0.00	0.00	0.00	40
8	0.36	0.34	0.35	103
9	0.29	0.29	0.29	34
Accuracy			0.38	575
Macro-average	0.30	0.34	0.31	575
Micro-average	0.38	0.38	0.38	575

Beat-synced mel spectrograms test dataset				
Class	Precision	Recall	F1-score	Support
0	0.40	0.10	0.16	40
1	0.55	0.53	0.54	40
2	0.39	0.70	0.50	80
3	0.36	0.60	0.45	80
4	0.40	0.20	0.27	40
5	0.06	0.03	0.04	40
6	0.47	0.63	0.54	78
7	0.00	0.00	0.00	40
8	0.38	0.37	0.37	103
9	0.12	0.03	0.05	34
Accuracy			0.39	575
Macro-average	0.31	0.32	0.29	575
Micro-average	0.39	0.39	0.39	575

Chromagrams test dataset				
Class	Precision	Recall	F1-score	Support
0	0.00	0.00	0.00	40
1	0.00	0.00	0.00	40
2	0.00	0.00	0.00	80
3	0.23	0.64	0.34	80
4	0.00	0.00	0.00	40
5	0.00	0.00	0.00	40
6	0.25	0.38	0.30	78
7	0.00	0.00	0.00	40
8	0.19	0.41	0.26	103
9	0.00	0.00	0.00	34
Accuracy			0.21	575
Macro-average	0.07	0.14	0.09	575
Micro-average	0.21	0.21	0.21	575

Beat-synced chromagrams test dataset				
Class	Precision	Recall	F1-score	Support
0	0.00	0.00	0.00	40
1	0.00	0.00	0.00	40
2	0.00	0.00	0.00	80
3	0.20	0.69	0.31	80
4	0.00	0.00	0.00	40
5	0.00	0.00	0.00	40
6	0.31	0.42	0.35	78
7	0.00	0.00	0.00	40
8	0.18	0.33	0.23	103
9	0.00	0.00	0.00	34
Accuracy			0.21	575
Macro-average	0.07	0.14	0.09	575
Micro-average	0.21	0.21	0.21	575

Beat-synced fused spectrograms test dataset				
Class	Precision	Recall	F1-score	Support
0	0.17	0.12	0.14	40
1	0.54	0.53	0.53	40
2	0.43	0.68	0.53	80
3	0.35	0.50	0.41	80
4	0.32	0.28	0.30	40
5	0.27	0.10	0.15	40
6	0.59	0.60	0.60	78
7	0.18	0.20	0.19	40
8	0.48	0.38	0.42	103
9	0.17	0.06	0.09	34
Accuracy			0.40	575
Macro-average	0.35	0.34	0.34	575
Micro-average	0.40	0.40	0.40	575


Με βάση τους παραπάνω πίνακες, συμπεραίνουμε ότι τα φασματογραφήματα (τόσο τα αρχικά όσο και τα beat-synced) είναι πιο αποτελεσματικά στην αναγνώριση μουσικού είδους από ότι τα χρωμογραφήματα. Συγκεκριμένα, με χρήση **Spectrograms** λαμβάνουμε Accuracy για το μοντέλο μας γύρω στο **40%**, σε αντίθεση με τα **Chromagrams** όπου το ποσοστό ανέρχεται κοντά στο **20%**. Αξίζει να σημειωθεί πως η συνένωση (**concatenation**) φασματογραφημάτων και χρωμογραφημάτων δεν φαίνεται να προσδίδει περαιτέρω βελτίωση στην απόδοση του μοντέλου, καθώς αυτή είναι ανάλογη της επίδοσης των φασματογραφημάτων (**40%**). Τα ποσοστά, σε κάθε περίπτωση, είναι αρκετά χαμηλά, γεγονός που οφείλεται στην αυξημένη δυσκολία του συγκεκριμένου classification task αλλά και στην αδυναμία των LSTMs να το προσεγγίσουν με επιτυχία. Θα δούμε σε επόμενο βήμα πως τα Βαθιά Συνελικτικά Νευρωνικά Δίκτυα (CNNs) μπορούν να ανταποκριθούν αρκετά πιο αποτελεσματικά σε ένα τέτοιο πρόβλημα κατηγοριοποίησης.

