Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Ροή Σ: Αναγνώριση Προτύπων

(9ο Εξάμηνο)

Εργαστηριακή Άσκηση 3

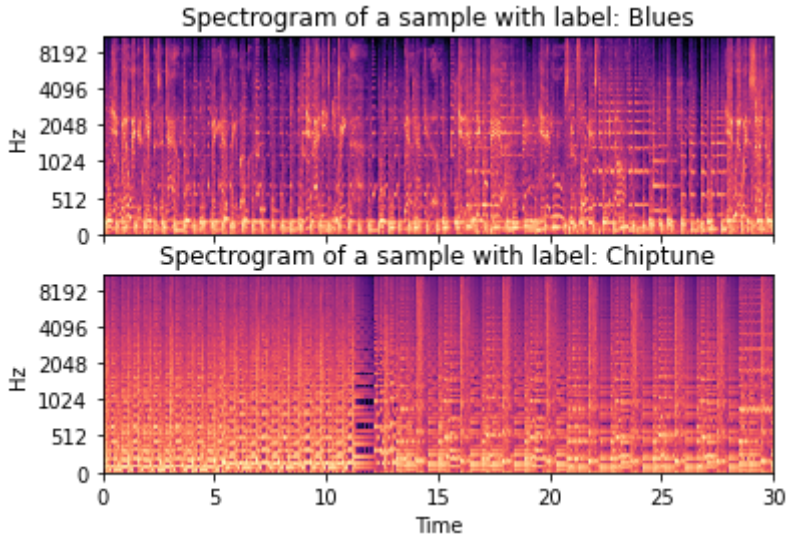
Αλέξης Μάρας 03118074

Δημήτρης Μπακάλης 03118163

Βήμα 1: Εξοικείωση με φασματογραφήματα στην κλίμακα mel

Το FMA (Free Music Archive) είναι μια βάση δεδομένων με 3834 δείγματα μουσικής χωρισμένα σε 20 κατηγορίες. Σε πρώτη φάση, θα επιλέξουμε 2 τυχαία δείγματα από 2 διαφορετικές κατηγορίες και θα εμφανίσουμε τα φασματογραφήματα τους, προκειμένου να σχολιάσουμε τις πληροφορίες που αυτά μας δίνουν.

Τα 2 φασματογραφήματα, που βλέπουμε στην παρακάτω εικόνα ανήκουν στις μουσικές κατηγορίες Blues και Chiptune.



Μέσω του φασματογραφήματος, μπορούμε να λάβουμε πληροφορίες σχετικά με το χρονο-συχνοτικό περιεχόμενο του μουσικού σήματος. Έτσι, μπορούμε καθόλη την διάρκεια του, να γνωρίζουμε την ένταση των συχνοτικών συνιστωσών του.

Από τα παραπάνω διαγράμματα, παρατηρούμε ότι το Chiptune κομμάτι, λαμβάνει μεγαλύτερες τιμές (έχει μεγαλύτερη ένταση) σε όλο το εύρος των φασματικών συχνοτήτων, σε αντίθεση με το κομμάτι της Blues, όπου φαίνεται να συγκεντρώνει το μεγαλύτερο μέρος του συχνοτικού του περιεχομένου στις χαμηλές συχνότητες.

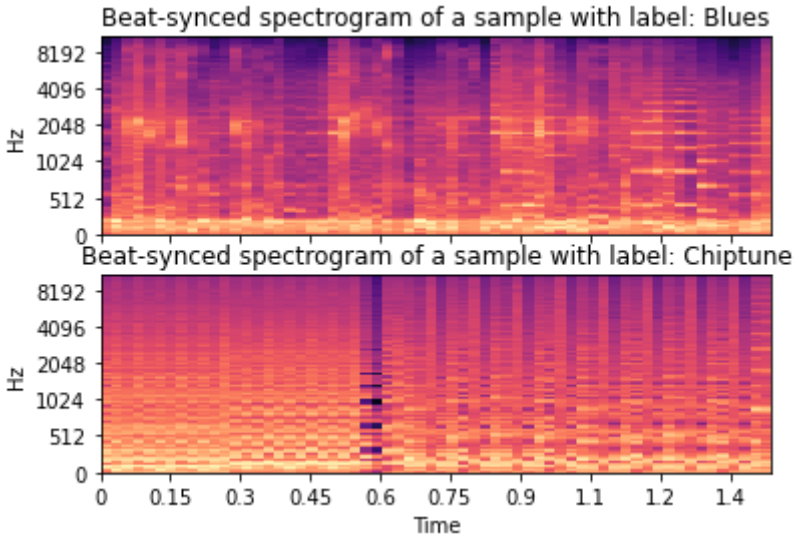
Βήμα 2: Συγχρονισμός φασματογραφημάτων στο ρυθμό της μουσικής (beat-synced spectrograms)

Από το Shape των δεδομένων που εμφανίσαμε (και τα 2 είναι 2d πίνακες μεγέθους (128,1293)), παρατηρούμε ότι το κάθε μουσικό δείγμα έχει χωριστεί σε 1293 χρονικά βήματα. Δεν θα ήταν, λοιπόν, καθόλου αποδοτικό από άποψη χρόνου και υπολογιστικού φόρτου (και μνήμης) να εκπαιδεύαμε ένα LSTM για την αναγνώριση

του είδους μουσικής των κομματιών, χρησιμοποιώντας το δεδομένο dataset, χωρίς πρώτα να το τροποποιήσουμε.

Για να περιορίσουμε τον αριθμό των χρονικών βημάτων, παίρνουμε την διάμεσο (median) ανάμεσα στα σημεία, που χτυπάει το beat της μουσικής, συγχρονίζοντας έτσι τα φασματογραφήματα πάνω στον ρυθμό της μουσικής.

Το αποτέλεσμα που λαμβάνουμε είναι το εξής:

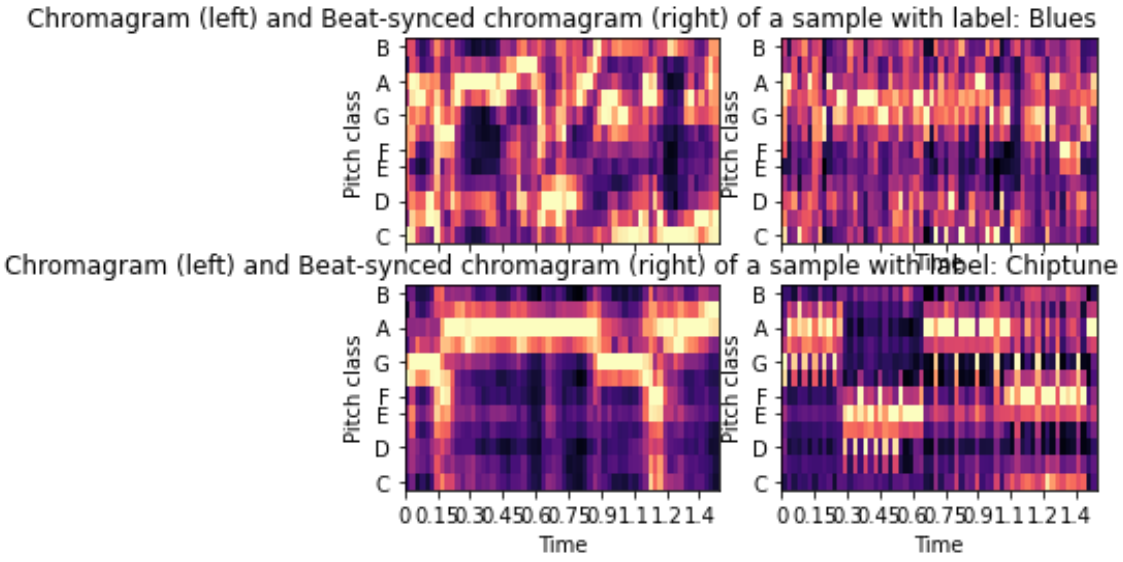


Τα νέα συγχρονισμένα φασματογραφήματα είναι μεγέθους (128,62). Παρατηρούμε ότι με αυτόν τον τρόπο μειώσαμε την ποιότητα - ανάλυση των φασματογραφημάτων, αφαιρώντας μεγάλο αριθμό από τις λεπτομέρειες του αρχικού. Παρόλα αυτά, δεν χάνεται κάποια ουσιαστική πληροφορία από το περιεχόμενο τους. Επομένως, χρησιμοποιώντας τα νέα αυτά φασματογραφήματα, μπορούμε να μειώσουμε σημαντικά την διάρκεια εκπαίδευσης, χωρίς ταυτόχρονα να επηρεάσουμε την επίδοση του μοντέλου μας.

