

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Χειμερινό εξάμηνο 2022-23



Αναγνώριση Προτύπων

3η Εργαστηριακή Άσκηση: Αναγνώριση Είδους
και Εξαγωγή Συναισθήματος από Μουσική

Δημήτριος Κοκκίνης: 03118896

Χριστίνα Ρεντίφη: 03118217

Περιγραφή εργαστηρίου

Στόχος της εργαστηριακής άσκησης είναι η αναγνώριση του είδους και η εξαγωγή συναισθημάτων από φασματογραφήματα μουσικών κομματιών. Θα εργαστούμε με δύο σύνολα δεδομένων:

- Το Free Music Archive το οποίο περιέχει 3834 δείγματα χωρισμένα σε 20 κλάσεις
- Το multitask music, που περιέχει 1497 δείγματα με labels συναισθηματικών διαστάσεων (valence, energy, danceability)

Στην εργασία, θα ασχοληθούμε με την ανάλυση φασματογραφημάτων με τη χρήση βαθιών αρχιτεκτονικών με συνελκτικά δίκτυα (CNN) και αναδρομικά νευρωνικά δίκτυα (RNN).

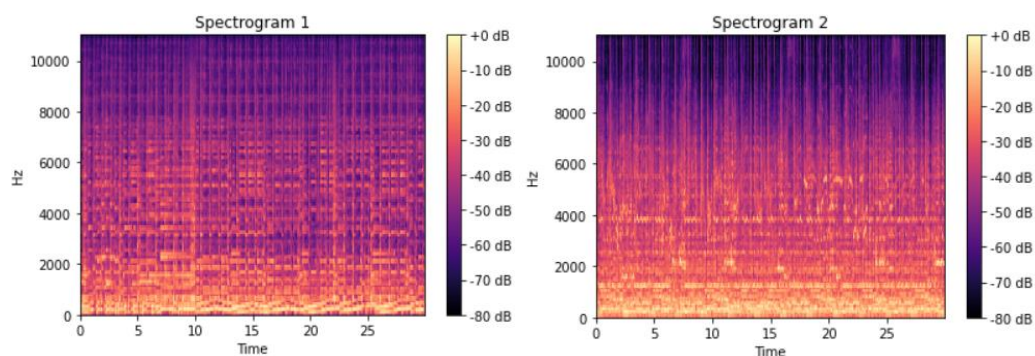
Βήμα 1

Για τα πρώτα βήματα της εργασίας χρησιμοποιούμε το FMA dataset. Αυτό είναι μια βάση δεδομένων από ελεύθερα δείγματα μουσικής.

α) Αρχικά επιλέγουμε δύο τυχαίες γραμμές με διαφορετικά labels από το σύνολο δεδομένων εκπαίδευσης `./input/patreco3-multitask-affective-music/data/fma_genre_spectrogram/train_labels.txt`

β) Στην συνέχεια διαβάζουμε τα αρχεία. Λαμβάνουμε τα φασματογραφήματα σε κλίμακα mel για να μπορέσουμε να παρατηρήσουμε το φάσμα των συχνοτήτων με την πάροδο του χρόνου.

γ) Τυπώνοντας τα φασματογραφήματα, παίρνουμε τα εξής σχήματα:



Παρατηρούμε ότι για διαφορετικά δείγματα τα φασματογραφήματα είναι διαφορετικά. Συγκεκριμένα, κάθε είδος συγκεντρώνει σε διαφορετικές ζώνες συχνοτήτων την ενέργεια του, η οποία απεικονίζεται με χρώμα, για αυτό τα δύο σχήματα έχουν σε διαφορετικά σημεία πιο σκούρο χρώμα και σε άλλα πιο ανοιχτό.

Γενικά, τα φασματογραφήματα μας δείχνουν την φασματική πυκνότητα του ήχου. Έχουμε δύο διαστάσεις (χρόνος και τόνος) και η τρίτη διάσταση η οποία αντιπροσωπεύει το πλάτος (ενέργεια), απεικονίζεται όπως είπαμε με το χρώμα.

Βήμα 2

α) Οι διαστάσεις των δειγμάτων που επιλέξαμε είναι:

```
sample 1 : (128, 1291)
sample 2: (128, 1291)
```

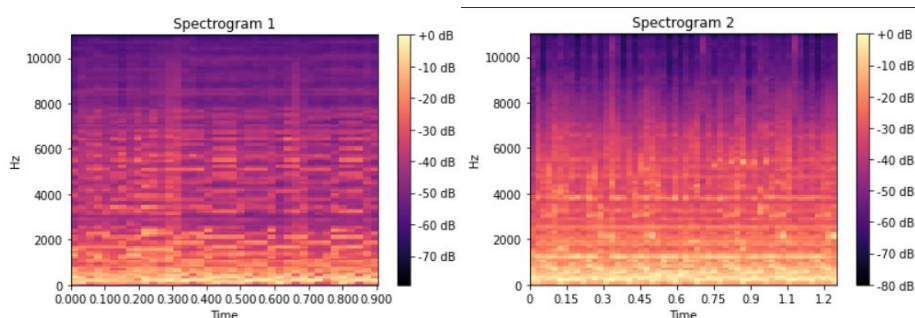
Έχουν 1291 χρονικά βήματα και για τα δύο δείγματα. Τα LSTM βασίζονται σε τένσορες που είναι του ίδιου μήκους. Συνεπώς, αν τα φασματογραφήματα μας δεν είχαν το ίδιο μήκος θα αναγκαζόμασταν να λύσουμε αυτό το πρόβλημα κόβοντας πληροφορία από τα κομμάτια που υπερβαίνουν το ελάχιστο μήκος.