Βήμα 7: 2D CNN

Ένας άλλος τρόπος για την κατασκευή ενός μοντέλου για την επεξεργασία ηχητικών σημάτων είναι να δούμε το φασματογράφημα σαν εικόνα και να χρησιμοποιήσουμε συνελικτικά δίκτυα (Convolutional Neural Networks-CNNs)

- (α) Αξιοποιώντας το σύνδεσμο [αυτό](#) εκπαιδεύουμε ένα δίκτυο στο MNIST και παρατηρούμε τη λειτουργία των ενεργοποιήσεων κάθε επιπέδου.

Αρχικά, το νευρωνικό δέχεται ως είσοδο μια εικόνα διαστάσεων $(24 \times 24 \times 1)$, δηλαδή μια grayscale εικόνα διαστάσεων 24×24 .



() () () () () () () ()

14

pool (12x12x8)
pooling size 2x2, stride 2
max activation: 3.6729, min: 0
max gradient: 0.00005, min: -0.00005

Activations:



Activation Gradients:



Σχήμα 11: Εφαρμογή Max Pooling μεγέθους 2×2 για μείωση των διαστάσεων

Την έξοδο του pooling layer περνάμε μέσα από ένα 2ο convolution layer, το οποίο συνελίσει με 16 κατάλληλα επιλεγμένα φίλτρα, διαστάσεων $5 \times 5 \times 8$. Έτσι, προκύπτουν συνολικά 16 διαφορετικά activation maps (και κατά συνέπεια η έξοδος του συνελικτικού αυτού επιπέδου έχει διάσταση $12 \times 12 \times 16$). Να σημειωθεί πως έχει πραγματοποιηθεί padding μεγέθους 2 για τη διατήρηση της διάστασης των εικόνων.

conv (12x12x16)
filter size 5x5x8, stride 1
max activation: 6.96083, min: -13.05398
max gradient: 0.00003, min: -0.00007
parameters: $16 \times 5 \times 5 \times 8 + 16 = 3216$

Activations:



Activation Gradients:



Weights:

()()()()()()()()()()()()()()()

Weight Gradients:

()()()()()()()()()()()()()()()

Σχήμα 12: 2ο Convolutional Layer με 16 activation maps

Η έξοδος του 2ου συνελικτικού επιπέδου, διέρχεται από μια συνάρτηση ενεργοποίησης ReLU, η οποία μηδενίζει όλες τις τιμές που είναι μικρότερες του μηδενός ενώ αφήνει ίδιες όσες είναι θετικές. Μετασχηματίζει λοιπόν τις επιμέρους εικόνες ως εξής:

relu (12x12x16)
max activation: 6.96083, min: 0
max gradient: 0.00003, min: -0.00007

Activations:





Activation Gradients:



Σχήμα 13: Εφαρμογή ReLU activation function στην έξοδο του 2ου συνελικτικού επιπέδου

Ακολουθεί Max Pooling Layer, το οποίο υποδειγματοληπτεί τις εικόνες. Όμοια με πριν, από κάθε block 3×3 κρατάμε την μέγιστη τιμή, μειώνοντας έτσι τις διαστάσεις των επιμέρους εικόνων σε 4×4 . Η έξοδος του επιπέδου αυτού είναι λοιπόν $4 \times 4 \times 16$.



pool (4x4x16)
pooling size 3x3, stride 3
max activation: 6.96083, min: 0
max gradient: 0.00003, min: -0.00007

Activations:

Activation Gradients:


Σχήμα 14: Εφαρμογή Max Pooling μεγέθους 3×3 για μείωση των διαστάσεων

Τέλος, τοποθετούμε ένα fully connected layer με πλήθος νευρώνων ίσο με τον αριθμό των κλάσεων, δηλαδή 10.


fc (1x1x10)
max activation: 7.41849, min: -17.99077
max gradient: 0.0001, min: -0.00012
parameters: $10 \times 256 + 10 = 2570$

Activations:

Activation Gradients:


Σχήμα 15: Fully Connected επίπεδο με 10 νευρώνες

Στην έξοδο του FC επιπέδου τοποθετούμε μια softmax, η οποία μετασχηματίζει τις τιμές στο εύρος $[0,1]$ (πιθανότητες).

softmax (1x1x10)
max activation: 0.99988, min: 0
max gradient: 0, min: 0

Activations:


Σχήμα 16: Softmax activation function για μετατροπή τιμών στο εύρος $[0,1]$

Είναι προφανές πως το δίκτυο προβλέπει επιτυχημένα ότι η συγκεκριμένη εικόνα αφορά το ψηφίο '2'.

(β) Υλοποιούμε τώρα ένα 2D CNN με 4 επίπεδα (layers) που επεξεργάζεται το φασματογράφημα σαν μονοκάναλη εικόνα. Κάθε επίπεδο πραγματοποιεί τις εξής λειτουργίες:

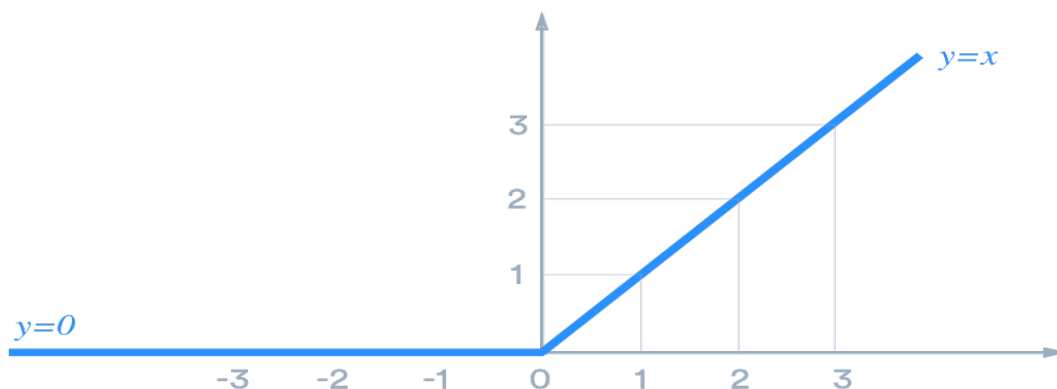
- 2D Convolution
- Batch normalization
- ReLU activation
- Max pooling

Για το 1ο 2D Convolution επίπεδο χρησιμοποιούμε 32 φίλτρα, για το 2ο 64, για το 3ο χρησιμοποιούμε 96 ενώ για το 4ο 128. Για το Batch normalization αλλά και για το Max pooling επιλέγουμε τις default τιμές παραμέτρων.

(γ) Στα προηγούμενα ερωτήματα χρησιμοποιήσαμε έννοιες όπως **Convolutions**, **Batch normalization**, **ReLU** και **Max Pooling**. Οι έννοιες αυτές αναλύονται στη συνέχεια:

- **Convolution layers:** Όπως αναφέραμε και προηγούμενως, τα συνελικτικά επίπεδα (Convolution layers) εφαρμόζουν κατάλληλα επιλεγμένα φίλτρα πάνω σε εικόνες για την εξαγωγή σημαντικών χαρακτηριστικών. Το πλήθος αλλά και το μέγεθος των φίλτρων μπορεί να διαφέρει και απαιτούνται δοκιμές για την βέλτιστη επιλογή αυτών των παραμέτρων. Το αποτέλεσμα της συνέλιξης μιας εικόνας με ένα φίλτρο ονομάζεται feature map και μπορεί να οδηγήσει σε μια εικόνα μικρότερων διαστάσεων. Κάτι τέτοιο άλλωστε είναι επιθυμητό και άλλωστε όχι. Αν θέλουμε να το αποφύγουμε, εφαρμόζουμε κατάλληλο padding στις αρχικές εικόνες πριν από τη συνέλιξη, έτσι ώστε οι διαστάσεις τους να μην αλλοιωθούν καθώς αυτές διέρχονται μέσα από το συνελικτικό επίπεδο.
- **Batch Normalization:** Χρησιμοποιείται ως τεχνική βελτίωσης της ταχύτητας εκπαίδευσης, της σταθερότητας αλλά και της επίδοσης στα νευρωνικά δίκτυα. Ουσιαστικά χρησιμοποιείται ως ένα μέσο κανονικοποίησης του επιπέδου εισόδου, πραγματοποιώντας κατάλληλο scaling των activations. Η χρησιμότητα αυτού του layer είναι αδιαμφισβήτητη, καθώς χρησιμοποιείται σε πολυάριθμες εφαρμογές τα τελευταία χρόνια. Παρόλα αυτά, ο λόγος στον οποίο έγκειται αυτή η αποτελεσματικότητα δεν έχει πλήρως εξακριβωθεί. Η πιο πιθανή αιτία, φαίνεται πως έχει να κάνει με το πρόβλημα του “internal covariate shift”, που επηρεάζει το learning rate του νευρωνικού λόγω της αρχικοποίησης των παραμέτρων. Η χρήση του batch normalization δείχνει να αμβλύνει το πρόβλημα αυτό.
- **ReLU:** Η Rectified Linear Unit (ReLU) αποτελεί συνάρτηση ενεργοποίησης και δίνεται από τη σχέση:

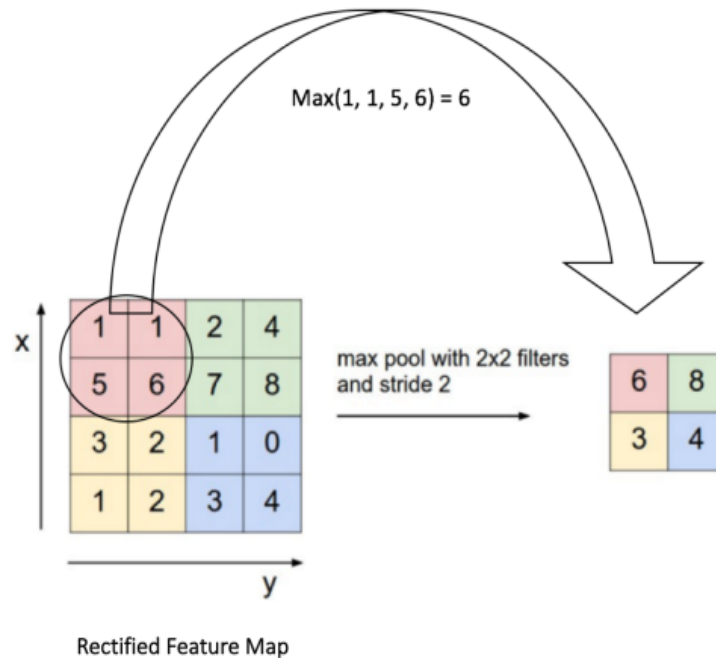
$$f(z) = \max(0, z)$$



Σχήμα 17: Rectified Linear Unit (ReLU)

Ο λόγος που επιλέγουμε αυτή είναι κυρίως η απλότητά της που μας γλιτώνει από τον επιπλέον υπολογιστικό φόρτο μίας πιο σύνθετης συνάρτησης ενεργοποίησης.

- **Max Pooling:** Γενικά ένα επίπεδο pooling μειώνει τη διάσταση της εισόδου του μέσω κάποιου είδους χωρικής υποδειγματοληψίας. Υπάρχουν αρκετά είδη pooling (max, sum, average, κτλ). Η λειτουργία του max-pooling μπορεί να γίνει εύκολα κατανοητή μέσω της ακόλουθης εικόνας:



Σχήμα 18: Max-Pooling layer

Όπως φαίνεται, υπάρχει ένα παράθυρο το οποίο μετακινείται και επιλέγει κάθε φορά το μέγιστο στοιχείο μιας περιοχής του πίνακα (της εικόνας). Οι παράμετροι που καλούμαστε να επιλέξουμε στο επίπεδο αυτό είναι το μέγεθος του παραθύρου (2×2 στο παράδειγμα) καθώς και το βήμα του. Το layer αυτό δεν έχει εκπαιδευσιμες παραμέτρους και υπάρχει μόνο για να μειώσει τη διάσταση της εισόδου του διατηρώντας τη χωρική πληροφορία. Πιο συγκεκριμένα βοηθάει:

- στη μείωση των συνολικών παραμέτρων του δικτύου.
- στο να κάνει την αναπαράσταση αμετάβλητη σε μικροαλλαγές στην είσοδο.
- στο να κάνει την αναπαράσταση ανεξάρτητη της κλίμακας (equivariance).

