ΗΜΜΥ ΕΜΠ

Αναγνώριση Προτύπων (2022-2023)

3η Εργαστηριακή Άσκηση

Ηλιόπουλος Γεώργιος:	03118815	giliopoulos301@gmail.com
Σερλής Εμμανουήλ Αναστάσιος:	03118125	manosserlis@gmail.com

Θέμα: Αναγνώριση Είδους και Εξαγωγή Συναισθήματος από Μουσική

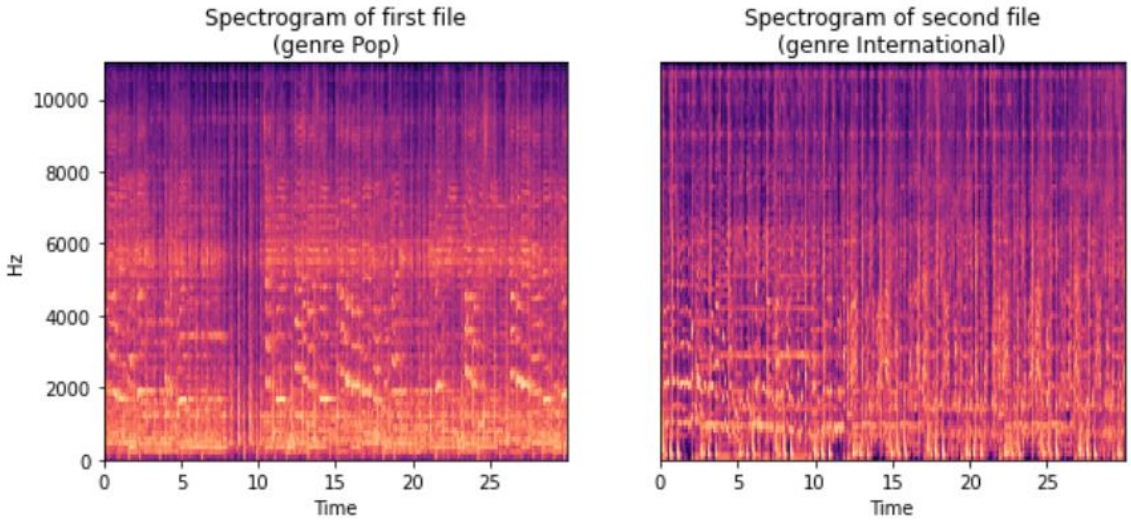
Σκοπός είναι η αναγνώριση του είδους και η εξαγωγή συναισθηματικών διαστάσεων από φασματογραφήματα μουσικών κομματιών. Στο στάδιο της προπαρασκευής υλοποιείται η εξαγωγή φασματογραφημάτων (mel και beat-synced) και χρωμογραφημάτων από τα δεδομένα του Free Music Archive (FMA) dataset. Στην συνέχεια τα φασματογραφήματα αποτελούν είσοδοι σε 5 διαφορετικούς τύπους LSTM δικτύων, τα οποία αξιολογούνται με βάση διάφορες μετρικές (accuracy, recall, F1-score κ.α.)

Βήμα 0: Εξοικείωση με Kaggle kernels

Στο βήμα αυτό, έγινε εξοικείωση με την πλατφόρμα Kaggle και τις διάφορες λειτουργικότητές της, από το διάβασμα των επιμέρους subfolders των δοθέντων datasets, μέχρι την λειτουργία των private kernels και την χρήση της GPU.

Βήμα 1: Εξοικείωση με φασματογραφήματα στην κλίμακα mel

Στο βήμα αυτό, πραγματοποιείται επιλογή 2 τυχαίων mel-spectrograms από το FMA dataset, τα οποία προέρχονται από διαφορετικά μουσικά labels. Η απεικόνιση των spectrograms ακολουθεί παρακάτω:



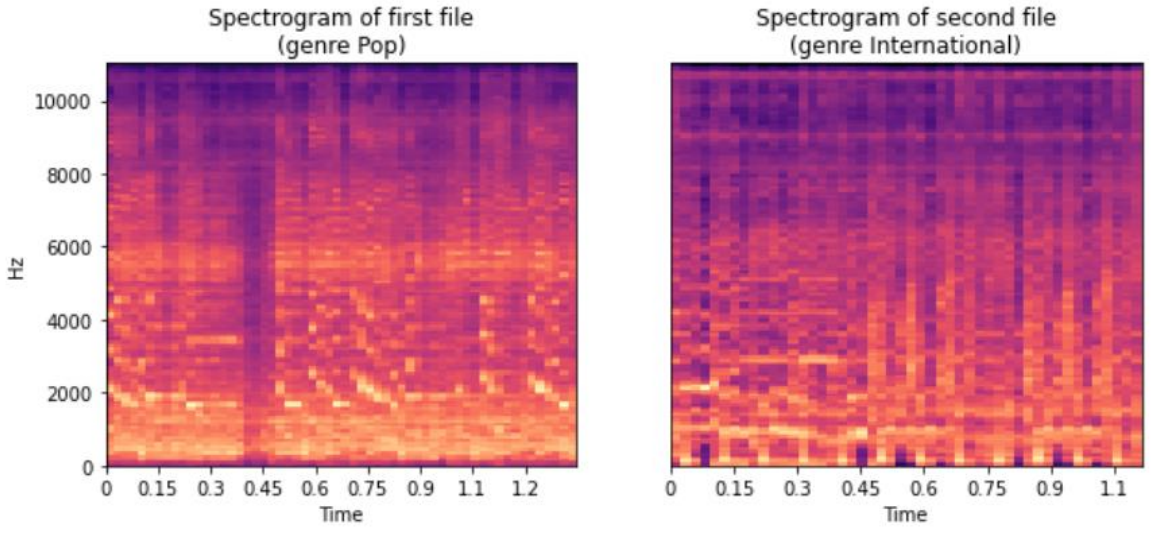
Εικόνα 1: Spectrograms ενός pop και ενός international sample

Όσον αφορά τα spectrograms, αυτά αποτελούν έναν οπτικό τρόπο αναπαράστασης της ισχύος του σήματος στον χρόνο, για τις διαφορετικές συχνότητες που υπάρχουν στην εκάστοτε κυματομορφή. Ο οριζόντιος άξονας αντιστοιχεί στα επιμέρους χρονικά παράθυρα (εν προκειμένω από 0-30sec) ενώ ο κατακόρυφος άξονας στις επιμέρους μπάντες συχνοτήτων. Στα άνωθι spectrograms, παρατηρούμε ότι η κύρια διαφορά ανάμεσα στην pop και στην international μουσική-τουλάχιστον στα άνωθι δείγματα- έγκειται στο γεγονός ότι στην pop οι συχνοτικές κατανομές είναι αισθητά πιο συνεχείς για τα διαφορετικά time windows από ό,τι στην international μουσική. Επιπλέον, στην περίπτωση της pop, παρατηρείται μια έντονη μπάντα συχνοτήτων γύρω από τα 6kHz η οποία απουσιάζει στην περίπτωση του international sample. Τα παραπάνω μας κάνουν να συμπεράνουμε ότι το spectrogram αποτελεί ένα εν δυνάμει ικανοποιητικό διαχωριστικό εργαλείο για την διάκριση μεταξύ των διαφορετικών ειδών μουσικής.

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

α) Τα φασματογραφήματα από το βήμα 1 έχουν και τα 2 διαστάσεις (128,1291), με την 2η διάσταση να αποτελεί το πλήθος των διακριτών χρονικών βημάτων. Η παραπάνω ανομοιογένεια μεταξύ του frequency και του time space θα οδηγούσε σε μη αποδοτική εκπαίδευση LSTM δικτύων, τα οποία θα καλούνταν να αποθηκεύσουν αχρείαστα μεγάλο όγκο πληροφορίας στο πεδίο του χρόνου.

