ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧ/ΚΩΝ & ΜΗΧ/ΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΗΜΑΤΩΝ, ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ

Μάθημα: "Αναγνώριση Προτύπων"

9ο εξάμηνο, Ακαδημαϊκό Έτος 2020-21

3η ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ

Αναγνώριση Είδους και Εξαγωγή Συναισθήματος από Μουσική

ΓΕΩΡΓΙΟΣ ΜΑΓΚΑΦΩΣΗΣ, 03116125

ΗΛΙΑΣ ΤΡΙΑΝΤΑΦΥΛΛΟΠΟΥΛΟΣ, 03116028

ΒΗΜΑ 0

Στο τρέχον βήμα εξοικειωθήκαμε με τα kaggle kernels και φορτώσαμε το dataset με το οποίο θα δουλέψουμε. Συγκεκριμένα, για την υλοποίηση της εργαστηριακής άσκησης θα χρησιμοποιήσουμε δύο 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.

Ακολουθούμε τις οδηγίες του notebook και φορτώνουμε τα δεδομένα μας.

Να σημειώσουμε επίσης ότι εν συνεχεία λόγω κάποιων προβλημάτων με το kaggle, μεταφερθήκαμε στο google colab, όπου εκεί εκτελέσαμε τα βήματα της εργασίας.

ΒΗΜΑ 1

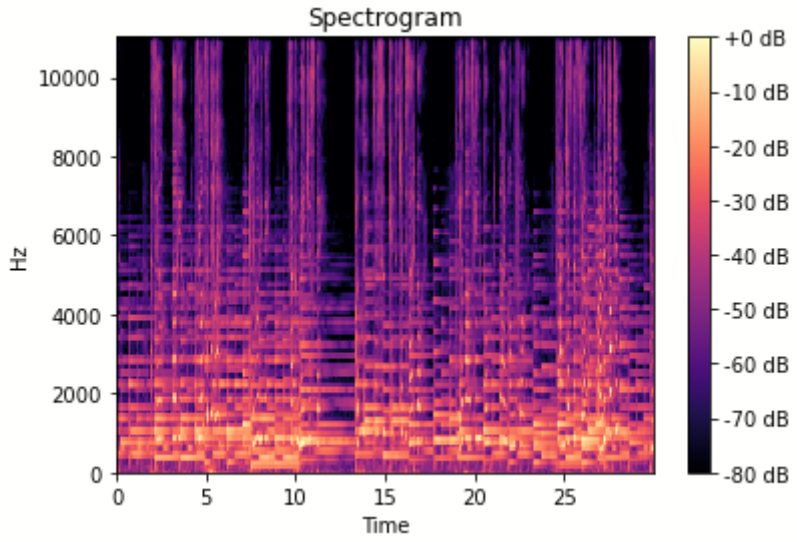
a) Διαλέγουμε δύο τυχαίες γραμμές από τον φάκελο fma_genre_spectrograms/train με διαφορετικά μεταξύ τους labels. Συγκεκριμένα, επιλέγουμε :

- `123947.fused.full.npy.gz` με ετικέτα `Blues`
- `14780.fused.full.npy.gz` με ετικέτα `Electronic`

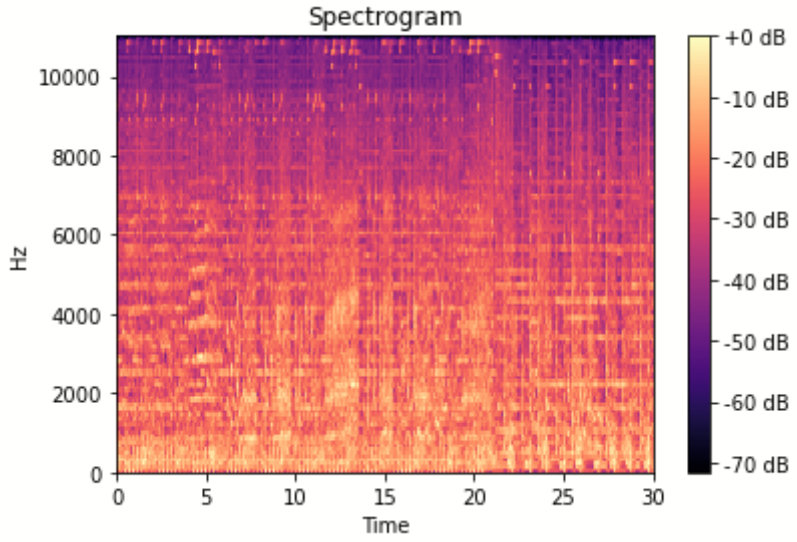
Σκοπός είναι να προβούμε σε απαραίτητες συγκρίσεις μεταξύ δύο διαφορετικών ειδών μουσικής.

b) Ακολουθώντας τις οδηγίες του notebook που μας δίνεται, παίρνουμε τα φασματογραφήματα σε κλίμακα mel.

c) Το φασματογράφημα του δείγματος της κατηγορίας Blues είναι το εξής :



και της κατηγορίας Electronic :



Τα spectrograms αποτελούν μια οπτική αναπαράσταση του spectrum των συχνοτήτων ενός σήματος, καθώς αυτό μεταβάλλεται με τον χρόνο. Το χρώμα που παρατηρείται ακολουθεί τις τιμές του πλαινού πίνακα και αναπαριστά το amplitude μιας συγκεκριμένης συχνότητας σε μία συγκεκριμένη χρονική στιγμή. Στις συγκεκριμένες αναπαραστάσεις, παρατηρούμε ότι το πάνω σπεκτρόγραμμα που αναπαριστά ένα κομμάτι blues, έχει πολλές μεταβολές από τις υψηλές συχνότητες στις χαμηλές, αλλά με μια τάση στις χαμηλές συχνότητες. Αντίθετα, στην περίπτωση της ηλεκτρονικής μουσικής, έχουμε χρήση και υψηλότερων συχνοτήτων, καθώς επίσης παρατηρείται και μία συνέχεια.

ΒΗΜΑ 2

a) Τυπώνουμε τις διαστάσεις των φασματογραφημάτων :

```
(128, 1291)
(128, 1293)
```

(το πρώτο αντιστοιχεί στο Blues και το δεύτερο στο Electronic).

Τα χρονικά βήματα ποικίλλουν ανά δείγμα. Στις εξεταζόμενες περιπτώσεις έχουμε 1291 χρονικά βήματα στο Blues και 1293 στο Electronic. Τα spectrograms έχουν επίσης έναν σταθερό αριθμό χαρακτηριστικών ίσο με 128.

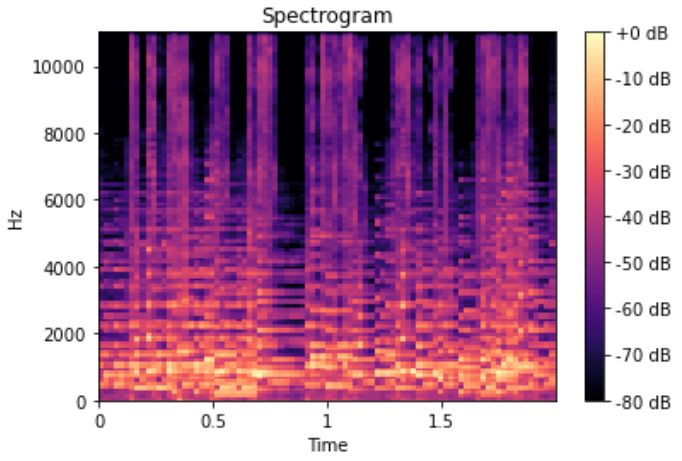
Ο αριθμός των χρονικών βημάτων είναι αρκετά μεγάλος και δεν είναι ικανός για μια αποδοτική εκπαίδευση ενός LSTM, καθώς προσφέρει μεγάλη υπολογιστική επιβάρυνση στο νευρωνικό, αυξάνοντας σημαντικά την χρονική πολυπλοκότητα. Επιπλέον, όπως γνωρίζουμε, το LSTM μπορεί να αντιμετωπίζει το πρόβλημα του vanishing gradient που παρατηρείται στα RNN, αλλά για μεγάλες ακολουθίες, αυξάνεται η μνήμη του μοντέλου και υπάρχει ακόμα η πιθανότητα να “ξεχαστεί” πληροφορία. Έτσι, βλέπουμε αμέσως την ανάγκη μείωσης των χρονικών βημάτων, αποσπώντας χρήσιμη πληροφορία ακόμα και με λιγότερες διαστάσεις.

b) Ένας αποδοτικός, λοιπόν, τρόπος μείωσης των χρονικών βημάτων μας είναι να συγχρονίσουμε τα φασματογραφήματα πάνω στον ρυθμό, παίρνοντας τη διάμεσο ανάμεσα στα σημεία που χτυπάει το beat της μουσικής. Τα αρχεία που περιέχουν τέτοια πληροφορία μας δίνονται έτοιμα στον φάκελο fma_genre_spectrograms_beat και ακολουθούμε παρόμοια διαδικασία με το βήμα 1 για να καταλήξουμε σε συμπεράσματα.

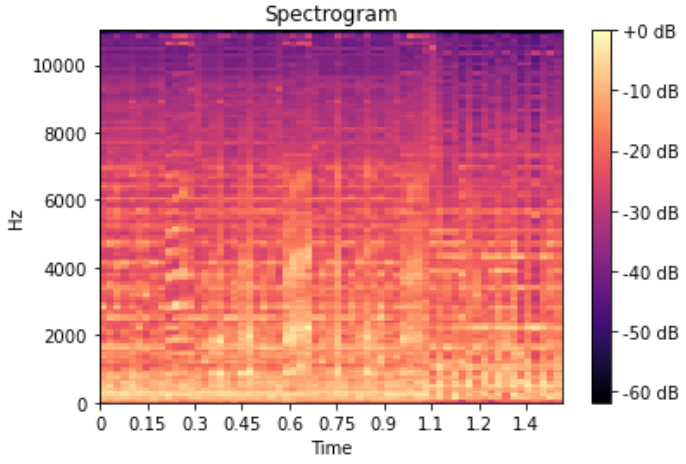
```
(128, 86)
(128, 63)
```

Παρατηρούμε ότι τα χρονικά βήματα μειώθηκαν σε 86 και 63 για το Blues και την Electronic αντίστοιχα. Τυπώνουμε πάλι τα spectrograms :

Blues



Electronic



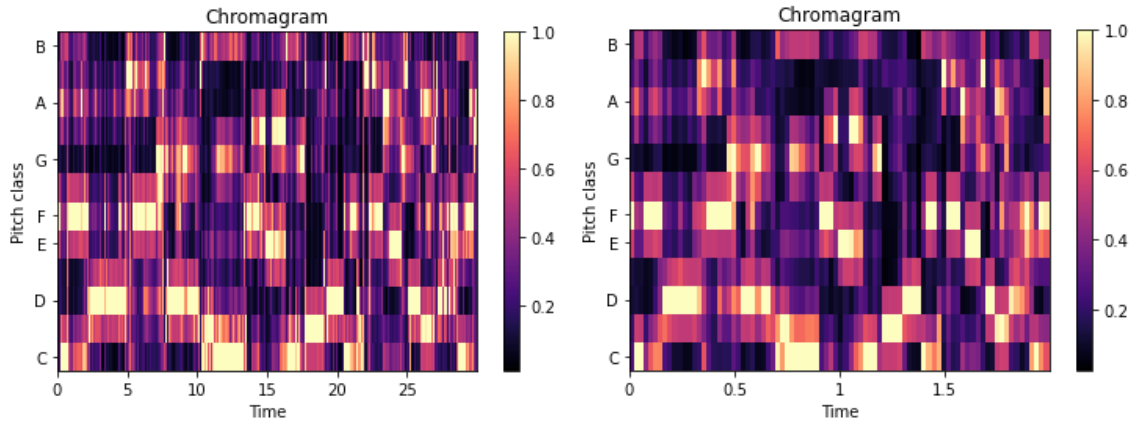
Αρχικά, μπορούμε να επαληθεύσουμε την μείωση των χρονικών βημάτων, απλά παρατηρώντας τον άξονα του χρόνου. Επίσης, και τα δύο spectrograms μοιάζουν αρκετά με εκείνα που τυπώσαμε στο προηγούμενο βήμα, διατηρώντας τις σημαντικές πληροφορίες, οι οποίες είναι ικανές να διαφοροποιήσουν τις διάφορες κλάσεις αναμεταξύ τους. Οι εικόνες φαίνεται να είναι ελάχιστα πιο θολές απ' ότι προηγουμένως, πράγμα λογικό αφού έχει συμπυκνωθεί ο χρόνος.