β) Ακολουθώντας την ίδια διαδικασία με το βήμα 1 παρατηρούμε ότι πλέον, για καθένα από τα δύο αρχεία το σχήμα του mel spectrogram είναι:

```
sample 1: (140, 39)
sample 2: (140, 54)
```

Παρατηρούμε ότι υπάρχει μια τεράστια μείωση της διαστατικότητας.

Τα φασματογραφήματα που προκύπτουν είναι τα εξής:

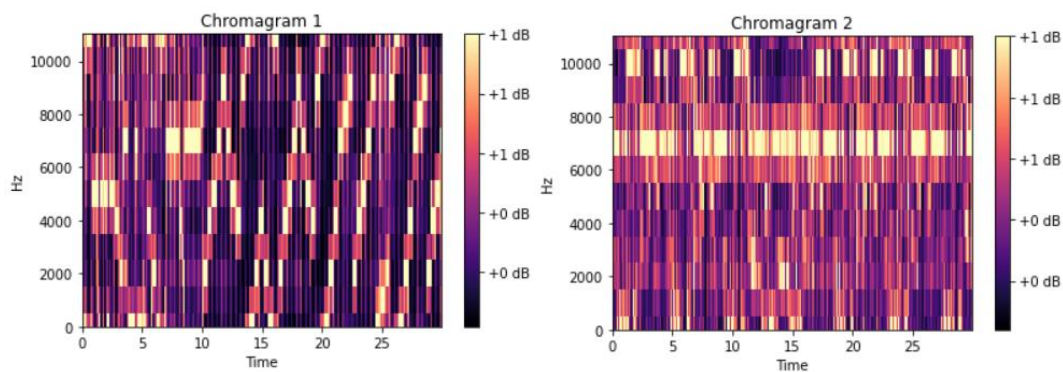


Παρατηρούμε τώρα ότι τα γραφήματα δεν είναι τόσο smooth, δεν έχουν τόσο μεγάλη ακρίβεια καθώς έχουν μειωθεί οι διαστάσεις με αποτέλεσμα να φαίνονται πιο "pixelated". Ωστόσο η πληροφορία δεν χάνεται σημαντικά επομένως μπορούμε να εκπαιδεύσουμε το LSTM πολύ πιο αποδοτικά.

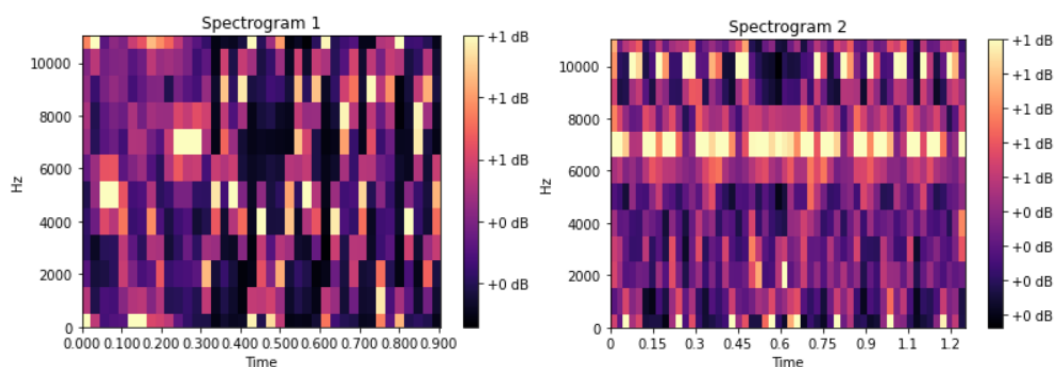
Βήμα 3

Τα χρωματογραφήματα απεικονίζουν την ενέργεια του σήματος μουσικής για τις ζώνες συχνοτήτων που αντιστοιχούν στις νότες μιας μουσικής οκτάβας. Παρουσιάζουν αρκετά καλά τα αρμονικά και μελωδικά χαρακτηριστικά του κομματιού και είναι σημαντικά στις αλλαγές του ηχοχρώματος και του οργάνου.

Τα χρωματογραφήματα που προκύπτουν είναι τα εξής:



Στη συνέχεια μειώνουμε τη διαστατικότητα όπως και προηγουμένως και λαμβάνουμε παρόμοια αποτελέσματα:



Όπως και προηγουμένως η ανάλυση των χρωματογραφημάτων χειροτερεύει με τη χρήση λιγότερων χαρακτηριστικών αλλά διατηρείται η συγκέντρωση της ενέργειας στις περιοχές στοιχείων.

Βήμα 4

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

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

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. Όσον αφορά την μεταβλητή seed (default=None), αυτή εάν λάβει οποιοδήποτε 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.

- **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 strings 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 επιταχύνουμε την εκπαίδευση των μοντέλων μας, αφού πλέον μπορούμε να τα εκπαιδεύουμε σε batches.