β) Ως αποτέλεσμα των παραπάνω, είναι αναγκαία η μείωση των time steps, κάτι που μπορεί να επιτευχθεί μέσω του συγχρονισμού των spectrograms πάνω στον ρυθμό. Συγκεκριμένα, χρησιμοποιούμε το fma_genre_spectrogram_beat, στο οποίο έχει ληφθεί η διάμεσος μεταξύ 2 διαδοχικών «χτύπων» του beat. Τέλος, ακολουθείται παρόμοιος τρόπος με το 1ο ερώτημα για την αναπαράσταση των beat-synced spectrograms, με διαστάσεις (128,58) το καθένα:



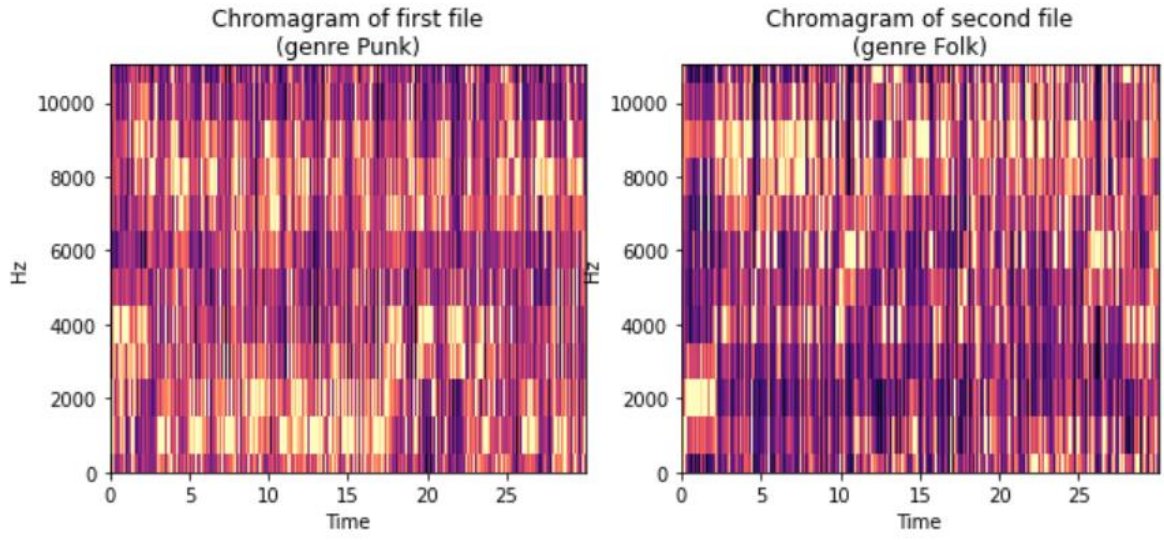
Εικόνα 2: Beat-synced Spectrograms ενός pop και ενός international sample

Παρατηρούμε ότι παρά την μείωση των time-steps, τα beat-synced spectrograms δεν διαφέρουν σημαντικά από τα αρχικά φασματογραφήματα του 1ου ερωτήματος, με παρεμφερή συμπεράσματα να εξάγονται εκ νέου (πιο συνεχές χαμηλόσυχνο φάσμα για την pop και πιο έντονη μπάντα συχνοτήτων στα 6kHz). Με αυτόν τον τρόπο, η χρονική διάρκεια προπόνησης των LSTM δικτύων αναμένεται να μειωθεί σημαντικά, χωρίς να αφαιρείται σημαντική χρονοσυχνοτική πληροφορία.

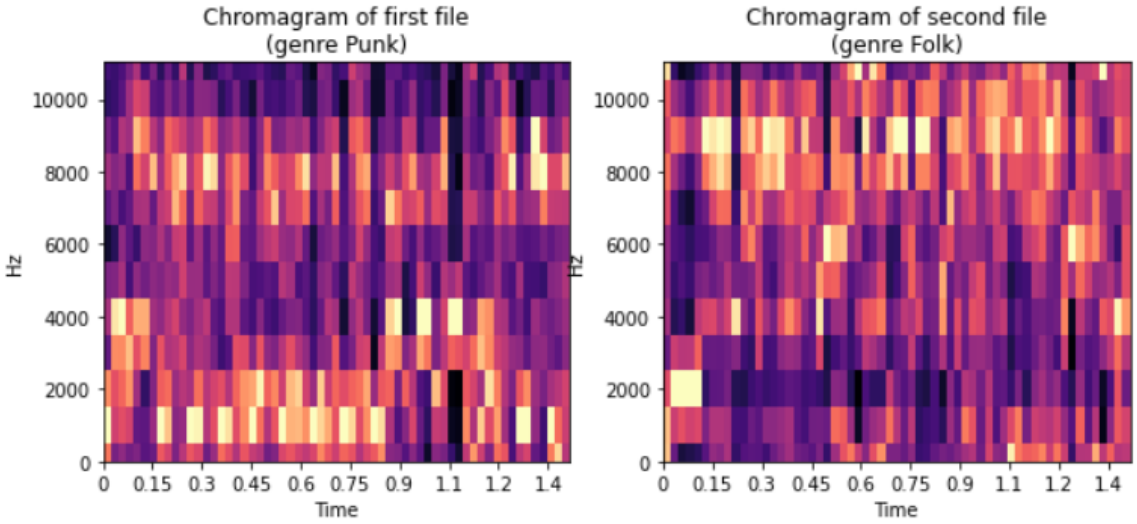
Βήμα 3: Εξοικείωση με χρωμογραφήματα

Ως χρωμογραφήματα, ορίζουμε τις 2D-αναπαραστάσεις της ενέργειας του εκάστοτε μουσικού σήματος για τις μπάντες συχνοτήτων που αντιστοιχούν στις 12 διαφορετικές νότες της κλασική μουσικής κλίμακας. Έτσι, αποτελούν ένα εναλλακτικό εργαλείο σε σχέση με τα spectrograms, ιδιαίτερα για την ανάλυση μελωδικών χαρακτηριστικών της μουσικής.

Ακολουθώντας την ίδια διαδικασία με αυτή των βημάτων 1 και 2, εξάγουμε τα χρωμογραφήματα (τόσο τα initial όσο και τα beat-synced) αρχείων για διαφορετικά labels (Punk & Folk). Σημειώνεται ότι και τα 2 αρχεία έχουν διαστάσεις (12, 1291) για τα initial chromagrams και διαστάσεις (12,128) για τα beat-synced chromagrams.



Εικόνα 3: Chromagrams ενός punk και ενός folk sample



Εικόνα 4: Beat-synced Chromagrams ενός punk και ενός folk sample

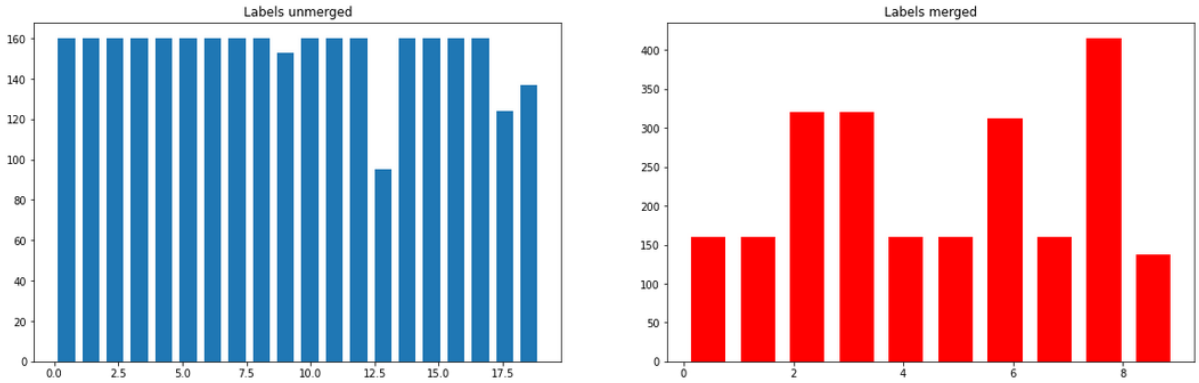
Για τον χαρακτηρισμό των μουσικών ειδών με βάση τα χρωμογραφήματά τους, παρατηρούμε ότι η βασική διαφορά ανάμεσα σε punk και folk σχετίζεται με την ύπαρξη χαμηλών συχνοτήτων υψηλής ενέργειας στο 1ο είδος τα οποία απουσιάζουν από το 2ο. Από την άλλη, και τα 2 είδη διαθέτουν σχετικά έντονες υψηλές συχνότητες, με το κομμάτι folk να προσεγγίζει την μπάντα των 10 kHz.

