Ε.Μ.Π., Σχολή Η.Μ.& Μ.Υ. **Αναγνώριση Προτύπων** Ακαδ.Ετος 2020-**2021**
Ημερ/νια: 11-01-2021 **Εργαστηριακή Άσκηση 3**

Εξηγήστε περιεκτικά και επαρκώς την εργασία σας. Κώδικας χωρίς σχόλια δεν θα βαθμολογηθεί. Επιτρέπεται η συνεργασία εντός ομάδων των 2 ατόμων εφόσον φοιτούν στο ίδιο πρόγραμμα σπουδών (είτε ομάδες προπτυχιακών, είτε ομάδες μεταπτυχιακών). Κάθε ομάδα 2 ατόμων υποβάλλει μια κοινή αναφορά που αντιπροσωπεύει μόνο την προσωπική εργασία των μελών της. Αν χρησιμοποιήσετε κάποια άλλη πηγή εκτός των βιβλίων και του εκπαιδευτικού υλικού του μαθήματος, πρέπει να το αναφέρετε. Η παράδοση της αναφοράς και του κώδικα της εργασίας θα γίνει ηλεκτρονικά στο moodle του μαθήματος (https://courses.pclab.ece.ntua.gr/course/view.php?id=18). Στη σελίδα αυτή μπορείτε επίσης να υποβάλετε απορίες και ερωτήσεις προς τους βοηθούς του μαθήματος με μορφή issues. Ερωτήσεις αναφορικά με το εργαστήριο που θα γίνονται μέσω mail δεν θα λαμβάνουν απάντηση.
Επισημαίνεται ότι απαγορεύεται η ανάρτηση των λύσεων των εργαστηριακών ασκήσεων στο github, ή άλλες ιστοσελίδες.

Θέμα: Αναγνώριση Είδους και Εξαγωγή Συναισθήματος από Μουσική

Σκοπός της άσκησης είναι η αναγνώριση του είδους και η εξαγωγή συναισθηματικών διαστάσεων από φασματογραφήματα μουσικών κομματιών. Για το σκοπό αυτό θα σας δοθούν 2 σύνολα δεδομένων:

- Το Free Music Archive (FMA) genre με 3834 δείγματα χωρισμένα σε 20 κλάσεις (είδη μουσικής).
- Η βάση δεδομένων (dataset) multitask music με 1497 δείγματα με επισημειώσεις (labels) για τις τιμές συναισθηματικών διαστάσεων όπως valence, energy και danceability.

Τα δείγματα είναι φασματογραφήματα (spectrograms), τα οποία έχουν εξαχθεί από clips 30 δευτερολέπτων από διαφορετικά τραγούδια. Για τη διευκόλυνση σας, έχουμε ανεβάσει τα δεδομένα σε μορφή dataset στο γνωστό ιστότοπο διαγωνισμών μηχανικής μάθησης kaggle στο link https://www.kaggle.com/geoparslp/patreco3-multitask-affective-music. Επιπλέον, έχουμε δημιουργήσει έναν διαγωνισμό στον οποίο μπορείτε προαιρετικά να συμμετάσχετε στο https://www.kaggle.com/c/multitask-music-classification-2021.

Υπόδειξη: Για να μπορέσετε να συμμετάσχετε στο διαγωνισμό, ενεργοποιήστε την πρόσβαση σας μέσω του link https://www.kaggle.com/t/8e7cc9f3d9c644618f263afe94c6eec3.

1 Προπαρασκευή

1.1 Περιγραφή

Θα ασχοληθούμε με την ανάλυση των φασματογραφημάτων με χρήση βαθιών αρχιτεκτονικών με συνελικτικά νευρωνικά δίκτυα (CNN) και αναδρομικά νευρωνικά δίκτυα (RNN). Η άσκηση χωρίζεται σε 5 μέρη:

1. Ανάλυση των δεδομένων και εξοικείωση με τα φασματογραφήματα.
2. Κατασκευή ταξινομητών για το είδος της μουσικής πάνω στη βάση δεδομένων (dataset) FMA.
3. Κατασκευή regression μοντέλων για την πρόβλεψη valence, energy και danceability πάνω στη Multitask βάση δεδομένων (dataset).
4. Χρήση προηγμένων τεχνικών εκπαίδευσης (transfer - multitask) learning για τη βελτίωση των αποτελεσμάτων
5. (Προαιρετικά) Υποβολή των μοντέλων στο Kaggle competition του εργαστηρίου και σύγκριση των αποτελεσμάτων

Για να τρέξετε τα δίκτυα σας είναι απαραίτητη η χρήση GPUs. Το kaggle σας προσφέρει πρόσβαση σε δωρεάν GPUs μέσω των kaggle notebooks, στα οποία είναι επίσης άμεση η προσθήκη των dataset που θα χρησιμοποιήσετε.

1.1.1 Βιβλιοθήκες Python

Χρήσιμες βιβλιοθήκες Python που θα χρειαστείτε: `librosa, numpy, pytorch, scikit-learn`.

1.2 Εκτέλεση

Στην προπαρασκευή θα ασχοληθούμε με την αναγνώριση είδους μουσικής με βάση το φασματογράφημα (spectrogram). Όπως είδαμε και στο εργαστήριο 2 το φασματογραφήματα είναι μια οπτική αναπαράσταση του συχνοτικού περιεχομένου ενός σήματος, όπου η εξαγώμενη εικόνα αναπαριστά τις διαφορετικές ζώνες συχνοτήτων ως προς το χρόνο.

Βήμα 0 **Εξοικείωση με Kaggle kernels**.

Ανοίξτε ένα (private) Kaggle notebook στη σελίδα του διαγωνισμού . Διερευνήστε τα δεδομένα και τους υποφακέλους, και δοκιμάστε να ενεργοποιήσετε και να απενεργοποιήσετε τη GPU. (**Προσοχή: Το kaggle σας προσφέρει έως 40 ώρες τη βδομάδα για την GPU.)**. Τέλος, κάντε commit τις αλλαγές σας. Σας παρέχουμε επίσης ένα notebook που σας δείχνει πως μπορείτε να προσθέσετε και να φορτώσετε τα δεδομένα στο link `https://www.kaggle.com/filby89/notebook44d591283b`.

Βήμα 1 **Εξοικείωση με φασματογραφήματα στην κλίμακα mel**.

Τα δεδομένα που θα χρησιμοποιήσετε στην προπαρασκευή είναι ένα υποσύνολο του Free Music Archive (FMA) dataset. Το FMA είναι μια βάση δεδομένων από ελεύθερα δείγματα (clips) μουσικής με επισημειώσεις ως προς το είδος της μουσικής. Έχουμε εξάγει τα φασματογραφήματα και τις επισημειώσεις τους στο φάκελο `fma_genre_spectrogram`.

Το αρχείο `fma_genre_spectrograms/train_labels.txt` περιέχει γραμμές του στη μορφή `spec_file label`.

(a) Διαλέξτε δύο τυχαίες γραμμές (με διαφορετικές επισημειώσεις). Τα αντίστοιχα αρχεία βρίσκονται στο φάκελο `fma_genre_spectrograms/train`.

(b) Διαβάστε τα αρχεία και πάρτε το φασματογράφημα σε κλίμακα mel σύμφωνα με τις οδηγίες του notebook.