Βήμα 3: Εξοικείωση με χρωμογραφήματα

Τα χρωμογραφήματα παρέχουν πληροφορίες σχετικά με την ενέργεια των συχνοτήτων που αντιστοιχούν στις 12 μουσικές νότες της κλίμακας κλασικής μουσικής. Επιπρόσθετα, μπορούν να αξιοποιηθούν, προκειμένου να εξάγουμε αρμονικά και μελωδικά χαρακτηριστικά, ενώ τέλος συμβάλλουν στην αναγνώριση των αλλαγών των ηχοχρωμάτων και των οργάνων μουσικής.

Τα χρωμογραφήματα των 2 μουσικών δειγμάτων, που απεικονίσαμε και προηγουμένως είναι τα εξής:



Παρατηρούμε ότι το κομμάτι της Blues, έχει πιο έντονο περιεχόμενο σε όλο το εύρος του χρωμογραφήματος του σε σχέση με το κομμάτι Chiptune, το οποίο είναι λογικό, λόγω των περισσότερων αλλαγών που παρατηρούνται στην μελωδία και τα όργανα μεταξύ των 2 ειδών μουσικής.

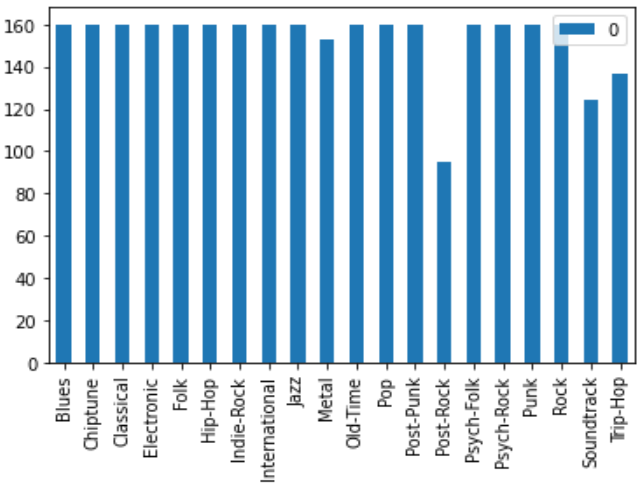
Βήμα 4: Φόρτωση και ανάλυση δεδομένων

Σε αυτό το σημείο, εισάγουμε τα train και test δεδομένα με την χρήση της κλάσης SpectogramDataset. Μάλιστα, μας δίνεται η δυνατότητα, κατά την αρχικοποίηση της κλάσης να λάβουμε είτε το φασματογράφημα του κάθε δείγματος, είτε το χρωμογράφημα, είτε και τα 2 μαζί.

Επίσης, παρέχει την δυνατότητα Zero Padding στα δεδομένα, έτσι ώστε να έχουν όλα το ίδιο μέγεθος (ίδιο αριθμό χρονικών βημάτων), προτού ξεκινήσουμε την διαδικασία της εκπαίδευσης.

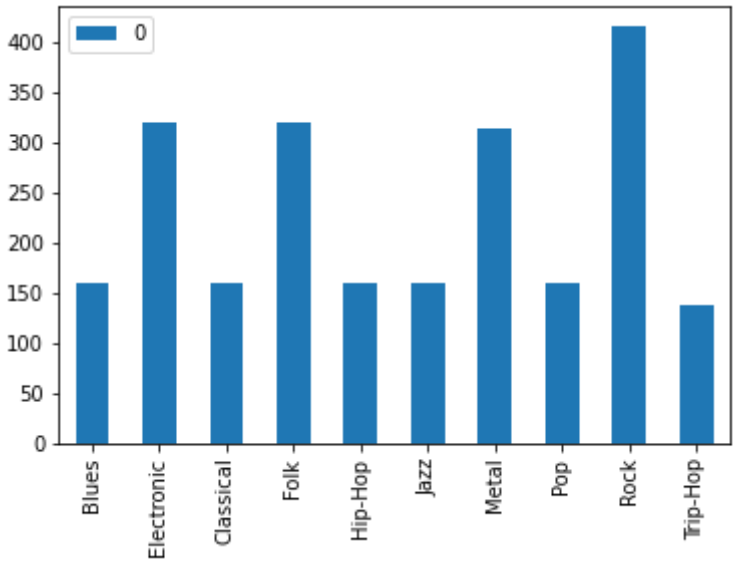
Τέλος, χρησιμοποιεί έναν Label Transformer, έτσι ώστε να κωδικοποιήσει τις κλάσεις του συνόλου δεδομένων, δίνοντας τους αριθμητικές τιμές, το οποίο είναι απαραίτητο για την εκτίμηση του σφάλματος (loss) στην συνέχεια.

Το ιστόγραμμα με τον αριθμό των δεδομένων που ανήκουν σε κάθε μία από τις κλάσεις, για το σύνολο εκπαίδευσης φαίνεται στην παρακάτω εικόνα:



Λόγω του μεγάλου αριθμού κλάσεων, επιλέγουμε να συγχωνεύσουμε κατηγορίες, των οποίων τα δείγματα εμφανίζουν πολλές ομοιότητες μεταξύ τους, ενώ ταυτόχρονα αφαιρούμε εκείνες που αντιπροσωπεύονται από πολύ λίγα δείγματα. Με αυτόν τον τρόπο, επιταχύνουμε την διαδικασία εκπαίδευσης του δικτύου και επιπλέον αυξάνουμε την ακρίβεια του, καθώς κλάσεις με πολλά όμοια χαρακτηριστικά είναι δύσκολο να τις διαχωρίσουμε μεταξύ τους, ενώ εκείνες με μικρό αριθμό δειγμάτων είναι δύσκολο να τις προβλέψουμε χωρίς την χρήση κάποιου είδους τεχνικής εξισορρόπησης του αριθμού των δεδομένων των κλάσεων. Την ίδια διαδικασία επαναλαμβάνουμε και για τα testing δεδομένα.

Το νέο ιστόγραμμα, που προκύπτει μετά την συγχώνευση & αφαίρεση κλάσεων είναι το εξής:

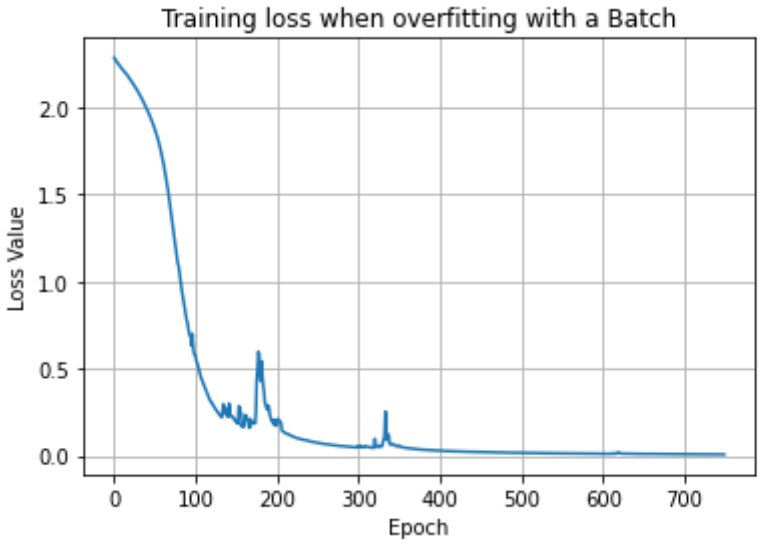


Βήμα 5: Αναγνώριση μουσικού είδους με LSTM

Για το πρόβλημα της ταξινόμησης των δειγμάτων στις νέες κλάσεις, που είδαμε προηγουμένως, θα χρησιμοποιήσουμε το LSTM της προηγούμενης σειράς ασκήσεων. Πιο συγκεκριμένα, θα χρησιμοποιήσουμε ένα Bidirectional LSTM, με 3 Hidden Layers, μεγέθους rnn_size = 100, καθώς και με την δυνατότητα Early Stopping σε περίπτωση που δεν παρατηρείται βελτίωση του validation loss, κατά την διάρκεια της εκπαίδευσης.

Ένας τρόπος για να βεβαιωθούμε ότι το νευρωνικό μας μπορεί να εκπαιδευτεί είναι μέσω της υπερεκπαίδευσης του δικτύου σε ένα batch. Εκπαιδεύοντας το δίκτυο για πολλές εποχές, αναμένουμε το σφάλμα εκπαίδευσης να τείνει προς το 0. Για αυτόν τον λόγο, προσθέτουμε μία boolean παράμετρο overfit_batch, όταν δημιουργούμε το μοντέλο, η οποία όταν είναι False, το δίκτυο εκπαιδεύεται κανονικά, ενώ όταν είναι True, το δίκτυο εκπαιδεύεται μονάχα με ένα batch.