ΒΗΜΑ 3

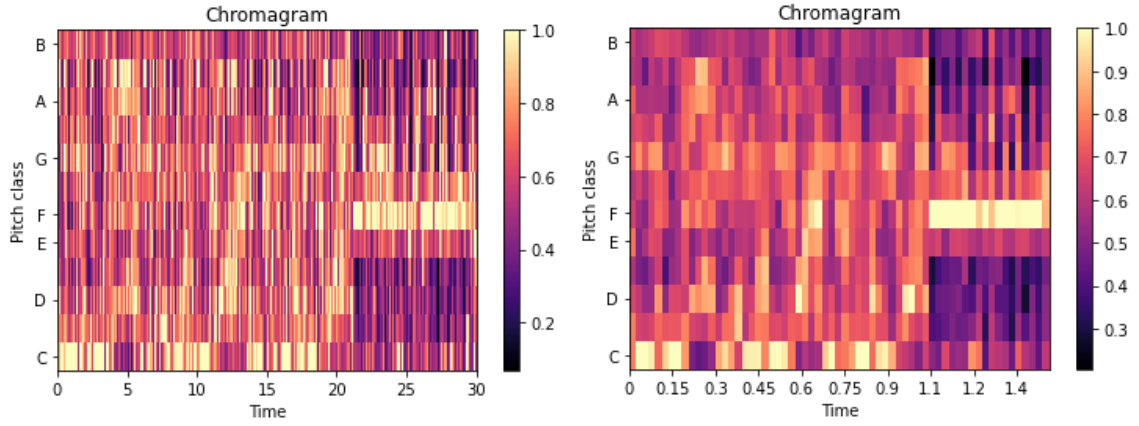
Σε αυτό το βήμα θα ασχοληθούμε με τα χρωμογραφήματα και θα προχωρήσουμε σε παρόμοιες αναλύσεις με αυτήν που ακολουθήσαμε στα spectrograms. Τα χρωμογραφήματα (chromagrams) σχετίζονται με δώδεκα διαφορετικές νότες (ημιτόνια) C, C\, D, D\, E, F, F\, G, G\, A, A\, B και μπορούν να χρησιμοποιηθούν ως εργαλείο 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.

Για τα ίδια δείγματα που επιλέξαμε παραπάνω, τυπώνουμε τα χρωμογραφήματα τόσο για τις αρχικές χρονικές ακολουθίες (αριστερά) όσο και για τις beat-synchronised εκδοχές (δεξιά) :

Blues



Electronic



Ουσιαστικά αυτό που μας δείχνουν τα χρωμογραφήματα μέσω της κλίμακας του χρώματος που φαίνεται και στα δεξιά του κάθε διαγράμματος είναι το πόση ενέργεια έχουμε στην κάθε πιθανή νότα. Συνεπώς, βλέπουμε και εδώ ότι μπορούμε να δούμε σημαντικές διαφορές μεταξύ των δύο διαφορετικών ειδών μουσικής σχετικά με το σε ποιες νότες σημειώνεται περισσότερη ενέργεια κάθε φορά, αφού τα δύο διαγράμματα των δύο κλάσεων διαφέρουν σημαντικά μεταξύ τους. Παρατηρούμε και εδώ ότι τα beat synchronized αρχεία μας δίνουν παρόμοια εικόνα με τα πρωτότυπα, παρά την μείωση των χρονικών βημάτων. Βλέπουμε ότι απλώς γίνεται πιο "θολή" η εικόνα, διατηρώντας όμως την πληροφορία που μας παρέχεται από τους χρωματισμούς στις αντίστοιχες νότες.

ΒΗΜΑ 4

a) Για τα επόμενα βήματα, μας παρέχεται έτοιμος κώδικας για την υλοποίηση ενός PyTorch Dataset η οποία διαβάζει τα δεδομένα και μας επιστρέφει τα δείγματα.

Αρχικά, έχουμε τις τρεις παρακάτω συναρτήσεις :

```
# Helper functions to read fused, mel, and chromagram
def read_fused_spectrogram(spectrogram_file):
    spectrogram = np.load(spectrogram_file)
    return spectrogram.T


def read_mel_spectrogram(spectrogram_file):
    spectrogram = np.load(spectrogram_file)[:128]
    return spectrogram.T


def read_chromagram(spectrogram_file):
    spectrogram = np.load(spectrogram_file)[128:]
    return spectrogram.T
```

όπου χρησιμοποιούνται για να διαβάσουμε, όλα τα δεδομένα κάνοντας concatenate στα spectrograms και chromograms, τα πρώτα 128 στοιχεία που αντιστοιχούν στα spectrograms και τα τελευταία 12 στοιχεία που αντιστοιχούν στα χρωμογράμματα αντίστοιχα κατά την σειρά εμφάνισης στο screenshot.

Στη συνέχεια, χρησιμοποιούμε την συνάρτηση torch_train_val_split για να κάνουμε διαχωρισμό των δεδομένων σε train και validation. Όπως φαίνεται και στον κώδικα, έχουμε μια συγκεκριμένη παράμετρο seed κατά την αρχικοποίηση των δεικτών σε τυχαίες τιμές. Αυτή ορίζεται ως none όταν τρέχουμε το μοντέλο, ενώ μπορούμε να την ορίσουμε με συγκεκριμένη τιμή όταν κάνουμε debugging. Αυτό συμβαίνει, καθώς, αν ορίσουμε το seed, θα προσθέσουμε ένα είδος bias στο μοντέλο μας, αφού θα το εκπαιδεύουμε συνεχώς με κάποιον συγκεκριμένο τρόπο και επομένως δεν θα είναι καλά εκπαιδευμένο για τα φαινόμενα τυχαιότητας που θέλουμε. Αν λοιπόν, υλοποιήσουμε όλη την διαδικασία μας με βάση έναν συγκεκριμένο διαχωρισμό των δεδομένων, τότε είναι πολύ πιθανό να έχουμε μεγάλα σφάλματα σε διαφορετικές περιπτώσεις (overfitting).

Έτσι, για τον διαχωρισμό των δεδομένων, παίρνουμε τους δείκτες, τους ανακατεύουμε τυχαία και επιλέγουμε όσους θέλουμε για training data και τους υπόλοιπους για validation. Έπειτα, με βάση αυτούς τους δείκτες, κρατάμε τα κατάλληλα δεδομένα και σχηματίζουμε τους loaders μέσω της Dataloader.

Ακολουθούν δύο ακόμα συναρτήσεις που θα μας βοηθήσουν στην κλάση δημιουργίας του dataset μας :

LabelTransformer : μετατρέπει τα δεδομένα επισημειώσεων σε κατάλληλη μορφή για τον Dataloader μας.

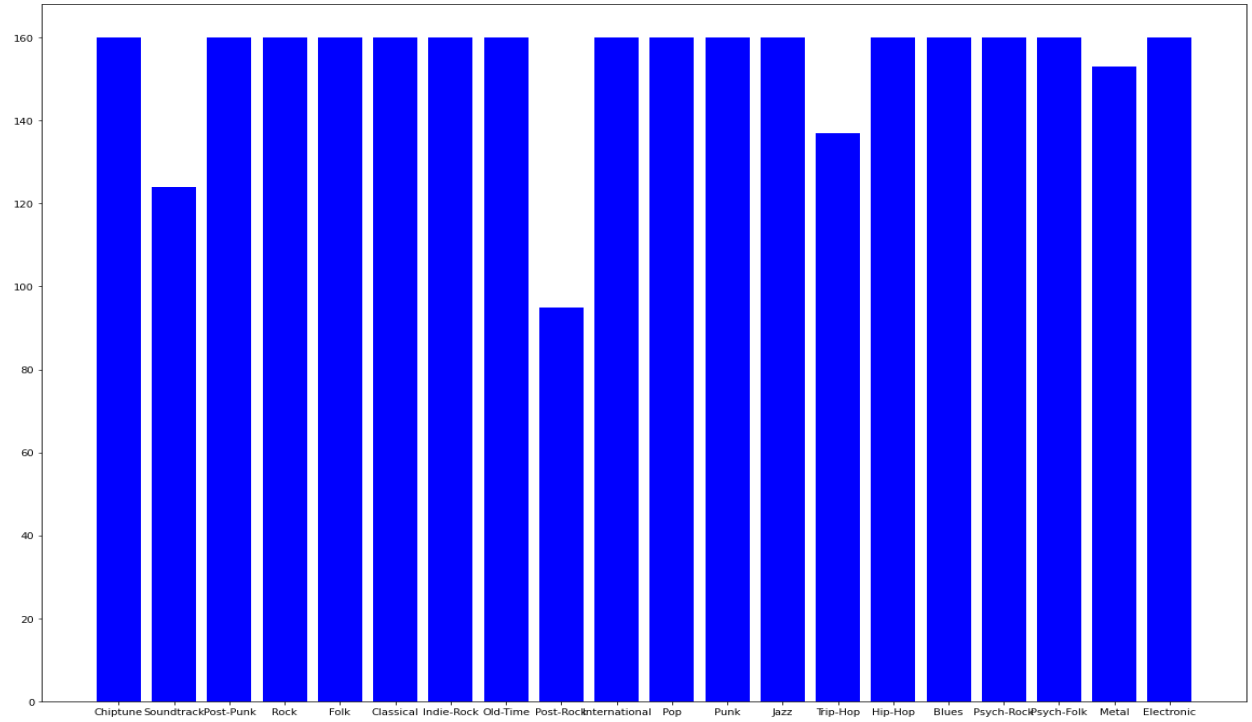
PaddingTransform : όπως είδαμε και στην προηγούμενη εργαστηριακή άσκηση, έτσι και εδώ έχουμε στην διάθεση μας ακολουθίες δειγμάτων διαφορετικού μήκους. Για να εκπαίδευσουμε το LSTM, θα πρέπει όλα τα δείγματα μας να έχουν ίσο μήκος. Έτσι, με την βοήθεια αυτής της συνάρτησης δημιουργούμε padding. Δηλαδή, βρίσκουμε (ή επιλέγουμε κάποιο αυθαίρετα) το μέγιστο μήκος των ακολουθιών και σε όλες τις υπόλοιπες προσθέτουμε μηδενικά ώστε να τις φέρουμε στις ίδιες διαστάσεις.