(c) Απεικονίστε τα φασματογραφήματα για τα διαφορετικά labels με χρήση της συνάρτησης `librosa.display.specshow`. Σχολιάστε τι πληροφορία σας δίνουν και τις διαφορές για δείγματα που αντιστοιχούν σε διαφορετικές επισημειώσεις (labels). (υπόδειξη: συχνότητα στον κατακόρυφο άξονα, χρόνος στον οριζόντιο).

Βήμα 2 **Συχρονισμός φασματογραφημάτων στο ρυθμό της μουσικής (beat-synced spectrograms)**

(a) Τυπώστε τις διαστάσεις των φασματογραφημάτων του Βήματος 1.
- Πόσα χρονικά βήματα έχουν;
- Είναι αποδοτικό να εκπαιδεύσετε ένα LSTM πάνω σε αυτά τα δεδομένα;
- Γιατί;

(b) Ένας τρόπος να μειώσουμε να τα χρονικά βήματα είναι να συγχρονίσουμε τα φασματογραφήματα πάνω στο ρυθμό. Για αυτό το λόγο παίρνουμε τη διάμεσο (median) ανάμεσα στα σημεία που χτυπάει το beat της μουσικής. Τα αντίστοιχα αρχεία δίνονται στο φάκελο `fma_genre_spectrograms_beat`. Επαναλάβετε τα βήματα του Βήματος 1 για αντίστοιχα beat-synced spectrograms και σχολιάστε τις διαφορές με τα αρχικά.

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 αρκετά εύρωστα και στην αναγνώριση των αλλαγών του ηχοχρώματος και των οργάνων. Επαναλάβετε τα υποερωτήματα από τα Βήματα 1 και 2 για τα χρωμογραφήματα των αντίστοιχων αρχείων.

Βήμα 4 **Φόρτωση και ανάλυση δεδομένων**

(a) Στο βοηθητικό notebook που είδατε και παραπάνω σας παρέχεται έτοιμη μια υλοποίηση ενός PyTorch Dataset η οποία διαβάζει τα δεδομένα και σας επιστρέφει τα δείγματα. Μελετήστε τον κώδικα και τα δείγματα που επιστρέφει και σχολιάστε τις λειτουργίες που εκτελούνται.

(b) Στον κώδικα που σας δίνουμε συγχωνεύουμε κλάσεις που μοιάζουν μεταξύ τους και αφαιρούμε κλάσεις που αντιπροσωπεύονται από πολύ λίγα δείγματα. Σχολιάστε γιατί είναι αναγκαίο να γίνει αυτό.

(c) Σχεδιάστε δύο ιστογράμματα που θα δείχνουν πόσα δείγματα αντιστοιχούν σε κάθε κλάση, ένα πριν από τη διαδικασία του βήματος 4β και ένα μετά.

Βήμα 5 **Αναγνώριση μουσικού είδους με Long Short-Term Memory (LSTM) Network**

Με τη βοήθεια του κώδικα που υλοποιήσατε στη δεύτερη άσκηση:

(a) εκπαιδεύστε ένα LSTM [1] δίκτυο, το οποίο θα δέχεται ως είσοδο τα φασματογραφήματα του συνόλου εκπαίδευσης (train set) και θα προβλέπει τις διαφορετικές κλάσεις (μουσικά είδη) του συνόλου δεδομένων (dataset).

(b) εκπαιδεύστε ένα LSTM δίκτυο, το οποίο θα δέχεται ως είσοδο τα beat-synced spectrograms (train set) και θα προβλέπει τις διαφορετικές κλάσεις (μουσικά είδη) του dataset.

(c) εκπαιδεύστε ένα LSTM δίκτυο, το οποίο θα δέχεται ως είσοδο τα χρωμογραφήματα (train set) και θα προβλέπει τις διαφορετικές κλάσεις (μουσικά είδη) του dataset.

(d) (extra credit) εκπαιδεύστε ένα LSTM δίκτυο, το οποίο θα δέχεται ως είσοδο τα ενωμένα (concatenated) χρωμογραφήματα και φασματογραφήματα (train set) και θα προβλέπει τις διαφορετικές κλάσεις (μουσικά είδη) του dataset.

Υπόδειξη: Για την εκπαίδευση χρησιμοποιήστε και σύνολο επαλήθευσης (validation set).

Υπόδειξη: Για την εκπαίδευση ενεργοποιήστε τη GPU.

Υπόδειξη: Για να επισπεύσετε τη διαδικασία ανάπτυξης και αποσφαλμάτωσης των μοντέλων σας προτείνονται οι ακόλουθες 2 τεχνικές (δείτε και τα [13, 2, 16]):

- Εκπαίδευση και επαλήθευση για λίγες εποχές σε λίγα (4-5) batches: Στόχος αυτού είναι να βεβαιωθείτε ότι το δίκτυο μπορεί να τρέξει απροβλημάτιστα ένα κύκλο εκπαίδευσης (2-3 εποχών) σε ένα πολύ μικρό υποσύνολο των πραγματικών δεδομένων. Χρήσιμο για να πιάσουμε μικρά λάθη νωρίς.
- Υπερεκπαίδευση του δικτύου σε ένα batch: Μια καλή πρακτική κατά την ανάπτυξη νευρωνικών είναι να βεβαιωθούμε ότι το δίκτυο μπορεί να εκπαιδευτεί (τα gradients γυρνάνε πίσω κτλ). Ένας γρήγορος τρόπος για να γίνει αυτό είναι να επιλέξουμε τυχαία ένα πολύ μικρό υποσύνολο των δεδομένων (ένα batch) και να εκπαιδεύσουμε το δίκτυο για πολλές εποχές πάνω σε αυτό. Αυτό που περιμένουμε να δούμε είναι το σφάλμα εκπαίδευσης να πάει στο 0 και το δίκτυο να κάνει overfit.

Βήμα 6 **Αξιολόγηση των μοντέλων**

Αναφέρετε τα αποτελέσματα των μοντέλων από το Βήμα 5 στο δύο ακόλουθα σύνολα αξιολόγησης (test sets)

- `fma_genre_spectrograms_beat/test_labels.txt`
- `fma_genre_spectrograms/test_labels.txt`

Συγκεκριμένα:

(a) υπολογίστε το accuracy

(b) υπολογίστε το precision, recall και F1-score για κάθε κλάση

(c) υπολογίστε το macro-averaged precision, recall και F1-score για όλες τις κλάσεις

(d) υπολογίστε το micro-averaged precision, recall και F1-score για όλες τις κλάσεις Αναφέρετε την ερμηνεία των μετρικών αυτών και σχολιάστε ποια από αυτές τις μετρικές θα επιλέγατε για την αξιολόγηση ενός ταξινομητή σε αυτό το πρόβλημα. Συγκεκριμένα εστιάστε στις ερωτήσεις:

- Τι δείχνει το accuracy / precision / recall / f1 score;
- Τι δείχνει το micro / macro averaged precision / recall / f1 score;
- Πότε μπορεί να έχω μεγάλη απόκλιση ανάμεσα στο accuracy / f1 score και τι σημαίνει αυτό;
- Πότε μπορεί να έχω μεγάλη απόκλιση ανάμεσα στο micro/macro f1 score και τι σημαίνει αυτό; Υπάρχουν προβλήματα όπου το precision με ενδιαφέρει περισσότερο από το recall και αντίστροφα; Είναι ένα καλό accuracy / f1 αρκετό σε αυτές τις περιπτώσεις για να επιλέξω ένα μοντέλο;

Υπόδειξη: Χρησιμοποιήστε τη συνάρτηση `sklearn.metrics.classification_report`

Υπόδειξη: Δείτε τα [12, 9, 11].

