Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών

Αναγνώριση Προτύπων

$9^ο$ Εξάμηνο - Ροή Σ

Τρίτη Εργαστηριακή Αναφορά

Αναγνώριση Είδους και Εξαγωγή
Συναισθήματος από Μουσική

Δημήτρης Δήμος - 031 17 165
dimitris.dimos647@gmail.com

Κωνσταντίνος Κοψίνης - 031 17 062
kon.kopsinis@gmail.com

Αθήνα
Φεβρουάριος, 2022

Περιεχόμενα

Περιγραφή Εργαστηρίου

Σκοπός της άσκησης είναι η αναγνώριση του είδους και η εξαγωγή συναισθηματικών διαστάσεων από φασματογραφήματα (spectrograms) μουσικών κομματιών. Χρησιμοποιούμε δύο σύνολα δεδομένων, το Free Music Archive (FMA) genre με 3834 δείγματα χωρισμένα σε 20 κλάσεις (είδη μουσικής) και τη βάση δεδομένων (dataset) multitask music με 1497 δείγματα με επισημειώσεις (labels) για τις τιμές συναισθηματικών διαστάσεων όπως valence, energy και danceability. Τα δείγματα είναι spectrograms, τα οποία έχουν εξαχθεί από clips 30 δευτερολέπτων από διαφορετικά τραγούδια.

Συνοπτικά, ασχολούμαστε με την ανάλυση των spectrograms με χρήση βαθιών αρχιτεκτονικών με CNNs και RNNs. Τα πέντε μέρη στα οποία χωρίζεται η άσκηση είναι:

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.
4. Χρήση προηγμένων τεχνικών εκπαίδευσης (transfer - multitask) learning για τη βελτίωση των αποτελεσμάτων.
5. Υποβολή των μοντέλων στο Kaggle competition του εργαστηρίου και σύγκριση των αποτελεσμάτων

Τα δεδομένα που θα χρησιμοποιήσουμε είναι διαθέσιμα στο [1]

$$\left[\text{Τα πρώτα έξι (6) βήματα αποτελούν την προπαρασκευή του εργαστηρίου}\right]$$

Βήμα 0. Εξοικείωση με Kaggle Kernels

Σε αυτό το βήμα εξοικειωνόμαστε με την πλατφόρμα Kaggle. Ουσιαστικά, εξερευνούμε τους τρόπους με τους οποίους ανοίγουμε private kernels, φορτώνουμε δεδομένα, περιηγούμαστε στους υποφακέλους και δοκιμάζουμε να απ/ενεργοποιήσουμε τη διαθέσιμη GPU κάνοντας commit τις αλλαγές μας.

Βήμα 1. Εξοικείωση με φασματογραφήματα στην κλίμακα mel

Τα δεδομένα που χρησιμοποιούμε 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.

Επιλέγουμε τυχαία δύο αρχεία από το σύνολο δεδομένων και απεικονίζουμε τα φασματογραφήματά τους στην κλίμακα mel.

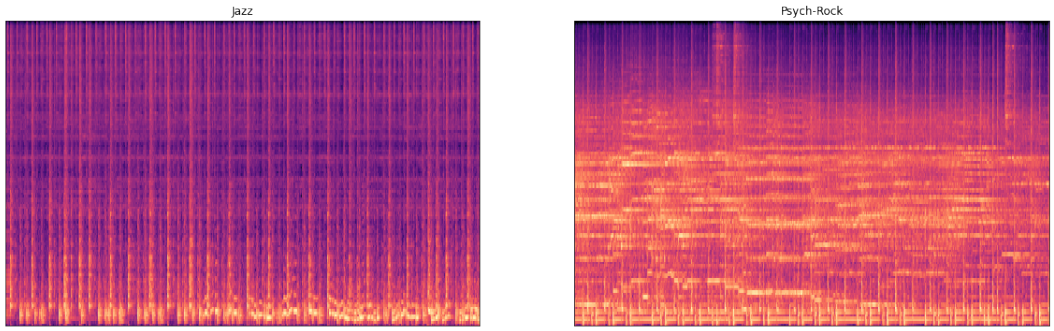


Figure 1: Mel spectrograms of a jazz and a psych-rock sample

Εν γένει ένα spectrogram προκύπτει εφαρμόζοντας τον μετασχηματισμό STFT ή, αλλιώς, Gabor σε ένα χρονικό σήμα εισόδου. Το spectrogram έχει στον οριζόντιο άξονα τον χρόνο (τα χρονικά βήματα ή παράθυρα που χρησιμοποιήθηκαν κατά τον μετασχηματισμό) και στον κατακόρυφο τη συχνότητα. Το χρώμα που βλέπουμε αντιστοιχεί στο πλάτος. Συνοπτικά, ένα spectrogram απεικονίζει το κατά πόσο υπάρχει μια συχνοτική συνιστώσα σε ένα χρονικό παράθυρο. Έχει πρακτική εφαρμογή σε πραγματικά σήματα, όπως ένα σήμα φωνής ή μουσικής, με δεδομένο οτι σε αυτά τα σήματα συνυπάρχουν πολλές συχνοτικές συνιστώσες οι οποίες μεταβάλλονται με το πέρασμα του χρόνου.

Παρατηρούμε στα spectrograms που απεικονίζουμε (τα οποία αντιστοιχούν σε κομμάτια διαφορετικού είδους) ότι στο Jazz κομμάτι επικρατούν γενικά σε όλη τη διάρκεια του χρόνου χαμηλές συχνότητες, ενώ στο psych-rock κομμάτι υπάρχουν επικρατούσες συχνότητες σε ένα μεγάλο εύρος. Σκεφτόμαστε, λοιπόν, ότι η διαφορά αυτή στο χρονοσυχνοτικό περιεχόμενο μπορεί να είναι ένα καλό διαχωριστικό κριτήριο ανάμεσα στα είδη μουσικής. Διαισθητικά, δηλαδή, θα περιμέναμε τα Jazz κομμάτια να έχουν γενικά χαμηλές συχνότητες, ενώ τα psych-rock ένα ευρύ φάσμα. Αν η υπόθεσή μας ισχύει, τότε αυτά τα χρονοσυχνοτικά μοτίβα μπορούν να αποτελέσουν ένα είδος ταυτότητας για το κάθε είδος και κατ' επέκταση να συντελέσουν στην εκπαίδευση μοντέλων που μπορούν να κάνουν την εν λόγω ταξινόμηση.

Βήμα 2. Συγχρονισμός φασματογραφημάτων στον ρυθμό της μουσικής

Οι διαστάσεις των φασματογραφημάτων του βήματος 1 είναι: (128×1293) και (128×1291), αντίστοιχα. Τα χρονικά βήματα είναι η δεύτερη διάσταση.

Τα δεδομένα μας έχουν πολύ μεγάλη διάσταση time steps. Περιμένουμε, για αυτόν τον λόγο, τα μοντέλα LSTM που θέλουμε να εκπαιδεύσουμε σε αυτά να μην είναι αποδοτικά, πρώτα και κύρια, επειδή θα χρειάζονται πάρα πολύ χρόνο για να εσωτερικεύσουν την υπερμεγέθη αυτή πληροφορία. Ταυτόχρονα, ο μεγάλος αριθμός από features που έχει το κάθε δείγμα θα έχει ως πιθανή συνέπεια το vanishing ή το explosion του gradient, λόγω του μεγάλου βάθους που θα έχει το νευρωνικό δίκτυο προς εκπαίδευση.