Μέσω αυτών των συναρτήσεων, προχωράμε στην δημιουργία του SpectrogramDataset. Στην init αρχικοποιούμε τις ποσότητες που θα χρειαστούμε. Κάνουμε χρήση της συνάρτησης get_files_labels για να εφαρμόσουμε την σωστή επεξεργασία, ώστε να εξάγουμε δύο λίστες που θα περιέχουν τα files και το αντίστοιχο label τους. Όπως φαίνεται κάνουμε χρήση και του LabelTransformer και του Padding Transform για την εξαγωγή έγκυρων αποτελεσμάτων.

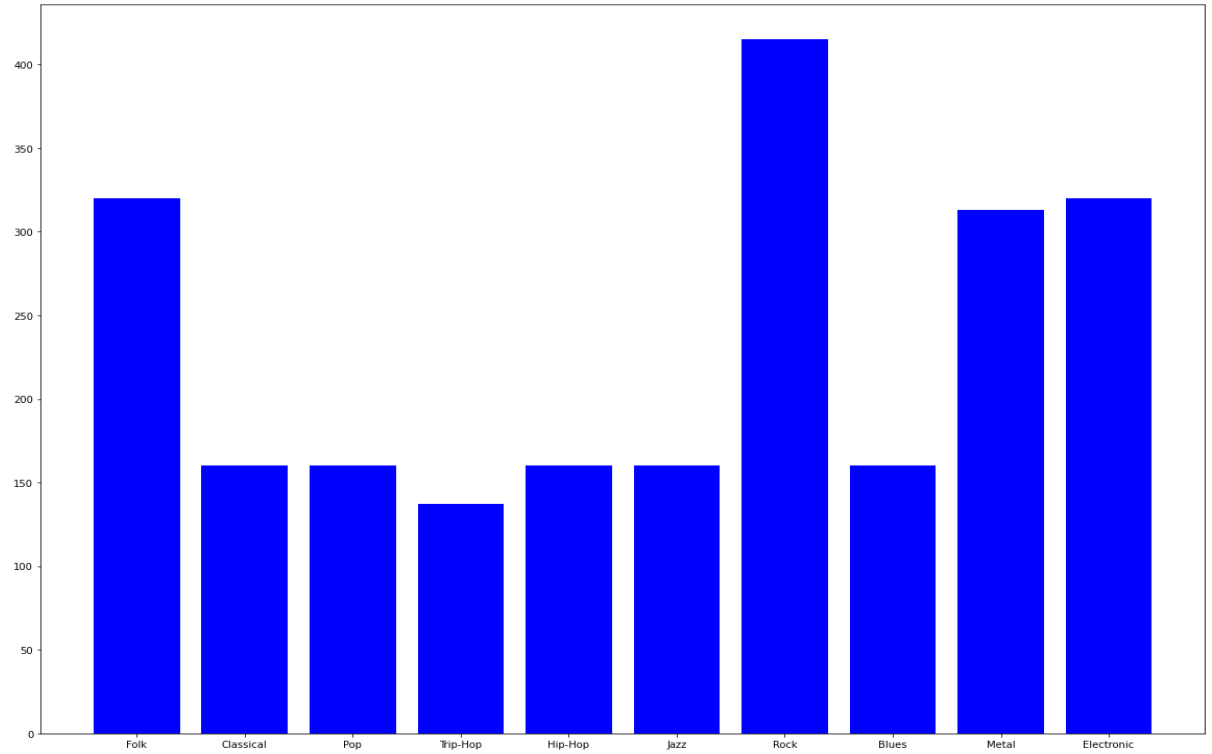
Διαθέτει επίσης, μία συνάρτηση __len__() που μας επιστρέφει το μέγεθος του dataset μας μέσω του μεγέθους του πίνακα των labels (που ισούται με το πλήθος των δειγμάτων μας). Επίσης, έχει την συνάρτηση __getitem__(item), η οποία είναι η κύρια συνάρτηση που θα μας επιστρέψει τα αντικείμενα που θέλουμε κατά την επαναληπτική διαδικασία που θα υποβληθεί ο loader μας. Επιστρέφει 3 πράγματα : τα δείγματα μας, έχοντας περάσει από padding, τις επισημειώσεις αυτών και τέλος μια λίστα με την πραγματική τους διάσταση χωρίς το padding. Όπως είδαμε και στην δεύτερη εργαστηριακή άσκηση και τα τρία αυτά αποτελούν απαραίτητα στοιχεία για το lstm μας.

b) Στον κώδικα που μας δίνεται συγχωνεύονται κλάσεις που μοιάζουν μεταξύ τους και αφαιρούνται κλάσεις που αντιπροσωπεύονται από πολύ λίγα δείγματα. Αυτό γίνεται μέσω της class_mapping. Σε περιπτώσεις που έχουμε λίγα δείγματα σε μία κλάση, θα δημιουργηθεί πρόβλημα κατά την διαδικασία της εκπαίδευσης, αφού με λίγα δείγματα, δεν θα γίνει σωστή ανίχνευση αυτών των κλάσεων και επομένως μπορεί να μην έχουμε σωστή γενίκευση του μοντέλου μας. Επίσης, είναι πολύ πιθανό κατά το σπάσιμο σε sets να μην ανατεθούν αρκετά στο training ώστε να έχουμε έναν καλό διαμοιρασμό. Επίσης σε κλάσεις που μοιάζουν αρκετά μεταξύ τους (όπως πχ Rock με Psych-Rock) θα έχουμε αρκετά παρόμοια δείγματα και έτσι η συγχώνευση αυτών μπορεί να είναι μια αναπόφευκτη διαδικασία.

c) Πριν από την εκτέλεση του βήματος 4b, έχουμε το εξής ιστόγραμμα :



και μετά την εκτέλεση του :



ΒΗΜΑ 5

Για την εκπαίδευση όλων των ζητούμενων LSTM, χρησιμοποιήσαμε τον κώδικα για τα LSTM της δεύτερης εργαστηριακής άσκησης, προσαρμόζοντας κατάλληλα τις αλλαγές που έπρεπε να κάνουμε εξαιτίας του dataset μας στις συναρτήσεις train_lstm και eval_lstm. Για την εκπαίδευση αυτών, φτιάξαμε ένα μοντέλο με rnn size = 128, 2 layers, regularization = 0.001, χωρίς Dropout layer, με loss function την crossentropy και optimizer τον Adam με learning rate = 0.0001 και το εκπαιδεύσαμε σε 15 εποχές.

a) Χρησιμοποιώντας τα φαματογραφήματα, προκύπτει κατά την εκπαίδευση :

```
epoch :  0 ,training loss =  0.08925924146053743
Accuracy in validation set =  17.500000000000004 %
epoch :  0 ,validation loss =  0.08885798044502735
epoch :  1 ,training loss =  0.08854318957547752
Accuracy in validation set =  17.500000000000004 %
epoch :  1 ,validation loss =  0.08838968630880117
epoch :  2 ,training loss =  0.0880831169838808
Accuracy in validation set =  17.63157894736842 %
epoch :  2 ,validation loss =  0.08806400063137214
epoch :  3 ,training loss =  0.08791932478851201
Accuracy in validation set =  18.026315789473692 %
epoch :  3 ,validation loss =  0.08775534232457478
epoch :  4 ,training loss =  0.08759824977237352
Accuracy in validation set =  21.184210526315784 %
epoch :  4 ,validation loss =  0.08759241458028555
epoch :  5 ,training loss =  0.08745973025049482
Accuracy in validation set =  21.184210526315784 %
epoch :  5 ,validation loss =  0.08718060608953238
epoch :  6 ,training loss =  0.08696755432352728
Accuracy in validation set =  22.631578947368425 %
epoch :  6 ,validation loss =  0.08699510960529248
epoch :  7 ,training loss =  0.08702813681899285
Accuracy in validation set =  19.86842105263158 %
epoch :  7 ,validation loss =  0.0869047399610281
epoch :  8 ,training loss =  0.08664453774690628
Accuracy in validation set =  20.65789473684211 %
epoch :  8 ,validation loss =  0.08692931849509478
epoch :  9 ,training loss =  0.08656675733473836
Accuracy in validation set =  22.368421052631582 %
epoch :  9 ,validation loss =  0.08619199817379315
epoch :  10 ,training loss =  0.08599771863343764
Accuracy in validation set =  23.15789473684211 %
epoch :  10 ,validation loss =  0.08578669745475054
epoch :  11 ,training loss =  0.08579019396280756
Accuracy in validation set =  23.42105263157894 %
epoch :  11 ,validation loss =  0.08554528125872214
epoch :  12 ,training loss =  0.08527983861918352
Accuracy in validation set =  24.868421052631582 %
epoch :  12 ,validation loss =  0.08520093653351068
epoch :  13 ,training loss =  0.08505875297955104
Accuracy in validation set =  20.921052631578952 %
epoch :  13 ,validation loss =  0.08788429976751407
epoch :  14 ,training loss =  0.08534238441866271
Accuracy in validation set =  25.657894736842103 %
epoch :  14 ,validation loss =  0.08538623744000991
```

b) Χρησιμοποιώντας τα beat-synced spectrograms :

```
epoch :  0 ,training loss =  0.08936779657188727
Accuracy in validation set =  20.13157894736842 %
epoch :  0 ,validation loss =  0.08827554589758317
epoch :  1 ,training loss =  0.08812395969823915
Accuracy in validation set =  20.921052631578952 %
epoch :  1 ,validation loss =  0.08690169484664996
epoch :  2 ,training loss =  0.08695059832261533
Accuracy in validation set =  26.315789473684205 %
epoch :  2 ,validation loss =  0.08577114498863618
epoch :  3 ,training loss =  0.08585006257100981
Accuracy in validation set =  25.789473684210527 %
epoch :  3 ,validation loss =  0.08454171754419804
epoch :  4 ,training loss =  0.08400106004306249
Accuracy in validation set =  28.421052631578945 %
epoch :  4 ,validation loss =  0.0831855967019995
epoch :  5 ,training loss =  0.08308628019021481
Accuracy in validation set =  29.73684210526315 %
epoch :  5 ,validation loss =  0.08202099893242121
epoch :  6 ,training loss =  0.08183894580116077
Accuracy in validation set =  33.94736842105264 %
epoch :  6 ,validation loss =  0.08097383597244819
epoch :  7 ,training loss =  0.08091864582835412
Accuracy in validation set =  31.447368421052623 %
epoch :  7 ,validation loss =  0.08058792166411877
epoch :  8 ,training loss =  0.08153458350166982
Accuracy in validation set =  32.36842105263158 %
epoch :  8 ,validation loss =  0.08035911930104096
epoch :  9 ,training loss =  0.08044843390888097
Accuracy in validation set =  30.263157894736835 %
epoch :  9 ,validation loss =  0.08166193185995023
epoch :  10 ,training loss =  0.07989927943871945
Accuracy in validation set =  31.57894736842105 %
epoch :  10 ,validation loss =  0.08008684497326612
epoch :  11 ,training loss =  0.07918039998229669
Accuracy in validation set =  33.026315789473685 %
epoch :  11 ,validation loss =  0.07883754186332226
epoch :  12 ,training loss =  0.07958447598681158
Accuracy in validation set =  28.421052631578945 %
epoch :  12 ,validation loss =  0.08220984848837058
epoch :  13 ,training loss =  0.0788685574519391
Accuracy in validation set =  35.131578947368425 %
epoch :  13 ,validation loss =  0.07854381389915943
epoch :  14 ,training loss =  0.07840672850000616
Accuracy in validation set =  33.81578947368421 %
epoch :  14 ,validation loss =  0.07880387765665849
```