Για την περίπτωση του Overfit Batch, η γραφική του training loss συναρτήσει των εποχών είναι η εξής:



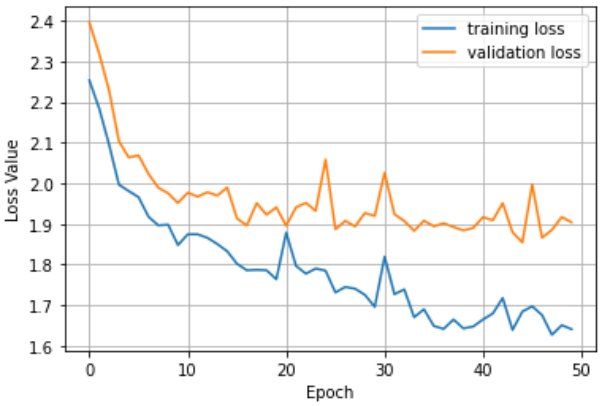
Παρατηρούμε, ότι καθώς αυξάνεται ο αριθμός των εποχών, το σφάλμα εκπαίδευσης ολοένα και μειώνεται και τείνει προς το 0, επιβεβαιώνοντας ότι μπορεί να “μαθαίνει”.

Σε αυτό το σημείο θα εκπαιδεύσουμε 4 LSTM δίκτυα, για να προβλέπουμε τις 10 διαφορετικές κλάσεις - μουσικά είδη του συνόλου δεδομένων FMA . Σε κάθε ένα από αυτά, θα χρησιμοποιήσουμε ως σύνολα εκπαίδευσης διαφορετικά δεδομένα.

Η εκπαίδευση του κάθε LSTM γίνεται για 50 συνολικά εποχές, με δυνατότητα Early Stopping. Ως optimizer χρησιμοποιείται ο αλγόριθμος Adam, ο οποίος αν και υπολογιστικά ακριβός, δίνει πολύ καλά αποτελέσματα και επιπλέον συγκλίνει γρήγορα. Learning Step μετά από δοκιμές επιλέγεται η τιμή 0.0001, ενώ τέλος ως Loss Function, χρησιμοποιείται η Cross Entropy, μίας και το πρόβλημα μας αφορά την ταξινόμηση σε πολλαπλές κατηγορίες.

(α) Εκπαίδευση με χρήση των αρχικών φασματογραφημάτων

Οι γραφικές παραστάσεις των training και validation loss δίνονται στο παρακάτω διάγραμμα:

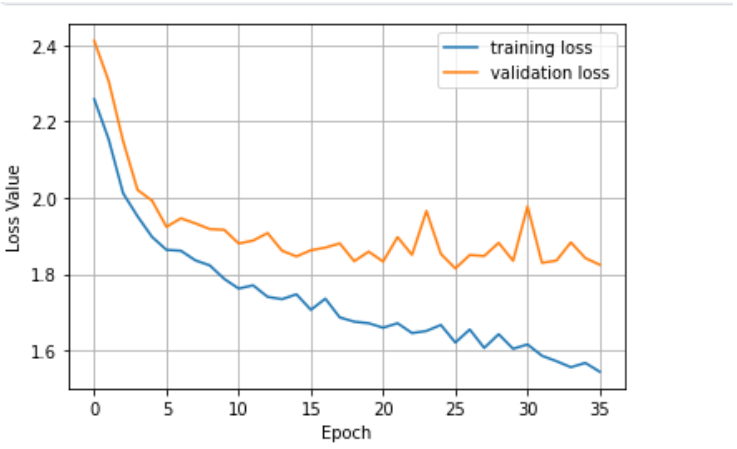


Το τελικό loss στα training και validation δεδομένα είναι το εξής:

- ☐ Training Loss: 1.656
- ☐ Validation Loss: 1.903

(β) Εκπαίδευση με χρήση των beat-synced φασματογραφημάτων

Οι γραφικές παραστάσεις των training και validation loss δίνονται στο παρακάτω διάγραμμα:

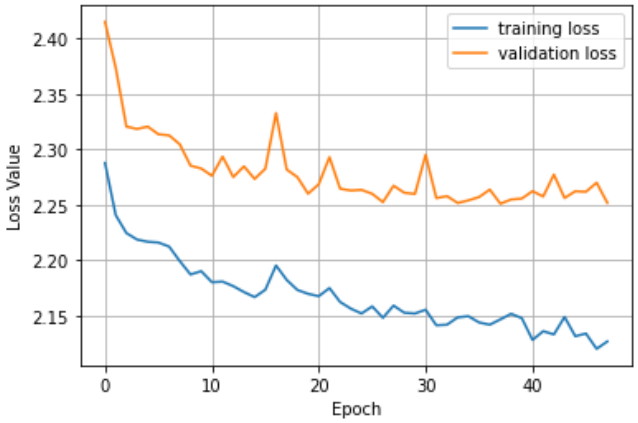


Το τελικό loss στα training και validation δεδομένα είναι το εξής:

- ☐ Training Loss: 1.552
- ☐ Validation Loss:1.824

(γ) Εκπαίδευση με χρήση των χρωμογραφημάτων

Οι γραφικές παραστάσεις των training και validation loss δίνονται στο παρακάτω διάγραμμα:

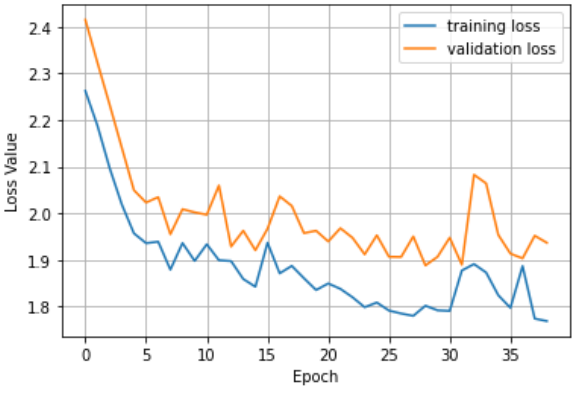


Το τελικό loss στα training και validation δεδομένα είναι το εξής:

- ☐ Training Loss: 2.006
- ☐ Validation Loss: 2.225

(δ) Εκπαίδευση με χρήση των ενωμένων φασματογραφημάτων & χρωμογραφημάτων

Οι γραφικές παραστάσεις των training και validation loss δίνονται στο παρακάτω διάγραμμα:



Το τελικό loss στα training και validation δεδομένα είναι το εξής:

- ☐ Training Loss: 1.784
- ☐ Validation Loss: 1.936

Βήμα 6: Αξιολόγηση των μοντέλων

Για την αξιολόγηση των LSTM, που δημιουργήσαμε, θα χρησιμοποιήσουμε τα testing dataset, που μας δίνονται και θα ελέγξουμε την επίδοση των μοντέλων μας. Για τον έλεγχο της αποτελεσματικότητας τους, θα χρησιμοποιήσουμε τις εξής μετρικές:

- ☐ **accuracy**: Μας δείχνει το ποσοστό των επιτυχημένων προβλέψεων του μοντέλου και συνιστά έναν πολύ απλό και αποτελεσματικό τρόπο, για να κρίνουμε την απόδοση του μοντέλου.

- ☐ **precision**: Είναι μια ποσότητα ενδεικτική του αριθμού των δειγμάτων, που ταξινομήθηκαν λανθασμένα σε κάθε μία από τις διαφορετικές κλάσεις.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

- ☐ **recall**: Είναι παρόμοια με την precision, με την διαφορά ότι είναι ενδεικτική του αριθμού των δειγμάτων, που δεν ταξινομήθηκαν σε μία συγκεκριμένη κλάση, ενώ θα έπρεπε.

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

- ☐ **f1-score**: Αποτελεί τον στατιστικό μέσο όρο των μετρικών precision και recall, δίνοντας μας μία γενικότερη εικόνα για την αποτελεσματικότητα του μοντέλου να προβλέπει τα δείγματα μίας συγκεκριμένης κλάσης.

- ☐ **macro - averaged**: Υπολογίζει τον μέσο όρο των αποτελεσμάτων κάθε μετρικής, δίνοντας την ίδια βαρύτητα σε κάθε κατηγορία, χωρίς να λαμβάνει υπόψη την πιθανότητα ύπαρξης imbalanced δεδομένων.

- ☐ **micro - averaged**: Υπολογίζει την μέση τιμή των True Positives, False Positives και False Negatives και με αυτόν τον τρόπο λαμβάνει υπόψη και τα πιθανά imbalances μεταξύ των κλάσεων.

Ο λόγος για τον οποίο εισάγουμε αυτές τις επιπλέον μετρικές, είναι έτσι ώστε να λάβουμε μια πιο αντικειμενική συνολική αξιολόγηση του μοντέλου.