Βήμα 4: Φόρτωση και ανάλυση δεδομένων

Σε αυτό το βήμα, πραγματοποιείται φόρτωση των δεδομένων τα οποία θα χωριστούν σε train,validation και test sets για την εκπαίδευση μοντέλων LSTM. Όσον αφορά τον κώδικα που χρησιμοποιείται, αρχικά κατασκευάζει ένα dataset συμβατό με pytorch frameworks, ενώ στην συνέχεια διαβάζει τα labels (με διαφορετικό τρόπο ανάλογα με το αν χρησιμοποιείται το dictionary CLASS_MAPPING ή όχι) και τα κωδικοποιεί σε αριθμούς. Τέλος, πραγματοποιείται zero-padding ώστε όλα τα δεδομένα εισόδου να έχουν ίδιο μήκος.

Αφού γίνει η διαμόρφωση του dataset, στην συνέχεια γίνεται το train-validation split μέσω της αντίστοιχης εντολής της scikit-learn. Αξίζει να σημειωθεί ότι στην συνάρτηση ‘torch_train_val_split()’ η initial τιμή του seed έχει τεθεί ως None, έτσι ώστε να πραγματοποιείται διαφορετικό splitting των δεδομένων κάθε φορά και κατ’επέκταση testing των μοντέλων σε διαφορετικές συνθήκες προπόνησης ανά εποχή.

Παρακάτω ακολουθούν τα labels του dataset, με και χωρίς την χρήση συγχωνευμένων labels.



Εικόνα 5: Ιστογράμματα χωρίς και με την χρήση 'CLASS_MAPPING' merge.

Παρατηρούμε ότι έπειτα από το merge των labels που ανήκουν σε παρόμοια μουσικά είδη (π.χ. Psych-folk & folk) και την πλήρη διαγραφή labels με ελλιπή αριθμό δεδομένων (π.χ. old-time), έχουμε ένα dataset με συνολικά 10 labels. Επιπλέον, το dataset με τα merged labels είναι αισθητά πιο imbalanced από το αρχικό, με κυριότερο παράδειγμα την κλάση 8 η οποία διαθέτει παραπάνω από 400 δείγματα, ενώ καθεμιά από τις υπόλοιπες λιγότερα από 350 δείγματα. Με βάση τα άνωθι, συμπεραίνουμε ότι το νέο dataset προσφέρει την δυνατότητα γενίκευσης της ικανότητας αναγνώρισης διαφορετικών μουσικών ειδών (π.χ. pop από rock), με το μειονέκτημα όμως του ότι είναι imbalanced, γεγονός που θα μπορούσε να οδηγήσει σε αυξημένο bias προς κλάσεις με περισσότερα δεδομένα (π.χ 2,3,6,8).

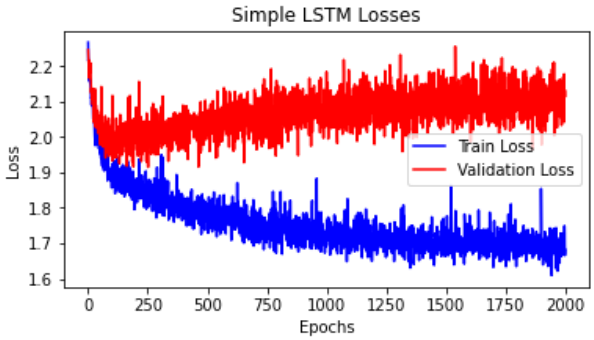
Βήμα 5: Αναγνώριση μουσικού είδους με LSTM

Για να δημιουργήσουμε, εκπαιδεύσουμε και αξιολογήσουμε τα LSTM μοντέλα χρησιμοποιήσαμε τον κώδικα μας από το 2ο εργαστήριο με ελάχιστες διαφορές λόγω της διαφορετικής σύνθεσης των δεδομένων εισόδου.

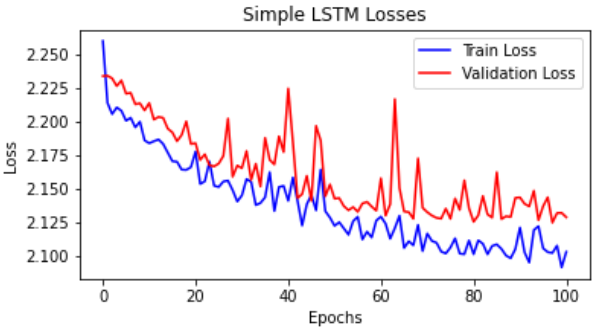
Συνολικά δημιουργήσαμε συνολικά 6 μοντέλα LSTM, 5 απλά LSTM με διαφορετικές εισόδους κάθε φορά και ένα πιο σύνθετο. Τα απλά μοντέλα LSTM έχουν 1 επίπεδο, δεν είναι bidirectional και δεν έχουν dropout. Ενώ το σύνθετο έχει 1 επίπεδο, είναι bidirectional και έχει dropout (0.2). Το πιο σύνθετο μοντέλο δημιουργήθηκε λόγω των χαμηλών ποσοστών επιτυχίας των απλών μοντέλων αλλά ούτε σε αυτό παρουσιάστηκε ιδιαίτερη βελτίωση. Επειδή για την ίδια μία είσοδο οι διαφορές μεταξύ του απλού και του σύνθετου μοντέλου δεν ήταν μεγάλες και λόγω του πολύ χρόνου που απαιτείται για να εκπαιδευτεί ένα μοντέλο δεν συνεχίσαμε με την εκπαίδευση σύνθετων μοντέλων για όλες τις εισόδους.

Στο πρώτο μοντέλο (μοντέλο β εικόνας 6) προσπαθήσαμε βάζοντας όλα τα δεδομένα σε ένα batch να πετύχουμε overfitting αλλά δεν τα καταφέραμε (ποσοστό επιτυχίας της τάξης του $0.3 - 0.4$) ίσως λόγω του μεγάλου μεγέθους των σημάτων εισόδου.

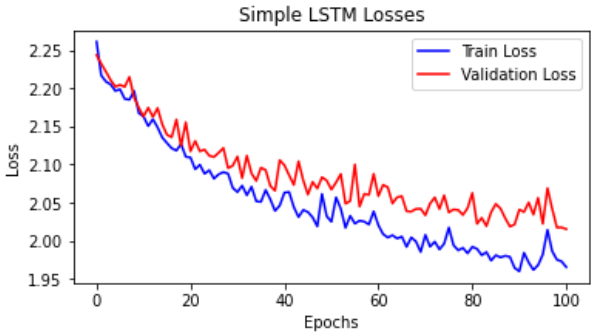
Στην εικόνα 6 φαίνονται τα losses κατά την εκπαίδευση των 6 μοντέλων.



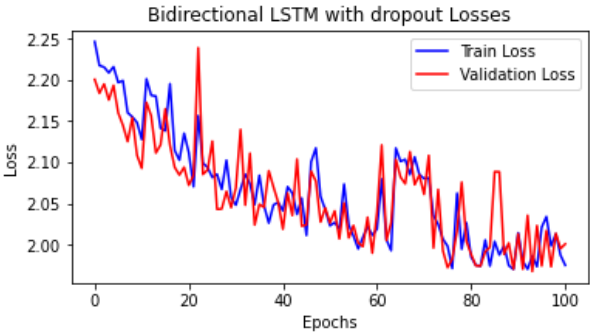
β) απλό LSTM με είσοδο τα συγχρονισμένα φασματογραφήματα σε ένα batch ώστε να πετύχουμε overfitting



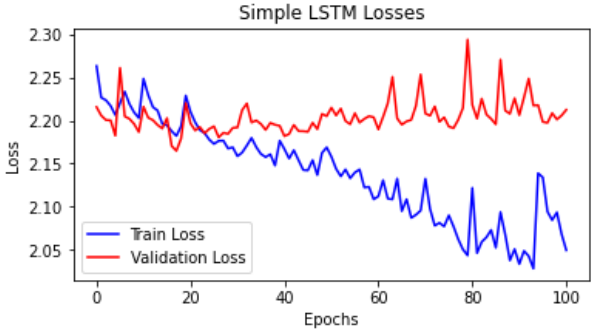
γ1) απλό LSTM με είσοδο τα φασματογραφήματα



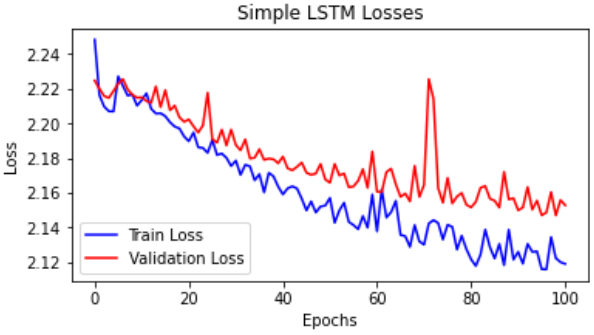
γ2) σύνθετο LSTM με είσοδο τα φασματογραφήματα



δ) απλό LSTM με είσοδο τα συγχρονισμένα φασματογραφήματα



ε) απλό LSTM με είσοδο τα χωρογραφήματα



στ) απλό LSTM με είσοδο τα ενωμένα φασματογραφήματα και χωρογραφήματα

Εικόνα 6: losses κατά την εκπαίδευση των 6 μοντέλων.