(δ) Εκπαιδεύουμε τώρα το CNN μοντέλο με είσοδο τα φασματογραφήματα.

Στους ακόλουθους πίνακες αναγράφονται οι μετρικές **Precision**, **Recall** και **F1-score** για κάθε κλάση, καθώς επίσης και τα αντίστοιχα **micro** και **macro averaged metrics**. Τέλος υπάρχει και μια στήλη **Support** όπου αναγράφεται το πλήθος των δειγμάτων κάθε κλάσης (καθώς και ο συνολικός αριθμός από samples). Ο πρώτος πίνακας αφορά την

αξιολόγηση του μοντέλου στο σύνολο επικύρωσης (**validation set**) ενώ ο δεύτερος αφορά τα αποτελέσματα πάνω στο σύνολο ελέγχου (**test set**).

Mel spectrograms validation dataset				
Class	Precision	Recall	F1-score	Support
0	0.24	0.12	0.16	48
1	0.66	0.82	0.73	55
2	0.57	0.65	0.61	99
3	0.39	0.55	0.45	99
4	0.76	0.67	0.71	51
5	0.39	0.22	0.28	55
6	0.57	0.64	0.60	115
7	0.28	0.16	0.21	55
8	0.38	0.36	0.37	143
9	0.39	0.40	0.40	40
Accuracy			0.48	760
Macro-average	0.46	0.46	0.45	760
Micro-average	0.48	0.48	0.48	760

Mel spectrograms test dataset				
Class	Precision	Recall	F1-score	Support
0	0.18	0.10	0.13	40
1	0.46	0.65	0.54	40
2	0.55	0.64	0.59	80
3	0.38	0.55	0.45	80
4	0.68	0.47	0.56	40
5	0.24	0.10	0.14	40
6	0.57	0.59	0.58	78
7	0.21	0.15	0.17	40
8	0.41	0.38	0.39	103
9	0.25	0.29	0.27	34
Accuracy			0.43	575
Macro-average	0.39	0.39	0.38	575
Micro-average	0.43	0.43	0.43	575

Με βάση τους παραπάνω πίνακες, παρατηρούμε πως το CNN υπερτερεί έναντι του LSTM μοντέλου. Αυτό συμβαίνει καθώς, όπως εξηγήσαμε, το CNN έχει την ικανότητα να εξάγει σημαντικά χαρακτηριστικά των εικόνων εισόδου (spectrograms). Η βελτίωση ωστόσο, δεν είναι ιδιαίτερα μεγάλη, με το ποσοστό να εξακολουθεί να είναι χαμηλό, λόγω της δυσκολίας του συγκεκριμένου classification task. Για ακόμα καλύτερα αποτελέσματα θα μπορούσε να γίνει χρήση Μεταφοράς Μάθησης (Transfer Learning) καθώς επίσης και συνδυασμός CNN και LSTM, μέσω της από κοινού αποκωδικοποίησης τόσο των ακολουθιακών στοιχείων που διέπουν τη μουσική όσο και των χαρακτηριστικών που μας προσφέρουν τα Spectrograms.