Ήδη παρατηρούμε μία σημαντική βελτίωση σε σχέση με την προηγούμενη περίπτωση, κάτι που περιμέναμε να δούμε όπως είχαμε αναλύσει και σε προγενέστερα βήματα, αφού τα μειωμένα χρονικά βήματα των ακολουθιών βοηθούν ένα LSTM να εκπαιδευτεί καλύτερα.

c) Με βάση τα χρωμογραφήματα :

```
epoch :  0 ,training loss =  2.299966865656327
Accuracy in validation set =  18.81578947368421 %
epoch :  0 ,validation loss =  2.2976003189881644
epoch :  1 ,training loss =  2.2833302750879403
Accuracy in validation set =  18.81578947368421 %
epoch :  1 ,validation loss =  2.2606093088785806
epoch :  2 ,training loss =  2.2591638467749773
Accuracy in validation set =  18.81578947368421 %
epoch :  2 ,validation loss =  2.2592745820681253
epoch :  3 ,training loss =  2.2584565512988033
Accuracy in validation set =  18.81578947368421 %
epoch :  3 ,validation loss =  2.257890353600184
epoch :  4 ,training loss =  2.2586051395961215
Accuracy in validation set =  18.81578947368421 %
epoch :  4 ,validation loss =  2.258551557858785
epoch :  5 ,training loss =  2.2573157719203403
Accuracy in validation set =  18.81578947368421 %
epoch :  5 ,validation loss =  2.258167843023936
epoch :  6 ,training loss =  2.256175980275991
Accuracy in validation set =  18.81578947368421 %
epoch :  6 ,validation loss =  2.2567914724349976
epoch :  7 ,training loss =  2.257677448039152
Accuracy in validation set =  18.81578947368421 %
epoch :  7 ,validation loss =  2.257265696922938
epoch :  8 ,training loss =  2.257079129316369
Accuracy in validation set =  18.81578947368421 %
epoch :  8 ,validation loss =  2.2579858005046844
epoch :  9 ,training loss =  2.258327386817154
Accuracy in validation set =  18.81578947368421 %
epoch :  9 ,validation loss =  2.2571663856506348
epoch :  10 ,training loss =  2.2564298182117697
Accuracy in validation set =  18.81578947368421 %
epoch :  10 ,validation loss =  2.2558971842130027
epoch :  11 ,training loss =  2.2577376122377357
Accuracy in validation set =  18.81578947368421 %
epoch :  11 ,validation loss =  2.258311688899994
epoch :  12 ,training loss =  2.257231702609938
Accuracy in validation set =  18.81578947368421 %
epoch :  12 ,validation loss =  2.2575447460015616
epoch :  13 ,training loss =  2.2591547722719154
Accuracy in validation set =  18.81578947368421 %
epoch :  13 ,validation loss =  2.257505108912786
epoch :  14 ,training loss =  2.2579070159367154
Accuracy in validation set =  18.81578947368421 %
epoch :  14 ,validation loss =  2.257362405459086
```

d) Και τέλος με βάση τα ενωμένα χρωμογραφήματα και spectrograms :

```
epoch :  0 ,training loss =  0.08947776014707526
Accuracy in validation set =  18.157894736842106 %
epoch :  0 ,validation loss =  0.08866289320091407
epoch :  1 ,training loss =  0.08813690074852534
Accuracy in validation set =  19.605263157894736 %
epoch :  1 ,validation loss =  0.08792229772855838
epoch :  2 ,training loss =  0.08754439697581895
Accuracy in validation set =  20.52631578947368 %
epoch :  2 ,validation loss =  0.08745270843307178
epoch :  3 ,training loss =  0.08703137432434122
Accuracy in validation set =  22.236842105263154 %
epoch :  3 ,validation loss =  0.08672747885187466
epoch :  4 ,training loss =  0.08584869075186398
Accuracy in validation set =  21.842105263157897 %
epoch :  4 ,validation loss =  0.08622921910136938
epoch :  5 ,training loss =  0.08464208078019474
Accuracy in validation set =  25.0 %
epoch :  5 ,validation loss =  0.08477593368540208
epoch :  6 ,training loss =  0.08437908608086254
Accuracy in validation set =  25.526315789473685 %
epoch :  6 ,validation loss =  0.08373224393775065
epoch :  7 ,training loss =  0.08343086984692788
Accuracy in validation set =  26.57894736842106 %
epoch :  7 ,validation loss =  0.08414315277089675
epoch :  8 ,training loss =  0.08318692232881274
Accuracy in validation set =  25.131578947368418 %
epoch :  8 ,validation loss =  0.0839892290532589
epoch :  9 ,training loss =  0.08217235411308249
Accuracy in validation set =  27.236842105263158 %
epoch :  9 ,validation loss =  0.08308827225118876
epoch :  10 ,training loss =  0.08174000513188694
Accuracy in validation set =  30.263157894736835 %
epoch :  10 ,validation loss =  0.08241043022523324
epoch :  11 ,training loss =  0.08067133916275841
Accuracy in validation set =  29.86842105263158 %
epoch :  11 ,validation loss =  0.08165222220122814
epoch :  12 ,training loss =  0.08042705576030576
Accuracy in validation set =  27.763157894736846 %
epoch :  12 ,validation loss =  0.08229559691001971
epoch :  13 ,training loss =  0.08192683984430468
Accuracy in validation set =  28.815789473684216 %
epoch :  13 ,validation loss =  0.08148411537210147
epoch :  14 ,training loss =  0.0802706253467774
Accuracy in validation set =  30.263157894736835 %
epoch :  14 ,validation loss =  0.0811287872493267
```

Παρατηρούμε καλύτερα ποσοστά από τις μεμονωμένες περιπτώσεις των χρωμογραφημάτων και των φασματογραφημάτων. Αυτό οφείλεται στην υπάρξη παραπάνω χρήσιμης πληροφορίας για να καταλήξουμε σε κρίσιμα συμπεράσματα ταξινόμησης των δειγμάτων.

ΒΗΜΑ 6

Για να αξιολογήσουμε τα μοντέλα μας θα χρησιμοποιήσουμε τα ακόλουθα test sets:

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

Συγκεκριμένα υπολογίζουμε
α) το accuracy
β) το precision, recall και F1-score για κάθε κλάση
γ) το macro-averaged precision, recall και F1-score για όλες τις κλάσεις
δ) το micro-averaged precision, recall και F1-score για όλες τις κλάσεις

Ας δούμε τι σημαίνει κάθε μία από αυτές τις μετρικές.

Το accuracy δείχνει το ποσοστό του test set που ταξινομήθηκε στην σωστή κλάση.

Ως precision ορίζεται ο λόγος των true positive προβλέψεων του μοντέλου προς το άθροισμα των true positive και των false positive προβλέψεων. Διαισθητικά, περιγράφει την ικανότητα του μοντέλου να μην ταξινομεί σε μια κλάση ένα δείγμα που δεν ανήκει σε αυτή.

Η μετρική recall είναι ο λόγος των true positive προβλέψεων του μοντέλου προς το άθροισμα των true positive και των false negative προβλέψεων. Δηλαδή, είναι η ικανότητα του μοντέλου να βρίσκει όλα τα δείγματα μίας κλάσης, χωρίς να του “ξεφεύγουν”.

Η f1 μετρική είναι κατά κάποιο τρόπο ένα σταθμισμένο μέσο των recall και precision με τον εξής τρόπο: f1-score = 2 * (precision * recall) / (precision + recall).

Οι micro/macro εκδόσεις των μετρικών αυτών αφορούν τις επιδόσεις του μοντέλου πάνω σε όλες τις κλάσεις. Στην μεν micro περίπτωση υπολογίζεται η μετρική, πχ το recall, με χρήση των συνολικών true positive, false positive και false negative για όλες τις κλάσεις. Στην δε macro περίπτωση έχουμε μία μέση τιμή της μετρικής, πχ του recall, για όλες τις κλάσεις. Πρέπει να παρατηρήσουμε στο σημείο αυτό ότι οι micro εκδοχές αυτών των μετρικών είναι όλες ίδιες μεταξύ τους στην περίπτωση πολλών κλάσεων και ίδιες με το συνολικό accuracy.

Γενικά για τις μετρικές:

Υπάρχει περίπτωση να έχουμε μεγάλη απόκλιση μεταξύ precision και f1-score. Κάτι τέτοιο μπορεί να συμβεί για παράδειγμα αν το μοντέλο έχει υψηλό precision και χαμηλό recall. Σε αυτό το ενδεχόμενο το μοντέλο μπορεί να προβλέπει πολύ καλά και χωρίς λάθη πότε ένα δείγμα δεν ανήκει στην κλάση υπό μελέτη αλλά δυσκολεύεται στο να βρει και τα δείγματα που όντως ανήκουν στην κλάση.

Ακομη, υπάρχει πιθανότητα σε ένα μοντέλο να υπάρχει απόκλιση μεταξύ micro και macro f1-score. Αυτό μπορεί να συμβεί αν το σύνολο των δειγμάτων μας είναι μη ισορροπημένο. Για παράδειγμα, μπορεί σε ένα πρόβλημα ταξινόμησης 2 κλάσεων για την μία και πιο πολυπληθή σε δείγματα κλάση να έχουμε μεγάλο precision και recall και άρα υψηλό f1-score ενώ για την άλλη το αντίθετο. Σε αυτή την περίπτωση

το micro-f1-score θα βγει όπως και το accuracy υψηλό λόγω της επιτυχίας στην μεγαλύτερη σε δείγματα κλάση ενώ το macro-f1-score θα βγει χαμηλότερο διότι δεν λαμβάνει υπόψη του την ανισότητα των κλάσεων.

Τέλος, υπάρχουν κατηγορίες προβλημάτων όπου θα πρέπει να βελτιστοποιήσουμε συγκεκριμένα ως προς το precision ή το recall αντίστοιχα. Θα δώσουμε ένα παράδειγμα για το καθένα. Για την πρώτη περίπτωση ας φανταστούμε ότι θέλουμε να δημιουργήσουμε ένα ασφαλές σύστημα που να αναγνωρίζει την αυθεντικότητα ενός χρήστη/προσώπου. Σε αυτήν την περίπτωση μας ενδιαφέρει για λόγους ασφαλείας να έχουμε υψηλό precision, δηλαδή να μπορεί να αναγνωρίσει και να απορρίψει απόπειρες από μη αυθεντικούς χρήστες. Για ένα πρόβλημα ιατρικής διάγνωσης αντίθετα επιθυμούμε να έχουμε υψηλό recall, δηλαδή να μη ξεφεύγουν στο σύστημα δείγματα που είναι θετικά στην ασθένεια/νόσο. Και στις δύο παραπάνω περιπτώσεις ένα απλό υψηλό accuracy ή f1-score δεν είναι κατάλληλη μετρική για την αξιολόγηση του μοντέλου διότι δεν ανταποκρίνονται στις ανάγκες του μοντέλου και στις ιδιαιτερότητες του προβλήματος.