Βήμα 6: Αξιολόγηση των μοντέλων

Για να υπολογίσουμε τις ζητούμενες μετρικές χρησιμοποιήσαμε την συνάρτηση `sklearn.metrics.classification_report`. Στην εικόνα 7 φαίνονται τα μετρικά των 5 μοντέλων.

	precision	recall	f1-score	support
0	0.00	0.00	0.00	40
1	0.00	0.00	0.00	40
2	0.19	0.72	0.30	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	0.27	0.71	0.39	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

γ1) απλό LSTM με είσοδο τα φασματογραφήματα

	precision	recall	f1-score	support
0	0.00	0.00	0.00	40
1	0.17	0.78	0.28	40
2	0.29	0.46	0.35	80
3	0.34	0.41	0.37	80
4	0.00	0.00	0.00	40
5	0.00	0.00	0.00	40
6	0.29	0.03	0.05	78
7	0.00	0.00	0.00	40
8	0.32	0.50	0.39	103
9	0.00	0.00	0.00	34
accuracy			0.27	575
macro avg	0.14	0.22	0.14	575
weighted avg	0.19	0.27	0.20	575

γ2) σύνθετο LSTM με είσοδο τα φασματογραφήματα

	precision	recall	f1-score	support
0	0.00	0.00	0.00	40
1	0.21	0.65	0.31	40
2	0.26	0.61	0.36	80
3	0.27	0.25	0.26	80
4	0.00	0.00	0.00	40
5	0.00	0.00	0.00	40
6	0.42	0.40	0.41	78
7	0.00	0.00	0.00	40
8	0.32	0.34	0.33	103
9	0.00	0.00	0.00	34
accuracy			0.28	575
macro avg	0.15	0.22	0.17	575
weighted avg	0.20	0.28	0.22	575

δ) απλό LSTM με είσοδο τα συγχρονισμένα φασματογραφήματα

	precision	recall	f1-score	support
0	0.14	0.07	0.10	40
1	0.16	0.20	0.18	40
2	0.13	0.15	0.14	80
3	0.19	0.31	0.23	80
4	0.00	0.00	0.00	40
5	0.00	0.00	0.00	40
6	0.23	0.31	0.26	78
7	0.00	0.00	0.00	40
8	0.22	0.31	0.26	103
9	0.08	0.06	0.07	34
accuracy			0.18	575
macro avg	0.12	0.14	0.12	575
weighted avg	0.14	0.18	0.16	575

ε) απλό LSTM με είσοδο τα χωρογραφήματα

	precision	recall	f1-score	support
0	0.00	0.00	0.00	40
1	0.00	0.00	0.00	40
2	0.16	0.74	0.26	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	0.28	0.56	0.37	103
9	0.00	0.00	0.00	34
accuracy			0.20	575
macro avg	0.04	0.13	0.06	575
weighted avg	0.07	0.20	0.10	575

στ) απλό LSTM με είσοδο τα ενωμένα φασματογραφήματα και χωρογραφήματα

Εικόνα 7: μετρικές των 5 μοντέλων (δεν υπολογίστηκαν για το μοντέλο β της εικόνας 6).

Στην περίπτωσή μας, παρατηρώντας το ιστόγραμμα της εικόνας 5 μπορούμε να δούμε ότι οι κλάσεις δεν είναι απόλυτα ισορροπημένες, αλλά ούτε και εντελώς ανισόρροπες. Έτσι, το accuracy εξακολουθεί να είναι μια αξιόπιστη μετρική. Το accuracy φαίνεται να είναι πιο σημαντική από το recall, αλλά χρειαζόμαστε κυρίως τη συνολική επιτυχία. Έτσι, βασιζόμαστε κυρίως στο accuracy και το f1-score.

Συγκρίνοντας τα μοντέλα μας, το μοντέλο LSTM με τα με είσοδο τα συγχρονισμένα φασματογραφήματα είχε καλύτερα αποτελέσματα από τα υπόλοιπα. Ο λόγος για τον οποίο αποδείχθηκε καλύτερο από αυτό με τα φασματογραφήματα mel θα μπορούσε να είναι ότι τα φασματογραφήματα μικρότερου μεγέθους είχαν ως αποτέλεσμα λιγότερη υπερπροσαρμογή. Το μοντέλο με τα χρωμογραφήματα ως είσοδο είχε τα χειρότερα αποτελέσματα, κάτι που διαισθητικά είναι λογικό, παρατηρώντας την πληροφορία που παρέχουν από τα διαγράμματα

στα παραπάνω βήματα (εικόνα 3 και 4). Το μοντέλο με ενωμένα φασματογραφήματα και χρωμογράμματα ως είσοδο, τα πήγε καλύτερα από εκείνα με τις αντίστοιχες εισόδους μόνο, όπως αναμενόταν, δεδομένου ότι είχε περισσότερα χρήσιμα δεδομένα για την ταξινόμηση.

Γενικά, η ταξινόμηση μουσικών ειδών είναι ένα δύσκολο πρόβλημα και τα απλά νευρωνικά δίκτυα LSTM είναι δεν είναι ιδανικά. Ως εκ τούτου βλέπουμε μη ικανοποιητικά αποτελέσματα.

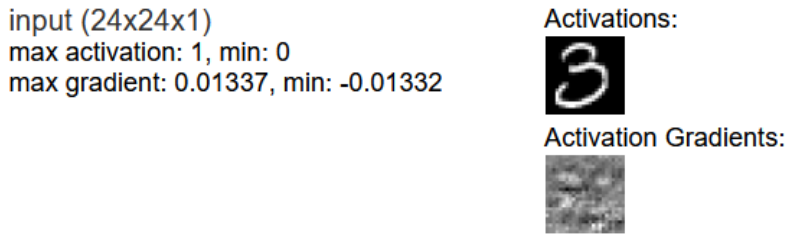
Ερμηνεία μετρικών:

- Το **Accuracy** εκφράζει το συνολικό ποσοστό των δειγμάτων που ταξινομήθηκαν σε σωστή κλάση.
- Αντίθετα το **Precision** αναφέρεται σε μια μόνο κλάση και ισούται με $\frac{TP}{TP+FP}$. Διαισθητικά αξιολογεί την ικανότητα του μοντέλου να διακρίνει άσχετα δείγματα από τη δεδομένη κλάση.
- Το **Recall** αναφέρεται και αυτό σε μια μόνο κλάση και ισούται με το λόγο $\frac{TP}{TP+FN}$. Διαισθητικά, αξιολογεί την ικανότητα του μοντέλου να διακρίνει την δεδομένη κλάση μέσα στα δείγματα.
- Το **F1-score** είναι ο σταθμισμένος αρμονικός μέσος όρος των precision και recall για μια κλάση, δίνοντας έτσι μια γενική αξιολόγηση του μοντέλου για μια συγκεκριμένη κλάση.
- Οι **Macro-** μετρικές επιστρέφουν το μέσο όρο των αποτελεσμάτων αγνοώντας αν οι κλάσεις είναι imbalanced.
- Οι **Micro-** μετρικές μας βοηθούν να αξιολογήσουμε imbalanced datasets αφού συγκεντρώνουν τα TP, FP και FN για όλες τις κλάσεις και υπολογίζουν μία μέση μετρική.
- Μεγάλες διαφορές μεταξύ του F1-score και του accuracy προκύπτει όταν έχουμε imbalanced datasets. Για παράδειγμα αν έχουμε πολλά δεδομένα από μία κλάση Α και πολύ λίγα από τις κλάσεις Β, Γ και Δ η κλάση Α θα έχει μεγάλο F1-score ενώ το γενικό accuracy του μοντέλου θα είναι χαμηλό. Αντίστοιχα αν έχουμε πολύ λίγα δεδομένα από μία κλάση Α και αρκετά από τις κλάσεις Β, Γ και Δ η κλάση Α θα έχει μικρό F1-score ενώ το γενικό accuracy του μοντέλου θα είναι υψηλό. Έτσι για τη σωστή αξιολόγηση ενός μοντέλου πρέπει να ελέγχουμε φυσικά το accuracy αλλά και τα επιμέρους F1-scores.
- Μεγάλες διαφορές μεταξύ των micro- και macro- μετρικών μπορεί να υπάρξουν όταν έχω πολύ imbalanced datasets.
- Ένα πρόβλημα όπου με ενδιαφέρει περισσότερο το recall από το precision είναι ένα ιατρικό σύστημα, για παράδειγμα ένα σύστημα πρόβλεψης καρκίνου. Αντίστοιχα προβλήματα υπάρχουν για το αντίστροφο όπως για παράδειγμα σε ένα πρόβλημα πρότασης ταινιών προς θέαση όπου μας ενδιαφέρει η ταινία να είναι αρεστή αλλά δεν πειράζει να μην προταθεί μία άλλη επίσης αρεστή ταινία.
- Η επιλογή απλώς του accuracy ή του F1-score σε τέτοιες ευαίσθητες περιπτώσεις είναι αναποτελεσματική.

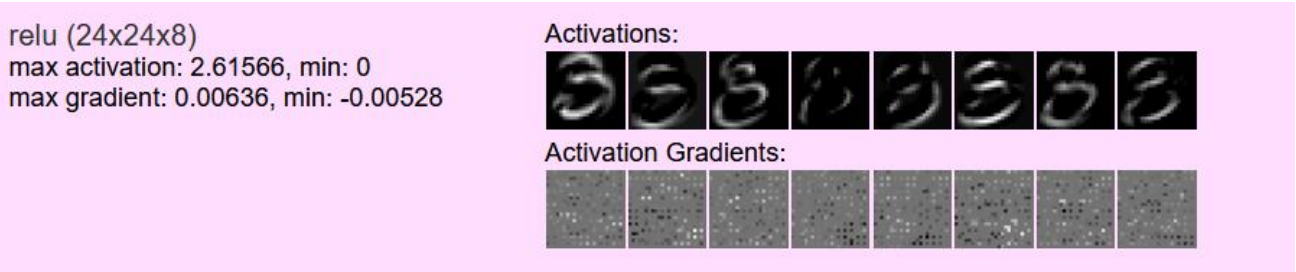
Βήμα 7: 2D CNN

Αρχικά, πραγματοποιήθηκε εκπαίδευση ενός δικτύου τύπου CNN με σκοπό την αναγνώριση χειρόγραφων ψηφίων. Το εν λόγω δίκτυο-που παρέχεται από το πανεπιστήμιο του Standford- αποτελείται από 2 ενδιάμεσα convolutional layers τα οποία μειώνουν τις διαστάσεις σε 24x24 και 12x12 αντίστοιχα. Έπειτα από κάθε layer υπάρχει η συνάρτηση ενεργοποίησης relu καθώς και max pooling. Η έξοδος του δικτύου περνά από μια συνάρτηση ενεργοποίησης τύπου softmax για να ολοκληρωθεί η ταξινόμηση της εκάστοτε εικόνας.

Όσον αφορά τις εξόδους κάθε layer αυτές απεικονίζονται παρακάτω:

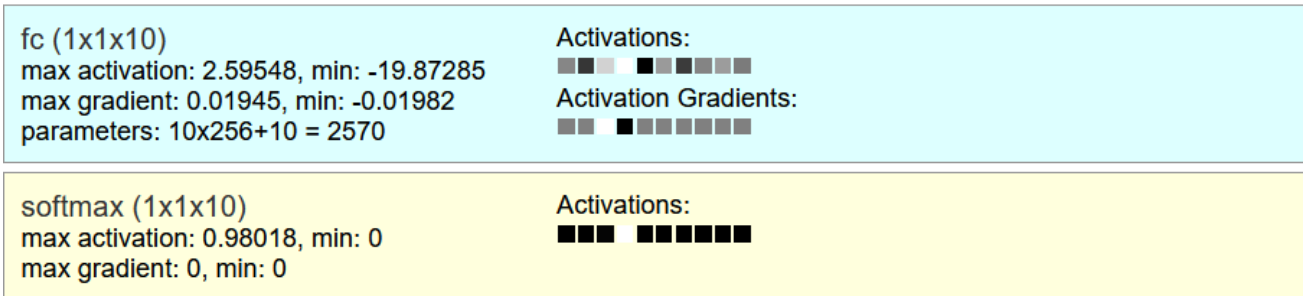


Εικόνα 8: Είσοδος CNN δικτύου για ταξινόμηση εικόνων



Εικόνα 9: Έξοδος 1ου layer CNN δικτύου για ταξινόμηση εικόνων

Εικόνα 10: Έξοδος 2ου layer CNN δικτύου για ταξινόμηση εικόνων



Εικόνα 11: Έξοδος CNN δικτύου για ταξινόμηση εικόνων

Παρατηρούμε ότι στο πρώτο layer του δικτύου, στόχος είναι η αναγνώριση high level χαρακτηριστικών του ψηφίου, που σχετίζονται με το περίγραμμα του. Στην συνέχεια, πραγματοποιείται max pooling για μείωση των διαστάσεων της εικόνας ενώ αυξάνεται το πλήθος των διαφορετικών φίλτρων από 8 σε 16, με στόχο την εξαγωγή low-level

χαρακτηριστικών στην έξοδο του επόμενου layer (κατανομή pixels στο εσωτερικό του ψηφίου, ενίσχυση περιοχών με ακμές και γωνίες κ.α.). Τέλος, τα εν λόγω χαρακτηριστικά υφίστανται εκ νέου max pooling, πρωτού περάσουν στο τελικό layer ταξινόμησής τους.

Στην συνέχεια, πραγματοποιήθηκε η υλοποίηση ενός 2D νευρωνικού δικτύου τύπου CNN, το οποίο χρησιμοποιήθηκε για το task αναγνώρισης μουσικών ειδών (σε αντιστοιχία με το βήμα 5). Τα χαρακτηριστικά κάθε layer για το CNNBackbone που χρησιμοποιήθηκε αναγράφονται και αναλύονται παρακάτω

```
Classifier(
  (backbone): CNNBackbone(
    (conv1): Sequential(
      (0): Conv2d(1, 32, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
      (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU()
      (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    )
    (conv2): Sequential(
      (0): Conv2d(32, 64, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
      (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU()
      (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    )
    (conv3): Sequential(
      (0): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU()
      (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    )
    (conv4): Sequential(
      (0): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU()
      (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    )
    (fc1): Linear(in_features=163840, out_features=1000, bias=True)
  )
  (output_layer): Linear(in_features=1000, out_features=10, bias=True)
  (criterion): CrossEntropyLoss()
)
```

Εικόνα 12: Δομή CNN δικτύου για ταξινόμηση μουσικών ειδών

Convolution: Στόχος της συνέλιξης στα νευρωνικά δίκτυα είναι το φιλτράρισμα της πληροφορίας εισόδου μέσω κατάλληλων φίλτρων/kernels για την εξαγωγή ενός feature map, δηλαδή ενός συνόλου χαρακτηριστικών. Αυτό επιτυγχάνεται μέσω της εφαρμογής πολλαπλών διαφορετικών φίλτρων σε κάθε layer του δικτύου, καθένα εκ των οποίων εξάγει ένα διακριτό χαρακτηριστικό για το input set. Έτσι, ένας βασικός στόχος του training είναι ο κατάλληλος προσδιορισμός των βαρών των εν λόγω συνελικτικών φίλτρων.