Ένας τρόπος να μειώσουμε τα χρονικά βήματα είναι να συγχρονίσουμε τα φασματογραφήματα πάνω στον ρυθμό. Για αυτόν τον λόγο, παίρνουμε τη διάμεσο ανάμεσα στα σημεία που χρυπάει το beat της μουσικής. Επαναλαμβάνοντας, λοιπόν, τα προηγούμενα βήματα, έχουμε τα ακόλουθα αποτελέσματα:

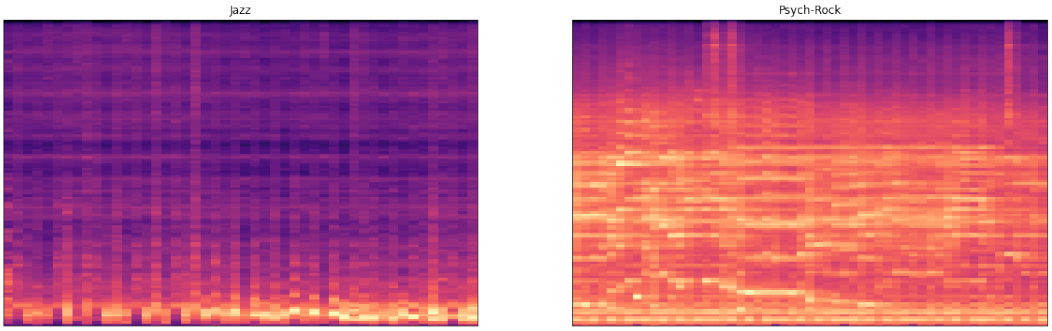


Figure 2: Beat-synced Mel spectrograms of a jazz and a psych-rock sample

Ο νέος αριθμός time steps είναι 48 και 55, αντίστοιχα.

Παρατηρούμε ότι κάνοντας την παραπάνω επεξεργασία τα τελικά φασματογραφηματα αποτελούνται πρώτον από σημαντικά λιγότερα time steps, πράγμα που θα βελτιώσει σημαντικά την χρονική απόδοση των εκπαιδευόμενων μοντέλων και θα ελαττώσει την πιθανότητα του vanishing/exploding gradient, χωρίς ταυτόχρονα να αλλοιώνει την χρονοσυχνοτική πληροφορία πολύ. Είναι φανερό ότι το Jazz κομμάτι διατηρεί τις χαμηλές συχνότητές του ενώ το psych-rock κομμάτι εξακολουθεί να έχει ευρύ φάσμα. Τα δύο spectrograms διατηρούν τον ξεχωριστό χαρακτήρα τους.

Βήμα 3. Εξοικείωση με χρωμογραφήματα

Τα χρωμογραφήματα (chromagrams) απεικονίζουν την ενέργεια του σήματος μουσικής για τις ζώνες συχνοτήτων που αντιστοιχούν στις δώδεκα διαφορετικές νότες της κλίμακας κλασσικής μουσικής 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 (μπορεί να θεωρηθεί ότι το χρωμογράφημα είναι ένα spectrogram modulo την οκτάβα).

Επαναλαμβάνουμε τα παραπάνω βήματα για τα χρωμογραφήματα των αντίστοιχων κομματιών με πριν.

Τα time steps των αντίστοιχων chromagrams είναι ίδια με αυτά των αντίστοιχων spectrograms (1293 και 1291), και οι συχνοτικές ζώνες είναι 12. Μετά το beat-syncing, τα timesteps γίνονται 48 και 55, αντίστοιχα.

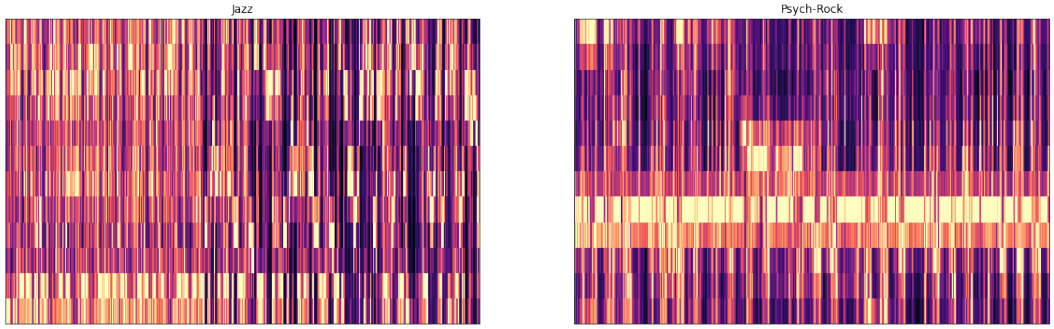


Figure 3: Chromagrams of a Jazz and a psych-rock sample

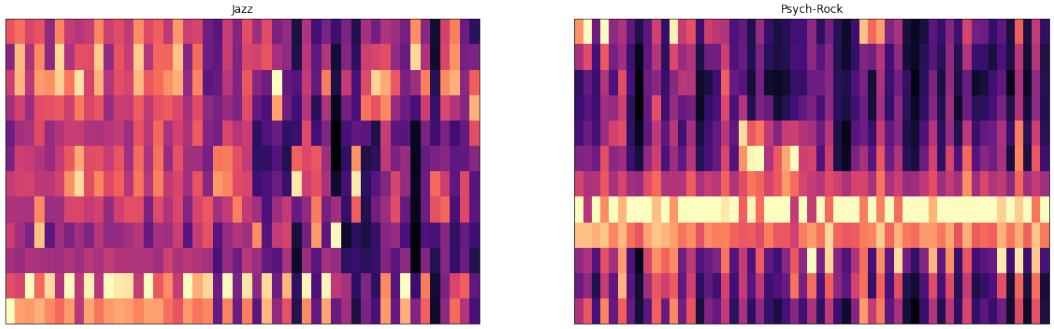


Figure 4: Beat-synced Chromagrams of a Jazz and a psych-rock sample

Για τα beat-synced chromagrams συνάγουμε ανάλογα συμπεράσματα με τα προηγούμενα συμπεράσματα των beat-synced spectrograms.

Βήμα 4. Φόρτωση και ανάλυση δεδομένων

Χρησιμοποιώντας τον έτοιμο κώδικα που δίνεται, φορτώνουμε τα δεδομένα μας με σκοπό να τα χρησιμοποιήσουμε ως train, validation και test sets στην εκπαίδευση μοντέλων LSTM.

Εν ολίγοις, ο κώδικας που δίνεται εκτελεί τις εξής λειτουργίες:

- κατασκευάζει ένα dataset συμβατό με το pytorch framework, με χρήση συναρτήσεων που επιτελούν τα παρακάτω
- χωρίζει τα δεδομένα σε train και validation sets
- διαβάσει τα spectrograms, chromagrams και fused data
- κωδικοποιεί τα labels σε αριθμούς
- κανει zero-pad στα δεδομένα εισόδου, ώστε να έχουν ολα το ίδιο μήκος

Προτού προχωρήσουμε στην εκπαίδευση κάνουμε μια επιπλέον τροποποίηση του dataset. Συγχωνεύουμε στην ίδια κλάσση δεδομένα που ανοίκουν σε παρεμφερές μουσικό είδος. Έτσι διευκολύνουμε τα μοντέλα μας, καθώς η διάκριση μεταξύ ενός κομματιού psych-rock από ένα κλασσικό rock θα ήταν εν γένει δύσκολη, με δεδομένο ότι τα είδη μοιάζουν πολύ. Άλλωστε, είναι ένα καλό πρώτο βήμα να μπορούμε να εκπαιδεύσουμε ένα μοντέλο στις πιο σημαντικές κλάσσεις (όπως η rock και η Jazz), παρά σε λιγότερο σημαντικές (υπό την έννοια του ότι είναι πολύ εξειδικευμένες ως προς το είδος τους, όπως η Punk).