Τέλος Προπαρασκευής

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 - CNN).

(a) Στο σύνδεσμο [8] μπορείτε να εκπαιδεύσετε απλά συνελικτικά δίκτυα και να δείτε την εσωτερική λειτουργία του δικτύου οπτικοποιώντας τις ενεργοποιήσεις (activations) των επιμέρους επιπέδων του δικτύου χωρίς προγραμματιστικό κόπο. Εκπαιδεύστε ένα δίκτυο στο MNIST και παρατηρήστε τη λειτουργία των ενεργοποιήσεων κάθε επιπέδου. Σχολιάστε τις επιμέρους λειτουργίες, τι φαίνεται να μαθαίνει το δίκτυο και δώστε κατάλληλα screenshots στην αναφορά.

(b) Υλοποιήστε ένα 2D CNN με 4 επίπεδα (layers) που θα επεξεργάζεται το φασματογράφημα σαν μονοκάναλη εικόνα, να το εκπαιδεύσετε στο train + validation set και να αναφέρετε τα αποτελέσματα στο test set. Κάθε επίπεδο θα πραγματοποιεί τις εξής λειτουργίες (operations) με αυτή τη σειρά:

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

(c) Εξηγήστε τη λειτουργία και τον ρόλο των convolutions, batch normalization, ReLU και Max pooling. Παραπέμπουμε στις αναφορές [6, 14, 7].

(d) Χρησιμοποιήστε αυτή την αρχιτεκτονική για την αναγνώριση μουσικού είδους με φασματογραφήματα και συγκρίνετε με το μοντέλο 5α.

Υπόδειξη: Εκτελέστε το μοντέλο σε διαφορετικό kernel για να αποφύγετε προβλήματα μνήμης.

Υπόδειξη: Ισχύουν όλες οι υποδείξεις του Βήματος 5

Υπόδειξη: Μην σπαταλήσετε πολύ χρόνο στη ρύθμιση των υπερπαραμέτρων του δικτύου. Απλά δείτε κάποιες έτοιμες online υλοποιήσεις από CNNs και βάλτε κάποιες "λογικές" τιμές (πχ kernel size ~ 3 ή 5) κτλ. Αν το δίκτυο σας δε λειτουργεί όπως θα έπρεπε, είναι πιο πιθανό να οφείλεται σε κάποιο λάθος (bug) στον κώδικα από την κακή επιλογή παραμέτρων, ειδικά αν δεν αποκλίνουν πολύ τις προεπιλεγμένες (default) τιμές.

Βήμα 8 **Εκτίμηση συναισθήματος - συμπεριφοράς με παλινδρόμηση**

Σε αυτό το βήμα θα χρησιμοποιήσετε το multitask dataset:

`multitask_dataset/train_labels.txt`.

Εδώ σας 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, χορευτικό κοντά στο 1.

(a) Προσαρμόστε το καλύτερο μοντέλο του Βήματος 5 και το μοντέλο του Βήματος 7 για παλινδρόμηση (regression) αλλάζοντας τη συνάρτηση κόστους.

(b) Εκπαιδεύστε τα μοντέλα του 8α για την εκτίμηση του valence.

(c) Επαναλάβετε για την εκτίμηση του energy.

(d) Επαναλάβετε για την εκτίμηση του danceability.

(e) Η τελική μετρική είναι το μέσο Spearman correlation ανάμεσα στις πραγματικές (ground truth) τιμές και στις προβλεπόμενες τιμές για όλους τους άξονες.

Υπόδειξη: Προσοχή. Σε αυτό το σύνολο δεδομένων δε σας παρέχονται οι επισημειώσεις για το test set, οπότε η εκτίμηση του πόσο καλά γενικεύει το μοντέλο θα πρέπει να γίνει παίρνοντας ένα υποσύνολο από τα δεδομένα που σας δίνονται.

Τα βήματα 9α και 9β είναι προαιρετικά για τους μεταπτυχιακούς φοιτητές και υποχρεωτικά για τους προπτυχιακούς φοιτητές

- **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. Για αυτό το λόγο

(a) Δείτε τα links [15, 10, 3]. Περιγράψτε με 2 προτάσεις τα βασικά συμπεράσματα του [3].

(b) Επιλέξτε ένα μοντέλο από τα βήματα 5, 7. Εξηγήστε γιατί επιλέξατε αυτό το μοντέλο.

(c) Εκπαιδεύστε αυτό το μοντέλο στο `fma_genre_spectrograms` dataset και αποθηκεύστε τα βάρη του δικτύου στην εποχή που έχει την καλύτερη επίδοση (checkpoint)

(d) Αρχικοποιήστε ένα μοντέλο με αυτά τα βάρη για το πρόβλημα του ερωτήματος 10 και εκπαιδεύστε το για λίγες εποχές (fine tuning) στο multitask dataset. Για ευκολία μπορείτε να αναφέρετε τα αποτελέσματα μόνο για έναν από τους 3 άξονες.

(e) Συγκρίνετε τα αποτελέσματα με αυτά από το Βήμα 8

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

Στο Βήμα 8 εκπαιδεύσατε 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.

(a) Δείτε τα links [15, 4, 5] και περιγράψτε με 2 προτάσεις τα βασικά συμπεράσματα του [5].

(b) Εκπαιδεύστε ένα μοντέλο στο multitask dataset χρησιμοποιώντας σαν συνάρτηση κόστους το άθροισμα από τα κόστη (losses) για το valence, energy και danceability. Μπορείτε να χρησιμοποιήσετε βάρη για να φέρετε τα επιμέρους κόστη στην ίδια τάξη μεγέθους.

(c) Συγκρίνετε τα αποτελέσματα με αυτά από το Βήμα 8

Βήμα 10 **(Προαιρετικό): Υποβολή στο Kaggle**

(a) Επιλέξτε το καλύτερο μοντέλο σας για το multitask dataset και πραγματοποιήστε προβλέψεις για το valence, energy και danceability στα test δεδομένα.

(b) Διαμορφώστε ένα αρχείο solution.txt στη μορφή

```
Id.fused.full.npy.gz,valence,energy,danceability
123212738.fused.full.npy.gz,0.153,0.961,0.013
```

(c) Υποβάλετε το solution.txt στο διαγωνισμό στο kaggle και δείτε τα αποτελέσματα στο leaderboard.

(d) Σχολιάστε πόσο κοντά είναι τα αποτελέσματά σας με αυτά που περιμένατε.

ΠΑΡΑΔΟΤΕΑ

1. Σύντομη αναφορά σε pdf που θα περιγράφει τη διαδικασία που ακολουθήθηκε σε κάθε βήμα, καθώς και τα σχετικά αποτελέσματα. Τα αποτελέσματα πρέπει να συνοδεύονται και από ερμηνεία – σχολιασμό.
2. Κώδικας Python (συνοδευόμενος από σύντομα σχόλια). Προσπαθήστε να κάνετε vectorized υλοποιήσεις.

Συγκεντρώστε τα (1) και (2) σε ένα .zip αρχείο το οποίο πρέπει να αποσταλεί μέσω του moodle του μαθήματος (https://courses.pclab.ece.ntua.gr/course/view.php?id=18).

References

[1] https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[2] http://karpathy.github.io/2019/04/25/recipe/.

[3] http://papers.nips.cc/paper/5347-how-transferable-are-features-in-deep-neural-networks.pdf.

[4] http://ruder.io/multi-task/.

[5] https://arxiv.org/pdf/1706.05137.pdf.