Για παράδειγμα, είναι αρκετά πιθανό να υπάρχουν αποκλίσεις μεταξύ των accuracy και f1 - score, σε περιπτώσεις που τα δεδομένα εκπαίδευσης και ελέγχου είναι imbalanced. Το μοντέλο δίνει έμφαση στην κλάση με τα περισσότερα δείγματα, ενώ τείνει να αγνοεί τις κλάσεις μειονότητας. Επομένως, ενώ το accuracy μπορεί να είναι ικανοποιητικό, το f1 - score θα λαμβάνει χαμηλότερες τιμές, καθιστώντας το πιο κατάλληλο για την αξιολόγηση του μοντέλου. Λόγω της ύπαρξης imbalanced δεδομένων μεταξύ των διαφόρων κλάσεων, σημαντικές αποκλίσεις μπορεί να παρατηρηθούν μεταξύ των macro - averaged και micro - averaged μετρικών.

Ιδιαίτερη έμφαση δίνουμε στις μετρικές precision και recall σε εφαρμογές της βιοϊατρικής, όπως στην διάγνωση αρρυθμιών και καρκίνων. Μάλιστα, σε αυτές τις περιπτώσεις, μεγαλύτερη έμφαση δίνεται στην βελτίωση της μετρικής recall, καθώς το ρίσκο του false negative (να μην έχει καρκίνο πχ ο ασθενής) είναι πολύ μεγαλύτερο από το αντίστοιχο false positive της precision.

Στο πρόβλημα που εξετάζουμε, μεγαλύτερη έμφαση θα δώσουμε κυρίως στην micro - averaged μετρική, η οποία μας δίνει και τα πιο αντικειμένικα αποτελέσματα. Ακόμη, μία σχετικά καλή εικόνα για την επίδοση του μοντέλου μας, μπορούμε να πάρουμε και από το accuracy.

Παρακάτω παραθέτουμε τα classification reports, που περιέχουν την τιμή των παραπάνω μετρικών, για κάθε ένα από τα 4 LSTM.

(α) LSTM που εκπαιδεύτηκε με φασματογραφήματα :

	precision	recall	f1-score	support
0	0.22	0.05	0.08	40
1	0.43	0.57	0.49	40
2	0.38	0.54	0.45	80
3	0.32	0.53	0.40	80
4	0.62	0.20	0.30	40
5	0.24	0.10	0.14	40
6	0.40	0.37	0.39	78
7	0.00	0.00	0.00	40
8	0.38	0.51	0.44	103
9	0.12	0.09	0.10	34
accuracy			0.36	575
macro avg	0.31	0.30	0.28	575
weighted avg	0.33	0.36	0.33	575

(β) LSTM που εκπαιδεύτηκε με beat - synced φασματογραφήματα :

	precision	recall	f1-score	support
0	0.00	0.00	0.00	40
1	0.36	0.72	0.48	40
2	0.23	0.80	0.36	80
3	0.56	0.11	0.19	80
4	0.33	0.03	0.05	40
5	0.18	0.15	0.16	40
6	0.62	0.23	0.34	78
7	0.00	0.00	0.00	40
8	0.38	0.46	0.41	103
9	0.25	0.06	0.10	34
accuracy			0.31	575
macro avg	0.29	0.26	0.21	575
weighted avg	0.34	0.31	0.25	575

(γ) LSTM που εκπαιδεύτηκε με χρωμογραφήματα :

	precision	recall	f1-score	support
0	0.00	0.00	0.00	40
1	0.00	0.00	0.00	40
2	0.20	0.29	0.24	80
3	0.21	0.65	0.32	80
4	1.00	0.03	0.05	40
5	0.00	0.00	0.00	40
6	0.34	0.14	0.20	78
7	0.00	0.00	0.00	40
8	0.23	0.41	0.29	103
9	0.00	0.00	0.00	34
accuracy			0.22	575
macro avg	0.20	0.15	0.11	575
weighted avg	0.21	0.22	0.16	575

(δ) LSTM που εκπαιδεύτηκε με ενωμένα φασματογραφήματα - χρωμογραφήματα :

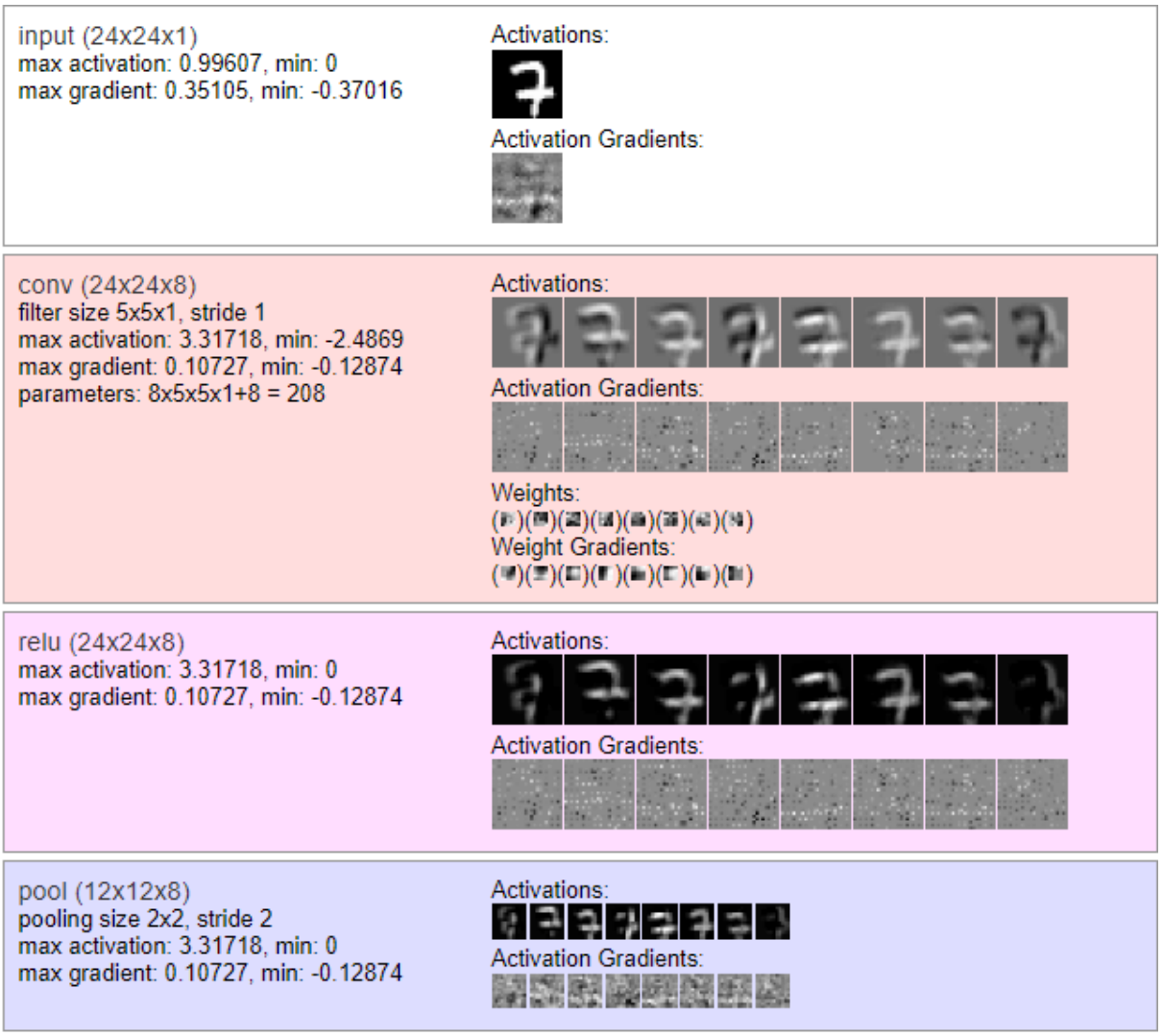
	precision	recall	f1-score	support
0	0.00	0.00	0.00	40
1	0.30	0.57	0.40	40
2	0.32	0.66	0.43	80
3	0.33	0.59	0.42	80
4	0.17	0.33	0.23	40
5	0.25	0.03	0.05	40
6	0.50	0.01	0.03	78
7	0.00	0.00	0.00	40
8	0.37	0.29	0.33	103
9	0.14	0.12	0.13	34
accuracy			0.30	575
macro avg	0.24	0.26	0.20	575
weighted avg	0.28	0.30	0.23	575