Batch Normalization: Αποτελεί βασική μέθοδο προπόνησης DNNs κατά την οποία κάθε training batch κανονικοποιείται έτσι ώστε να αντιστοιχεί σε Gaussian κατανομή με μέση τιμή 0 και τυπική απόκλιση 1. Ως αποτέλεσμα, μειώνονται οι απαιτούμενες εποχές προπόνησης ενώ βελτιώνεται και η ικανότητα γενίκευσής του (μέσω μείωσης του generalization error).

ReLU: Η βασική συνάρτηση ενεργοποίησης στα ενδιάμεσα layers των DNNs, η οποία αντιστοιχεί στην συνάρτηση max(0,x). Ο λόγος για την διαδεδομένη χρήση της τα τελευταία χρόνια έγκειται στο γεγονός ότι δεν ενεργοποιεί όλους τους νευρώνες ενός layer ταυτόχρονα, παρά μόνο όσους έχουν τιμή εισόδου x>0. Έτσι, επιταχύνεται η διαδικασία προπόνησης του

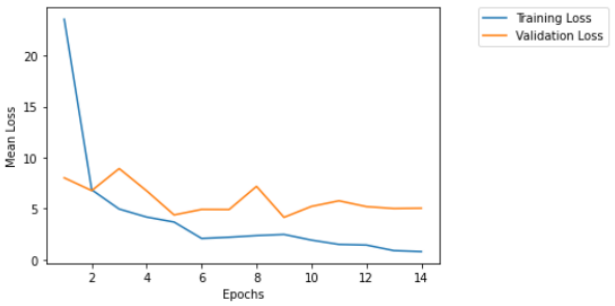
δικτύου. Παράλληλα, προσφέρει βελτιωμένο gradient propagation μιας και έχει τιμή παραγώγου ίση με 1 (για x>0), περιορίζοντας έτσι τα φαινόμενα vanishing gradient που παρατηρούνται σε άλλα activation functions (π.χ. sigmoid). Τέλος, η απλότητα της συνάρτησης (απουσία εκθετικών όρων και διαίρεσης) την καθιστά υπολογιστικά αποδοτική.

Max pooling: Αποτελεί μέθοδο pooling, η οποία υπολογίζει την μέγιστη τιμή εντός ενός παραθύρου nxn οδηγώντας σε downsampling της εκάστοτε πληροφορίας εισόδου. Συνήθως χρησιμοποιείται ανάμεσα στην έξοδο ενός layer και στην είσοδο του επόμενου. Η εν λόγω διαδικασία προσφέρει τόσο μείωση του overfitting (λόγω αύξησης του translation invariance) όσο και μείωση του υπολογιστικού κόστους και επιτάχυνση training & inference (λόγω μείωση της διαστατικότητας της πληροφορίας εισόδου).

Η προπόνηση του δικτύου έγινε στα non-beat-synced spectrograms, ενώ το σύνολο υπερπαραμέτρων που χρησιμοποιήθηκαν αναγράφεται παρακάτω:

- Loss: Cross Entropy
- Optimizer: Adam
- Learning rate: 1e-4
- Epochs: 50

Παρακάτω παρατίθενται training & validation losses καθώς και το classification report, στο οποίο επιτευχθεί accuracy score στο 35%, δηλαδή καλύτερο από όλες τις υλοποιήσεις LSTM των ερωτημάτων 5 και 6.



	precision	recall	f1-score	support
0	0.50	0.03	0.05	40
1	0.50	0.42	0.46	40
2	0.76	0.28	0.40	80
3	0.38	0.47	0.42	80
4	0.42	0.57	0.48	40
5	0.14	0.17	0.16	40
6	0.40	0.76	0.52	78
7	0.15	0.33	0.20	40
8	0.24	0.11	0.15	103
9	0.32	0.24	0.27	34
accuracy			0.35	575
macro avg	0.38	0.34	0.31	575
weighted avg	0.39	0.35	0.32	575

Εικόνα 13: Αποτελέσματα training & testing CNN δικτύου στο non-beat-synced sprectrogram datase

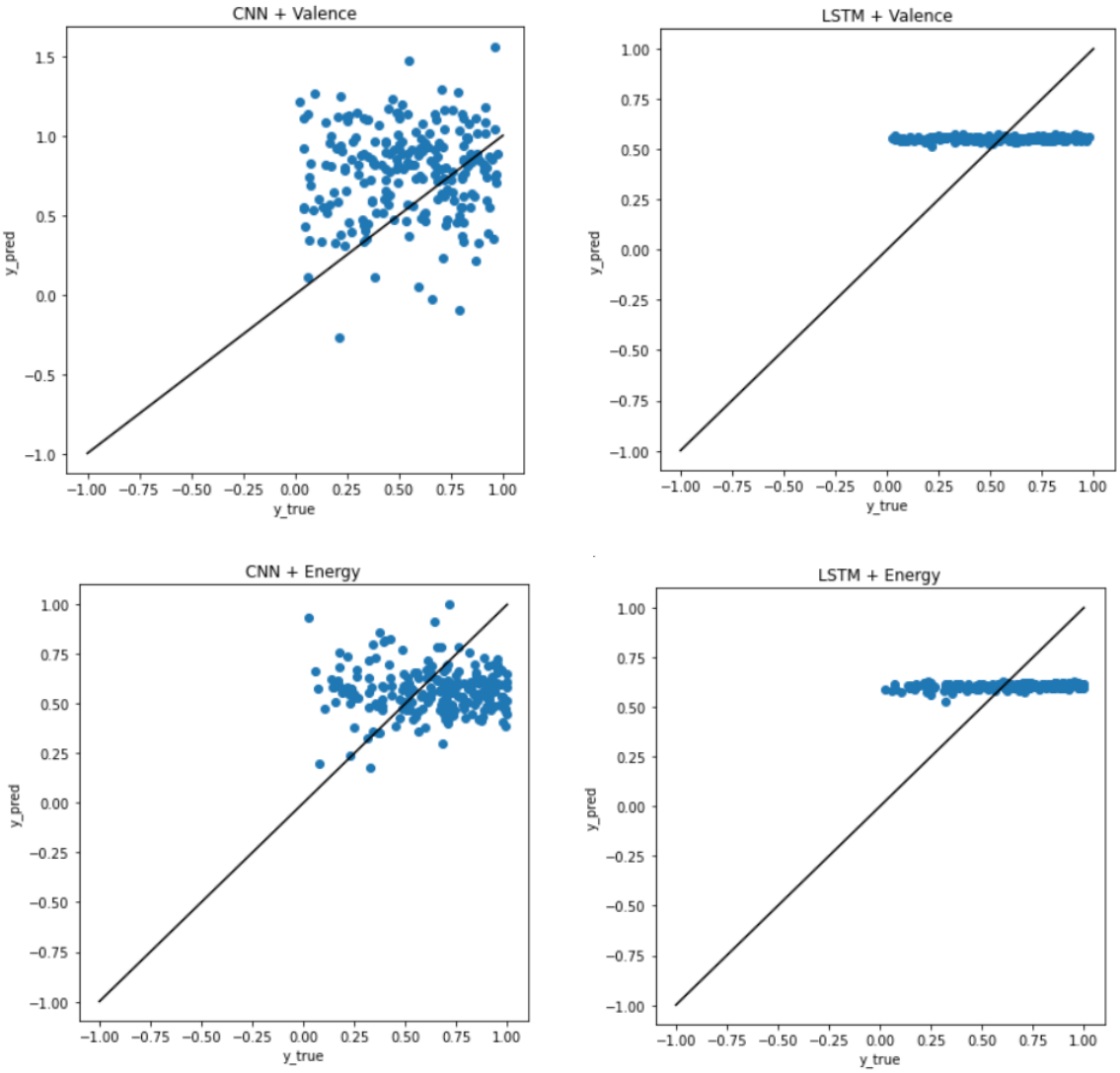
Βήμα 8: Εκτίμηση συναισθήματος – συμπεριφοράς με παλινδρόμιση