[6] https://blog.xrds.acm.org/2016/06/convolutional-neural-networks-cnns-illustrated-explanation/.

[7] https://colah.github.io/posts/2014-07-Understanding-Convolutions/.

[8] https://cs.stanford.edu/people/karpathy/convnetjs/.

[9] https://medium.com/@george.drakos62/how-to-select-the-right-evaluation-metric-for-machine-learning-models-part-3-classification-3eac420ec991.

[10] https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a.

[11] https://towardsdatascience.com/metrics-for-imbalanced-classification-41c71549bbb5.

[12] https://towardsdatascience.com/multi-class-metrics-made-simple-part-ii-the-f1-score-ebe8b2c2ca1.

[13] https://twitter.com/karpathy/status/1013244313327681536.

[14] https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/.

[15] https://www.coursera.org/lecture/machine-learning-projects/transfer-learning-WNPap.

[16] https://www.reddit.com/r/MachineLearning/comments/5pidk2/d_is_overfitting_on_a_very_small_data_set_a/.

Αναγνώριση Προτύπων

Αναγνώριση Είδους και Εξαγωγή Συναισθήματος από Μουσική

Τρυφωνόπουλος Δημήτρης

ΕΔΕΜ

0370034

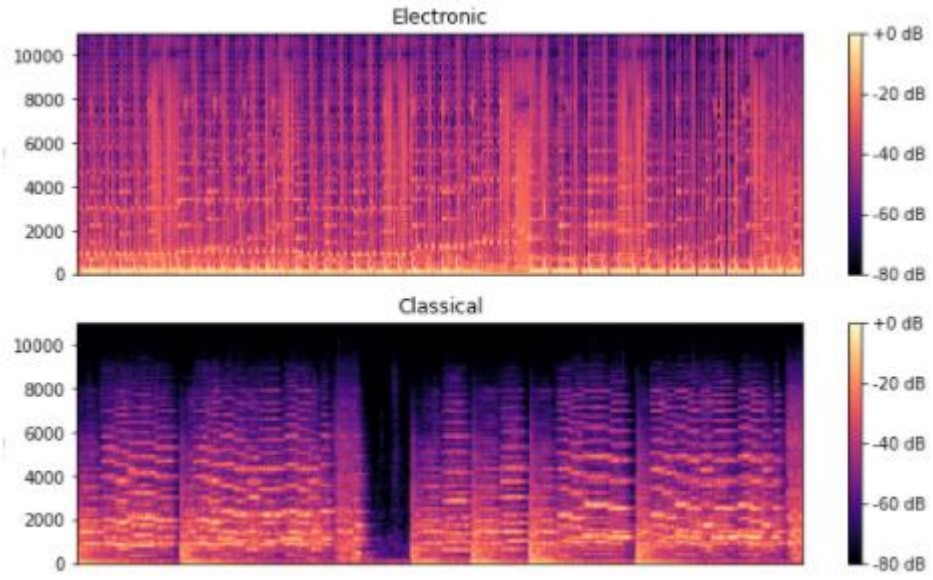
ΒΗΜΑ 1:

a. Σε αυτό το εργαστήριο χρησιμοποιήσαμε ένα υποσύνολο του Free Music Archive Dataset, το οποίο αποτελείται από υψηλής ποιότητας ακουστικά δεδομένα καθώς επίσης και από διάφορα χαρακτηριστικά που έχουν υπολογιστεί εκ των προτέρων. Περισσότερα από 106 χιλιάδες τραγούδια από >16 χιλιάδες καλλιτέχνες και με πάνω από 14 χιλιάδες άλμπουμ, σε μορφή MP3. Από αυτά θα χρησιμοποιήσουμε 8.000 κομμάτια από 8 διαφορετικά είδη. Θα κατηγοριοποιήσουμε τα δείγματα με βάση το είδος τους.

   Θα χρησιμοποιήσουμε τα spectrograms – αναπαραστάσεις συχνοτήτων- που προκύπτουν χρησιμοποιώντας τον μετασχηματισμό Fourier του σήματος. Επίσης τα mel-spectrograms η ίδια αναπαράσταση αλλά προβαλλόμενη στην κλίμακα mel, στην οποία οι συχνότητες είναι εγγύτερα σε αυτές που αντιλαμβάνεται το ανθρώπινο αυτί. Τέλος θα χρησιμοποιήσουμε και τα chromagrams τα οποία σχετίζονται με τις μουσικές νότες. Τα chromagrams περιέχουν πληροφορίες για τα αρμονικά και μελωδικά χαρακτηριστικά ανεξαρτήτως μουσικού οργάνου.

b. Εξάγουμε τα spectrograms και mel-spectrograms.

c.

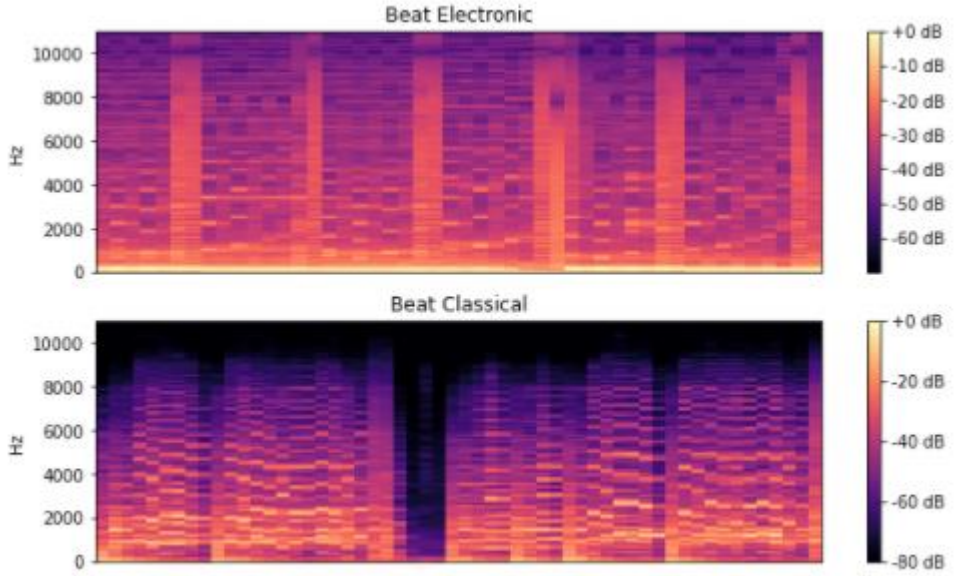


   Παρατηρούμε μια αναπαράσταση των spectrum frequencies κατά την πάροδο του χρόνου. Παρατηρούμε τους διαφορετικούς τόνους (κίτρινες τελείες) που αναπαράγονται σε μικρά χρονικά διαστήματα.
   Τα είδη που επιλέχθηκαν ήταν Electronic και Classical music. Για την Electronic music παρατηρούμε κάθετες μπάρες που ίσως ανταποκρίνονται στο beat και κατά κάποιο τρόπο τα μοτίβα αυτά φαίνεται να επαναλαμβάνονται (χαρακτηριστικο της ηλεκτρονικής μουσικής). Από την άλλη στην Classical music βλέπουμε πιο συνεχείς και χαμηλότερης συχνότητας ήχους που μάλλον οφείλονται στην ύπαρξη περισσότερων μουσικών οργάνων και μάλλον εγχόρδων εξού και η συνεχής συμπεριφορά της συχνότητας.