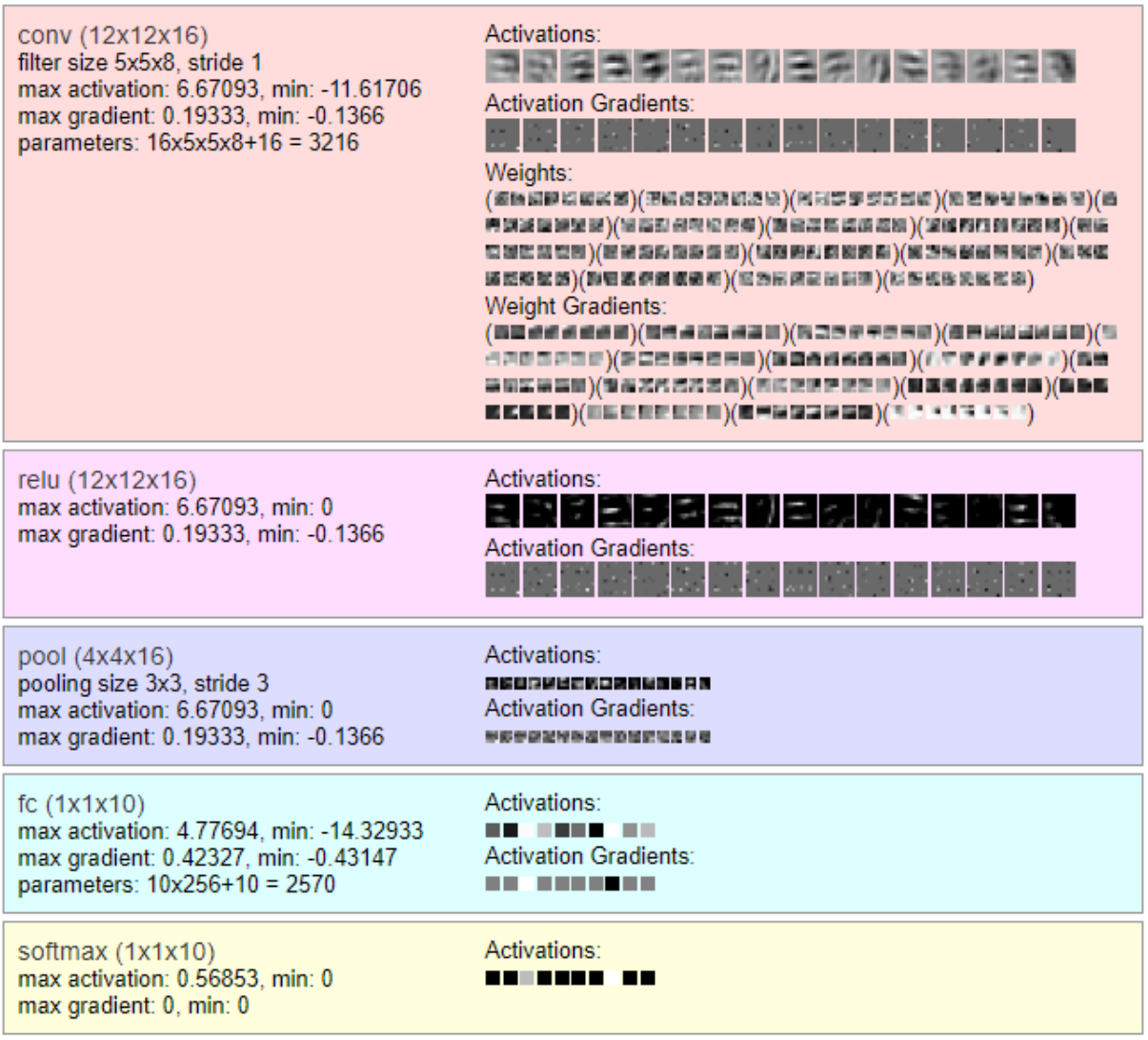
Παρατηρώντας λοιπόν, τις μετρικές weighted avg και accuracy, που μας ενδιαφέρουν πιο πολύ στο συγκεκριμένο πρόβλημα, παρατηρούμε ότι για την ταξινόμηση των FMA αρχείων, καλύτερα αποτελέσματα μας δίνει το πρώτο LSTM, που εκπαιδεύτηκε στα φασματογραφήματα (χωρίς τα χρωμογραφήματα).

Βήμα 7: 2D CNN

Προτού υλοποιήσουμε το δικό μας συνελικτικό νευρωνικό δίκτυο, για τις ανάγκες του προβλήματος ταξινόμησης του FMA, θα δούμε μέσω του MNIST την εσωτερική λειτουργία ενός CNN, οπτικοποιώντας τις ενεργοποιήσεις των επιμέρους επιπέδων.

Για την οπτικοποίηση αυτή, το MNIST χρησιμοποιεί ένα σύνολο δεδομένων, αποτελούμενο από ψηφία 0 - 9, τα οποία είναι γραμμένα χειρόγραφα και επιχειρεί να τα ταξινομήσει στην σωστή κατηγορία.





Από τις παραπάνω εικόνες, παρατηρούμε ότι με την βοήθεια των συνελικτικών επιπέδων, εξάγουμε χαρακτηριστικά σχετικά με τα δεδομένα εισόδου. Τέτοια χαρακτηριστικά είναι οι ακμές των ψηφίων ή το πάχος τους. Με αυτόν τον τρόπο, διευκολύνεται η ταξινόμηση των δειγμάτων στις επιμέρους κατηγορίες. Βέβαια με την χρήση επιπλέον συνελικτικών επιπέδων, το μοντέλο γίνεται πιο σύνθετο και εξάγει επιπλέον χαρακτηριστικά για την κάθε κλάση, τα οποία ωστόσο δεν μπορούν να οπτικοποιηθούν αποτελεσματικά, όπως φαίνεται και στις παραπάνω εικόνες.

Το CNN, που θα υλοποιήσουμε θα αποτελείται από επίπεδα, το καθένα από τα οποία θα επιτελεί τις παρακάτω λειτουργίες:

2D Convolution: Θεωρώντας μια εικόνα (φασματογράφημα), που λαμβάνουμε ως είσοδο, ως έναν πίνακα από pixels. Η λειτουργία αυτού του επιπέδου είναι να εφαρμόσουμε ένα φίλτρο (ή kernel), ώστε να παράγουμε το Feature Map (ή Convolved Feature). Στόχος μας με αυτή την διαδικασία, είναι να εξάγουμε

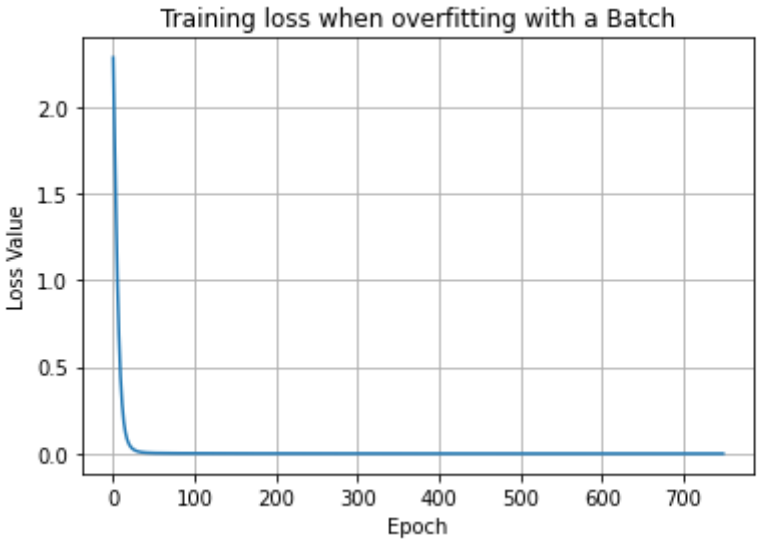
χαρακτηριστικά για την εικόνα και να κατανοήσουμε περισσότερο τις συσχετίσεις μεταξύ των pixels. Αξίζει να σημειωθεί πως μπορούμε να χρησιμοποιήσουμε πολλαπλά φίλτρα, για να έχουμε πολλαπλά Feature Maps και κατ' επέκταση καλύτερα αποτελέσματα.

Batch Normalization: Στο συγκεκριμένο βήμα κανονικοποιούμε τα δεδομένα του κάθε batch, με σκοπό να επιταχύνουμε την διαδικασία την εκπαίδευσης του μοντέλου μας.

ReLU Activation: Η συνάρτηση ενεργοποίησης ReLU (ReLU(x) = max{0, x} συντελεί στην προσθήκη μη-γραμμικότητας στο CNN μας, καθώς τα πραγματικά δεδομένα συνήθως δεν παρουσιάζουν γραμμικότητα (και η συνέλιξη είναι γραμμικός μετασχηματισμός).