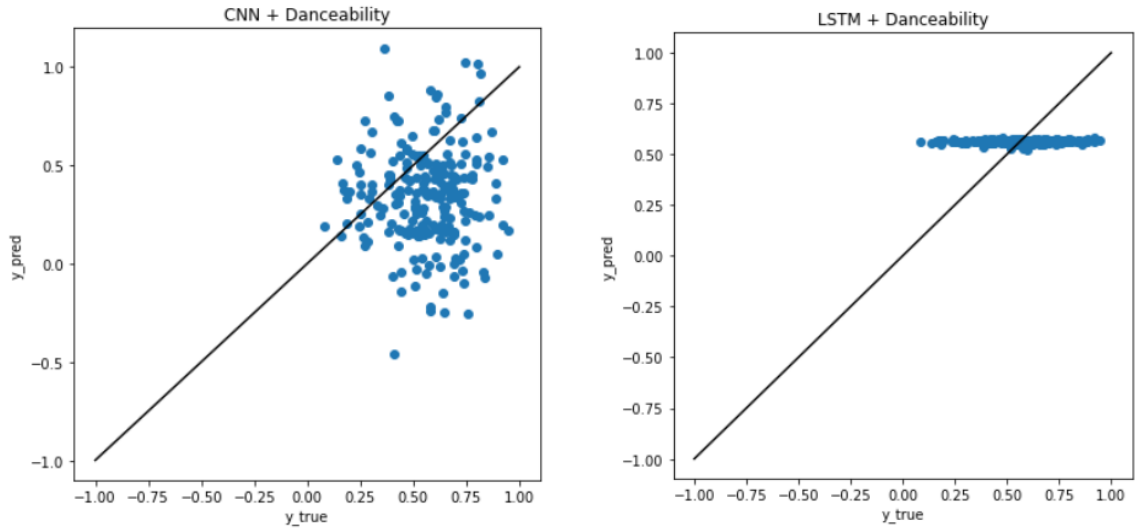
Στο βήμα αυτό έγινε χρήση του multitask dataset, και πιο συγκεκριμένα των 3 αξόνων που αφορούν το συναίσθημα του κάθε τραγουδιού (valence, energy, danceability). Για κάθε άξονα, πραγματοποιήθηκε training & validation 2 μοντέλων παλινδρόμησης, ενός τύπου CNN (όμοιο με αυτό του βήματος 7) και ενός τύπου LSTM (όμοιο με αυτό του βήματος 5). Τελική μετρική για την αξιολόγηση των 6 μοντέλων που προέκυψαν είναι το μέσο Spearman correlation μεταξύ πραγματικών και προβλεπόμενων τιμών που δίνεται από την σχέση:

$$\rho = \frac{cov\big(R(X), R(Y)\big)}{\sigma_{R(X)} * \sigma_{R(Y)}} \in [-1,1]$$

Συγκεκριμένα, όσο μεγαλύτερη είναι η τιμή της μετρικής ρ, τόσο πιο ισχυρά σχετιζόμενα είναι τα ground truth values με τα προβλεπόμενα, γεγονός που συνεπάγεται καλύτερο regression.

Σε αντίθεση με τα ερωτήματα 5 και 7, το loss function που χρησιμοποιήθηκε στο βήμα αυτό ήταν το MSELoss μιας και είναι καταλληλότερο για προβλήματα τύπου regression. Κάθε δίκτυο προπονήθηκε για 100 εποχές με τα αποτελέσματα να αναγράφονται παρακάτω:





Εικόνα 14 : Scatter plots CNN-LSTM δικτύων

	CNN	**LSTM**
Valence	0.022	0.086
Energy	−0.025	0.222
Danceability	0.004	0.152
Mean	0.0003	0.153

Πίνακας 1: Spearman correlation values για όλους του συνδυασμούς αξόνων - δικτύων

Παρατηρούμε ότι και οι 2 τύποι δικτύου αδυνατούν να πραγματοποιήσουν επαρκή γενίκευση για το regression task, έτσι ώστε να προσφέρουν ικανοποιητικά υψηλές τιμές της μετρικής ρ (>0.3). Από την μία πλευρά, τα μοντέλα τύπου CNN έχουν την τάση να δίνουν τιμές y_pred < y_true, ενώ σε ορισμένα validation danceability δεδομένα έχουμε y_pred < 0. Από την άλλη πλευρά, τα LSTM δίκτυα τείνουν να βγάζουν παρεμφερείς τιμές y_pred για όλα τα δεδομένα εισόδου (μεταξύ 0.5 και 0.75), συνεπώς η υψηλότερη μέση τιμή του spearman correlation factor δεν αντιστοιχεί σε επαρκή ικανότητα γενίκευσης του προβλήματος παλινδρόμησης.

Στην συνέχεια, θα γίνει προσπάθεια βελτίωσης των άνωθι μοντέλων με τεχνικές transfer & multitask learning.

Βήμα 9: Μεταφορά γνώσης (Transfer Learning)

Όσον αφορά τα συμπεράσματα του cited paper, αυτό αναφέρεται στην μεταφερσιμότητα των χαρακτηριστικών των νευρωνικών δικτύων σε νέα μοντέλα. Η ανάγκη για μελέτη του φαινομένου αυτού έγκειται στο γεγονός ότι τα DNNs τείνουν να αναγνωρίζουν γενικά χαρακτηριστικά στα πρώτα layers επεξεργασίας της πληροφορίας εισόδου, με την ειδίκευσή τους στο εκάστοτε dataset να πραγματοποιείται στα επόμενα layers. Έτσι, η δυνατότητα μεταφοράς παραμέτρων μεταξύ μοντέλων επηρεάζεται αρνητικά από 2 παράγοντες: την αυξημένη ειδίκευση των νευρώνων αρχικών layers στο πρωτεύον task και το «σπάσιμο» του αρχικού δικτύου σε co-adapted layer. Παρά τα άνωθι προβλήματα, οι συγγραφείς

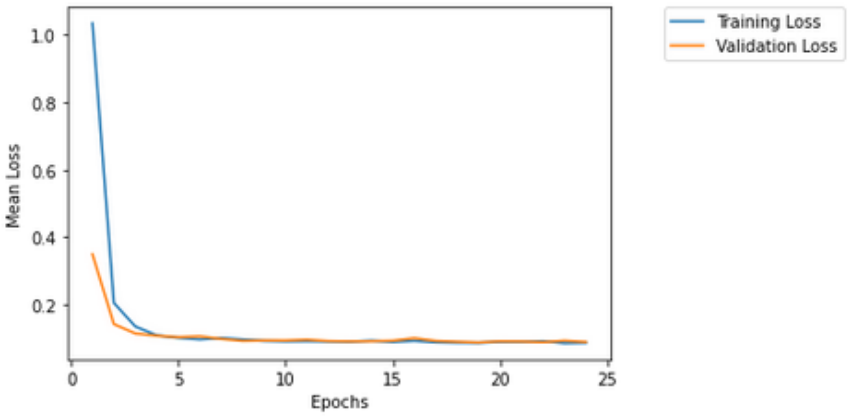
επισημαίνουν ότι το transfer learning ακόμα και από δίκτυα που ειδικεύονται σε φαινομενικά μακρινά tasks προτιμάται σε σχέση με την τυχαία αρχικοποίηση των βαρών του νέου δικτύου.

Σε παρεμφερή λογική με αυτή του paper, καλούμαστε να χρησιμοποιήσουμε το βέλτιστο classification μοντέλο, το οποίο είναι το CNN δίκτυο του βήματος 7 και να το τροποποιήσουμε κατάλληλα για να επιλύσει το regression πρόβλημα του βήματος 8. Έτσι, διατηρούμε τα αρχικά layers επεξεργασίας όπως είναι και καθιστούμε trainable μόνο τελικό linear layer του δικτύου, μιας και αυτό πρέπει να έχει 1 έξοδο αντί για 10 και κριτήριο κόστους MSELoss αντί για CrossEntropy.