ΒΗΜΑ 2:

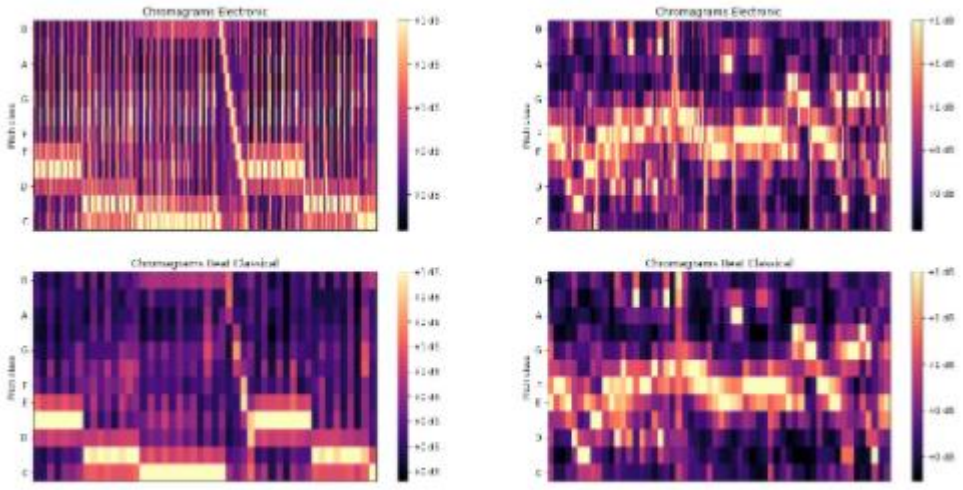
a. Από τις διαστάσεις βλέπουμε ότι έχουμε 140 διαφορετικές συχνότητες αλλά τα 2 είδη είναι διαφορετικής διάρκειας – μήκους (1291 & 1292 αντίστοιχα). Κατά την γνώμη μου αυτή η αναπαράσταση περιέχει μεγάλο όγκο πληροφορίας και πιθανόν πλεονάζουσας γεγονός που καθιστά την χρήση LSTMs' ιδιαίτερα ακριβή και σύνθετη υπολογιστικά.

b. Μία λύση για μείωση της χρονικής διάρκειας θα μπορούσε να ήταν μέσω της χρήσης του beat όπου το μήκος τώρα είναι 48 και 56 αντίστοιχα.



ΒΗΜΑ 3:

Τέλος με την χρήση των Chromagrams, χρησιμοποιώντας τις μουσικές νότες C, C#, D, D#, E, F, F#, G, G#, A, A#, B) μπορούμε να έχουμε μια άλλη αναπαράσταση. Παρατηρούμε τα Chromagrams και beat-synced chromagrams για τα 2 είδη.



ΒΗΜΑ 4:

a. Τα functions που μας δόθηκαν για βοήθεια επιτελούν τις παρακάτω λειτουργίες.

Torch_train_var_split: χωρίζει το dataset σε train και validation

Read_mel_spectrogram: διαβάζει μόνο τα 128 στοιχεία της εισόδου τα οποία απαρτίζουν το spectrogram.

Read_chromagram: διαβάζει τα 128 τελευταία στοιχεία που αποτελούν το chromagram.

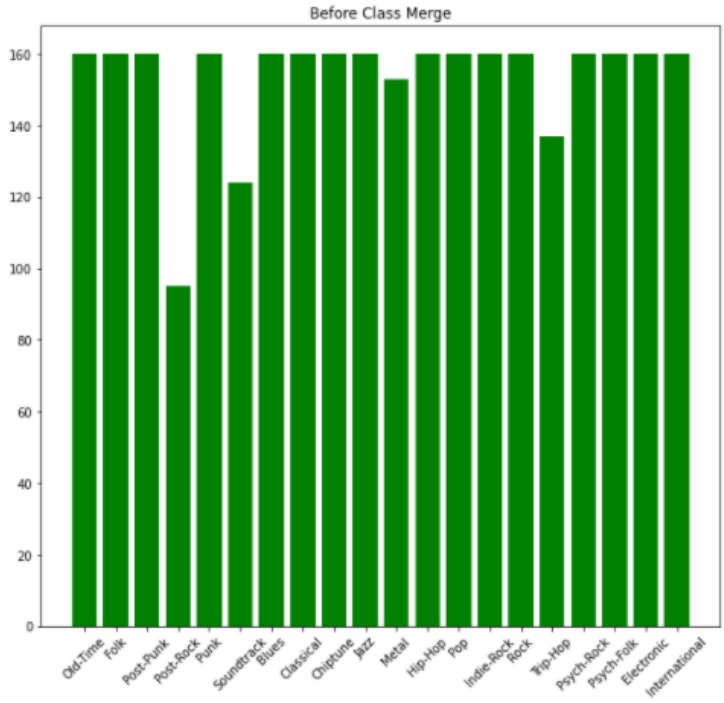
Read_fused_sptectrogram: διαβάζει όλα τα 256 στοιχεία της εισόδου τα οποία δημιουργούν τα spectrogram και chromagram.

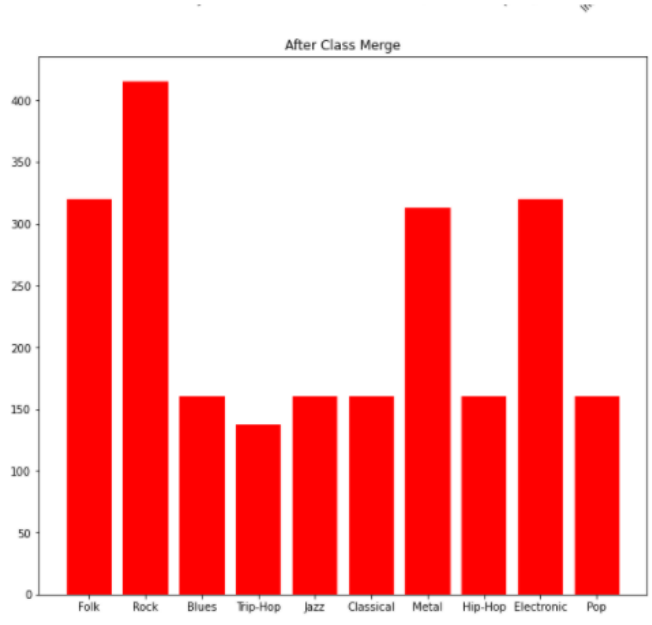
Get_files_labels: βρίσκει το είδος για το κάθε dataset.