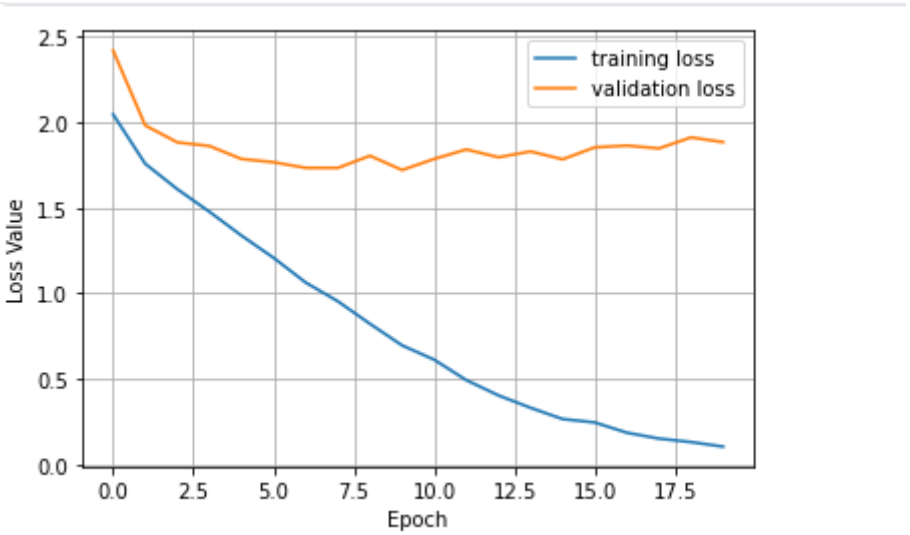
Max Pooling: Στο max pooling (ή downsampling) κρατάμε μόνο τα pixels με τις μεγαλύτερες τιμές, με σκοπό την συμπίεση της εικόνας. Με αυτή την τεχνική χάνουμε πληροφορία, αλλά βελτιώνουμε σημαντικά την υπολογιστική πολυπλοκότητα του CNN. Μάλιστα, λόγω της περιορισμένης πληροφορίας που διατηρείται, πολλές φορές το max pooling βοηθά στην αποφυγή του overfit. Επομένως, πρόκειται για ένα trade-off μεταξύ σε χρόνο και ακρίβεια δεδομένων.

Πραγματοποιώντας την διαδικασία overfit_batch διαπιστώνουμε πως μετά τις πρώτες 20 εποχές το loss του μοντέλου μας μηδενίζεται και, επομένως, το μοντέλο μας μπορεί να εκπαιδευτεί. Παρακάτω παρατίθεται και το διάγραμμα με το loss για τις πρώτες 750 εποχές:



Επαναλαμβάνοντας τη διαδικασία που ακολουθήσαμε για την εκπαίδευση του LSTM στο βήμα 5, έχουμε τα παρακάτω αποτελέσματα για το loss στις πρώτες 50 εποχές (ενεργοποιείται και εδω πολλές φορές το Early Stopping). Για κάθε ένα από τα CNN, θα πρέπει να προσαρμόζουμε τις εισόδους, ώστε να μην εμφανίζεται error:

(α) Εκπαίδευση με χρήση των αρχικών φασματογραφημάτων



Το τελικό loss στα training και validation δεδομένα είναι το εξής:

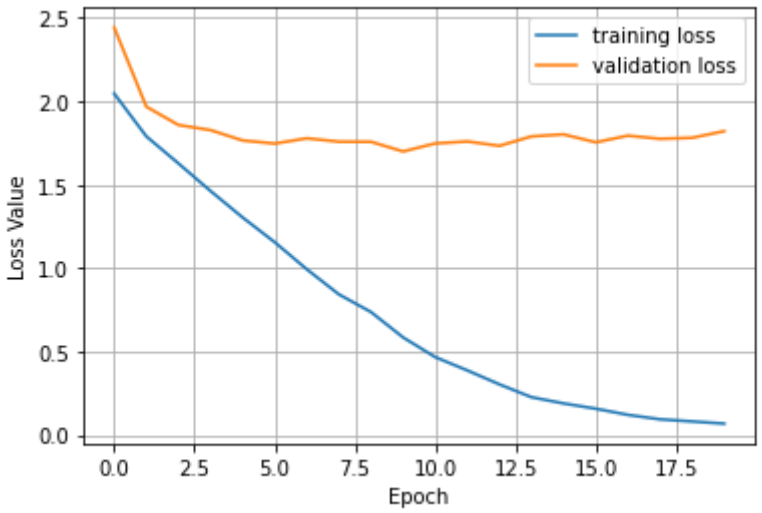
- ☐ Training Loss: 0.112
- ☐ Validation Loss: 1.883

Το classification report με τα αποτελέσματα στα test δεδομένα είναι:

	precision	recall	f1-score	support
0	0.19	0.07	0.11	40
1	0.61	0.50	0.55	40
2	0.47	0.45	0.46	80
3	0.43	0.46	0.44	80
4	0.35	0.35	0.35	40
5	0.16	0.12	0.14	40
6	0.55	0.60	0.58	78
7	0.06	0.03	0.04	40
8	0.34	0.56	0.42	103
9	0.39	0.21	0.27	34
accuracy			0.40	575
macro avg	0.35	0.34	0.34	575
weighted avg	0.38	0.40	0.38	575

(β) Εκπαίδευση με χρήση των ενωμένων φασματογραφημάτων - χρωμογραφημάτων

Οι γραφικές παραστάσεις των training και validation loss δίνονται στο παρακάτω διάγραμμα:



Το τελικό loss στα training και validation δεδομένα είναι το εξής:

- ☐ Training Loss: 0.048
- ☐ Validation Loss: 1.821

Το classification report με τα αποτελέσματα στα test δεδομένα είναι:

	precision	recall	f1-score	support
0	0.43	0.07	0.13	40
1	0.37	0.60	0.46	40
2	0.49	0.49	0.49	80
3	0.36	0.51	0.42	80
4	0.44	0.40	0.42	40
5	0.17	0.12	0.14	40
6	0.52	0.47	0.50	78
7	0.00	0.00	0.00	40
8	0.38	0.54	0.44	103
9	0.47	0.26	0.34	34
accuracy			0.40	575
macro avg	0.36	0.35	0.33	575
weighted avg	0.38	0.40	0.37	575

Παραλείπεται η εκπαίδευση CNN, με τα beat synced spectrograms και με τα χρωμογραφήματα, καθώς όπως είδαμε και προηγουμένως, εμφανίζουν χαμηλή απόδοση. Επιπλέον, στα άλλα 2 set δεδομένων, οι τιμές των weighted avg και accuracy, στις οποίες δίνουμε έμφαση, είναι παρόμοιες.

Παρατηρούμε πως η μείωση του training loss με χρήση CNN, είναι αισθητά μεγαλύτερη σε σχέση με LSTM, χωρίς, ωστόσο αυτή η διαφορά να αποτυπώνεται και ούτε στο accuracy ούτε και στο validation loss/accuracy, το οποίο παρουσιάζει μικρότερη αύξηση (και στα 2 set δεδομένων).

Βήμα 8: Εκτίμηση συναισθήματος - συμπεριφοράς με παλινδρόμηση

Στο συγκεκριμένο βήμα θα πραγματοποιήσουμε μερικές αλλαγές στα μοντέλα των βημάτων 5, 7, καθώς πλέον καλούμαστε να εκτιμήσουμε 3 παραμέτρους (valence, energy, danceability) με πεδίο τιμών [0,1]. Επομένως, το πρόβλημα μετατρέπεται από classification σε regression. Για να γίνει η παραπάνω αλλαγή, αλλάζουμε τη συνάρτηση κόστους των νευρωνικών από cross entropy σε MSE (||y' - y||^2). Η τελική μετρική για το evaluation του κάθε μοντέλου είναι η μέση τιμή των spearman correlations (περιγράφεται παρακάτω), που προκύπτει για κάθε μία από τις 3 παραμέτρους.

Spearman Correlation: Είναι μία μετρική, η οποία εκφράζει την συσχέτιση μεταξύ δύο μεταβλητών (όσο πιο κοντά στο +-1 τόσο μεγαλύτερη συσχέτιση) και περιγράφεται από την παρακάτω μαθηματική σχέση:

$$r_s = \rho_{\mathrm{R}(X),\mathrm{R}(Y)} = \frac{\mathrm{cov}(\mathrm{R}(X), \mathrm{R}(Y))}{\sigma_{\mathrm{R}(X)} \sigma_{\mathrm{R}(Y)}}$$

Στο πρόβλημα μας, υπολογίζουμε το Spearman Correlation, μεταξύ των προβλεπόμενων από το μοντέλο τιμών, και των πραγματικών. Για ευνόητους λόγους επιθυμούμε ιδανικά η τιμή αυτή να τείνει προς την μονάδα.