- **SpectrogramDataset():** 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() αντίστοιχα. 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:

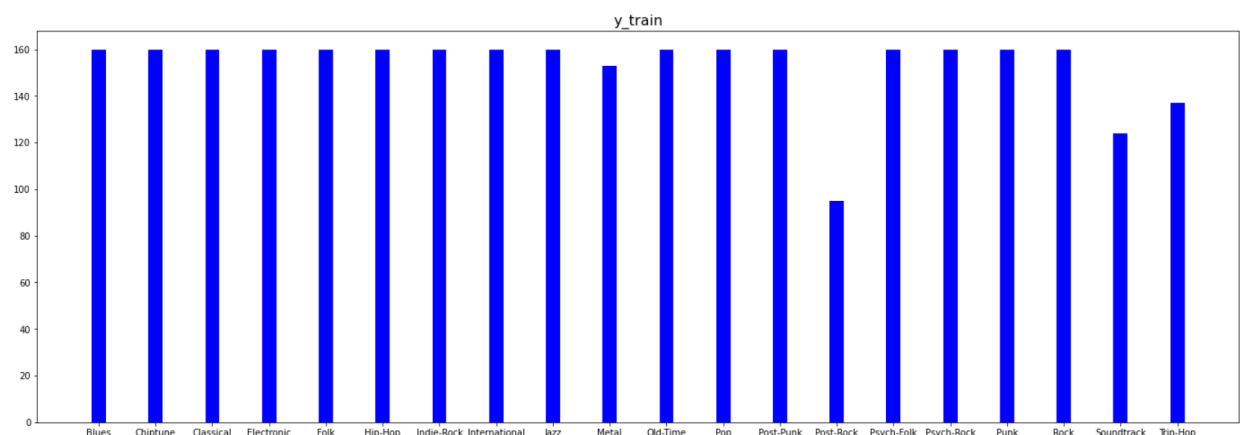
1. Την zero padded ακολουθία των δεδομένων εισόδου η οποία μάλιστα είναι μιας διάστασης
2. Τα labels των δεδομένων εισόδου
3. Το ελάχιστο μήκος ανάμεσα στο μέγεθος της ακολουθίας των δεδομένων εισόδου και του μέγιστου μήκους ακολουθίας που έχουμε επιλέξει. Αυτό είναι απαραίτητο ώστε κατά την εκπαίδευση του LSTM να μην λαμβάνονται υπόψιν στην ανανέωση των βαρών τα βήματα τα οποία περιλαμβάνουν 0 που οποία υπάρχουν λόγω του zero-padding

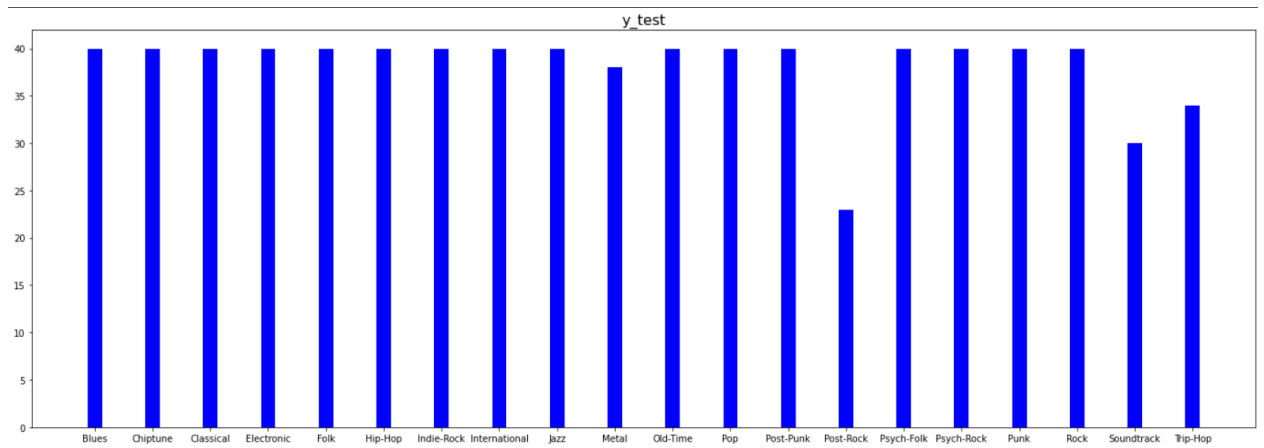
- **len ():** η συνάρτηση αυτή επιστρέφει το μήκος των labels.

Η αναγκαιότητα της συγχώνευσης των κλάσεων και η διαγραφή αυτών που έχουν λίγα στοιχεία, γίνεται για την καλύτερη ταξινόμηση των στοιχείων.