Επιπλέον, αφαιρούμε underrepresented δεδομένα, δηλαδή κομμάτια που ανήκουν σε κλάσεις με λίγα δεδομένα, καθώς η αναγνώρισή τους θα περιμέναμε να ήταν σχεδόν αδύνατη, όταν έχουμε ένα σύνολο 3834 δειγμάτων εκ των οποίων ελάχιστα αντιπροσωπεύουν την εν λόγω κλάσση.

Η κατανομή των δεδομένων σε κλάσσεις μπορεί αν φανέι στα παρακάτω ιστογράμματα πριν και μετά την τροποποίησή μας:

Παρατηρούμε ότι τα περισσότερα δεδομένα στο τέλος ανήκουν στην κλάσση υπ' αριθμόν 8. Μια (ίσως αφελής) πρόβλεψη είναι ότι με τα τωρινά δεδομένα, το μοντέλο θα έχει την τάση να προβλέπει συχνά την κλάσση 8, αφού τα δεδομένα μας δεν είναι όσα θα θέλαμε, αλλά ούτε είναι ισοκατανεμημένα.

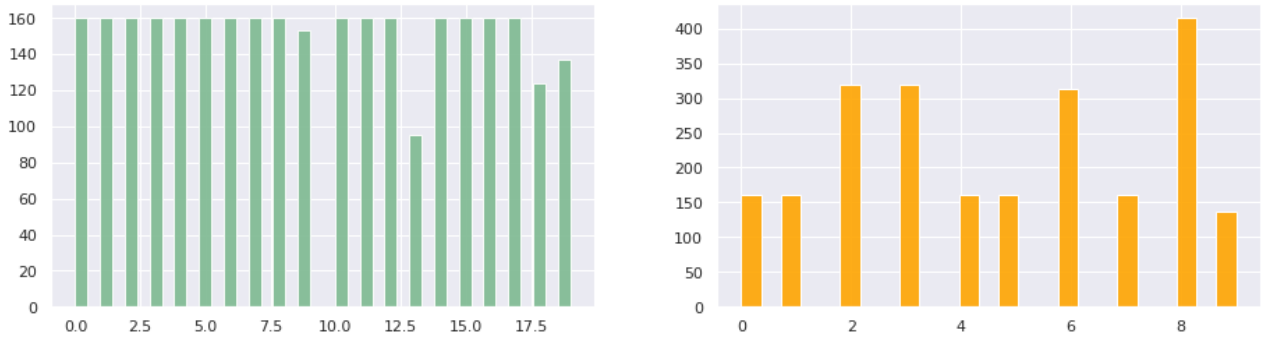


Figure 5: Before - After Histograms

Βήμα 5. Αναγνώριση μουσικού είδους με LSTM

Τροποποιούμε την υλοποίηση του **bidirectional LSTM** μοντέλου από το εργαστήριο 2 ώστε να δέχεται ως είσοδο τα spectrograms. Το μοντέλο μας αποτελείται από **ένα hidden layer** που περιέχει **32 νευρώνες**.

Αρχικά, δοκιμάζουμε αν το μοντέλο μας μπορεί να πάθει overfit, εσκεμμένα ώστε να κάνουμε debugging και να ελέγξουμε την εκπαιδευσιμότητα του μοντέλου. Εκπαιδεύουμε, λοιπόν, το μοντέλο μας για 8000 εποχές σε ένα batch με 32 samples και, όντως, το μοντέλο υπερεκπαιδεύεται και παθαίνει overfit, με το κριτήριο κόστους cross entropy να φτάνει στο 0 (πρακτικά πολύ κοντά στο μηδέν, έχει τιμή της τάξης του 10^{-3})

Στη συνέχεια, έχοντας διαπιστώσει ότι η αρχιτεκτονική μας είναι ευσταθής εκπαιδεύουμε τα παρακάτω τέσσερα μοντέλα. Εκπαιδεύουμε για **80 εποχές** με **batch size ίσο με 30** και χρησιμοποιούμε **ρυθμό μάθησης ίσο με 10^{-3}**. Ακόμη, ως συνάρτηση κόστους ορίζεται η **Cross Entropy Loss**. Στην αρχή, εκπαιδεύουμε τα μοντέλα χωρίς Early Stopping.

- ένα εκπαιδευμένο στα non-beat-synced spectrograms
- ένα εκπαιδευμένο στα beat-synced spectrograms
- ένα εκπαιδευμένο στα non-beat-synced chromagrams (160 εποχές)
- ένα εκπαιδευμένο στα non-beat-synced fused data

Figure 6: Left: Trained on non-beat-synced mel spectrograms, Right: Trained on beat-synced spectrograms

Figure 7: Left: Trained on non-beat-synced chromagrams, Right: Trained on non-beat-synced fused spectrograms

Μέχρι στιγμής παρατηρούμε ότι στο μοντέλο που εκπαιδεύεται στα chromagrams δεν μειώνει το Validation Loss, όπως στα υπόλοιπα. Αυτό είναι μια αρχική ένδειξη για το ότι τα chromagrams δεν είναι από μόνα τους τόσο περιεκτικά σε διακριτική πληροφορία σε σχέση με τα spectrograms.

Επιπλέον, τα μοντέλα που εκπαιδεύονται σε non-beat-synced data απαιτούν υπερβολικά μεγάλο πραγματικό χρόνο για να εκπαιδευτούν και, εν τέλει τα αποτελέσματα δεν ειναι όσο καλά θα θέλαμε. Σχολιάζουμε εκτενέστερα την απόδοση των μοντέλων στο επόμενο βήμα.

Βήμα 6. Αξιολόγηση των μοντέλων

- **Accuracy.** Πρόκειται για μια από τις πιο συνήθεις και γενικές μετρικές απόδοσης. Είναι το ποσοστό των σωστά ταξινομημένων δειγμάτων:

$$accuracy = \frac{TP + TN}{ALL\ DATA}$$

- **Precision.** Η μετρική αυτή ορίζεται ως:

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

- **Recall.** Η μετρική αυτή ορίζεται ως:

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

  Ανάλογα με το αν αποδίδεται μεγαλύτερη σημαντικότητα στα False Negatives (περιπτώσεις λανθασμένης διάγνωσης, για παράδειγμα) ή στα False Positives, προτιμάται το Recall σαν μετρική ή το Precision, αντίστοιχα. Εν γένει, υπάρχουν περιπτώσεις στις οποίες αύξηση του precision οδηγεί σε μείωση του recall και αντίστροφα.

- **f1_score.** Η μετρική αυτή ορίζεται ως:

$$f1\_score = 1 \cdot \frac{recall \cdot precision}{recall + precision}$$

  Η f1_score προκύπτει από τον συνδυασμό των precision και recall, τα οποία εν γένει επιθυμούμε να είναι υψηλά.

Γενικά, αυτές οι μετρικές απόδοσης υπόκεινται σε ελαττώματα του εκάστοτε προβλήματος και κάποιες φορές οδηγούν σε παραπλανητικά συμπεράσματα. Κάτι τέτοιο θα μπορούσε να συμβεί στην περίπτωση ενός προβλήματος με unbalanced dataset, κατά την οποία η μετρική του accuracy ενδέχεται να είναι μικρή αφού δε λαμβάνει υπόψη τις underrepresented κλάσεις.

Σε μια περίπτωση σαν την ανωτέρω, η μετρική του f1 score είναι πιο σωστή.