Βήμα 8: Εκτίμηση συναισθήματος - συμπεριφοράς με παλινδρόμηση

Στο βήμα αυτό χρησιμοποιούμε το multitask dataset: `multitask_dataset/train_labels.txt`. Εδώ μας δίνονται τα φασματογραφήματα, καθώς και επισημειώσεις σε 3 άξονες που αφορούν το συναίσθημα του τραγουδιού. Οι επισημειώσεις είναι πραγματικοί αριθμοί μεταξύ 0 και 1:

- Valence (πόσο θετικό ή αρνητικό είναι το συναίσθημα), όπου αρνητικό κοντά στο 0, θετικό κοντά στο 1.
- Energy (πόσο ισχυρό είναι το συναίσθημα), όπου ασθενές κοντά στο 0, ισχυρό κοντά στο 1.
- Danceability (πόσο χορευτικό είναι το τραγούδι), όπου μη χορευτικό κοντά στο 0, και χορευτικό κοντά στο 1.

Αρχικά, προτού εκπαιδεύσουμε οποιοδήποτε μοντέλο, τροποποιούμε την έτοιμη υλοποίηση της `SpectrogramDataset` και ορίζουμε την νέα κλάση `MultitaskDataset`. Επειδή στο σύνολο αυτό των δεδομένων δεν παρέχονται οι επισημειώσεις για το test set, η εκτίμηση του πόσο καλά γενικεύει το μοντέλο θα πρέπει να γίνει παίρνοντας ένα υποσύνολο από τα δεδομένα που δίνονται. Για τον λόγο αυτό τροποποιούμε επιπλέον την συνάρτηση `torch_train_val_split` την οποία τώρα μετονομάζουμε σε `torch_train_val_test_split`. Συγκεκριμένα, υιοθετούμε το σχήμα 70-15-15 για training, validation και testing αντίστοιχα. Τέλος, χρειάζεται να αλλάξουμε και την συνάρτηση `get_files_labels`, δίνοντάς της ένα επιπλέον όρισμα `emotion_type` το οποίο καθορίζει ποιός από τους 3 άξονες (Valence, Energy ή Danceability) θα χρησιμοποιηθεί σαν επισημείωση. Η παράμετρος `emotion_type` μπορεί επιπλέον να λάβει και την τιμή `All`, οπότε τότε χρησιμοποιούνται και οι 3 τιμές των αξόνων.

Προσαρμόζουμε το καλύτερο μοντέλο του Βήματος 5 και το μοντέλο του Βήματος 7 για παλινδρόμηση αλλάζοντας τη συνάρτηση κόστους. Για προβλήματα τύπου regression, το loss που χρησιμοποιούμε είναι το **MSELoss**. Ακόμα, αντί των Classification metrics που υπολογίζαμε και τυπώναμε πριν, επιστρέφουμε το αντίστοιχο **Spearman Correlation**. Σημειώνουμε πως το καλύτερο μοντέλο (με βάση το Validation Accuracy) του Βήματος 5 προέκυψε για τα Beat-synced mel spectrograms και για τον λόγο αυτό σε όλα τα ερωτήματα του Βήματος 8 δουλεύουμε με το αντίστοιχο `multitask_dataset_beat`. Εκπαιδεύουμε και τα δύο μοντέλα αρχικά για την εκτίμηση του Valence. Αφού αποθηκεύσουμε το προκύπτον correlation προχωράμε με την εκπαίδευσή τους για το Energy. Αποθηκεύουμε τα αποτελέσματα και επαναλαμβάνουμε για το Danceability. Στον ακόλουθο πίνακα φαίνονται τα ενδιαμέσα αποτελέσματα τόσο για το validation όσο και για το test set για κάθε μοντέλο (LSTM/CNN) και για κάθε συναισθηματικό άξονα.

	LSTM		CNN	
Emotion type	Validation set	Test set	Validation set	Test set
Valence	0.207	0.199	0.573	0.574
Energy	0.655	0.694	0.754	0.738
Danceability	0.575	0.43	0.654	0.587

Η τελική μετρική για κάθε μοντέλο φαίνεται στον ακόλουθο πίνακα και είναι το **μέσο Spearman Correlation** ανάμεσα στις πραγματικές (ground truth) τιμές και στις προβλεπόμενες τιμές για όλους τους άξονες.