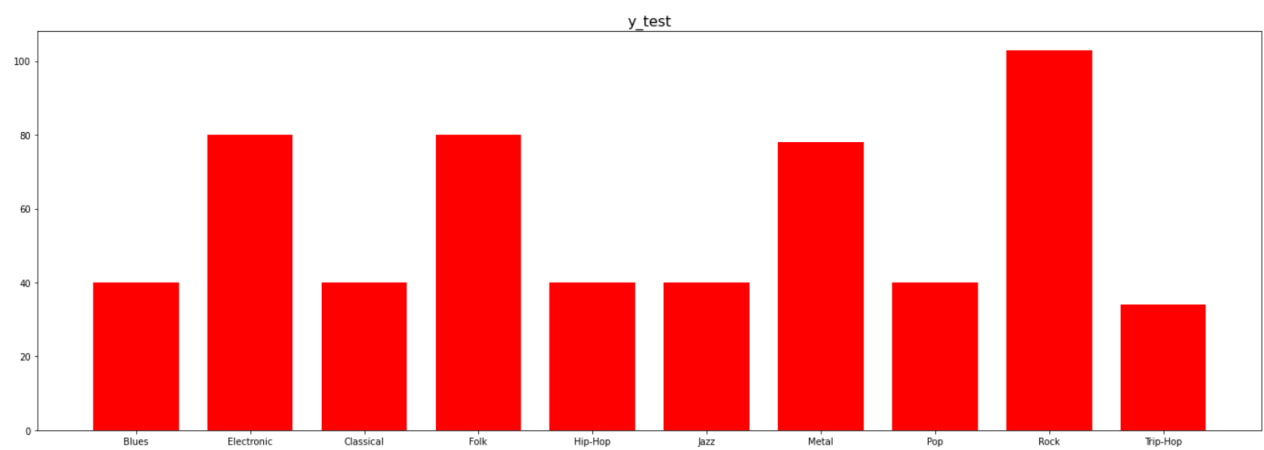
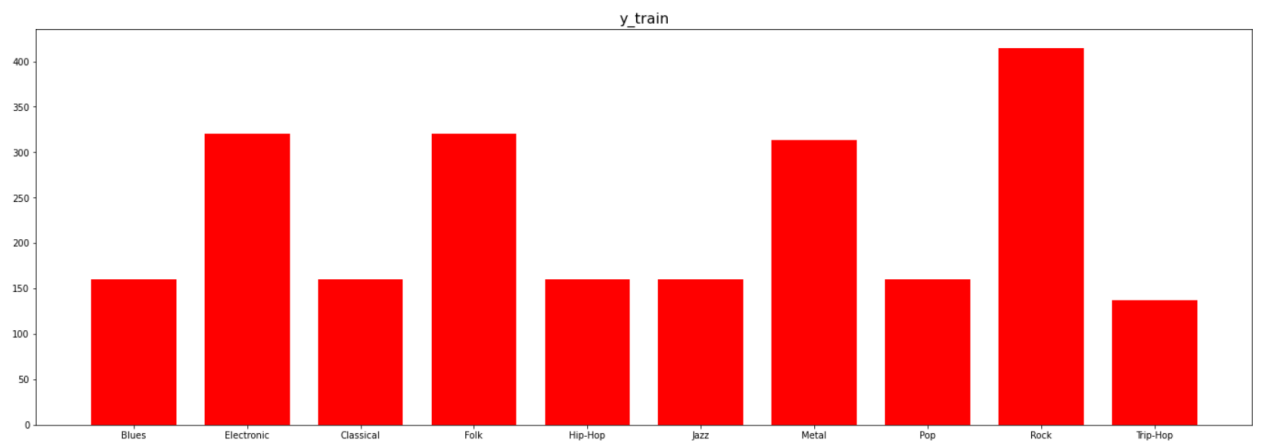
Τα ιστογράμματα που προκύπτουν πριν και μετά τη συγχώνευση παρουσιάζονται παρακάτω:

ΠΡΙΝ





META



Βήμα 5-6

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

```
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
```

Παρακάτω θα παρουσιάσουμε τις γραφικές παραστάσεις που προέκυψαν από την εκπαίδευση του μοντέλου LSTM για 100 εποχές πάνω στα:

1. *Synced mel spectrograms*

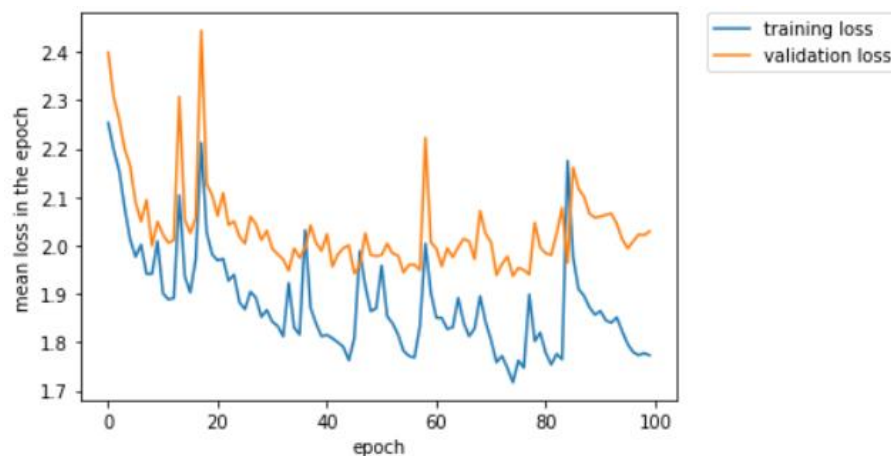
2. *Beat synced mel spectrograms*

3. *Beat synced chromagrams*

4. *Beat-fused synced*

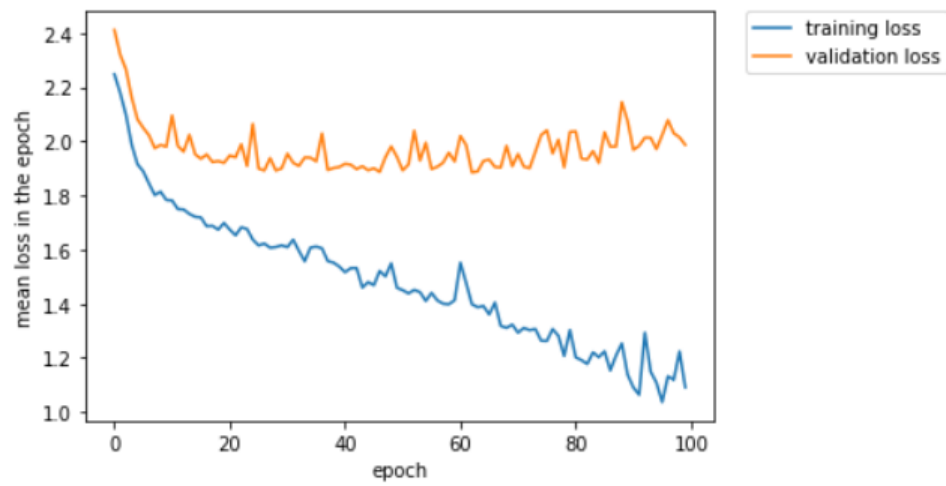
Αλλά επίσης και τις μετρικές που ζητούνται στο βήμα 6.