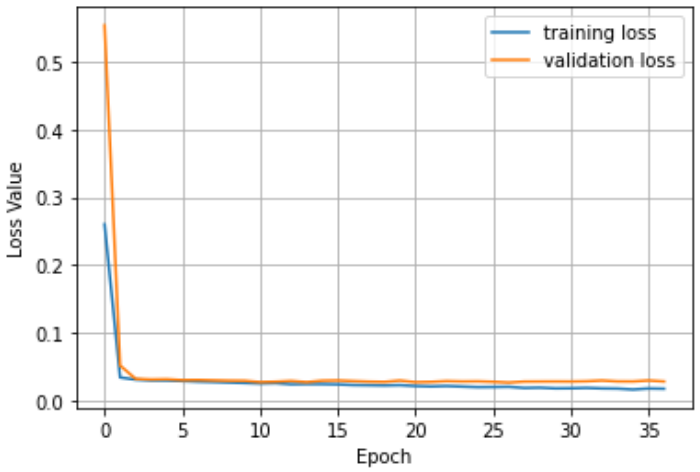
Για τον υπολογισμό των 3 παραμέτρων, που μας ενδιαφέρουν, θα χρησιμοποιήσουμε ένα μοντέλο LSTM και 1 CNN, με υπερ- παραμέτρους, έτσι ώστε να βελτιστοποιείται η επίδοση τους και ως set δεδομένων ένα εκ των :

- Spectrogram Dataset
- Fused Spectrogram & Chromogram Dataset

καθώς μας δίνουν τα καλύτερα αποτελέσματα. Επιλέγουμε εν τέλει την 2η επιλογή, καθώς έπειτα από δοκιμές, έδωσε ελάχιστα καλύτερα αποτελέσματα στο regression πρόβλημα.

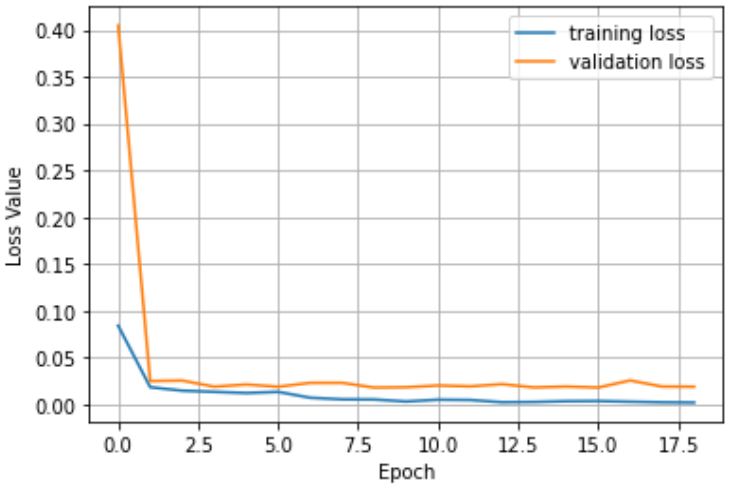
1. Valence

☐ LSTM



Spearman Correlation = 0.4093

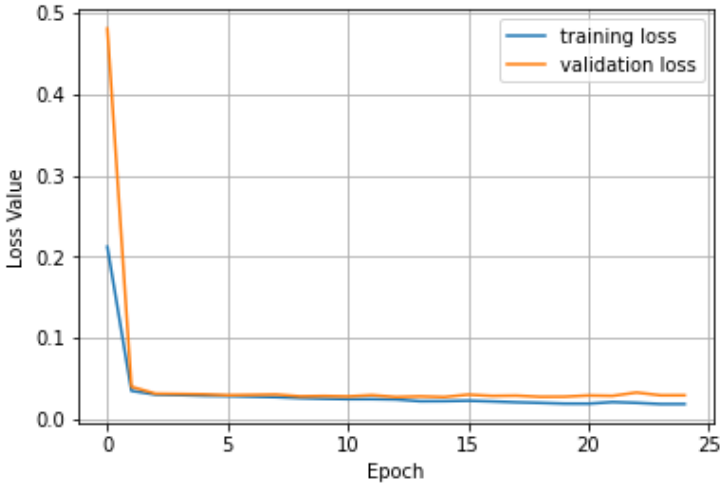
☐ CNN



Spearman Correlation = 0.6790

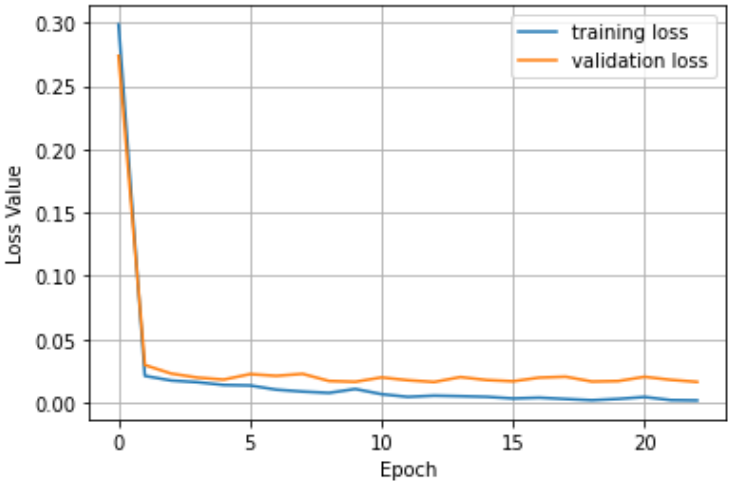
2. Energy

☐ LSTM



Spearman Correlation = 0.3731

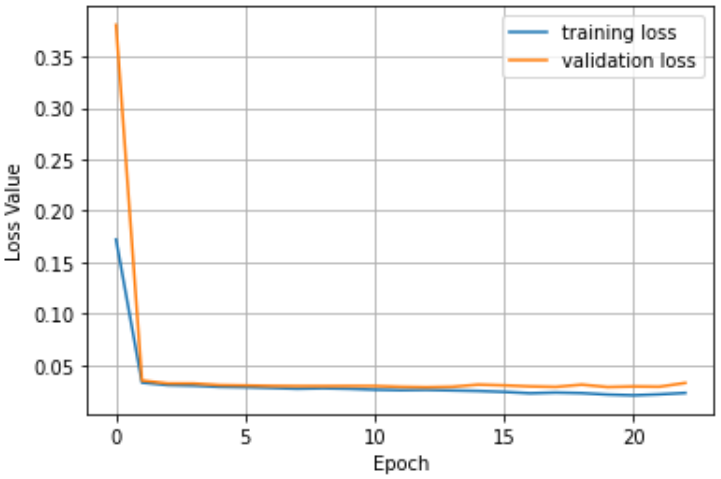
☐ CNN



Spearman Correlation = 0.6792

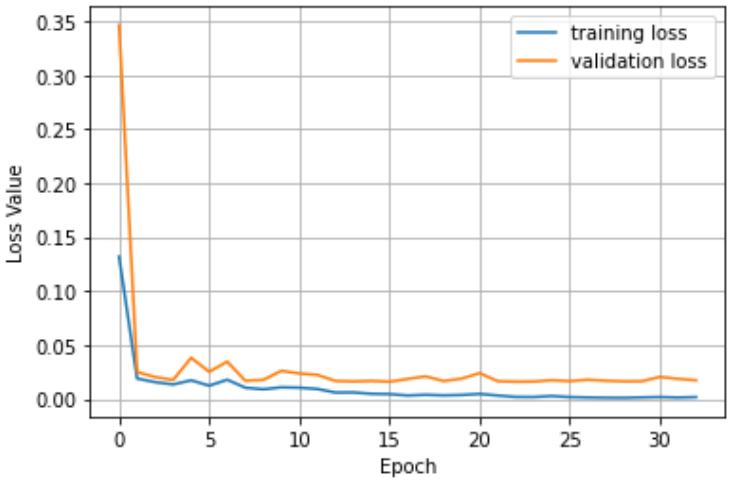
3. Danceability

☐ LSTM



Spearman Correlation = 0.3703

☐ CNN



Spearman Correlation = 0.7037

Η απότομη πτώση του Loss, πιθανότατα οφείλεται στην χρήση του Adam Optimizer. Φαίνεται ξεκάθαρα από τις τιμές των Spearman Correlation ότι τα CNN εμφανίζουν πολύ καλύτερη απόδοση από τα LSTM (σχεδόν διπλάσιο) σε κάθε μία από τις 3 παραμέτρους.