PaddingTransform: φέρνει με την χρήση padding όλα τα χαρακτηριστικά στο ίδιο μήκος.

b. Πριν την εκπαίδευση συμπτύσσουμε ομοιόμορφα είδη μουσικής.

c. Παρατηρούμε την αρχική κατανομή των ειδών μουσικής και αυτή μετά την σύμπτυξη





ΒΗΜΑ 5:

Χρησιμοποιούμε το LSTM με διαστάσεις 128, 128 ,12 και 140 για τα spectrograms, beat-synced spectrograms, chromagrams και concatenated spectro/chromograms αντίστοιχα. Πραγματοποιούμε την εκπαίδευση και έχουμε τα παρακάτω αποτελέσματα:

Μπλε – training learning curve Πορτοκαλί - validation learning curve

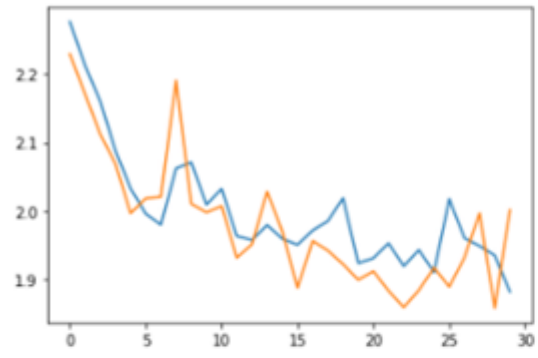


Figure 1: Beat Synced Spectrogram

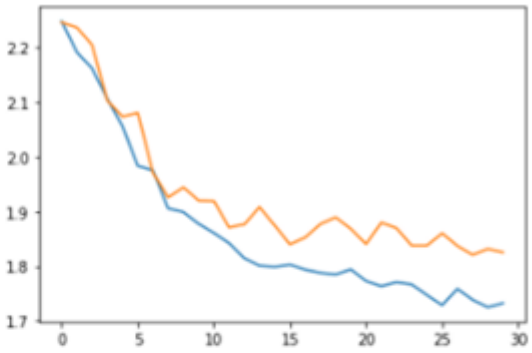


Figure 2: Spectrogram

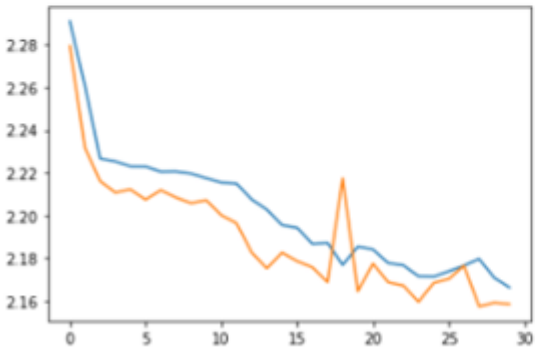


Figure 3: Chromograms

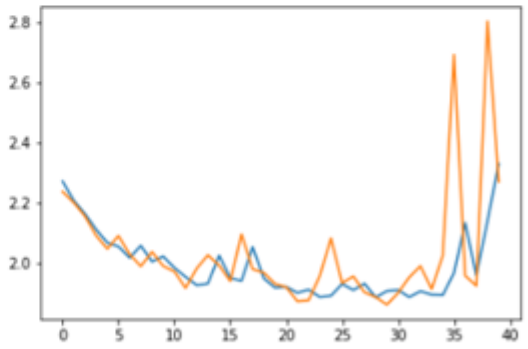


Figure 4: Chromograms & Spectrograms Together

ΒΗΜΑ 6:

Χρησιμοποιώντας την συνάρτηση ***sklearn.metrics.classification_report*** πραγματοποιούμε το validation του μοντέλου μας, σε αντιστοιχία με τα παραπάνω διαγράμματα.

Βασισμένοι στις επόμενες κατηγορίες δειγμάτων κατασκευάζουμε και τις ζητούμενες μετρικές:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	40
1	0.33	0.62	0.43	40
2	0.28	0.50	0.36	80
3	0.24	0.51	0.33	80
4	0.50	0.03	0.05	40
5	0.00	0.00	0.00	40
6	0.48	0.28	0.35	78
7	0.00	0.00	0.00	40
8	0.34	0.47	0.39	103
9	0.00	0.00	0.00	34
accuracy			0.31	575
macro avg	0.22	0.24	0.19	575
weighted avg	0.26	0.31	0.25	575

	precision	recall	f1-score	support
0	0.00	0.00	0.00	40
1	0.49	0.53	0.51	40
2	0.36	0.68	0.47	80
3	0.34	0.61	0.44	80
4	0.22	0.20	0.21	40
5	0.20	0.10	0.13	40
6	0.48	0.62	0.54	78
7	0.00	0.00	0.00	40
8	0.39	0.27	0.32	103
9	0.12	0.03	0.05	34
accuracy			0.37	575
macro avg	0.26	0.30	0.27	575
weighted avg	0.30	0.37	0.32	575

	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	0.68	0.31	80
4	0.00	0.00	0.00	40
5	0.00	0.00	0.00	40
6	0.23	0.29	0.26	78
7	0.00	0.00	0.00	40
8	0.19	0.36	0.25	103
9	0.00	0.00	0.00	34
accuracy			0.20	575
macro avg	0.06	0.13	0.08	575
weighted avg	0.09	0.20	0.12	575

	precision	recall	f1-score	support
0	0.00	0.00	0.00	40
1	0.39	0.60	0.48	40
2	0.30	0.75	0.42	80
3	0.36	0.40	0.38	80
4	0.00	0.00	0.00	40
5	0.00	0.00	0.00	40
6	0.44	0.36	0.39	78
7	0.00	0.00	0.00	40
8	0.33	0.50	0.40	103
9	0.00	0.00	0.00	34
accuracy			0.34	575
macro avg	0.18	0.26	0.21	575
weighted avg	0.24	0.34	0.27	575

- TP – true positive
- TN – true negative
- FP – false positive
- FN – false negative

- **Accuracy:** το οποίο μας δείχνει την αναλογία μεταξύ σωστών και λάθος ταξινομημένων δειγμάτων.

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

- **Precision:** Τα πραγματικά σωστά δείγματα ως προς το σύνολο των δειγμάτων που ταξινομήθηκαν ως σωστά.

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

- **Recall:** Τα πραγματικά σωστά δείγματα ως προς το σύνολο των δειγμάτων που ήταν πραγματικά σωστά ταξινομημένα (P and N)

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

- **F1-Score:** επειδή το precision και το recall όταν αλλάζουν αλληλοεπηρεάζονται το F1-Score είναι μια μέτρηση που συμπεριλαμβάνει και τα 2.

$$F1-Score = \frac{precision - recall}{precision + recall}$$

Τα Accuracy, Precision και Recall αναφέρονται σε κάθε κλάση συγκεκριμένα από την άλλη μετρικές με weighted average μπορούν να χρησιμοποιηθούν για την βέλτιστη κατανομή των βαρών μεταξύ διαφορετικών κλάσεων (micro and macro respectively). Τα macro scores θα αναλογιστούν ισόνομα όλες τις κλάσεις.

Ανάλογα με την περίσταση τα metrics μπορεί να είναι περισσότερο ή λιγότερο κατάλληλα. Το accuracy είναι σημαντικό όταν μας ενδιαφέρουν κυρίως ο σωστός αριθμός προβλέψεων. Από την άλλη το F1-score είναι πιο κατάλληλο όταν μας ενδιαφέρουν ο αριθμός των λανθασμένων προβλέψεων. Ένα χαρακτηριστικό παράδειγμα είναι σε διαγνώσεις ασθενειών όπου η λανθασμένη πρόβλεψη – ‘δεν έχει την ασθένεια’ – θα είχε καταστροφικές επιπτώσεις για τον ασθενή. Συνεπώς θα θέλαμε υψηλό recall για τον εντοπισμό των πραγματικά ασθενών. Αντίθετα σε περιπτώσεις αλγορίθμων για προτεινόμενες ταινίες ή μουσική το precision θα μας ενδιέφερε περισσότερο ώστε οι προτάσεις μας να ήταν πιο σωστές.

Στην περίπτωση της άσκησης αυτής λοιπόν, και μιας και τα δείγματα για τα είδη της μουσικής δεν είναι ομοιόμορφα κατανεμημένα το accuracy και το recall διαφέρουν σημαντικά για τις διαφορετικές κλάσεις. Επίσης ανάλογα με την πληροφορία που χρησιμοποιούμε στα LSTMs οι γενικότερες αποδόσεις είναι καλύτερες (beat-synced mel spectrograms). Μία εξήγηση θα μπορούσε να ήταν το μικρότερο μέγεθος των δειγμάτων σε σχέση με τις άλλες εισόδους. Πιθανών τα LSTM να μην είναι το κατάλληλο μοντέλο για αυτού του είδους classification (μουσική).