Οι macro-averaged metrics είναι γενίκευση των παραπάνω μετρικών σε περισσότερες των δύο κλάσεων. Είναι η μέση τιμή των αντίστοιχων μετρικών κλάσεων και θεωρούν πως κάθε κλάση έχει ίση βαρύτητα με τις υπόλοιπες. Αντίθετα, οι weighted-average μετρικές θεωρούν πως το βάρος για κάθε κλάση εξαρτάται από το πλήθος των δειγμάτων που αυτή περιέχει.

Οι micro-averaged metrics είναι οι ίδιες μετρικές με τις κλασσικές, θεωρώντας ως positive τα δείγματα που ανήκουν σε μια κλάσση και negative όλα τα υπόλοιπα.

Κάθε μετρική είναι περισσότερη χρήσιμη από τις άλλες ανάλογα το πρόβλημα που μας ενδιαφέρει. Ένα παράδειγμα σχετικά με την χρησιμότητα του precision έναντι του recall αναφέρθηκε παραπάνω. Το accuracy είναι γενικά καλή μετρική όταν έχουμε banaced dataset.

Στις περιπτώσεις unbalanced datasets, συχνά παρατηρείται μεγάλη απόκλιση ανάμεσα σε accuracy και f1-score, όπου η δεύτερη είναι πιο χρήσιμη και σωστή.

Αντίστοιχα, οι μετρικές macro και micro f1-scores μπορούν να παρουσιάζουν αποκκλίσεις σε unbalanced datasets, καθώς η πρώτη αντιμετωπίζει ισάξια όλες τις κλάσεις, λαμβάνοντας έτσι σοβαρά υπόψη και αυτές με τα λιγότερα δείγματα, ενώ η δεύτερη ενδιαφέρεται μόνο για το συνολικό αριθμό ορθών και εσφαλμένων ταξινομήσεων σε ολόκληρο το dataset.

Συγκεκριμένα για το πρόβλημά μας, εφόσον χρησιμοποιήσαμε class mapping, καταφέρνουμε να εξισόρροπήσουμε σημαντικά το dataset μας. Συνεπώς, η μετρική του accuracy είναι ικανοποιητικά αντιπροσωπετυτική της ικανότητας του μοντέλου μας να γενικεύει. Η weighted average F1-score είναι, επίσης, αξιόπιστη αφού δεν δίνεται έμφαση στο είδος των εσφαλμένων ταξινομήσεων.

Μετά το πέρας της εκπαίδευσης των τεσσάρων μοντέλων, εξάγουμε το classification report με βάσει το test set. Τα αποτελέσματα φαίνονται παρακάτω για κάθε μοντέλο:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	40
1	0.00	0.00	0.00	40
2	0.17	0.72	0.28	80
3	0.50	0.01	0.02	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	0.26	0.60	0.37	103
9	0.00	0.00	0.00	34
accuracy			0.21	575
macro avg	0.09	0.13	0.07	575
weighted avg	0.14	0.21	0.11	575

(a) Model Trained on Spectrograms

	precision	recall	f1-score	support
0	0.00	0.00	0.00	40
1	0.00	0.00	0.00	40
2	0.30	0.56	0.39	80
3	0.23	0.70	0.35	80
4	0.00	0.00	0.00	40
5	0.00	0.00	0.00	40
6	0.39	0.58	0.46	78
7	0.00	0.00	0.00	40
8	0.24	0.11	0.15	103
9	0.00	0.00	0.00	34
accuracy			0.27	575
macro avg	0.12	0.19	0.13	575
weighted avg	0.17	0.27	0.19	575

(b) Model Trained on beat-synced Spectrograms

	precision	recall	f1-score	support
0	0.04	0.03	0.03	40
1	0.16	0.15	0.16	40
2	0.15	0.23	0.18	80
3	0.22	0.29	0.25	80
4	0.15	0.10	0.12	40
5	0.09	0.07	0.08	40
6	0.23	0.24	0.24	78
7	0.06	0.03	0.03	40
8	0.26	0.24	0.25	103
9	0.12	0.12	0.12	34
accuracy			0.18	575
macro avg	0.15	0.15	0.15	575
weighted avg	0.17	0.18	0.17	575

(c) Model Trained on Chromagrams

	precision	recall	f1-score	support
0	0.00	0.00	0.00	40
1	0.00	0.00	0.00	40
2	0.20	0.70	0.31	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	0.26	0.73	0.38	103
9	0.00	0.00	0.00	34
accuracy			0.23	575
macro avg	0.05	0.14	0.07	575
weighted avg	0.07	0.23	0.11	575

(d) Model Trained on fused data

Figure 8: Model Efficiency Stats

Όπως περιμέναμε, το μοντέλο που εκπαιδεύτηκε στα chromagrams δεν αποδίδει καλά, καθώς πετυχαίνει τα χαμηλότερα efficiency stats σε σχέση με τα υπόλοιπα μοντέλα. Το καλύτερο είναι μοντέλο που εκαπιδεύτηκε στα beat-synced-spectrograms με ποσοστό ευστοχίας ίσο με **27%**.

Συνολικά, συμπεραίνουμε ότι το LSTM μας, ανεξαρτήτως εισόδου εκπαίδευσης, δεν είναι αρκετά ικανοποιητικό. Το γεγονός αυτό οφείλεται καταρχάς στην αρχιτεκτονική του μοντέλου, η οποία είναι εν γένει δοκιμαστική. Αν καταστρώναμε ένα διαφορετικό LSTM τα αποτελέσματα ενδέχεται να ήταν και καλύτερα. Ωστόσο, αυτό θα είχε

κόστος ως προς τον πραγματικό χρόνο που απαιτείται για την εκπαίδευση, ενώ αυξάνεται η πιθανότητα να συμβεί vanishing/exploding gradient. Και στην περίπτωση του δικού μας μοντέλου, είναι αρκετά πιθανό να παρουσιάζεται αυτό το φαινόμενο και το μοντέλο να καταλήγει να μη "μαθαίνει" όσα θα θέλαμε. Ένα ακόμα πιθανό πρόβλημα μπορεί να είναι το γεγονός ότι το dataset μας δεν είναι αρκετά μεγάλο. Το πρόβλημα της ταξινόμησης μουσικού είδους είναι ένα από τα κατεξοχήν δύσκολα προβλήματα και θα μας εξέπληττε αν είχαμε πολύ καλά αποτελέσματα με ένα απλό LSTM. Στη συνέχεια, δοκιμάζουμε να εκπαιδεύσουμε το μοντέλο ξανά στα ίδια training datasets με πριν, χρησιμοποιώντας **early stopping**. Το μοντέλο που εκπαιδεύτηκε στα beat-synced mel spectrograms δεν παρουσίασε πρόωρο τερματισμό εκπαίδευσης.

Figure 9: Left: Trained on mel spectrograms

Figure 10: Left: Trained on non-beat-synced chromagrams, Right: Trained on non-beat-synced fused spectrograms

	precision	recall	f1-score	support
0	0.00	0.00	0.00	40
1	0.00	0.00	0.00	40
2	0.20	0.53	0.29	80
3	0.23	0.51	0.32	80
4	0.00	0.00	0.00	40
5	0.00	0.00	0.00	40
6	0.31	0.33	0.32	78
7	0.00	0.00	0.00	40
8	0.26	0.26	0.26	103
9	0.00	0.00	0.00	34
accuracy			0.24	575
macro avg	0.10	0.16	0.12	575
weighted avg	0.15	0.24	0.18	575

Figure 11: Left: Trained on mel spectrograms

	precision	recall	f1-score	support
0	0.00	0.00	0.00	40
1	0.08	0.03	0.04	40
2	0.21	0.26	0.24	80
3	0.22	0.53	0.31	80
4	0.14	0.03	0.04	40
5	0.17	0.03	0.04	40
6	0.26	0.55	0.36	78
7	0.00	0.00	0.00	40
8	0.15	0.15	0.15	103
9	0.00	0.00	0.00	34
accuracy			0.22	575
macro avg	0.12	0.16	0.12	575
weighted avg	0.15	0.22	0.16	575

	precision	recall	f1-score	support
0	0.00	0.00	0.00	40
1	0.00	0.00	0.00	40
2	0.24	0.69	0.36	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	0.25	0.84	0.38	103
9	0.00	0.00	0.00	34
accuracy			0.25	575
macro avg	0.05	0.15	0.07	575
weighted avg	0.08	0.25	0.12	575

Figure 12: Left: Trained on non-beat-synced chromagrams, Right: Trained on non-beat-synced fused spectrograms

Παρατηρούμε πως τα μοντέλα που εκπαιδεύουμε με early stopping έχουν καλύτερο accuracy σε σχέση με τα πρώτα. Σαν εμφανές συμπέρασμα προκύπτει πως η δομή του LSTM μας είναι τέτοια που μπορεί να μάθει περισσότερη χρήσιμη (γενικεύσιμη) πληροφορία από το συγκεκριμένο dataset με λιγότερες εποχές.