Βήμα 9: Μεταφορά γνώσης (Transfer Learning)

Σχετικά με το paper: Περιγράφεται το πως θα μπορέσουμε να έχουμε καλύτερα αποτελέσματα με Transfer Learning, ανάλογα με το πόσο ειδικό ή γενικό είναι ένα layer. Πιο συγκεκριμένα, επισημαίνεται, ως παράδειγμα, ότι η πλειοψηφία των NNs, μετά την εκπαίδευση, στο πρώτο layer έχουν features, που παραπέμπουν σε φίλτρο Gabor (general), ενώ τα τελευταία εξαρτώνται πολύ από το dataset (specific). Σκοπός του είναι, μέσω πειραμάτων, να απαντηθούν ερωτήσεις όπως το πως παραμετροποιούμε τη γενικότητα ενός layer, πόσο απότομα γίνεται η αλλαγή από general σε specific κλπ. Μετά από αυτές τις διαπιστώσεις το Transfer Learning εφαρμόζεται ως εξής: Στα general layers έχουμε αρχικοποιήσεις από άλλα datasets (πολλές φορές frozen, για να αποφύγουμε το overfitting) και στα specific τυχαίες αρχικοποιήσεις, για να γίνουν οι κατάλληλες αλλαγές με το κανονικό dataset.

Στο συγκεκριμένο βήμα χρησιμοποιούμε Transfer Learning, επειδή τα δεδομένα που έχουμε για το multitask data set δεν είναι αρκετά, ενώ στο fma data set έχουμε περισσότερα (3834 έναντι 1497 φασματογραφημάτων).

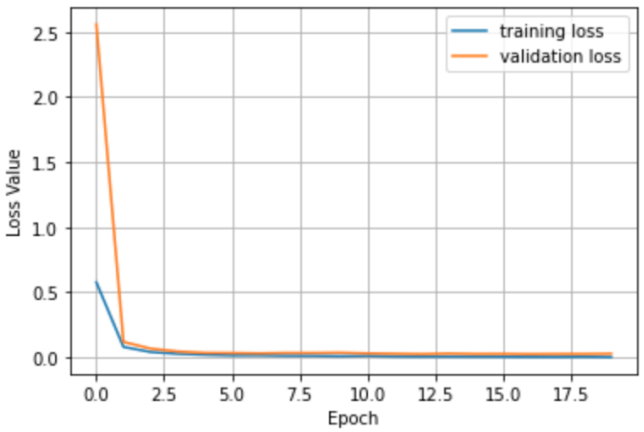
Επομένως, αφού επιλέξουμε CNN, λόγω της καλύτερης επίδοσης του σχέση με το LSTM, το εκπαιδεύουμε αρχικά με τα δεδομένα του FMA dataset, και στη συνέχεια, αλλάζουμε το τελευταίο layer στο κατάλληλο output size (από 10 σε 1), προτού προσθέσουμε και τα multitask δεδομένα. Αυτό συμβαίνει, επειδή για τα πρώτα δεδομένα έχουμε classification σε 10 κλάσεις (μουσικά είδη), ενώ για τα δεύτερα έχουμε regression για τους 3 συναισθηματικούς άξονες (valence, energy, danceability).

Ως dataset εκπαίδευσης, επιλέγουμε το fused spectogram - chromogram, καθώς όπως έχουμε αναφέρει, τα CNN, όταν εκπαιδεύονται με αυτό, εμφανίζουν πολύ καλά αποτελέσματα, ενώ ως μετρική κοιτάμε την weighted average, η οποία είναι και η πιο αντικειμενική σε προβλήματα ταξινόμησης.

Με αυτόν τον τρόπο πετυχαίνουμε fine tuning και απαιτούνται λιγότερες εποχές, για να εξασφαλίσουμε μικρότερο loss και μεγαλύτερο spearman correlation.

Το τελικό loss στα training και validation δεδομένα είναι το εξής:

- ☐ Training Loss: 0.00296
- ☐ Validation Loss: 0.02881



Spearman Correlation = 0.5869

Βήμα 10: Εκπαίδευση σε πολλαπλά προβλήματα (Multitask Learning)

Για το FMA dataset, που περιέχει πολλές επισημειώσεις, ένας αρκετά αποδοτικός τρόπος, για να εκπαιδεύσουμε το μοντέλο μας είναι το multitask learning. Αντί λοιπόν, να χρησιμοποιήσουμε 3 διαφορετικά δίκτυα για την εκτίμηση των παραμέτρων Valance , Energy και Danceability, όπως στο βήμα 8, θα εφαρμόσουμε την παραπάνω τεχνική, έτσι ώστε να υπολογίζουμε τις παραμέτρους αυτές με μονάχα 1 δίκτυο.

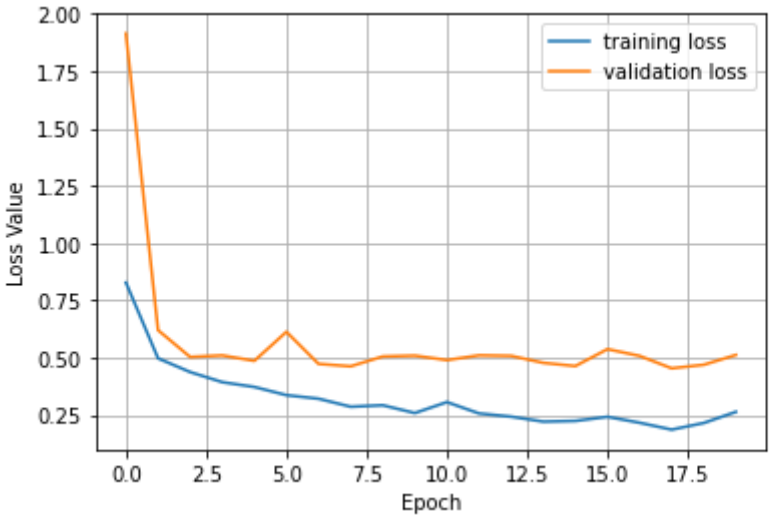
Σύμφωνα με το δοσμένο paper, η εκπαίδευση ενός δικτύου, ικανού να αντιμετωπίσει πολλά διαφορετικά tasks, είχε ικανοποιητικά αποτελέσματα. Μπορεί να μην εμφανίζει την βέλτιστη απόδοση που έχουν ορισμένα state of the art δίκτυα, αλλά η αποτελεσματικότητα του ξεπέρασε σε ορισμένες περιπτώσεις εκείνην που είχαν μοντέλα, προσαρμοσμένα στα αντίστοιχα προβλήματα. Μάλιστα, σημειώνεται πως για την δημιουργία του δικτύου, δεν εφαρμόστηκε ιδιαίτερο tuning στις παραμέτρους του, κάτι το οποίο μπορεί να επηρέασε εν τέλει αρνητικά την απόδοση του.

Δεδομένου πως για το νέο αυτό πρόβλημα, το label του κάθε δείγματος θα είναι ένα “διάνυσμα” 3 τιμών (1 για το valence, 1 για το energy και 1 για το danceability), θα χρησιμοποιήσουμε μία νέα συνάρτηση κόστους, η οποία θα λαμβάνει υπόψη και τις 3 αυτές τιμές κάθε φορά. Πιο συγκεκριμένα, θα υλοποιήσουμε μία Loss Function, η οποία υπολογίζει την L1 νόρμα, χρησιμοποιώντας και κάποια βάρη για τα επιμέρους κόστη των 3 παραμέτρων.

$$L1LossFunction = \sum_{i=1}^{n} |y_{true} - y_{predicted}|$$

Για το Multitask Learning, θα χρησιμοποιήσουμε ένα CNN, καθώς όπως είδαμε και στο ερώτημα 8, κρίνεται καταλληλότερο για τον υπολογισμό των 3 ζητούμενων παραμέτρων. Λαμβάνουμε τα εξής αποτελέσματα:

Οι γραφικές του training & validation Loss, της συνάρτησης σφάλματος που υλοποιήσαμε είναι οι εξής (L1 Loss Function):



Spearman Correlation for Valence : 0.5327

Spearman Correlation for Energy : 0.7319

Spearman Correlation for Danceability : 0.6306

Παρατηρούμε, λοιπόν, πως μέσα από το Multitask Learning, καταφέραμε με την χρήση ενός μόνο νευρωνικού δικτύου (και επομένως σε πολύ λιγότερο χρόνο και με λιγότερους πόρους) να πετύχουμε πολύ καλά αποτελέσματα όσον αφορά την μετρική του Spearman Correlation. Οι τιμές αυτές, είναι εμφανώς καλύτερες από των LSTM μοντέλων, όπως περιμέναμε, και μάλιστα είναι συγκρίσιμες με αυτές των CNN, που εκπαιδεύτηκαν αποκλειστικά σε εκείνα τα tasks. Μάλιστα στην περίπτωση του Energy, πετύχαμε καλύτερη απόδοση (αν και στις άλλες 2 περιπτώσεις είχαμε λίγο χειρότερες τιμές) .