ΒΗΜΑ 7:

a. Από την προτεινόμενη ιστοσελίδα παρατηρώντας το CNN εξοικειωθήκαμε τα την λειτουργία των εν' λόγω δικτύων. Πιο συγκεκριμένα τα φίλτρα στο 1^{ο} Layer που ενεργούν πάνω στα pixel απευθείας φαίνονται να αναγνωρίζουν χαμηλού-επιπέδου χαρακτηριστικά όπως lines, edges and corners. Για κάθε ένα από τα φίλτρα (8) έχουμε και ένα feature map (total-8). Η Relu στη συνέχεια αποκλείει τα pixel με χαμηλές τιμές και το pooling layer επισυνάπτει τα συνολικά χαρακτηριστικά από 2χ2 pixel-περιοχές (max pooling kernel-πυρήνα) οδηγώντας σε αναπαράσταση της μισής από την προηγούμενη διάσταση (24x24x8 -> 12x12x8). Η παραπάνω διαδικασία επαναλαμβάνεται (convolution/relu/pooling layers) και παρατηρούμε ότι όσο το δίκτυο γίνεται βαθύτερο εξάγει πιο δυσνόητα χαρακτηριστικά υψηλότερων-επιπέδων. Τέλος το τελευταίο layer προβάλλεται πάνω στις διαφορετικές κλάσεις και με την χρήση της softmax έχουμε πιθανότητες για την κλάση στην οποία ενδεχομένως ανήκει το κάθε ψηφίο με βάση την πρόβλεψη του μοντέλου.

b. Στο βήμα αυτό δημιουργούμε ένα 2D CNN με 4 layers το οποίο δέχεται σαν είσοδο τα spectrograms. Σε κάθε layer το CNN χρησιμοποιεί μια 2D convolution, batch normalization, ReLU and max-pooling. Στο τέλος η έξοδος έγινε flatten (μετατροπή σε ένα vector) πριν περάσει μέσα από ένα fully connected layer. Μετά την εκπαίδευση είχαμε τα παρακάτω αποτελέσματα.

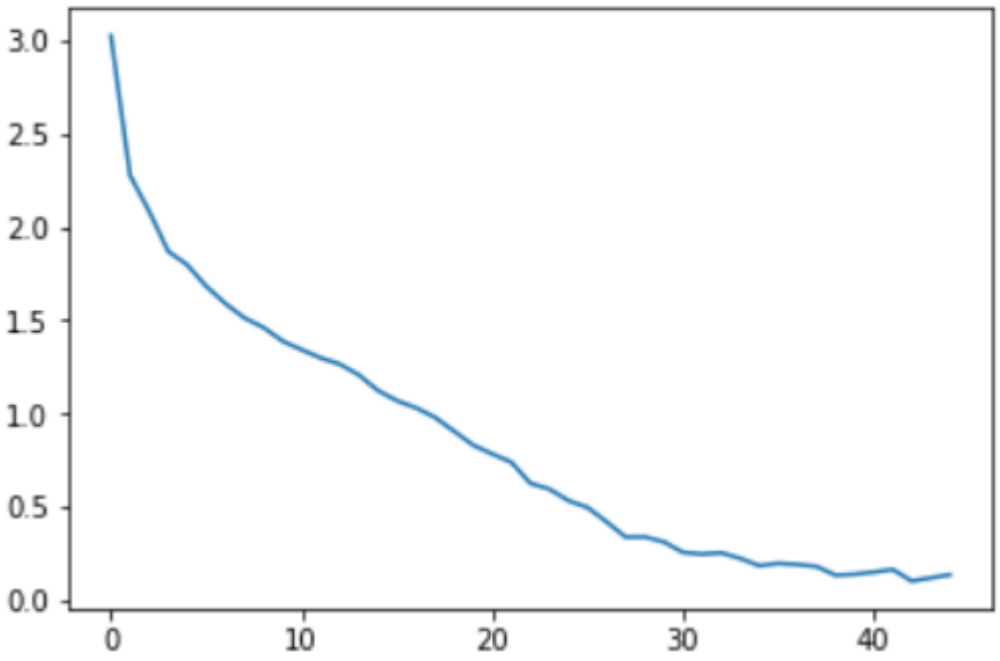


Figure 5: Learning Curve

	precision	recall	f1-score	support
0	0.29	0.33	0.31	40
1	0.68	0.65	0.67	40
2	0.66	0.56	0.61	80
3	0.45	0.35	0.39	80
4	0.65	0.70	0.67	40
5	0.39	0.40	0.40	40
6	0.54	0.59	0.56	78
7	0.11	0.07	0.09	40
8	0.39	0.53	0.45	103
9	0.38	0.26	0.31	34
accuracy			0.47	575
macro avg	0.45	0.45	0.45	575
weighted avg	0.47	0.47	0.46	575

c. Παρακάτω ακολουθεί μια συνοπτική ανάλυση στα χαρακτηριστικά που απαρτίζουν τα CNNs.

2D Convolution

Χρησιμοποιείται κατά κόρων στην Ανάλυση σήματος για τον συνδυασμό 2 ή περισσότερων χρονικών συναρτήσεων σε 1 ή περισσότερες διαστάσεις. Στην ανάλυση εικόνας μπορεί να περιγραφεί ως το πέρασμα της μιας εικόνας πάνω από την άλλη για την δημιουργία μιας τρίτης εικόνας. Τα περισσότερα από τα transformations σε Image analysis αφορούν convolutions με διαφορετικά kernels ή filters. Ένας kernel περνάει από όλα τα σημεία μιας εικόνας δημιουργώντας μια τιμή πραγματική τιμή για να συνοψίσει με διαφορετικές στρατηγικές τα pixel που έχει ήδη διαπεράσει οδηγώντας σε μια άλλη αναπαράσταση της παραπάνω εικόνας (fearure map). Ο kernel που διαπερνάει την εικόνα όπως αναφέραμε έχει διαφορετικές μεθόδους για την εξαγωγή της νέας τιμής του pixel, που χαρακτηρίζονται από διαφορετικά βάρη. Στο DL τα βάρη αυτά αναγνωρίζονται κατά την διαδικασία της εκπαίδευσης του μοντέλου. Τα διαφορετικά είδη convolution kernels που εφαρμόζονται σε μια εικόνα οδηγούν στην δημιουργία διαφορετικών απεικονίσεων χαρακτηριστικών (feature maps). Το κάθε convolution αναγνωρίζεται από τα χαρακτηριστικά του – hyperparameters. Μιας και δεν μπορούμε να συνδέσουμε όλα τα neurons από κάθε layer με όλες τις πιθανές περιοχές της εικόνας, έχουμε το receptive field, το οποίο είναι η διαστάσεις της περιοχής της εικόνας όπου τα περιεχόμενα pixel συνδέονται με την είσοδο του Neural Network (input layer).

Σε πολλές περιπτώσεις που επιθυμούμε η έξοδος του δικτύου να έχει την ίδια διάσταση με την είσοδο χρησιμοποιούμε zero-padding. Προσθέτουμε δηλαδή μηδενικά στην χαμηλότερης διάστασης εικόνα ώστε να την φέρουμε στο επιθυμητό μέγεθος.