Βήμα 7. 2D CNN

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.

Το δίκτυό μας αποτελείται από δύο επίπεδα, καθένα από τα οποία επιμερίζεται σε ένα συνελικτικό layer, μια ενεργοποίηση relu και ένα max pooling layer. Η έξοδος αποτελείται από ένα fully connected layer που καταλήγει σε μια softmax. Απεικονίζουμε τα παρακάτω για λόγους οπτικοποίησης και σχολιάζουμε στη συνέχεια:

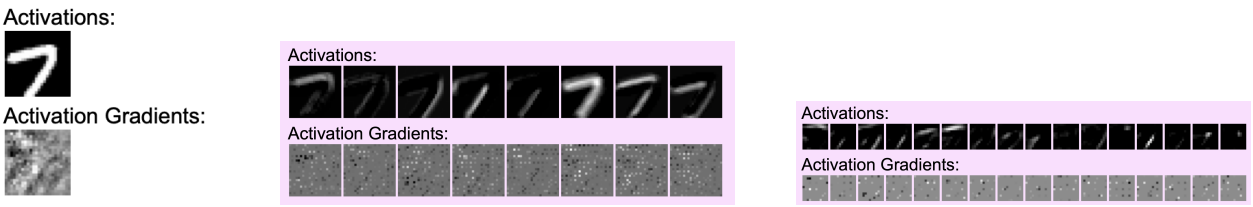


Figure 13: Input. Activations and Activation Gradients of the first two ReLUs

Figure 14: Activations and Activation gradients of FC layers. Softmax Output

Στο παράδειγμα για το οποίο παραθέτουμε τις εικόνες βλέπουμε τι επεξεργασία υφίσταται η εικόνα ενός "7" σε επιμέρους επίπεδα. Στο πρώτο επίπεδο, η εικόνα περνά από το συνελικτικό layer κι, εν συνεχεία, από τη συνάρτηση ενεργοποίησης relu. Πρακτικά, η εικόνα εισόδου συνελίσσεται με τους 8 πυρήνες και κάθε έξοδος περνά από τη συνάρτηση $max(0, x)$. 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 (αναλυτικότερα για τη λειτουργία ενός max pooling layer στη συνέχεια). 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, αυτά τα activations μειώνουν τη διαστατικότητά τους περνώτας από το max pooling layer και μετατρέπονται σε ένα flattened vector. Το vector αυτό περνά από το fully connected layer της εξόδου. Στα activations της εξόδου παρατηρούμε 10 τετραγωνίδια γκρι χρώματος. Όσο πιο λευκό είναι το τετραγωνίδιο τόσο μεγαλύτερη και η a-posteriori πιθανότητα που αποδίδει το νευρωνικό στο να ανήκει το ψηφίο εισόδου στην αντίστοιχη κλάση. Στην περίπτωσή μας, η θέση του ψηφίου 7 είναι λευκή (το νευρωνικό έκρινε μεγάλη την πιθανότητα η είσοδος να ήταν ένα "7"). Στο τελικό activation όλα τα τετραγωνίδια είναι μαύρα πλην της τελικής πρόβλεψης του νευρωνικού. Σωστά, το μοντέλο έκρινε ότι η είσοδος ανήκει στην κλάσση των "7".

Στη συνέχεια, κατασκευάζουμε κι εκπαιδεύουμε στα φασματογραφήματα (αντιμετωπίζοντάς τα σαν μονοκάναλη εικόνα) του FMA genre dataset, ένα custom CNN με τα εξής χαρακτηριστικά:

- 4 συνελικτικά επίπεδα κι ένα fully connected layer εξόδου
- κάθε επίπεδο αποτελείται από τα εξής 4 layers:
  - 2D convolution
  - batch normalization
  - ReLU activation
  - Max pooling
- Επίπεδο 1:
  - κανάλια εισόδου convLayer: 1, κανάλια εξόδου: 32, διάσταση πυρήνων: (3×3), stride: 1, pad: 1
  - Max pooling layer. Διάσταση πυρήνα: (2×2)
- Επίπεδο 2:
  - κανάλια εισόδου convLayer: 32, κανάλια εξόδου: 64, διάσταση πυρήνων: (3×3), stride: 1, pad: 1
  - Max pooling layer. Διάσταση πυρήνα: (2×2)
- Επίπεδο 3:
  - κανάλια εισόδου convLayer: 64, κανάλια εξόδου: 128, διάσταση πυρήνων: (3×3), stride: 1, pad: 1
  - Max pooling layer. Διάσταση πυρήνα: (4×4)
- Επίπεδο 4:
  - κανάλια εισόδου convLayer: 128, κανάλια εξόδου: 256, διάσταση πυρήνων: (4×4), stride: 1, pad: 1
  - Max pooling layer. Διάσταση πυρήνα: (4×4)
- Για το fully connected layer. Διάσταση: (10240×1)

Σε αυτό το σημείο εξηγούμε τη λειτουργία και τον ρόλο των επιμέρους layers κάθε επιπέδου του νευρωνικού μας δικτύου. Για αυτό βασιζόμαστε στις αναφορές [3], [4] και [5].

- **Convolution**: Η συνέλιξη είναι το βασικό συστατικό ενός CNN. Ένα convolutional layer αποτελείται από έναν σταθερό αριθμό από φίλτρα (πυρήνες) συνέλιξης και υλοποιεί την συνέλιξη της εισόδου (εν γένει πολυκάναλη εικόνα, αλλά όχι αποκλειστικά) με κάθε ένα από αυτά τα φίλτρα. Οι έξοδοι (που ονομάζονται και feature maps) των επιμέρους συνελίξεων "συνενώνονται" στην τρίτη διάσταση για να παράξουν μια νέα πολυκάναλη εικόνα. Περιγράφοντας τη διαδικασία της συνέλιξης σε ένα lower level, 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 διαδικασία οπτικοποιείται παρακάτω:

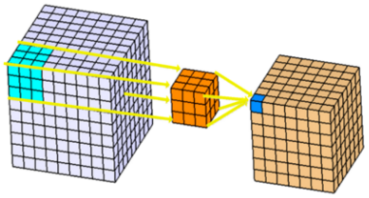


Figure 15: 3D Convolution Visualization

Εν γένει, η έξοδος της συνέλιξης έχει μικρότερες διαστάσεις από αυτές τις εισόδου. Γι' αυτό, αν επιθυμούμε να διατηρούνται, "συμπληρώνουμε" με μηδενικά, κάνοντας το γνωστό zero-padding πριν την εφαρμογή της συνέλιξης.
Τα βάρη των φίλτρων συνέλιξης είναι ένα από τα εκπαιδεύσιμα συστατικά ενός CNN.