Ας επικεντρωθούμε τώρα στο συγκεκριμένο πρόβλημα της αναγνώρισης του μουσικού είδους από φασματογραφήματα. Σε αυτή την περίπτωση δεν είναι κρίσιμο να αναγνωρίσουμε αν ένα δείγμα ανήκει σε ένα μουσικό είδος ή αν ένα δείγμα δεν ανήκει σε κάποιο. Επομένως, μπορούμε να αρκεστούμε στο precision του μοντέλου ή στο f1-score.

Για το πρώτο lstm δίκτυο εκπαιδευμένο πάνω στα φασματογραφήματα:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	40
1	1.00	0.03	0.05	40
2	0.19	0.90	0.32	80
3	0.20	0.05	0.08	80
4	0.00	0.00	0.00	40
5	0.00	0.00	0.00	40
6	0.00	0.00	0.00	78
7	0.00	0.00	0.00	40
8	0.32	0.55	0.41	103
9	0.00	0.00	0.00	34
accuracy			0.23	575
macro avg	0.17	0.15	0.09	575
weighted avg	0.18	0.23	0.13	575

Για το δεύτερο lstm δίκτυο πάνω στα beat-synced φασματογραφήματα:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	40
1	0.29	0.72	0.41	40
2	0.32	0.66	0.43	80
3	0.32	0.60	0.41	80
4	0.00	0.00	0.00	40
5	0.00	0.00	0.00	40
6	0.48	0.26	0.33	78
7	0.00	0.00	0.00	40
8	0.31	0.34	0.32	103
9	0.00	0.00	0.00	34
accuracy			0.32	575
macro avg	0.17	0.26	0.19	575
weighted avg	0.23	0.32	0.25	575

Για το τρίτο lstm δίκτυο πάνω στα χρωματογραφήματα:

	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	0.00	0.00	0.00	80
4	0.00	0.00	0.00	40
5	0.00	0.00	0.00	40
6	0.00	0.00	0.00	78
7	0.00	0.00	0.00	40
8	0.18	1.00	0.30	103
9	0.00	0.00	0.00	34
accuracy			0.18	575
macro avg	0.02	0.10	0.03	575
weighted avg	0.03	0.18	0.05	575

Για το τέταρτο lstm δίκτυο πάνω στα φασματογραφήματα και χρωματογραφήματα:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	40
1	0.30	0.60	0.40	40
2	0.28	0.74	0.41	80
3	0.37	0.24	0.29	80
4	0.00	0.00	0.00	40
5	0.00	0.00	0.00	40
6	0.36	0.55	0.43	78
7	0.00	0.00	0.00	40
8	0.27	0.30	0.29	103
9	0.00	0.00	0.00	34
accuracy			0.31	575
macro avg	0.16	0.24	0.18	575
weighted avg	0.21	0.31	0.23	575

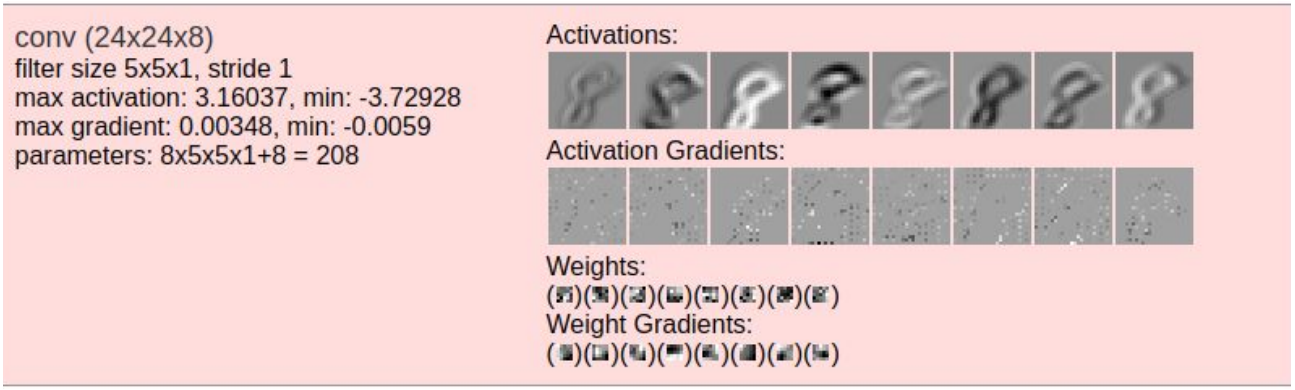
Επομένως, με βάση τις μετρικές που θέσαμε για το πρόβλημα θεωρούμε πώς την καλύτερη επίδοση συγκεντρώνει το lstm μοντέλο εκπαιδευμένο στα συγχρονισμένα με το ρυθμό φασματογραφήματα.

ΒΗΜΑ 7

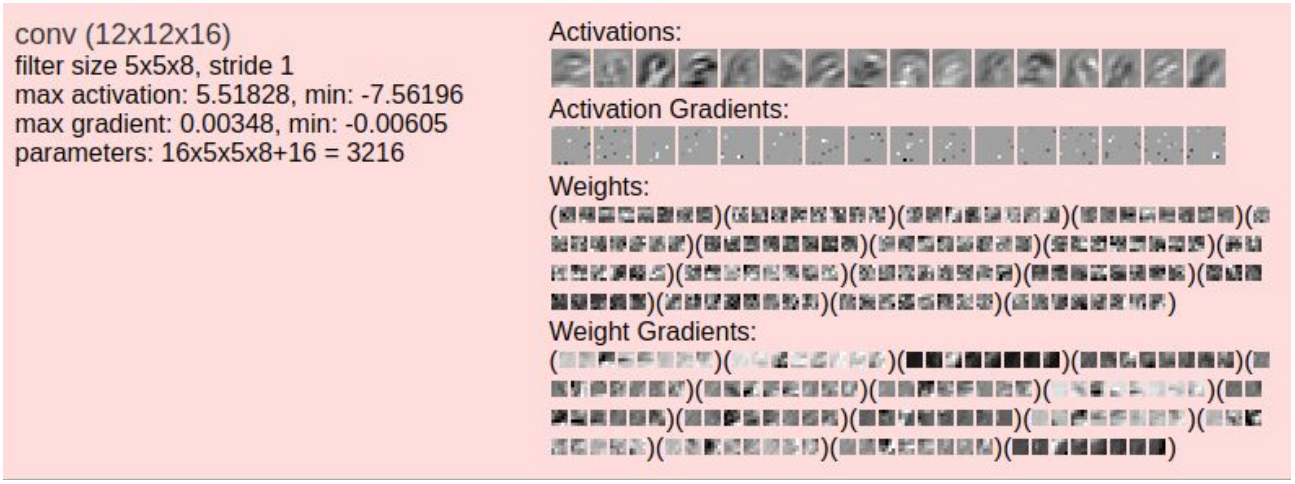
Για το 7ο βήμα της αναφοράς χρησιμοποιήσαμε convolutional neural networks.

a) Για εξοικείωση πειραματιστήκαμε με τον ιστότοπο https://cs.stanford.edu/people/karpathy/convnetjs/demo/mnist.html όπου δίνεται η ευκαιρία να εκπαιδεύσουμε ένα CNN πάνω σε ένα dataset εικόνων χειρόγραφων ψηφίων αφαιρώντας τις λεπτομέρειες της προγραμματιστικής υλοποίησης. Το συγκεκριμένο CNN αποτελείται από τα εξής επίπεδα:
- συνελικτικό επίπεδο 8 πυρήνων 5x5
- relu
- επίπεδο max pooling 2x2
- συνελικτικό επίπεδο 16 πυρήνων 5x5
- relu
- επίπεδο max pooling 3x3
- πλήρως συνδεδεμένο επίπεδο νευρώνων 10 εξόδων
- softmax

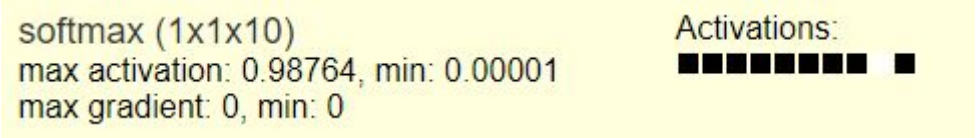
Από την οπτικοποίηση που προσφέρει το περιβάλλον βλέπουμε ότι τα βάρη στα οποία καταλήγουν οι πυρήνες μετά από εκπαίδευση καθώς και οι έξοδοι του κάθε επιπέδου φαίνεται να έχουν κάποια “φυσική” ερμηνεία. Για το πρώτο από τα συνελικτικά επίπεδα βλέπουμε ότι τα βάρη του δικτύου έχουν προσαρμοστεί ώστε να εντοπίζουν πρωτόλεια στοιχεία της απεικόνισης των ψηφίων.



Όμοια και το δεύτερο συνελικτικό επίπεδο.



Καθώς, πηγαίνουμε βαθύτερα στο δίκτυο, αναπαρίστανται όλο και πιο αφηρημένα χαρακτηριστικά στις εικόνες και απομακρύνονται από τις αρχικές εικόνες που έχουμε. Τελικώς, υπάρχει ένα επίπεδο softmax που μας δίνει την τελική ταξινόμηση του ψηφίου :



b) Υλοποιούμε κι εμείς ένα συνελικτικό δίκτυο με 4 στάδια που το κάθε ένα περιλαμβάνει
- 2D convolution
- Batch normalization
- ReLU activation
- Max pooling

και το εκπαιδεύουμε στο σύνολο των φασματογραφημάτων τα οποία τα βλέπουμε τώρα ως μονοκάναλες εικόνες αντί για ακολουθίες. Τα αποτελέσματα είναι τα εξής:

Κατά την διάρκεια του training:

```
epoch :  0 ,training loss =  2.322191997450225
Accuracy in validation set =  36.44736842105263 %
epoch :  1 ,training loss =  1.5035201019170332
Accuracy in validation set =  33.157894736842096 %
epoch :  2 ,training loss =  1.1349664403467763
Accuracy in validation set =  30.78947368421052 %
epoch :  3 ,training loss =  0.8392163332627744
Accuracy in validation set =  37.5 %
epoch :  4 ,training loss =  0.4912762654070951
Accuracy in validation set =  37.36842105263158 %
epoch :  5 ,training loss =  0.33280697890690397
Accuracy in validation set =  41.05263157894736 %
epoch :  6 ,training loss =  0.23151969727204771
Accuracy in validation set =  45.39473684210527 %
epoch :  7 ,training loss =  0.20587552779791307
Accuracy in validation set =  45.78947368421053 %
epoch :  8 ,training loss =  0.12027399667671748
Accuracy in validation set =  45.526315789473685 %
epoch :  9 ,training loss =  0.10920248013369892
Accuracy in validation set =  45.78947368421053 %
epoch :  10 ,training loss =  0.07441867355789457
Accuracy in validation set =  41.57894736842105 %
epoch :  11 ,training loss =  0.06085446971107503
Accuracy in validation set =  44.47368421052631 %
epoch :  12 ,training loss =  0.048462813454014916
Accuracy in validation set =  44.99999999999999 %
epoch :  13 ,training loss =  0.041556312265444775
Accuracy in validation set =  45.78947368421053 %
epoch :  14 ,training loss =  0.0386166409874449
Accuracy in validation set =  45.657894736842096 %
epoch :  15 ,training loss =  0.03641313333444449
Accuracy in validation set =  46.18421052631578 %
epoch :  16 ,training loss =  0.027865632456176136
Accuracy in validation set =  45.78947368421053 %
epoch :  17 ,training loss =  0.027256042259384175
Accuracy in validation set =  47.10526315789474 %
epoch :  18 ,training loss =  0.0255702445473598
Accuracy in validation set =  45.92105263157895 %
epoch :  19 ,training loss =  0.021266083404117703
Accuracy in validation set =  46.18421052631578 %
epoch :  20 ,training loss =  0.018247180504306237
Accuracy in validation set =  45.526315789473685 %
epoch :  21 ,training loss =  0.017228752602728045
Accuracy in validation set =  46.18421052631578 %
epoch :  22 ,training loss =  0.016719383586730276
Accuracy in validation set =  45.39473684210527 %
epoch :  23 ,training loss =  0.01678920569544544
Accuracy in validation set =  44.47368421052631 %
epoch :  24 ,training loss =  0.01456709154786504
Accuracy in validation set =  46.97368421052632 %
```