Επίσης έχουμε το stride το οποίο είναι το βήμα μετακίνησης του convolution kernel πάνω στην εικόνα. Τέλος, οι κύριες hyperparameters του Convolution είναι οι διαστάσεις του kernel (WxHxD – μήκος-πλάτος-βάθος).

Batch Activation (Normalization)

Είναι μια τεχνική που χρησιμοποιείται για την ευκολότερη εκπαίδευση βαθιών νευρωνικών δικτύων για την σταθεροποίηση την διαδικασίας εκπαίδευσης που οδηγεί σε μείωση των απαιτούμενων κύκλων εκπαίδευσης (epochs). Πιο συγκεκριμένα το normalization του κάθε batch γίνεται re-scaling ώστε να έχει mean = 0 και std =1.

ReLU activation

Rectified Linear Unit. Αντικαθιστά τις αρνητικές τιμές των gradients με 0 και συνεπώς μόνο neurons με θετικά gradients συγκαταλέγονται στην έξοδο κάθε layer.

Max Pooling

Χρησιμοποιείται για να μειώσει τις διαστάσεις ενός feature map, κρατώντας της σημαντικότερες πληροφορίες που περιέχει. Μια περιοχή επιλέγεται, περνά ώς παράθυρο από την εικόνα και μόνο το pixel με τη μεγαλύτερη τιμή της γειτονιάς αυτής αποθηκεύεται.

ΒΗΜΑ 8:

Χρησιμοποιούμε το MSE σαν loss-function μιας και πρόκειται για regression task. Μετά την εκπαίδευση του LSTM και του CNN έχουμε τα επόμενα χαρακτηριστικά.

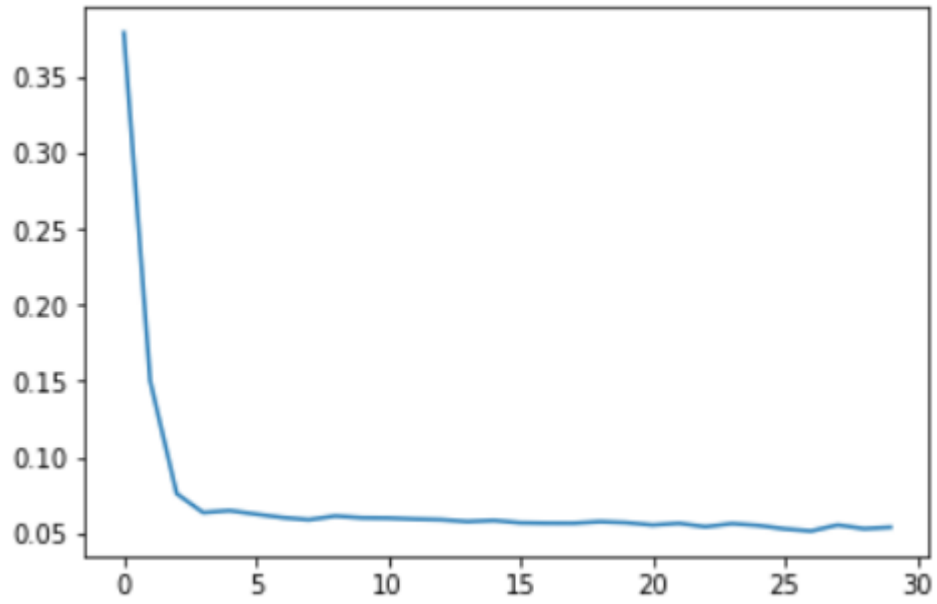


Figure 6: Valence1

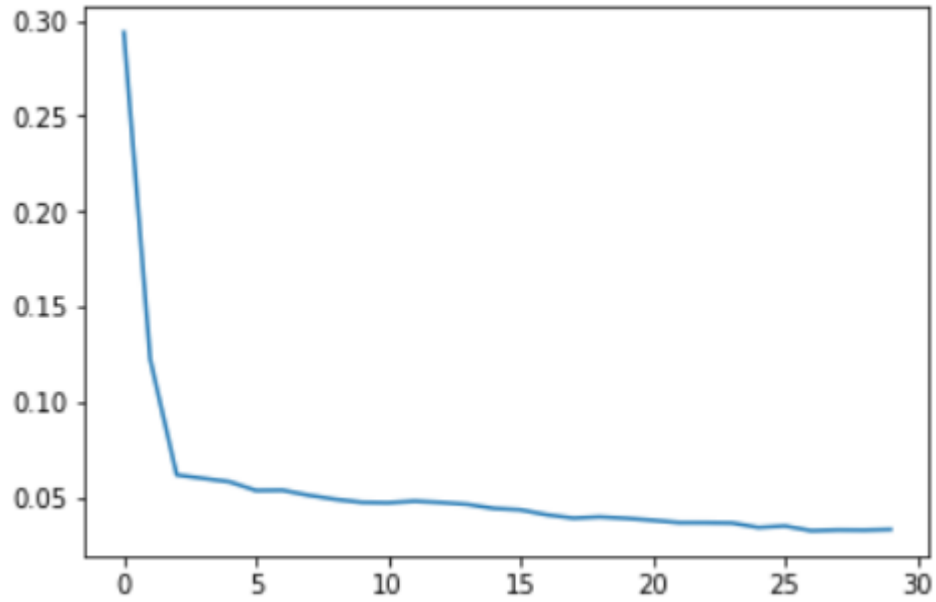


Figure 7: Energy

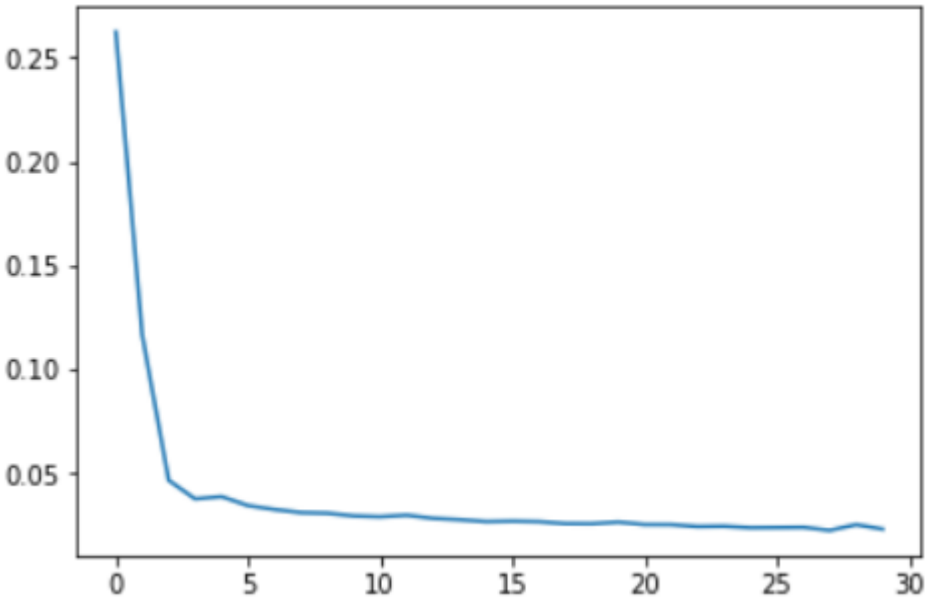


Figure 8: Danceability

	Spearman Correlation			
	Energy	Valence	Danceability	Mean
LSTM	0.715	0.422	0.604	0.580
CNN	0.757	0.570	0.621	0.650