- **Batch Normalization**: Πρόκειται για μια τεχνική που επιφέρει έναν αριθμό από πλεονεκτήματα για την εκπαίδευση Deep Neural Networks. Πρακτικά, το BatchNorm κανονικοποιεί τα δεδομένα (ένα batch), ώστε να έχουν zero mean και unit variance. Αυτό, φυσικά, το κάνει σε κάθε επίπεδο που εφαρμόζεται το BatchNorm. Κατ' αυτόν τον τρόπο, μειώνονται οι εποχές που χρειάζεται το δίκτυο για να εκπαιδευτεί και αυξάνεται η ικανότητά του στο να γενικεύει. Αυτό συμβαίνει επειδή αντιμετωπίζει το "internal covariate shift" και ομαλοποιεί τη χρησιμοποιούμενη συνάρτηση βελτιστοποίησης.

- **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. Αν το CNN μεγαλώσει σε μέγεθος, το υπολογιστικό κόστος της προσθήκης επιπλέον ReLUs αυξάνεται γραμμικά [6].

- **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 ενα feature map ξεχωριστά και παράγει ένα νέο feature map του οποίου κάθε pixel είναι το μέγιστο στοιχείο του αντίστοιχου παραθύρου του αρχικού feature map. Η διαδικασία οπτικοποιείται στην εικόνα:

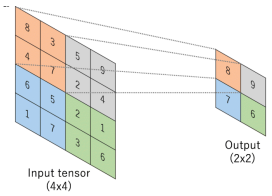


Figure 16: Max Pooling Visualization

Όπως είναι φυσικό, οι παράμετροι του max pooling πυρήνα (δηλαδή μόνο οι διαστάσεις του sliding παραθύρου) δεν είναι εκπαιδεύσιμοι. Η χρήση του max pooling συντελεί στην μείωση του overfitting παρέχοντας πιο αφηρημένες μορφές αναπαραστάσεων. Ταυτόχρονα, μειώνει το υπολογιστικό κόστος με το να ελαττώνει τον αριθμο των προς εκμάθηση παραμέτρων και παρέχει ένα βασικό "translation invariance" στην εσωτερική αναπαράσταση [7].

Υπάρχουν, επίσης, τα average και general pooling.

Προχωρώντας στο νευρωνικό μας, για αρχή πραγματοποιούμε overfit_batch για να διαβεβαιώσουμε ότι το μοντέλο μας είναι free of bugs και trainable. Ύστερα, το εκπαιδεύουμε στα non-beat-synced spectrograms. Χρησιμοποιήθηκαν:

- Cross Entropy Loss
- Adam Optimizer

- learning rate: 10^{-3}
- weight decay: 10^{-4}
- batch size: 32
- Early Stopping (τερματισμός στις 10 εποχές)

Το classification report για το νευρωνικό μετά την εκπαίδευσή του δίνεται παρακάτω:

	precision	recall	f1-score	support
0	0.50	0.05	0.09	40
1	0.59	0.60	0.59	40
2	0.58	0.55	0.56	80
3	0.35	0.75	0.48	80
4	0.53	0.23	0.32	40
5	0.18	0.25	0.21	40
6	0.58	0.53	0.55	78
7	0.07	0.05	0.06	40
8	0.39	0.29	0.34	103
9	0.24	0.24	0.24	34
accuracy			0.40	575
macro avg	0.40	0.35	0.34	575
weighted avg	0.42	0.40	0.38	575

Figure 17: CNN trained on non-beat-synced mel spectrograms: Classification Report

Σύμφωνα με το ανωτέρω report, το CNN μας πετυχαίνει accuracy ίσο με **40%** και ξεπερνά την απόδοση του βέλτιστου LSTM μας. Η υπεροχή του CNN στο παρόν task είναι εμφανής, όχι μόνο ως προς την τιμή των μετρικών απόδοσης, αλλά και ως προς (1) τον πραγματικό χρόνο εκπαίδευσης και (2) την δυνατότητα εκμετάλλευσης όλων των features κάθε sample (δεν χρειάζεται να χρησιμοποιήσουμε τα beat-synced data, καθώς οι δοκιμές μας στην ίδια αρχιτεκτονική δε φανέρωσαν καμία υπεροχή αυτής τους έναντι των non-beat synced, ενώ δεν κινδυνεύουμε από vanishing/exploding gradient λόγω μεγάλου βάθους).

Βήμα 8. Εκτίμηση συναισθήματος - συμπεριφοράς με παλινδρόμηση

Σε αυτό το βήμα χρησιμοποιούμε το multitask dataset, το οποίο αποτελείται από 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.

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.

$$\rho = \frac{cov(R(X), R(Y))}{\sigma_{R(X)}\sigma_{R(Y)}}$$

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].

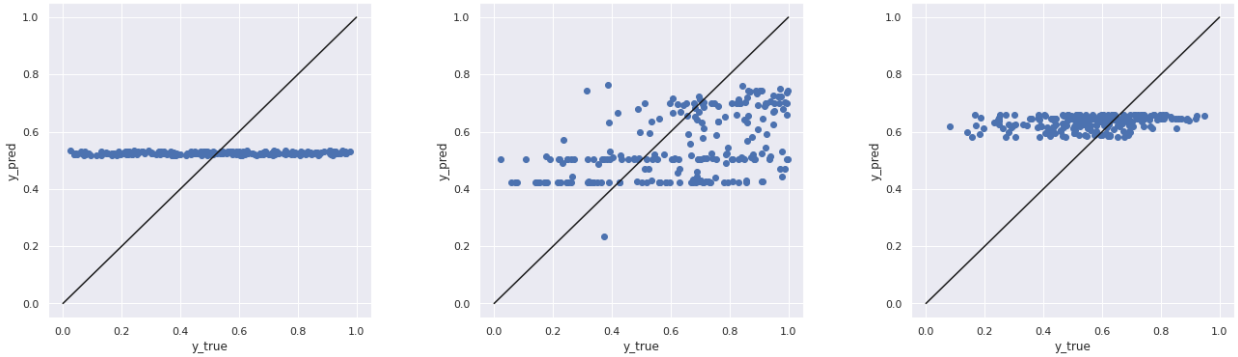
Εκπαιδεύσαμε τα μοντέλα μας για **30 εποχές** και κάναμε χρήση του **MSE Loss** ως κριτήριο κόστους.

Παραθέτουμε την απόδοση των δύο μοντέλων σε κάθε επιμέρους task.