Mean Spearman Correlation			
LSTM		CNN	
Validation set	Test set	Validation set	Test set
0.479	0.441	0.661	0.633

Βήμα 9α: Μεταφορά Γνώσης (Transfer Learning)

Ένας τρόπος για τη βελτίωση των βαθιών νευρωνικών όταν έχουμε λίγα διαθέσιμα δεδομένα είναι η μεταφορά της γνώσης από ένα άλλο μοντέλο, εκπαιδευμένο σε ένα μεγαλύτερο dataset. Πρόκειται για μία σύγχρονη και ευρέως διαδεδομένη τεχνική η οποία χρησιμοποιείται για την βελτίωση της μάθησης σε μία νέα εργασία μέσω της μεταφοράς γνώσεων από μία άλλη σχετική που έχει ήδη μελετηθεί και μαθητεί.

- (α) Το [1] αρχικά διακρίνει τα χαρακτηριστικά ενός εκπαιδευμένου μοντέλου σε general και specific και στην συνέχεια εξετάζει ποιά από αυτά είναι κατάλληλα για μεταφορά μάθησης. Συγκεκριμένα, general είναι τα first-layer features, ενώ specific τα last layers. Τα τελευταία εξαρτώνται σε μεγάλο βαθμό από το επιλεγμένο dataset και task. Για το transfer learning προτιμώνται τα general features τα οποία είναι κατάλληλα τόσο για το base task όσο και για το target. Τέλος, τεκμηριώνεται πως η απόδοση της μεταφοράς γνώσης μειώνεται καθώς αυξάνεται η απόσταση των 2 αυτών εργασιών. Ωστόσο, ακόμα και αν τα προ-εκπαιδευμένα βάρη προέρχονται από ένα task το οποίο απέχει αρκετά από αυτό του στόχου, εξακολουθούν να είναι καλύτερα από οποιαδήποτε τυχαία βάρη.
- (β) Επιλέγουμε να δουλέψουμε με το CNN αφού έχει αποδώσει καλύτερα σε όλα τα προηγούμενα πειράματα.

- (γ) Εκπαιδεύουμε το επιλεγμένο μοντέλο στο `fma_genre_spectrograms` dataset. Επειδή τα Βήματα 8-9β υλοποιήθηκαν σε ξεχωριστό kernel του Kaggle λόγω ζητημάτων χωρητικότητας μνήμης, στο σημείο αυτό ορίζουμε εκ νέου την κλάση `SpectrogramDataset` και την συνάρτηση `torch_train_val_split`. Στη συνέχεια, κατά τον ίδιο τρόπο με τα προηγούμενα ερωτήματα, εκπαιδεύουμε το CNN για τα beat-synced mel spectrograms και αποθηκεύουμε μέσω checkpoints τα βάρη του δικτύου στην εποχή που αυτό έχει την καλύτερη επίδοση, δηλαδή το μικρότερο validation loss.
- (δ) Φορτώνουμε τώρα το καλύτερο μοντέλο που προέκυψε. Για να το εφαρμόσουμε στο multitask dataset τροποποιούμε τα 3 τελευταία linear layers ώστε να ταιριάζουν οι διαστάσεις και να έχουμε μία μόνο έξοδο, αντί για 10 που είχαμε στο πρόβλημα της αναγνώρισης είδους μουσικής. Στη συνέχεια το εκπαιδεύουμε για λίγες εποχές κατά το πρότυπο του Βήματος 8. Για ευκολία επικεντρωνόμαστε στην εκτίμηση του **Energy** που ήταν εκείνο για το οποίο πετύχαμε τα καλύτερα αποτελέσματα. Τελικά το Spearman Correlation προκύπτει ίσο με **0.772** για τα validation data και **0.725** για τα test.
- (ε) Οι αντίστοιχες τιμές χωρίς την τεχνική της μεταφοράς γνώσης ήταν **0.754** και **0.738**. Συνεπώς, κρίνοντας από το validation set παρατηρείται μία μικρή αύξηση του Correlation με την εφαρμογή του Transfer Learning. Το γεγονός αυτό είναι αναμενόμενο καθώς το είδος ενός μουσικού κομματιού καθορίζει σε μεγάλο βαθμό τα προκλητά συναισθήματα και το Danceability του.