1.



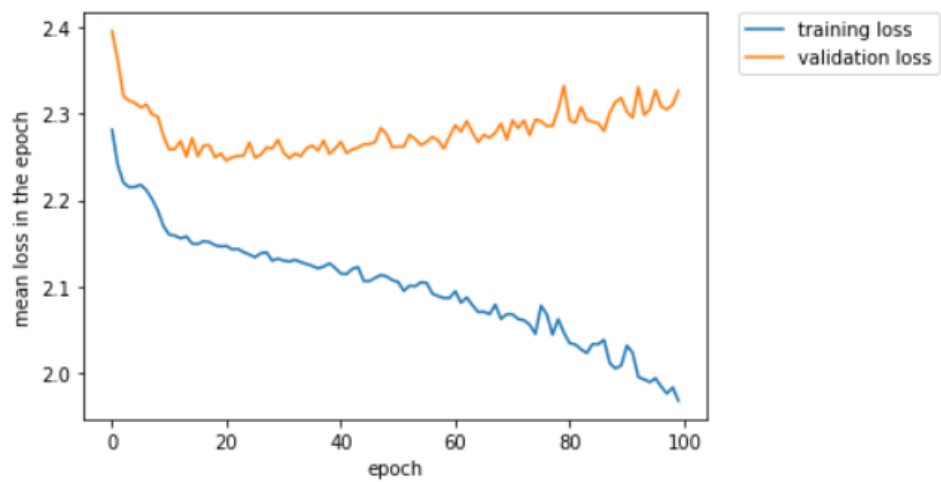
	precision	recall	f1-score	support
0	0.00	0.00	0.00	40
1	0.42	0.47	0.45	40
2	0.34	0.50	0.40	80
3	0.29	0.41	0.34	80
4	0.29	0.17	0.22	40
5	0.00	0.00	0.00	40
6	0.49	0.36	0.41	78
7	0.00	0.00	0.00	40
8	0.29	0.47	0.36	103
9	0.06	0.09	0.07	34
accuracy			0.31	575
macro avg	0.22	0.25	0.23	575
weighted avg	0.26	0.31	0.27	575

2.



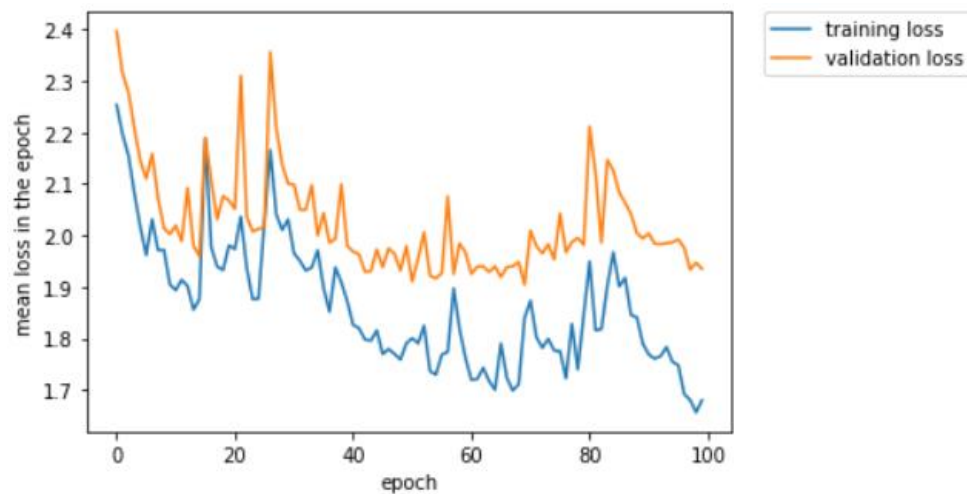
	precision	recall	f1-score	support
0	0.17	0.05	0.08	40
1	0.42	0.70	0.53	40
2	0.38	0.56	0.46	80
3	0.38	0.50	0.43	80
4	0.25	0.12	0.17	40
5	0.18	0.15	0.16	40
6	0.64	0.38	0.48	78
7	0.00	0.00	0.00	40
8	0.36	0.53	0.43	103
9	0.10	0.06	0.07	34
accuracy			0.37	575
macro avg	0.29	0.31	0.28	575
weighted avg	0.33	0.37	0.34	575

3.



	precision	recall	f1-score	support
0	0.00	0.00	0.00	40
1	0.00	0.00	0.00	40
2	0.09	0.01	0.02	80
3	0.20	0.55	0.30	80
4	0.00	0.00	0.00	40
5	0.00	0.00	0.00	40
6	0.29	0.42	0.35	78
7	0.00	0.00	0.00	40
8	0.17	0.40	0.24	103
9	0.00	0.00	0.00	34
accuracy			0.21	575
macro avg	0.08	0.14	0.09	575
weighted avg	0.11	0.21	0.13	575

4.



	precision	recall	f1-score	support
0	0.40	0.05	0.09	40
1	0.39	0.55	0.46	40
2	0.33	0.35	0.34	80
3	0.33	0.47	0.39	80
4	0.31	0.62	0.41	40
5	0.00	0.00	0.00	40
6	0.46	0.47	0.47	78
7	0.00	0.00	0.00	40
8	0.34	0.32	0.33	103
9	0.11	0.18	0.14	34
accuracy			0.33	575
macro avg	0.27	0.30	0.26	575
weighted avg	0.30	0.33	0.30	575

Συμπεράσματα:

Όπως μπορούμε να δούμε, τα chromagram 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.