Emotion/Model	LSTM	CNN
Valence	0.110805	0.521929
Energy	0.515467	0.731633
Danceability	0.275879	0.706031
Mean	0.300717	0.653198

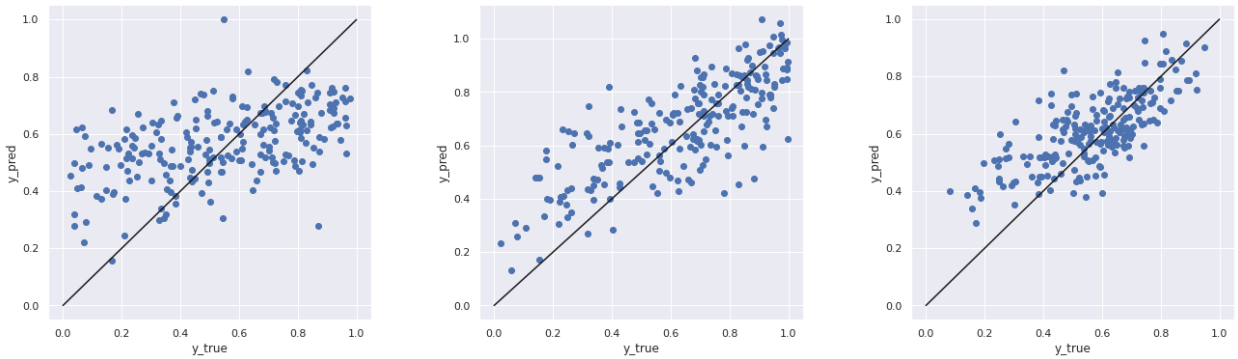
Table 1: Spearman Correlation for each emotion and each model

Από τα παραπάνω αποτελέσματα γίνεται φανερό πως το CNN πετυχαίνει πολύ καλύτερα αποτελέσματα στο συγκεκριμένο task για κάθε συναίσθημα. Στη συνέχεια, παραθέτουμε και για λόγους οπτικοποίησης τις τιμές που "κρίνει" το κάθε μοντέλο πως αντιστοιχούν σε κάθε sample εισόδου (από το validation set που ξεχωρίσαμε στην αρχή ώστε να αξιολογήσουμε το μοντέλο μας).



(a) Valence (b) Energy (c) Danceability

Figure 18: (LSTM) Scatter Plots



(a) Valence (b) Energy (c) Danceability

Figure 19: (CNN) Scatter Plots

Από τα scatter plots μπορούμε να παρατηρήσουμε ότι το CNN έχει καλύτερη ικανότητα να γενικέυει. 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.

Αντιθέτως, το LSTM αδυνατεί σε αυτό το task. Για το valence, το LSTM αποδίδει διαρκώς τιμές κοντά στο 0.5), γεγονός που υποδηλώνει πλήρη αδυναμία του LSTM να "μάθει" να κρίνει το valence από το συγκεκριμένο dataset. Αυτό είναι αποτέλεσμα και της δομής του νευρωνικού, του dataset, αλλά και της δυσκολίας αναγνώρισης ενός συναισθήματος σαν το valence. Παρόμοια συμπεράσματα συνάγονται και για την απόδοση του LSTM ως προς το συναίσθημα του danceability. Παρόλαυτά, η απόδοση του LSTM στο energy δεν ειναι όσο κακιά όσο στα υπόλοιπα συναισθήματα, γεγονός που αποτυπώνεται τόσο στο scatter plot όσο και στην αντίστοιχη τιμή του spearman correlation. Είναι αρκετά πιθανό, για την χαμηλή απόδοση του νευρωνικού να ευθύνεται και κάποιο φαινόμενο vanishing/exploding gradient.

Παρατηρώντας πως και τα δύο μοντέλα πετυχαίνουν τη μέγιστη απόδοσή τους στο energy, εύλογα συμπεραίνουμε πως αυτό είναι το πιο εύκολο συναίσθημα ανάμεσα στα τρία να ανιχνευθεί.

Βήμα 9. Μεταφορά γνώσης (Transfer Learning)

Αρχικά θα αναφερθούμε συνοπτικά στα βασικά συμπεράσματα του [8]. Στο εν λόγω paper εξετάζονται οι παράμετροι που επηρεάζουν τη μεταφερσιμότητα βαρών από προεκπαιδευμένα νευρωνικά δίκτυα σε νέα. Συγκεκριμένα, βάσει της ιδέας ότι τα νευρωνικά δίκτυα, εν γένει, εσωτερικεύουν γενικά χαρακτηριστικά στα ψηλά τους επίπεδα και πιο εξειδικευμένα στο εκάστοτε task χαρακτηριστικά στα βαθύτερα επίπεδα, εξετάζεται το κατά πόσο είναι δυνατό να μεταφερθούν βάρη δικτύου που έχει εκπαιδευτεί σε ένα task σε ένα δίκτυο που προορίζεται για ένα άλλο (παρεμφερές) task. Τα συμπεράσματα στα οποία καταλήγουν οι συγγραφείς είναι πως η μεταφερσιμότητα εξαρτάται κυρίως από το επίπεδο που επιλέγουμε να "κόψουμε" το προεκπαιδευμένο νευρωνικό και τον βαθμό στον οποίο τα features των higher layers του είναι εξειδικευμένα στο αρχικό task. Ακόμη κι αν το νέο μας task ειναι αρκετά "μακρινό" σε σχέση με αυτό στο οποίο έχουμε ένα προεκπαιδευμένο νευρωνικό, είναι προτιμότερο να αρχικοποιήσουν τα βάρη του νέου νευρωνικού με τιμές των βαρών του παλιού, αντι να τα αρχικοποιήσουμε με τυχαιες τιμές. Αυτή η τεχνική αυξάνει το generalization performance ακόμη και μετά από στοιχειώδες fine-tuning στο νέο task.

Για να εκπαιδεύσουμε στο νέο dataset (multitask) επιλέγουμε το βέλτιστο custom CNN μοντέλο από τα προηγούμενα βήματα. Η επιλογή μας βασίζεται στους εξής λόγους: (1) το CNN έχει ήδη πετύχει σημαντικά καλύτερα αποτελέσματα σε σχέση με το LSTM, (2) Θέλουμε να εκμεταλλευτούμε την γενική ιδιότητα των CNNs να εσωτερικεύουν high-level και low-level features στα υψηλότερα και βαθύτερα, αντίστοιχα, επίπεδά τους. Περιμένουμε ότι εφόσον το valence, energy και danceability ειναι ιδιότητες συσχετιζόμενες με το genre, θα ήταν ενδεχομένως καλή πρακτική να αξιοποιήσουμε τα συνελικτικά στρώματα που είναι εκπαιδευμένα να αναγνωρίζουν χαρακτηριστικά genre, (3) η εφαρμογή transfer learning σε LSTMs δε συνηθίζεται.

Παγώνουμε λοιπόν τα συνελικτικά στρώματα του CNN μας και τροποποιούμε τη διάσταση εξόδου του νευρωνικού. Συγκεκριμένα τροποποιούμε το νευρωνικό ώστε το fully connected layer να καταλήγει σε μία μόνο έξοδο (που θα δίνει τον αριθμό από 0 έως 1 ανάλογα με το κατά πόσο υπάρχει valence, energy, danceability σε ένα sample). Επιλέγουμε να μετεκπαιδεύσουμε το μοντέλο μας ώστε να δίνει στην έξοδο αξιολόγηση sample ως προς το energy. Εκπαιδεύουμε για **10 εποχές** και χρησιμοποιούμε για τις υπόλοιπες παραμέτρους εκπαίδευσης τις ήδη χρησιμοποιούμενες στις προηγούμενες εκπαιδεύσεις.

Επιλέξαμε το set των non-beat synced spectrograms, διότι δεν χρειάζεται να αφαιρέσουμε την πλεονάζουσα πληροφορία τους (και να χρησιμοποιήσουμε τα beat-synced). Κι αυτό, επειδή (όπως αναφέρθηκε και σε προηγούμενο μέρος της αναφοράς) διότι δεν υπάρχει ο κίνδυνος του vanishing/exploding gradient λόγω μεγάλου αριθμού timesteps (σε αντίθεση με τα LSTMs). Τα chromagrams απορρίφθηκαν εξ'αρχής λόγω της χείριστης επίδοσης κάθε προηγούμενου μοντέλου που εκπαιδεύτηκε σε αυτά (κι επιπλέον δοκιμάστηκαν και στο CNN, αλλά δεν απέδωσαν καλά). Φαίνεται πως η πληροφορία που ενσωματώνουν δεν είναι αρκετά περιεκτική για τα παρόντα tasks.