Βήμα 9β: Εκπαίδευση σε πολλά προβλήματα (Multitask Learning)

Στο Βήμα 8 εκπαιδεύσαμε ξεχωριστά ένα μοντέλο για κάθε συναισθηματική διάσταση. Ένας τρόπος για να εκπαιδεύσουμε πιο αποδοτικά μοντέλα όταν μας δίνονται πολλές επισημειώσεις είναι η χρήση multitask learning.

- (α) Το [2] εισάγει την MultiModel αρχιτεκτονική παρουσιάζοντας ένα πολυεπίπεδο νευρωνικό μοντέλο το οποίο εκπαιδεύεται σε 8 corpora και μπορεί ταυτόχρονα να μάθει διαφορετικά tasks ακόμα και από διαφορετικούς τομείς. Η αρχιτεκτονική του ενσωματώνει δομικά στοιχεία όπου το κάθε ένα είναι χρήσιμο για ένα υποσύνολο των εργασιών για τις οποίες εκπαιδεύεται. Μεταξύ άλλων περιέχει convolutional layers, attention mechanism και sparsely-gated layers. Σύμφωνα με το [2] ακόμη και αν ένα block δεν είναι χρήσιμο για μία εργασία, μετά την προσθήκη του δεν βλάπτει την απόδοσή της. Τέλος, επιβεβαιώνει πως το multitask learning επωφελεί σε μεγάλο βαθμό τα tasks εκείνα για τα οποία έχουμε λίγα δεδομένα. Από την άλλη, η επίδοση σε μεγαλύτερα tasks υποβαθμίζεται ελαφρώς και ίσως και καθόλου.
- (β) Δημιουργούμε το multitask dataset αλλά αυτή τη φορά με `emotion_type=All` και ορίζουμε τις συναρτήσεις **multitask_fit** και **find_multitask_spearman_correlation**. Η πρώτη διαφέρει από την **fit** στο ότι εκπαιδεύει ένα μοντέλο χρησιμοποιώντας σαν συνάρτηση

κόστους το άθροισμα από τα losses για το valence, energy και danceability. Στην `find_multitask_spearman_correlation` υπολογίζεται ένας παράγοντας correlation ανάμεσα στις πραγματικές τιμές και στις προβλεπόμενες για κάθε έναν από τους 3 άξονες. Τελικά, αυτό που επιστρέφει είναι το μέσο Spearman Correlation ενώ φροντίζουμε ώστε να τυπώνονται και οι επιμέρους τιμές. Εκπαιδεύουμε το συνελικτικό δίκτυο για 100 εποχές και αλλάζοντας πρώτα το `output_dim` από 1 σε 3.

Στον ακόλουθο πίνακα φαίνονται τα επιμέρους correlations για κάθε έναν από τους 3 άξονες.

Spearman Correlation					
Valence		Energy		Danceability	
Validation set	Test set	Validation set	Test set	Validation set	Test set
0.579	0.61	0.778	0.804	0.606	0.543

Το μέσο Spearman Correlation ισούται με **0.654** για τα validation data και **0.652** για τα test.

(γ) Συγκρίνοντας τα αποτελέσματα με αυτά του Βήματος 8 παρατηρούμε πως για τα δεδομένα επικύρωσης το mean spearman correlation μειώθηκε από 0.661 σε 0.654. Ωστόσο, τα αποτελέσματα αυτά δεν είναι πλήρως αντιπροσωπευτικά καθώς σε κάθε διαφορετική εκτέλεση του κώδικα αλλάζουν (σε μικρό βέβαια βαθμό) οι τελικές τιμές. Σε κάθε περίπτωση αν οι συναισθηματικές συντεταγμένες δεν είναι πλήρως ασυσχέτιστες τότε αναμένουμε μία μικρή αύξηση της απόδοσης.

Αναφορές

- [1] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'14). MIT Press, Cambridge, MA, USA, 3320–3328.,
- [2] Kaiser, Lukasz, Aidan N. Gomez, Noam Shazeer, Ashish Vaswani, Niki Parmar, Llion Jones and Jakob Uszkoreit. “One Model To Learn Them All.” ArXiv abs/1706.05137 (2017)