και τα αποτελέσματα στο test set, confusion matrix και accuracy:

```
(array([[ 3.,  1.,  0., 12.,  5.,  4.,  4.,  2.,  8.,  1.],
        [ 0., 24.,  5.,  6.,  0.,  3.,  1.,  0.,  0.,  1.],
        [ 0.,  2., 50.,  5.,  2.,  4.,  5.,  3.,  4.,  5.],
        [ 4.,  5.,  2., 48.,  2.,  1.,  4.,  2., 12.,  0.],
        [ 0.,  2.,  5.,  1., 16.,  0.,  1.,  0.,  6.,  9.],
        [ 1., 10.,  5.,  4.,  2.,  7.,  2.,  1.,  3.,  5.],
        [ 0.,  4.,  5.,  2.,  0.,  0., 52.,  1., 11.,  3.],
        [ 0.,  0.,  3., 14.,  2.,  1.,  4.,  2., 11.,  3.],
        [ 0.,  4., 11., 14.,  2.,  4., 28.,  3., 35.,  2.],
        [ 0.,  0.,  6.,  2.,  1.,  1.,  1.,  0.,  5., 18.]]),
 0.4434782608695652)
```

c) Σε αυτό το σημείο θα πρέπει να αναφερθούμε στην λειτουργία και τον ρόλο ορισμένων από τα βασικά επίπεδα ενός συνελικτικού νευρωνικού δικτύου.

Το συνελικτικό επίπεδο, από το οποίο παίρνει και το όνομα της όλη η αρχιτεκτονική, έχει ως πρωταρχικό σκοπό την εξαγωγή χαρακτηριστικών από την είσοδο-εικόνα. Η συνέλιξη διατηρεί την χωρική σχέση μεταξύ των pixels μαθαίνοντας χαρακτηριστικά της εικόνας χρησιμοποιώντας μικρά ορθογώνια, συνήθως τετράγωνα, των δεδομένων εισόδου. Αν θεωρήσουμε την αναπαράσταση της εικόνας ως πίνακα δύο διαστάσεων, τότε η πράξη της συνέλιξης συνίσταται στην δημιουργία ενός νέου πίνακα-εικόνας με κύλιση ενός πυρήνα, ενός μικρού τετραγώνου, σε όλες τις δυνατές θέσεις και εφαρμογή σε κάθε θέση της πράξης της συνέλιξης. Το σύνολο των αποτελεσμάτων σε κάθε θέση είναι η εικόνα αποτέλεσμα.

Το επόμενο επίπεδο που χρησιμοποιούμε είναι ένα επίπεδο batch normalization. Σε αυτό οι έξοδοι της συνέλιξης με τους διάφορους πυρήνες κανονικοποιούνται ως προς την μέση τιμή και την τυπική απόκλιση των αποτελεσμάτων του προηγούμενο επιπέδου. Με αυτό τον τρόπο κρατάμε τις τιμές του δικτύου σε ένα πλαίσιο και επιταχύνουμε την διαδικασία εκπαίδευσης κι επίσης εισάγουμε ένα είδους θόρυβο κατά αυτόν τον μετασχηματισμό και άρα αποφεύγουμε φαινόμενα overfitting. Τέλος, η υλοποίηση του batch normalization επιπέδου του pytorch έχει και δύο εκπαιδεύσιμες παραμέτρους, ένα συντελεστή και ένα bias προκειμένου να διατηρήσει την σταθερότητα του υπόλοιπου δικτύου.

Όπως και κάθε νευρωνικό δίκτυο θα πρέπει και το δικό μας να περιλαμβάνει ένα επίπεδο μη γραμμικότητας, δηλαδή ένα επίπεδο στο οποίο θα εφαρμόζεται σε κάθε τιμή - σημειακά μία μη γραμμική συνάρτηση. Στην περίπτωση μας αυτή θα είναι η ReLU δηλαδή η ταυτοτική για $x>0$ και η σταθερή μηδενική για $x<0$. Η συγκεκριμένη επιλογή είναι συχνή σε αρχιτεκτονικές βαθιάς μάθησης καθώς διευκολύνει την εκπαίδευση, έχει σταθερή παράγωγο, και αποφεύγει προβλήματα vanishing gradient τα οποία είναι αρκετά συχνά σε πολυεπίπεδες αρχιτεκτονικές με σιγμοειδής ή υπερβολικές εφαπτομένες ως μη γραμμικότητες.

Καθώς στοιβάζουμε όλο και περισσότερα επίπεδα αυξάνοντας το πλήθος των συνελικτικών πυρήνων παράλληλα παρατηρούμε ένα πολύ σημαντικό πρόβλημα: Το πλήθος των παραμέτρων αρχίζει να ξεφεύγει και η εκπαίδευση να γίνεται όλο και πιο δύσκολη και αργή. Για να το ξεπεράσουμε αυτό χωρίς να χάσουμε την ακρίβεια του μοντέλου τοποθετούμε ένα επίπεδο pooling το οποίο χρησιμοποιώντας πάλι την ιδέα του πυρήνα, για να εκμεταλλευτούμε την χωρική τοπικότητα, μειώνει κάθε φορά την ανάλυση της εικόνας στο μισό.

d) Παρατηρούμε ότι η συγκεκριμένη αρχιτεκτονική δίνει καλύτερα αποτελέσματα και στο validation και στο test set απ’ότι η αρχιτεκτονική lstm. Με accuracy 44-46% έχει πολύ καλύτερη επίδοση από την lstm λύση του ερωτήματος 5α, που έχει περίπου 25% στο validation set.

ΒΗΜΑ 8

Σε αυτό το βήμα θα προσπαθήσουμε να προβλέψουμε πραγματικές τιμές στο διάστημα [0, 1]. Συγκεκριμένα, με βάση τα φασματογραφήματα των κομματιών και τις επισημειώσεις τους για τις τιμές των
- valence
- energy
- danceability
θα προσπαθήσουμε να κάνουμε προβλέψεις για τις αντίστοιχες τιμές. Σε αυτό το βήμα θα χρησιμοποιήσετε το multitask_dataset/train_labels.txt.

a) Για τον σκοπό αυτό θα προσαρμόσουμε την διαδικασία εκπαίδευσης στα βήματα 5 και 7 ώστε να αποδίδουν καλύτερα σε πρόβλημα regression από το classification. Η αλλαγή που κάνουμε είναι η χρήση της συνάρτησης μέσου τετραγωνικού σφάλματος ως συνάρτηση κόστους προς ελαχιστοποίηση αντί αυτή της διασταυρούμενης εντροπίας.

b) Τα αποτελέσματα του training για το lstm για την πρόβλεψη του valence:

```
epoch :  0 ,training loss =  0.1485870030115951
epoch :  0 ,validation loss =  0.06654710056526321
epoch :  1 ,training loss =  0.07143332267349417
epoch :  1 ,validation loss =  0.06640007080776351
epoch :  2 ,training loss =  0.06650444069369273
epoch :  2 ,validation loss =  0.06479750520416669
epoch :  3 ,training loss =  0.06869294358925386
epoch :  3 ,validation loss =  0.06525066973907608
epoch :  4 ,training loss =  0.06805968640202825
epoch :  4 ,validation loss =  0.06475942262581416
epoch :  5 ,training loss =  0.06562553430822762
epoch :  5 ,validation loss =  0.06601510143705777
epoch :  6 ,training loss =  0.06614029255102981
epoch :  6 ,validation loss =  0.06502687877842359
epoch :  7 ,training loss =  0.0653492437506264
epoch :  7 ,validation loss =  0.0651736498943397
epoch :  8 ,training loss =  0.06933779404921965
epoch :  8 ,validation loss =  0.06466272579772132
epoch :  9 ,training loss =  0.06542549248446118
epoch :  9 ,validation loss =  0.065296221524477
epoch :  10 ,training loss =  0.07140219685706226
epoch :  10 ,validation loss =  0.06471036055258342
epoch :  11 ,training loss =  0.06547695703127167
epoch :  11 ,validation loss =  0.06532027838485581
epoch :  12 ,training loss =  0.06367541108788415
epoch :  12 ,validation loss =  0.06514829556856837
epoch :  13 ,training loss =  0.06718402914702892
epoch :  13 ,validation loss =  0.06567545235157013
epoch :  14 ,training loss =  0.0662531065331264
epoch :  14 ,validation loss =  0.06452211258666855
```

και τα αντίστοιχα για το cnn:

```
epoch :  0 ,training loss =  0.28241137821565976
epoch :  0 ,validation loss =  0.17545177893979208
epoch :  5 ,training loss =  0.04782540190287612
epoch :  5 ,validation loss =  0.09729160581316267
epoch :  10 ,training loss =  0.02826998954300176
epoch :  10 ,validation loss =  0.09476154031498092
epoch :  15 ,training loss =  0.015480867333032867
epoch :  15 ,validation loss =  0.09497664549521037
epoch :  20 ,training loss =  0.019156079929829997
epoch :  20 ,validation loss =  0.08485651016235352
epoch :  25 ,training loss =  0.008010165956379338
epoch :  25 ,validation loss =  0.09010611474514008
epoch :  30 ,training loss =  0.004691568489017134
epoch :  30 ,validation loss =  0.08899326462830816
epoch :  35 ,training loss =  0.004483387678522955
epoch :  35 ,validation loss =  0.0877743379345962
epoch :  40 ,training loss =  0.008432487967762758
epoch :  40 ,validation loss =  0.08988524547645024
epoch :  45 ,training loss =  0.003336557631634853
epoch :  45 ,validation loss =  0.08837686691965375
```

c) Έπειτα επαναλαμβάνουμε την εκπαίδευση αλλά για το energy:

```
epoch :  0 ,training loss =  0.11493013274263252
epoch :  0 ,validation loss =  0.08036307990550995
epoch :  1 ,training loss =  0.07298359037800269
epoch :  1 ,validation loss =  0.06700168283922332
epoch :  2 ,training loss =  0.06565985100513155
epoch :  2 ,validation loss =  0.06713560596108437
epoch :  3 ,training loss =  0.06534635961394418
epoch :  3 ,validation loss =  0.06660985467689377
epoch :  4 ,training loss =  0.06730341487987475
epoch :  4 ,validation loss =  0.0670383907854557
epoch :  5 ,training loss =  0.070839640091766
epoch :  5 ,validation loss =  0.06659511689628873
epoch :  6 ,training loss =  0.0682972613722086
epoch :  6 ,validation loss =  0.06689725071191788
epoch :  7 ,training loss =  0.06627874279564078
epoch :  7 ,validation loss =  0.06658808354820524
epoch :  8 ,training loss =  0.06467481063340198
epoch :  8 ,validation loss =  0.06727755442261696
epoch :  9 ,training loss =  0.06667064926163717
epoch :  9 ,validation loss =  0.06670022170458521
epoch :  10 ,training loss =  0.06667111424559896
epoch :  10 ,validation loss =  0.06658849865198135
epoch :  11 ,training loss =  0.0675410618158904
epoch :  11 ,validation loss =  0.06766992009111814
epoch :  12 ,training loss =  0.06899657794697718
epoch :  12 ,validation loss =  0.06657866707869939
epoch :  13 ,training loss =  0.06473832962695848
epoch :  13 ,validation loss =  0.06710645982197352
epoch :  14 ,training loss =  0.0651581058786674
epoch :  14 ,validation loss =  0.0666191428899765
```