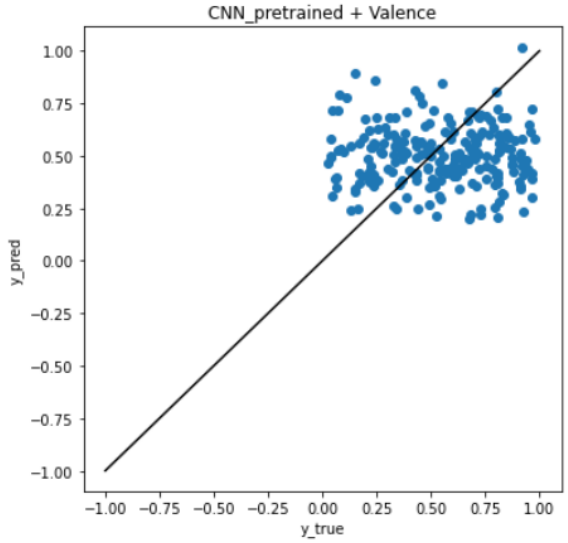
```
Classifier(
  (backbone): CNNBackbone(
    (conv1): Sequential(
      (0): Conv2d(1, 32, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
      (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU()
      (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    )
    (conv2): Sequential(
      (0): Conv2d(32, 64, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
      (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU()
      (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    )
    (conv3): Sequential(
      (0): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU()
      (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    )
    (conv4): Sequential(
      (0): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU()
      (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    )
    (fc1): Linear(in_features=163840, out_features=1000, bias=True)
  )
  (output_layer): Linear(in_features=1000, out_features=1, bias=True)
  (criterion): MSELoss()
)
```

Εικόνα 15 : Δομή CNN δικτύου με transfer learning για axis regression

Πραγματοποιήθηκε training για 50 εποχές για regression στον συναισθηματικό άξονα valence, με τα αποτελέσματα training και validation να ακολουθούν παρακάτω:



Εικόνα 16: Training & validation losses CNN δικτύου με transfer learning



Εικόνα 17 : Scatter plots CNN δικτύου με transfer learning

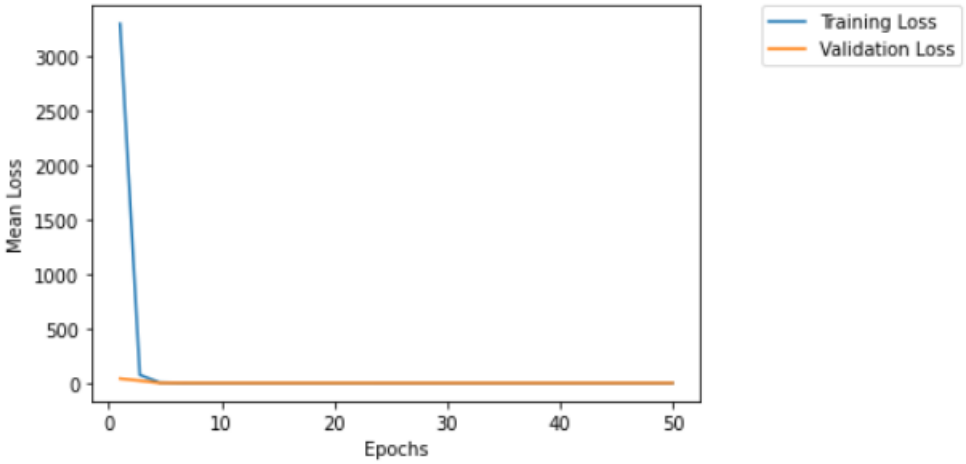
Με βάση το προκύπτον scatter plot, παρατηρούμε πως το νέο δίκτυο CNN αδυνατεί εκ νέου να ανταποκριθεί επαρκώς στο regression task, γεγονός που επαληθεύεται και από το spearman correlation value που ισούται με -0.014. Η μόνο πιθανή εξήγηση για το ανεπιτυχές validation του δικτύου έχει να κάνει με το Loss function το οποίο-μέχρι στιγμής-υπολογίζεται ανεξάρτητα για καθεμία από τους 3 άξονες. Αυτό πρόκειται να αλλάξει, ωστόσο, στο αμέσως επόμενο βήμα.

Βήμα 10: Εκπαίδευση σε πολλαπλά προβλήματα (Multitask learning)

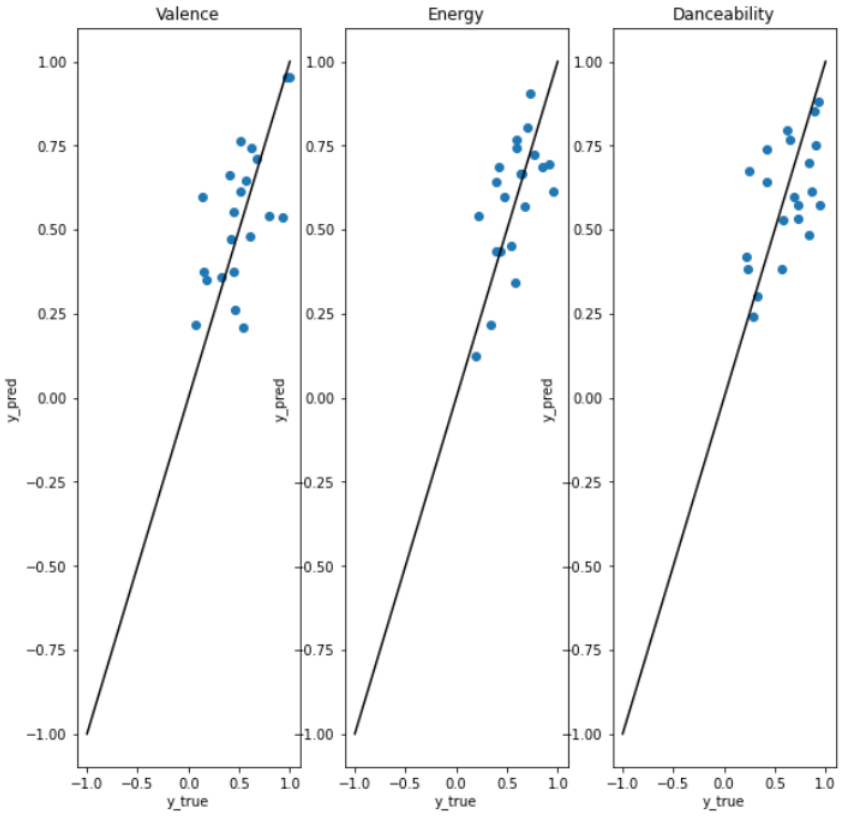
Σύμφωνα με το cited paper, υλοποιήθηκε ένα ενιαίο μοντέλο σε πολλαπλά διαφορετικά και τα αντίστοιχα datasets τους (από image captioning μέχρι speech recognition). Η απόδοση ενός τέτοιου δικτύου επωφελείται τόσο από την χρήση συγκεκριμένων computational blocks ανά task (μιας και αυτά βοηθούν και τα υπόλοιπα tasks) όσο και από την χρήση επαρκών δεδομένων ανά task (μιας και ένα task με λιγοστά δεδομένα μπορεί να «βοηθηθεί» από τα υπόλοιπα).

Με βάση την λογική του multitask learning, υλοποιήσαμε μια κοινή custom loss function για το regression πρόβλημα του βήματος 8, η οποία λαμβάνει τα logits και για τους 3 άξονες,

υπολογίζει τα MSELoss values ανά άξονα και επιστρέφει το άθροισμά τους. Έγινε προπόνηση ενός CNN δικτύου, ίδιου σε δομή με αυτό των ερωτημάτων 7 και 9 και με χρήση του συνολικού loss, με τα αποτελέσματα να παρατίθενται παρακάτω:



Εικόνα 18 : Training & validation losses CNN δικτύου με custom loss function



Εικόνα 19: Scatter plots CNN δικτύου με custom loss function

$\rho_{valence}$	ρ_{energy}	$\rho_{danc.}$	ρ_{mean}
0.557	0.658	0.522	0.579

Πίνακας 2: Spearman correlation values CNN δικτύου με custom loss function

Παρατηρούμε την εμφανή βελτίωση των αποτελεσμάτων στο regression task με την εφαρμογή multitask learning, σε σχέση με τις υλοποιήσεις των βημάτων 8 και 9. Μάλιστα, είναι η μοναδική υλοποίηση η οποία οδήγησε σε ρ values > 50% με την υψηλότερη τιμή συσχέτισης να επιτυγχάνεται στον energy axis (65.8%). Έτσι, οδηγούμαστε στο συμπέρασμα ότι η ταυτόχρονη προπόνηση του δικτύου σε διαφορετικά tasks οδηγεί στην συνολική βελτίωση των correlation scores και της ικανότητας γενίκευσής του.