Στο σημείο αυτό θα εξετάσουμε τις μετρικές που λάβαμε υπόψιν στον classification report.

Accuracy: εκφράζει το συνολικό ποσοστό των δειγμάτων που ταξινομήθηκαν σε σωστή κλάση και είναι πολύ άμεσος και αποτελεσματικός τρόπος να εκτιμήσουμε την επίδοση ενός μοντέλου.

Precision: αναφέρεται σε μια μεμονωμένη κλάση C και ισούται με το λόγο $tp / (tp + fp)$ όπου tp (true positive) είναι ο αριθμός των δειγμάτων που ταξινομήθηκαν ορθώς σε αυτή την κλάση και fp (false positive) ο αριθμός των δειγμάτων που ταξινομήθηκαν λανθασμένα σε αυτή την κλάση. Διαισθητικά αξιολογεί την ικανότητα του μοντέλου να διακρίνει άσχετα δείγματα από τη δεδομένη κλάση.

Recall: αναφέρεται σε μια μεμονωμένη κλάση C και ισούται με το λόγο $tp / (tp + fn)$ όπου fn (false negative) ο αριθμός των δειγμάτων που ανήκουν στην κλάση C αλλά ταξινομήθηκαν σε άλλη. Διαισθητικά, αξιολογεί την ικανότητα του μοντέλου να διακρίνει την δεδομένη κλάση μέσα στα δείγματα.

F1-score: σταθμισμένος αρμονικός μέσος όρος των precision και recall για μια κλάση, δίνοντας έτσι μια γενική αξιολόγηση του μοντέλου για μια συγκεκριμένη κλάση.

Macro: επιστρέφουν το μέσο όρο των αποτελεσμάτων, συνεπώς αντιμετωπίζουν ισότιμα κάθε κλάση και δεν λαμβάνουν υπόψιν το ενδεχόμενο του imbalanced dataset.

Micro: συγκεντρώνουν τα tp, fp και fn για όλες τις κλάσεις και υπολογίζουν ενοποιημένα την μέση μετρική. Έτσι, λαμβάνουν υπόψη το class imbalance, και αυτή είναι και η πιο συχνή αιτία όταν αποκλίνουν τα micro-macro.

Έτσι, μας βολεύει το micro average να λάβουμε υπόψιν, καθώς και το accuracy. Γενικώς, μεγάλη διαφορά ανάμεσα στα accuracy και F1 προκύπτει, όταν το μοντέλο τείνει να επικεντρώνεται σε μια, ή/και δύο κλάσεις αν έχουμε πολλές, εξαιτίας πιθανού imbalanced dataset ή underfitting. Με αυτό τον τρόπο, το accuracy θα επηρεάζεται μόνο από το μέγεθος των samples αυτής της μεγάλης κλάσης και δεν θα λαμβάνει σε μεγάλο βαθμό υπόψιν τις υπόλοιπες.

Τα precision και recall έχουν μεγάλη ισχύ σε περιπτώσεις που τα false positives/negatives έχουν σημαντικές συνέπειες στην μελέτη, πχ. στην διάγνωση καρκίνου.

Στην μελέτη μας, παρατηρούμε πως τα spectrograms ταξινομούνται αποτελεσματικότερα από τα chromograms. Το ίδιο συμβαίνει και με τα fused που ταξινομούνται σχεδόν όπως τα spectrograms. Βλέπουμε πως στα chromograms, το LSTM δεν καταφέρνει να εντοπίσει στο classification αρκετές κλάσεις, εξού και το χαμηλό αποτέλεσμα των μετρικών του.

Βήμα 7

Σε αυτό το βήμα επισκεπτόμαστε τον ιστότοπο [2], όπου και εκπαιδεύουμε ένα CNN δίκτυο στο MNIST dataset. Στον ίδιο χώρο απεικονίζονται οι ενεργοποιήσεις του κάθε επιπέδου.



Στο παράδειγμα για το οποίο παραθέτουμε τις εικόνες βλέπουμε τι επεξεργασία υφίσταται η εικόνα ενός "2" σε επιμέρους επίπεδα. Στο πρώτο επίπεδο, η εικόνα περνά από το συνελικτικό layer κι, εν συνεχεία, από τη συνάρτηση ενεργοποίησης. Πρακτικά, η εικόνα εισόδου συνελίσσεται με τους 8 πυρήνες. Στην έξοδο της relu (activations) παρατηρούμε να γίνονται "έντονα" χαρακτηριστικά όπως ακμές και γωνίες. Με απλά λόγια, το νευρωνικό "προσπαθεί" να "αναγνωρίσει" high level χαρακτηριστικά της εικόνας εισόδου, για αρχή. Τα activations του πρώτου αυτού επιπέδου περνούν από το max pooling layer του πρώτου επιπέδου και έτσι μειώνεται η διαστατικότητά τους. Τα activations του πρώτου επιπέδου αντιμετωπίζονται στο επόμενο στάδιο σαν μια "εικόνα" 8 καναλιών. Η νέα αυτή εικόνα περνά στο επόμενο στάδιο και υφίσταται ίδιας φύσης επεξεργασία με το πρώτο στάδιο. Στην έξοδο του δεύτερου αυτού επιπέδου παρατηρούμε ότι το νευρωνικό χρησιμοποιεί τους 16 πυρήνες του για να αναγνωρίσει χαρακτηριστικά που είναι πιο λεπτομερή (low-level).

Στη συνέχεια, κατασκευάζουμε κι εκπαιδεύουμε στα φασματογραφήματα του FMA genre dataset, ένα custom CNN.