και για το cnn:

```
epoch :  0 ,training loss =  0.11576225066726858
epoch :  0 ,validation loss =  0.08477152564695903
epoch :  5 ,validation loss =  0.07245108378784997
epoch :  10 ,training loss =  0.01318331450139257
epoch :  10 ,validation loss =  0.06764077820948192
epoch :  15 ,validation loss =  0.07070745155215263
epoch :  20 ,training loss =  0.010992686295966532
epoch :  20 ,validation loss =  0.0663531172488417
epoch :  25 ,validation loss =  0.0589728264936379
epoch :  30 ,training loss =  0.026432085972787303
epoch :  30 ,validation loss =  0.07090062754494804
epoch :  35 ,validation loss =  0.056242216378450394
epoch :  40 ,training loss =  0.005740945566106926
epoch :  40 ,validation loss =  0.048934080238853185
epoch :  45 ,validation loss =  0.05166240941200938
```

d) Τέλος, εκπαιδεύουμε για το danceability. Training στο lstm:

```
epoch :  0 ,training loss =  0.15884227187118746
epoch :  0 ,validation loss =  0.02945337550980704
epoch :  1 ,training loss =  0.03610380578108809
epoch :  1 ,validation loss =  0.029254933819174767
epoch :  2 ,training loss =  0.03311689689078114
epoch :  2 ,validation loss =  0.029334627890161107
epoch :  3 ,training loss =  0.031819385645741764
epoch :  3 ,validation loss =  0.028999279120138714
epoch :  4 ,training loss =  0.03178639074956829
epoch :  4 ,validation loss =  0.028730266594461033
epoch :  5 ,training loss =  0.0339926601472226
epoch :  5 ,validation loss =  0.028851472373519624
epoch :  6 ,training loss =  0.03313510289246386
epoch :  6 ,validation loss =  0.02854219318500587
epoch :  7 ,training loss =  0.03178759977560152
epoch :  7 ,validation loss =  0.028518148564866612
epoch :  8 ,training loss =  0.031059888953512364
epoch :  8 ,validation loss =  0.028921772592834065
epoch :  9 ,training loss =  0.03061633244876496
epoch :  9 ,validation loss =  0.029186419610466276
epoch :  10 ,training loss =  0.03295001218264753
epoch :  10 ,validation loss =  0.02844046854547092
epoch :  11 ,training loss =  0.03104885295033455
epoch :  11 ,validation loss =  0.02842135008956705
epoch :  12 ,training loss =  0.030387873655523766
epoch :  12 ,validation loss =  0.02841875808579581
epoch :  13 ,training loss =  0.03149779335680333
epoch :  13 ,validation loss =  0.028462395604167665
epoch :  14 ,training loss =  0.03164057873866775
epoch :  14 ,validation loss =  0.028515478329999105
```

και στο cnn:

```
epoch :  0 ,training loss =  0.17688481797548858
epoch :  0 ,validation loss =  0.061201839574745724
epoch :  5 ,validation loss =  0.05549345697675433
epoch :  10 ,training loss =  0.014680726030333475
epoch :  10 ,validation loss =  0.07347660139203072
epoch :  15 ,validation loss =  0.06200924675379481
epoch :  20 ,training loss =  0.016622815556316214
epoch :  20 ,validation loss =  0.07774682555879865
epoch :  25 ,validation loss =  0.04691955713289125
epoch :  30 ,training loss =  0.003946127105419609
epoch :  30 ,validation loss =  0.05182740278542042
epoch :  35 ,validation loss =  0.05018974840641022
epoch :  40 ,training loss =  0.00801420637766238
epoch :  40 ,validation loss =  0.04619627179844039
epoch :  45 ,validation loss =  0.048456271312066486
```

e) Για την τελική εκτίμηση του μοντέλου στο test set θα χρησιμοποιήσουμε ως μετρική το μέσο Spearman correlation ανάμεσα στις πραγματικές τιμές και το ground truth και για τους τρεις άξονες (valence, energy, danceability).

lstm:

```
The Spearman correlation between the ground true values and the predicted values for valence is  0.012364502171831717
The Spearman correlation between the ground true values and the predicted values for energy is  0.16034465501623185
The Spearman correlation between the ground true values and the predicted values for danceability is  0.210778828662883
The Mean Spearman correlation between the ground true values and the predicted values is  12.782932861698217 %
```

cnn:

```
The Spearman correlation between the ground true values and the predicted values for valence is  0.27917109987070177
The Spearman correlation between the ground true values and the predicted values for energy is  0.6176403857147156
The Spearman correlation between the ground true values and the predicted values for danceability is  0.32707513314810377
The Mean Spearman correlation between the ground true values and the predicted values is  40.796220624450704 %
```

Παρατηρούμε ότι και στους τρεις άξονες το συνελικτικό μοντέλο δίνει πολλαπλάσια καλύτερα αποτελέσματα από το lstm μοντέλο.

ΒΗΜΑ 9

9α

Μια μέθοδος που χρησιμοποιείται συχνά στην εκπαίδευση μοντέλων όταν δεν είναι διαθέσιμα αρκετά δεδομένα είναι η μεταφορά γνώσης. Δηλαδή, η εκπαίδευση του μοντέλου σε ένα παρεμφερές πρόβλημα, με μεγαλύτερο train set, και μετά η χρησιμοποιήση των εκπαιδευμένων παραμέτρων του στο αρχικό πρόβλημα. Με αυτό τον τρόπο η εκπαίδευση στο μικρό σύνολο δεδομένων εκπαίδευσης ξεκινάει με μη-τυχαίες παραμέτρους κι έτσι έχουμε καλύτερα αποτελέσματα και μικρότερη πιθανότητα overfitting. Το μοντέλο μας χρησιμοποιεί την γνώση που απέκτησε κατά την εκπαίδευσή του στο παρεμφερές πρόβλημα για να γενικεύει καλύτερα στο αρχικό.

a) Η μελέτη πολλών τύπων νευρωνικών δικτύων και κυρίως των συνελικτικών έχει δείξει ότι τα πρώτα επίπεδα του δικτύου φαίνεται να εκπαιδεύονται στο να μαθαίνουν να αναγνωρίζουν χαρακτηριστικά της εισόδου τα οποία στην πλειοψηφία τους χαρακτηρίζουν την είσοδο σε γενικότερο πλαίσιο από το εκάστοτε πρόβλημα. Με αφορμή αυτή την παρατήρηση, έχει γίνει έρευνα στο κατά πόσο είναι δυνατόν να μεταφερθεί γνώση από εκπαιδευμένους νευρώνες συναρτήσει του βάθους της θέσης τους στο δίκτυο. Αποδεικνύεται πειραματικά ότι η μεταφερσιμότητα της γνώσης των νευρώνων επηρεάζεται από την εξειδίκευση των τελικών νευρώνων και από την σχέση γειτονικών επιπέδων νευρώνων. Όμως, ακόμα και η γνώση αυτών των νευρώνων είναι πολλές φορές καλύτερη από την τυχαία αρχικοποίηση των παραμέτρων του μοντέλου.

b) Επιλέγουμε το μοντέλου του CNN έναντι αυτού του LSTM. Η επιλογή γίνεται λόγω των καλύτερων αποτελεσμάτων που μας έδωσε το cnn στα προηγούμενα ερωτήματα, καθώς πρόκειται για μοντέλο που παράγει καλύτερα αποτελέσματα στο πεδίο της επεξεργασίας εικόνων, το οποίο και αφορά την συγκεκριμένη εφαρμογή.

c) Εκπαιδεύουμε το μοντέλο στο fma_genre_spectrograms dataset, όπως κάναμε και στο ερώτημα 7b, χρησιμοποιώντας παράλληλα ένα checkpoint που θα αποθηκεύει κάθε φορά τα βάρη του δικτύου στην εποχή που έχουμε τα καλύτερα αποτελέσματα. Για πληρότητα, επισυνάπτουμε τα αποτελέσματα και αυτής της εκπαίδευσης του μοντέλου :

```
epoch :  0 ,training loss =  2.3794826439448764
Accuracy in validation set =  36.31578947368421 %
epoch :  1 ,training loss =  1.617509515918031
Accuracy in validation set =  33.55263157894737 %
epoch :  2 ,training loss =  1.2348788246816518
Accuracy in validation set =  35.0 %
epoch :  3 ,training loss =  0.8115654149833991
Accuracy in validation set =  34.078947368421055 %
epoch :  4 ,training loss =  0.5825166915144239
Accuracy in validation set =  39.21052631578947 %
epoch :  5 ,training loss =  0.37180917299523647
Accuracy in validation set =  35.65789473684211 %
epoch :  6 ,training loss =  0.2557648590632847
Accuracy in validation set =  41.31578947368422 %
epoch :  7 ,training loss =  0.177902631613673
Accuracy in validation set =  37.89473684210527 %
epoch :  8 ,training loss =  0.13908079114495492
Accuracy in validation set =  41.05263157894736 %
epoch :  9 ,training loss =  0.09832225975637533
Accuracy in validation set =  41.18421052631579 %
epoch :  10 ,training loss =  0.07924268379503367
Accuracy in validation set =  42.763157894736835 %
epoch :  11 ,training loss =  0.06470818650357578
Accuracy in validation set =  42.763157894736835 %
epoch :  12 ,training loss =  0.05762544912951333
Accuracy in validation set =  41.18421052631579 %
epoch :  13 ,training loss =  0.04884910279390763
Accuracy in validation set =  42.368421052631575 %
epoch :  14 ,training loss =  0.039622097979394755
Accuracy in validation set =  43.42105263157895 %
epoch :  15 ,training loss =  0.033755218686193834
Accuracy in validation set =  43.42105263157895 %
epoch :  16 ,training loss =  0.030006189583515634
Accuracy in validation set =  42.10526315789473 %
epoch :  17 ,training loss =  0.028277121978450795
Accuracy in validation set =  43.28947368421052 %
epoch :  18 ,training loss =  0.023511535887207304
Accuracy in validation set =  42.89473684210526 %
epoch :  19 ,training loss =  0.022812356203034217
Accuracy in validation set =  44.21052631578948 %
epoch :  20 ,training loss =  0.02113889463778053
Accuracy in validation set =  44.34210526315789 %
epoch :  21 ,training loss =  0.018044110672662452
Accuracy in validation set =  43.15789473684211 %
epoch :  22 ,training loss =  0.016639603130823494
Accuracy in validation set =  44.21052631578948 %
epoch :  23 ,training loss =  0.015643523720910355
Accuracy in validation set =  43.55263157894738 %
epoch :  24 ,training loss =  0.014363675131177416
Accuracy in validation set =  45.131578947368425 %
```