Η μετρική που χρησιμοποιούμε για την αξιολόγηση των μοντέλων μας είναι η Spearman Correlation, όπως και πριν. Είναι εμφανές ότι μετρικές όπως το accuracy δεν είναι έγκυρες σε αυτό το πρόβλημα, αφού το task μας είναι τύπου regression και όχι classification.

Το μοντέλο μας πετυχαίνει Spearman Correlation ίσο με 0.759614 και το scatter plot που του αντιστοιχεί είναι το παρακάτω:

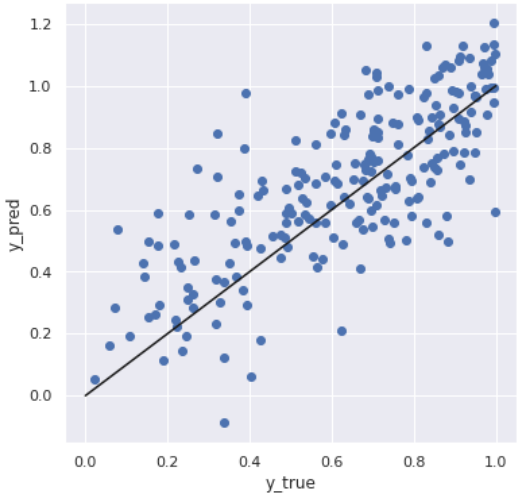


Figure 20: (CNN) Scatter Plot after Transfer Learning

Η απόδοση του νευρωνικού μας είναι λίγο χαμηλότερη από αυτή του CNN που εκπαιδεύτηκε πλήρως στο multitask dataset, γεγονός που αναδεικνύει την χρησιμότητα του transfer learning, αφού με πολύ σύντομη χρονικά εκπαίδευση ενός προϋπάρχοντος νευρωνικού πετυχαίνουμε ανταγωνιστικά αποτελέσματα με ένα μοντέλο που είναι specifically trained στο dataset ενδιαφέροντος.

Βήμα 10. Εκπαίδευση σε πολλαπλά προβλήματα (Multitask Learning)

Αρχικά θα αναφερθούμε συνοπτικά στα βασικά συμπεράσματα του [9]. Το εν λόγω paper επιδεικνύει πως η εκπαίδευση ενός μοντέλου για την ταυτόχρονη εκτέλεση πολλαπλών learning tasks αποδίδει ισότιμα ή και καλύτερα από την εκπαίδευση πολλαπλών μοντέλων ξεχωριστά για κάθε task. Ακόμη, αποδεικνύει ότι tasks με λιγοστά δεδομένα επιτελούνται με μεγαλύτερη απόδοση από την ταυτόχρονη εκπαίδευση με άλλα tasks, ενώ η απόδοση σε large tasks να πέφτει ελαφρώς ή καθόλου.

Υλοποιούμε μια custom συνάρτηση κόστους σαν nn.Module:

```python
class MyLoss(nn.Module):
    def forward(self, y_gold, y_pred):
        gold_valence = y_gold[:, 0]
        gold_energy  = y_gold[:, 1]
        gold_dance   = y_gold[:, 2]

        pred_valence = y_pred[:, 0]
        pred_energy  = y_pred[:, 1]
        pred_dance   = y_pred[:, 2]

        valence_loss = nn.MSELoss()(gold_valence, pred_valence)
        energy_loss  = nn.MSELoss()(gold_energy,  pred_energy)
        dance_loss   = nn.MSELoss()(gold_dance,   pred_dance)

        return valence_loss + energy_loss + dance_loss
```

Χρησιμοποιούμε την ανωτέρω συνάρτηση κόστους και εκπαιδεύουμε για **30 εποχές** το βέλτιστο CNN μοντέλο μας στο multitask dataset. Πρώτα, φυσικά, για να το σχεδιάσουμε ώστε να κάνει multitask learning, θέτουμε τη διάσταση εξόδου του fully connected layer ίση με 3 (ένας αριθμός για κάθε task: valence, energy, danceability).

Το CNN μας πετυχαίνει τις εξής Spearman Correlations:

Emotion/Model	**Multitask CNN**
Valence	0.557476
Energy	0.745879
Danceability	0.707885
Mean	0.670413

Table 2: Spearman Correlation for each emotion and each model

Ερχόμενοι σε συμφωνία με το [9], παρατηρούμε ότι το multitask CNN πετυχαίνει καλύτερη επίδοση στην αναγνώριση συναισθήματος σε σχέση με κάθε CNN που εκπαιδεύτηκε στο Βήμα 8 σε εξειδικευμένο task (στον χρόνο εκπαίδευσης ενός δικτύου).

Βήμα 11. Υποβολή στο Kaggle

Στο τελευταίο βήμα του εργαστηρίου πειραματιζόμαστε με την απόδοση του CNN μας στο task της multitask αναγνώρισης συναισθήματος. Ύστερα από πειράματα με τις παραμέτρους του μοντέλου μας, το διαχωρισμό του training set και το batch size, καταλήξαμε στα εξής:

Σχετικά με το dataset:

- batch_size = 32 (και για το train και για το validation set)
- ποσοστό των data που αποτελούν validation set: 15%

Σχετικά με την αρχικτονική του CNN:

```
class CustomCNN(nn.Module):
    def __init__(self):
        super(CustomCNN,self).__init__()

        self._cnn_module = nn.Sequential(
            # layer 1
            nn.Conv2d(in_channels=1, out_channels=32, kernel_size=3, stride=1, padding=1),
            nn.BatchNorm2d(32),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2),
            # layer 2
            nn.Conv2d(in_channels=32, out_channels=64, kernel_size=3, stride=1, padding=1),
            nn.BatchNorm2d(64),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2),
            # layer 3
            nn.Conv2d(in_channels=64, out_channels=64, kernel_size=5, stride=1, padding=1),
            nn.BatchNorm2d(64),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2),
            # layer 4
            nn.Conv2d(in_channels=64, out_channels=128, kernel_size=3, stride=1, padding=1),
            nn.BatchNorm2d(128),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2),
            # layer 5
            nn.Conv2d(in_channels=128, out_channels=32, kernel_size=3, stride=1, padding=1),
            nn.BatchNorm2d(32),
            nn.ReLU()
        )
        # fully connected layer
        self._fc_module = nn.Sequential(
            nn.ReLU(),
            nn.Dropout(),
            nn.Linear(in_features=20480, out_features=3)
        )

    def forward(self, x):
        x=x.transpose(1,2)
        x=torch.unsqueeze(x,1)
        x = self._cnn_module(x)
        x = x.view(x.size(0), -1)
        x = self._fc_module(x)
        return x
```

Υποβάλλοντας τα αποτελέσματα του μοντέλου μας με είσοδο τα unlabelled test data στο Kaggle, πετυχαίνουμε score ίσο με **0.69198** στο Leaderboard (στο 100% των test data). Παρατηρούμε ότι σε σχέση με το βέλτιστο μέσο Spearman Correlation που πετύχαμε στο βήμα 10, αυτό του Kaggle είναι λίγο καλύτερο. Τα δύο scores, ωστόσο, είναι πολύ κοντά μεταξύ τους, γεγονός που έρχεται σε συμφωνία με τις προσδοκίες μας. Ακόμη, αυτό μας υποδεικνύει ότι το μοντέλο μας μπορεί να κάνει ένα σχετικά ικανοποιητικό generalization σε τέτοιου τύπου data (δεδομένου του scale του προβλήματος).

Αναφορές

[1] https://www.kaggle.com/c/multitask-affective-music-lab-2022/data

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

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

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

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

[6] https://www.baeldung.com/cs/ml-relu-dropout-layers

[7] https://deepai.org/machine-learning-glossary-and-terms/max-pooling

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

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