Σε αυτό το σημείο εξηγούμε τη λειτουργία και τον ρόλο των επιμέρους layers κάθε επιπέδου του νευρωνικού μας δικτύου.

- **Convolution:** Η συνέλιξη είναι το βασικό συστατικό ενός CNN. Ένα convolutional layer αποτελείται από έναν σταθερό αριθμό από φίλτρα (πυρήνες) συνέλιξης και κάθε ένα από αυτά τα φίλτρα υλοποιεί την συνέλιξη της εισόδου. Οι έξοδοι (που ονομάζονται και feature maps) των επιμέρους συνελίξεων "συνενώνονται" στην τρίτη διάσταση για να παράγουν μια νέα πολυκάναλη εικόνα. Πρόκειται για μια διαδικασία κατά την οποία ο πυρήνας "ολισθαίνει" εντός της πολυκάναλης εικόνας εισόδου και πραγματοποιείται το άθροισμα των επιμέρους γινομένων μεταξύ των "pixel" του φίλτρου και της εικόνας. Η έξοδος της συνέλιξης έχει μικρότερες διαστάσεις από αυτές της εισόδου. Γι' αυτό, αν επιθυμούμε να διατηρούνται, "συμπληρώνουμε" με μηδενικά, κάνοντας το γνωστό zero-padding πριν την εφαρμογή της συνέλιξης.

- **Batch Normalization:** Πρακτικά, το BatchNorm κανονικοποιεί τα δεδομένα (ένα batch), ώστε να έχουν zero mean και unit variance. Κατ' αυτόν τον τρόπο, μειώνονται οι εποχές που χρειάζεται το δίκτυο για να εκπαιδευτεί και αυξάνεται η ικανότητά του στο να γενικεύει.

- **ReLU Activation:** Μια από τις συνηθέστερες συναρτήσεις ενεργοποίησης που χρησιμοποιούνται στα νευρωνικά δίκτυα, για πολλούς λόγους. Πρακτικά, πραγματοποιεί element-wise την πράξη $\max(0, x)$. Η χρήση της ReLU βοηθά στην αντιμετώπιση της εκθετικής αύξησης στους υπολογισμούς που χρειάζονται για να λειτουργήσει το νευρωνικό.

- **Max Pooling:** Το max pooling χρησιμοποιείται για να μειώσει τις διαστάσεις των feature maps που συναποτελούν κάποια έξοδο στρώματος. Πρόκειται για ένα φίλτρο που ολισθαίνει σε κάθε ένα feature map ξεχωριστά και παράγει ένα νέο feature map του οποίου κάθε pixel είναι το μέγιστο στοιχείο του αντίστοιχου παραθύρου του αρχικού feature map.

Μετά την εκπαίδευση του CNN για 50 εποχές λαμβάνουμε το εξής classification report:

	precision	recall	f1-score	support
0	0.15	0.05	0.08	40
1	0.47	0.42	0.45	40
2	0.56	0.29	0.38	80
3	0.32	0.42	0.37	80
4	0.39	0.28	0.32	40
5	0.10	0.03	0.04	40
6	0.55	0.41	0.47	78
7	0.07	0.05	0.06	40
8	0.21	0.48	0.29	103
9	0.29	0.21	0.24	34
accuracy			0.31	575
macro avg	0.31	0.26	0.27	575
weighted avg	0.34	0.31	0.30	575

Βήμα 8

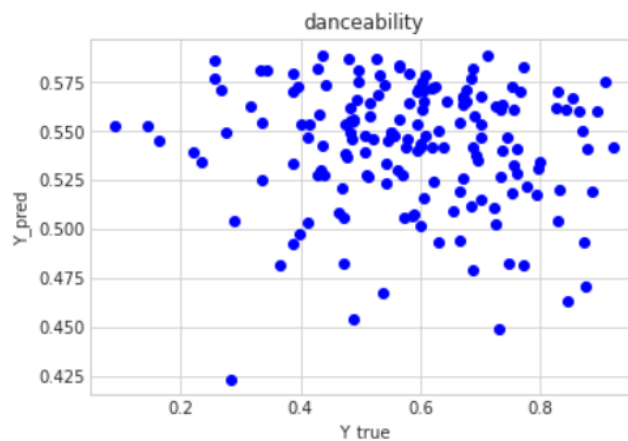
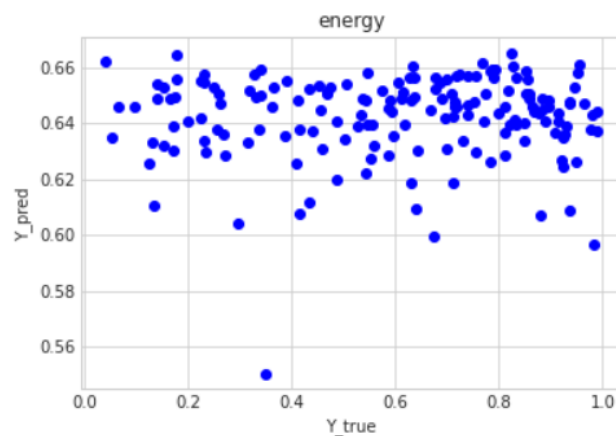
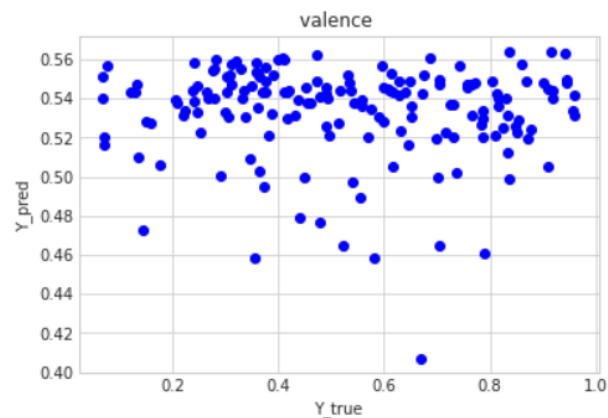
Εκπαιδεύουμε το CNN του βήματος 7 και το LSTM του βήματος 5 στο νέο μας dataset. Εφόσον δεν διαθέτουμε επισημειώσεις για το test set, αξιολογούμε τα μοντέλα μας χρησιμοποιώντας ένα υποσύνολο του train set. Η τελική μετρική απόδοσης για τα μοντέλα μας είναι το μέσο Spearman correlation ρ ανάμεσα