και στο test set :

```
(array([[ 4.,  0.,  1., 17.,  1.,  2.,  4.,  0., 10.,  1.],
        [ 0., 22.,  4.,  7.,  0.,  4.,  1.,  0.,  2.,  0.],
        [ 0.,  1., 51.,  5.,  6.,  2.,  5.,  1.,  7.,  2.],
        [ 1.,  5.,  0., 55.,  2.,  2.,  3.,  0., 12.,  0.],
        [ 0.,  1.,  8.,  3., 20.,  1.,  0.,  0.,  5.,  2.],
        [ 1.,  9.,  9.,  9.,  1.,  3.,  1.,  0.,  4.,  3.],
        [ 0.,  2.,  6.,  3.,  0.,  0., 43.,  1., 23.,  0.],
        [ 2.,  0.,  4., 15.,  2.,  1.,  0.,  2., 14.,  0.],
        [ 0.,  1.,  5., 13.,  1.,  3., 29.,  2., 48.,  1.],
        [ 0.,  0., 14.,  8.,  1.,  1.,  1.,  0.,  5.,  4.]]),
 0.43826086956521737)
```

d) Σε αυτό το βήμα θα εφαρμόσουμε την τεχνική του transfer learning μεταξύ των δύο διαφορετικών προβλημάτων που εξετάσαμε στα προηγούμενα βήματα. Έχουμε ήδη αποθηκευμένα τα βάρη του καλύτερου μοντέλου που πετυχαίνουμε κατά την εκπαίδευση στο fma_genre_spectrograms. Έτσι, δημιουργούμε ένα νέο μοντέλο που θα εκπαιδευτεί στο multitask dataset και αρχικά του περνάμε τα υπολογισμένα βάρη. Στο συγκεκριμένο βήμα θα ασχοληθούμε μόνο με την μετρική valence (για τις υπόλοιπες μπορούμε να ακολουθήσουμε παρόμοια διαδικασία). Στη συνέχεια, εφαρμόζουμε εκπαίδευση με λίγες εποχές (fine tuning) και τα αποτελέσματα που παίρνουμε είναι :

```
epoch :  0 ,training loss =  0.1478214135224169
epoch :  0 ,validation loss =  0.19842975054468429
epoch :  1 ,training loss =  0.06939574720507319
epoch :  1 ,validation loss =  0.07216924535376686
epoch :  2 ,training loss =  0.05262090037153526
epoch :  2 ,validation loss =  0.0850745460816792
epoch :  3 ,training loss =  0.0472526355561885
epoch :  3 ,validation loss =  0.0852866390986102
epoch :  4 ,training loss =  0.050435670566829766
epoch :  4 ,validation loss =  0.08369494176336698
epoch :  5 ,training loss =  0.03762472641061653
epoch :  5 ,validation loss =  0.07681218800800187
epoch :  6 ,training loss =  0.03632977681065148
epoch :  6 ,validation loss =  0.07444819488695689
epoch :  7 ,training loss =  0.03432422292164781
epoch :  7 ,validation loss =  0.07787303679755755
epoch :  8 ,training loss =  0.033181146887893025
epoch :  8 ,validation loss =  0.07315106210964066
epoch :  9 ,training loss =  0.02938474735922434
epoch :  9 ,validation loss =  0.08274818531104497
```

Παίρνοντας και εδώ την spearman μετρική, προκύπτει τελικώς :

```
The Spearman correlation between the ground true values and the predicted values for valence is  0.4294233743431847
```

e) Παρατηρούμε ότι η συσχέτιση spearman αυξήθηκε από 0.28 σε 0.43. Αυτή η αύξηση είναι αναμενόμενη, καθώς αυτός είναι ο βασικός στόχος της τεχνικής του transfer learning. Δηλαδή, να χρησιμοποιήσουμε την γνώση που αποκτήσαμε από ένα αρκετά μεγαλύτερο και παραπλήσιο μοντέλο, ώστε να συγκλίνουμε σε καλύτερη απόδοση στο νέο μοντέλο μας με λιγότερες εποχές (fine tuning). Για περαιτέρω βελτίωση της απόδοσης, θα έπρεπε να χρησιμοποιήσουμε ένα αρκετά μεγαλύτερο dataset για την εκπαίδευση των αρχικών βαρών, καθώς όπως γνωρίζουμε σε μεγάλες εφαρμογές της συγκεκριμένης τεχνικής τα προεκπαιδευμένα βάρη προέρχονται συνήθως από κάποια πολύ μεγάλα datasets (cifar100 κλπ).

9β

Όταν έχουμε πολλές επισημειώσεις στα δεδομένα μας πολλές φορές είναι αποδοτικότερο να εκπαιδεύσουμε ένα μοντέλο το οποίο να αναγνωρίζει ταυτόχρονα σε όλες τις διαστάσεις των επισημειώσεων.

a) Ερευνητικές προσπάθειες σε αυτή την ιδέα έχουν προσπαθήσει να εκπαιδεύσουν μοντέλα τα οποία φέρνουν καλά αποτελέσματα σε ένα πλήθος εφαρμογών και προβλημάτων. Συγκεκριμένα, στην βάση ImageNet, σε προβλήματα μετάφρασης, προσθήκης λεζάντας σε εικόνες, αναγνώρισης φωνής και επεξεργασίας γλώσσας ταυτόχρονα! Παρατήρησαν ότι η ταυτόχρονη εκπαίδευση βοηθησε στην επίδοση των επιμέρους task και ιδιαίτερα σε task με μικρό πλήθος δεδομένων εκπαίδευσης. Επιπλέον, μέσω της τεχνικής του multitask learning, μειώνεται σημαντικά το overfitting κατά την εκπαίδευση των επιμέρους tasks, αφού το δίκτυο δεν επικεντρώνεται στην εκμάθηση των βαρών μόνο για ένα πρόβλημα, αλλά για τον συνδυασμό διαφορετικών προβλημάτων.

b) Σε αυτό το υποερώτημα, θα εκπαιδεύσουμε ένα μοντέλο στο multitask dataset χρησιμοποιώντας σαν συνάρτηση κόστους το άθροισμα από τα κόστη (losses) για το valence, energy και danceability. Αρχικά, με νέους loaders φορτώνουμε τα συνολικά δεδομένα και για τις τρεις επισημειώσεις. Στη συνέχεια, εφαρμόζουμε συνολική εκπαίδευση, χρησιμοποιώντας κατάλληλα βάρη, τα οποία μπορούν να μεταβληθούν ώστε να φέρουμε τα επιμέρους κόστη στην ίδια τάξη μεγέθους. Τα ορίζουμε αρχικά όλα ίσα με 0.33. Παρουσιάζουμε ένα μέρος των αποτελεσμάτων της εκπαίδευσης :

```
Epoch:  0
Training
valence_loss: 2.5908724760467354, energy_loss: 3.4761291315609757, danceability_loss: 2.3947780525142495
Total training loss =  2.79238737848672
Validation
valence_loss: 0.16190234358821595, energy_loss: 0.20052284853799002, danceability_loss: 0.05069834499486855
Total validation loss =  0.13633077485220774
Epoch:  5
Training
valence_loss: 0.12066578678786755, energy_loss: 0.05945925787091255, danceability_loss: 0.041595006425103005
Total training loss =  0.07316761938008395
Validation
valence_loss: 0.10598519657339368, energy_loss: 0.09260144404002599, danceability_loss: 0.08717685458915574
Total validation loss =  0.09430195604051862
Epoch:  10
Training
valence_loss: 0.056876372630623256, energy_loss: 0.02690119749273766, danceability_loss: 0.023712781554257326
Total training loss =  0.035471817732534626
Validation
valence_loss: 0.10236443845289094, energy_loss: 0.09051733144692012, danceability_loss: 0.10554284176656178
Total validation loss =  0.09848012562308993
Epoch:  15
Training
valence_loss: 0.13632437485185536, energy_loss: 0.021203547927804968, danceability_loss: 0.04497546749189496
Total training loss =  0.0668261203576218
Validation
valence_loss: 0.11266701881374631, energy_loss: 0.08756444177457265, danceability_loss: 0.07507369880165372
Total validation loss =  0.09085070448262351
Epoch:  20
Training
valence_loss: 0.02210799376056953, energy_loss: 0.01173401774245907, danceability_loss: 0.009132595044899394
Total training loss =  0.014181620631874963
Validation
valence_loss: 0.11174128204584122, energy_loss: 0.0883654026048524, danceability_loss: 0.0738332314150674
Total validation loss =  0.09040017638887678
Epoch:  25
Training
valence_loss: 0.040925220755690876, energy_loss: 0.01522450990424576, danceability_loss: 0.014588406203653325
Total training loss =  0.02334358665922826
Validation
valence_loss: 0.15403026448828833, energy_loss: 0.08664798896227564, danceability_loss: 0.07707493273275239
Total validation loss =  0.10485855915716716
Epoch:  30
Training
valence_loss: 0.013830931941893969, energy_loss: 0.01042722389948639, danceability_loss: 0.011168158623728563
Total training loss =  0.011690684319050475
Validation
valence_loss: 0.091282136206116, energy_loss: 0.08776070975831576, danceability_loss: 0.0682976735489709
Total validation loss =  0.08162237757018634
```

c) Μέσω αυτής της τεχνικής, τα τελικά αποτελέσματα που παίρνουμε για την spearman συσχέτιση και για τις τρεις επισημειώσεις είναι :

```
The Spearman correlation between the ground true values and the predicted values for valence is
 0.1741903900675711
The Spearman correlation between the ground true values and the predicted values for energy is
 0.5485611146083195
The Spearman correlation between the ground true values and the predicted values for danceability is
 0.3660426744978142
```

Παρατηρούμε ότι σε γενικές γραμμές έχουμε παρόμοια αποτελέσματα με το βήμα 8 (συγκεκριμένα το valence και το energy είναι μειωμένα, ενώ το danceability αυξημένο). Έτσι, συμπεραίνουμε ότι δημιουργείται ένας μέσος όρος και αντισταθμίζονται τα αποτελέσματα μεταξύ τους, καθώς προσπαθούν και τα 3 ξεχωριστά tasks να βελτιώσουν ταυτόχρονα την επίδοση τους. Καλύτερη βελτίωση θα μπορούσαμε να πετύχουμε είτε με αύξηση των εποχών εκπαίδευσης είτε με αλλαγή των βαρών στα κόστη για να πετύχουμε καλύτερη συσχέτιση των επισημειώσεων στην ίδια τάξη μεγέθους. Γενικότερα, η τεχνική του multitask learning αποδεικνύεται ιδιαίτερα ικανοποιητική σε περιπτώσεις που έχουμε tasks που θεωρητικά είναι κοινά μεταξύ τους (μοιράζονται κάποια κοινά low-level features) και έχουμε ίδιο dataset εκπαίδευσης.