στις πραγματικές (ground truth) τιμές και στις προβλεπόμενες τιμές για όλους τους άξονες.

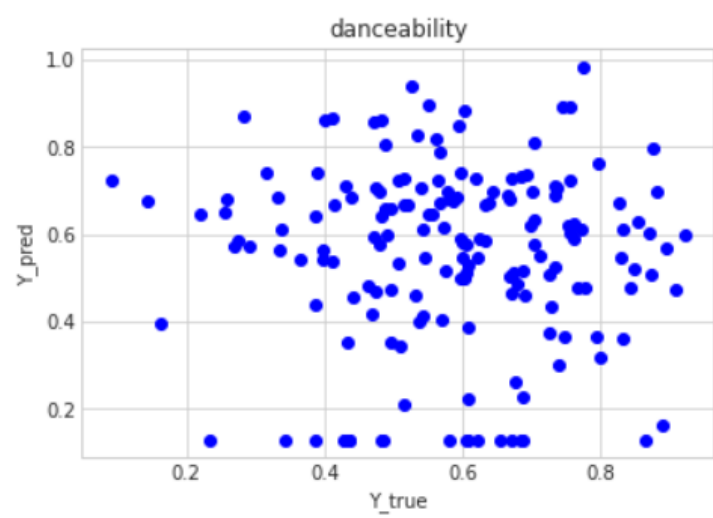
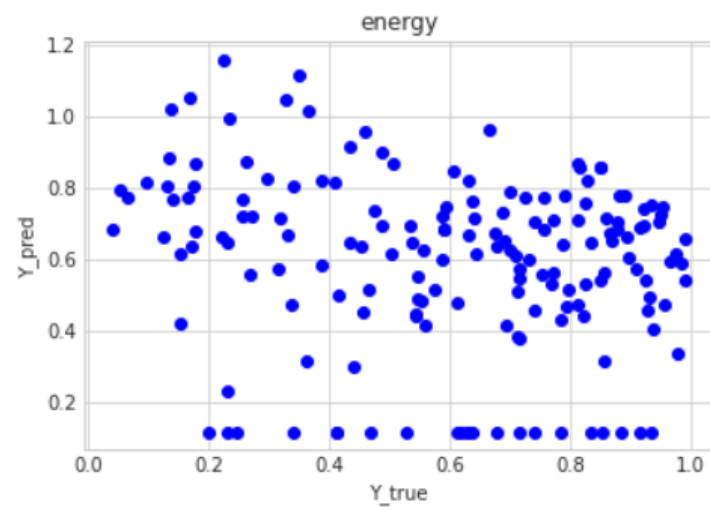
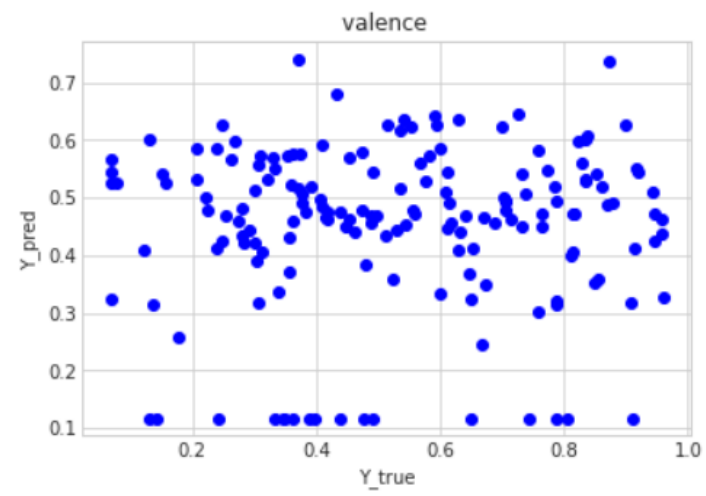
Η Spearman Correlation εκφράζει το κατά πόσο η σχέση δύο μεταβλητών μπορεί να προσεγγιστεί με μια μονοτονική συνάρτηση και λαμβάνει τιμές στο διάστημα $[-1,1]$.

Παραθέτουμε και για λόγους οπτικοποίησης τις τιμές που "κρίνει" το κάθε μοντέλο πως αντιστοιχούν σε κάθε sample εισόδου.

LSTM



CNN



Από τα scatter plots μπορούμε να παρατηρήσουμε ότι το CNN έχει καλύτερη ικανότητα να γενικεύει. Τα δείγματα στα plots τοποθετούνται με τρόπο τέτοιο που φαίνεται να υπάρχει μια γραμμική σχέση ανάμεσα στις προβλεπόμενες τιμές του νευρωνικού και τις πραγματικές. Αυτή η γραμμικότητα είναι εντονότερη στο scatter plot που αφορά το energy, το οποίο είναι και το συναίσθημα στο οποίο το CNN πετυχαίνει το μέγιστο Spearman correlation.