ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

Αυτόματη Σύνθεση Μουσικής Σε Συμβολική Μορφή Με Αναδρομικά Νευρωνικά Δίκτυα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Γεώργιος Β. Φιλανδριανός

Επιβλέπων : Γεώργιος Στάμου
Αν. Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2019

Εθνικό ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

Αυτόματη Σύνθεση Μουσικής Σε Συμβολική Μορφή Με Αναδρομικά Νευρωνικά Δίκτυα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Γεώργιος Β. Φιλανδριανός

Επιβλέπων : Γεώργιος Στάμου
Αν. Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 16η Ιουλίου 2019.

.....	.....	.....
Γ. Στάμου	Α. – Γ. Σταφυλοπάτης	Δ. Φωτάκης
Αν. Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Επ.Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2019

.....
Γεώργιος Β. Φιλανδριανός

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Η Αυτόματη Σύνθεση Μουσικής αποτελεί ίσως ένα από τα πλέον κομβικά αλλά και δύσκολα έργα στον τομέα της ανακατασκευής πληροφορίας. Για τους ειδικούς αποτελεί το αποδοτικότερο μέσο επικοινωνίας τους ενώ για τους υπολοίπους χρήστες είναι ένα από τα καλύτερα μέσα έκφρασης των συναισθημάτων τους.

Παρόλα αυτά η σύνθεση νέων και ενδιαφέροντών κομματιών είναι μια διεργασία η όποια απαιτεί βαθιά γνώση, εμπειρία και εξειδίκευση. Αντίστοιχη δυσκολία συναντάται και στους υπολογιστές όπου, παρόλες τις προσπάθειες, έχει αποδειχθεί μια εργασία ιδιαίτερα απαιτητική η όποια έχει γνωρίσει ως την ώρα επιτυχία μόνο σε μερικές κατηγορίες ακουσμάτων.

Όπως και με τα περισσότερα έργα ανάκτησης και ανακατασκευής πληροφορίας στον τομέα της μουσικής, έτσι και τα συστήματα αυτόματης σύνθεσης που κατασκευάστηκαν στα πλαίσια αυτής της διατριβής ακολουθούν την τάση να αντικαθιστούν τα στάδια επεξεργασίας σήματος και εξαγωγής χαρακτηριστικών από στατιστικά μοντέλα με αρχιτεκτονικές βαθιάς μηχανικής μάθησης. Για τον λόγο αυτόν στην παρούσα εργασία επιλέχθηκε ο παραδοσιακός δρόμος όσο αναφορά την αναπαράσταση της μουσικής, ο οποίος είναι η κωδικοποίησή της σε ακολουθιακή μορφή και συγκεκριμένα η κωδικοποίησή της με το πρωτόκολλο Midi.

Για το πειραματικό μέρος της εργασίας εκπαιδεύτηκαν διαφορετικές αρχιτεκτονικές νευρωνικών δικτύων με σκοπό δοσμένης μιας αρχικής μελωδίας να συνθέτουν κάποια πρωτότυπη συνέχειά της. Συγκεκριμένα χρησιμοποιήθηκαν: ένα Αναδρομικό Νευρωνικό Δίκτυο Βαθιάς Μακροπρόθεσμής Μνήμης (LSTM) με Πολλαπλά επίπεδα, μια αρχιτεκτονική Κωδικοποιητή- Αποκωδικοποιητή (LSTM Encoder- Decoder) καθώς και μια Αρχιτεκτονική Κωδικοποιητή- Αποκωδικοποιητή με Συγκέντρωση (LSTM Encoder- Decoder with Attention).

Παράλληλα με την αρχιτεκτονική άλλαζε και το σύνολό εκπαίδευσης όπου χρησιμοποιήθηκαν σύνολα: πιάνου, κιθάρας καθώς και συνδυασμοί αλλά και παραλλαγές αυτών. Τέλος στα παραπάνω δίκτυα αλλάχθηκαν και ορισμένες υπερπαράμετροι τους όπως: το μέγεθος της μνήμης του LSTM και η μέθοδος πρόβλεψης, με σκοπό μα διερευνηθεί ο ρόλος και η επίδρασή τους στις παραγόμενες συνθέσεις.

Λέξεις Κλειδιά: Τεχνητά Νευρωνικά Δίκτυα, Ακολουθιακή Μουσική, Βαθιά Μάθηση, Αναδρομικά Νευρωνικά Δίκτυα, Αρχιτεκτονική Κωδικοποιήτη- Αποκωδικοποιητή, Συγκέντρωση, Πόλωση, Μουσικά Όργανα, Νότες, Εξισορρόπηση Δεδομένων

Abstract

Composing music automatically is perhaps one of the most crucial, but also difficult, projects in the field of information reconstruction. For expert users it is the most effective means of communication, while for all the others is just one of the best ways of expressing their feelings.

However, the composition of new and interesting tracks is a process that requires deep knowledge, experience and expertise. A similar difficulty could also be met in computers, where, despite all efforts, it has been proved to be a particularly demanding task, which has been successful only in a few categories of hearings.

As it happens with most of the information retrieval and reconstruction projects in the field of music, systems of automatic composition that have been developed within this dissertation tend to replace the stages of signal processing and extraction from statistical models with architectures of deep learning architectures. For this reason, the traditional way of music representation has been chosen for the purposes of the current dissertation, which is encoding in sequential form and encoding by using the Midi protocol, if we want to be more specific.

For the experimental part of the work different neural network architectures were trained so that to create an original sequence of a given initial melody. More specifically, the following were used: a recurrent neural network of deep Long-Short Term Memory (LSTM) with multiple levels, an Encoder-Decoder architecture (LSTM Encoder-Decoder) as well as an Encoder-Decoder architecture with an attention mechanism (LSTM Encoder- Decoder with Attention).

Along with the architecture, also the whole training dataset had been changing, using songs from different instruments as: piano, guitar as well as different combinations and variations of them. Finally, some of the hyperparametrs of the above networks were changed, such as LSTM memory size and prediction method, in order to investigate their role and impact on the compositions.

Keywords: Neural Networks, Sequential, Deep Learning, Recurrent Neural Networks, Encoder, Decoder, Attention Mechanism, Bias, Instruments, Data Augmentation

Πίνακας Περιεχομένων

Κεφάλαιο 1 – Εισαγωγή

1.1 Ιστορική Αναδρομή

Από την απαρχή των υπολογιστών είχαν γίνει αρκετές προσπάθειες για την δημιουργία αλγορίθμων με σκοπό την αυτόματη σύνθεση μουσικής. Η πρώτη μελωδία που δημιουργήθηκε από υπολογιστή εμφανίστηκε μόλις το 1957. Ήταν μια μελωδία μήκους 17 δευτερολέπτων που ονομάστηκε 'The Silver Scale' από τον συγγραφέα της Newman Guttman και δημιουργήθηκε από ένα λογισμικό για ηχητική σύνθεση που ονομαζόταν Music I, το όποιο είχε κατασκευαστεί από τον Mathews στα Bell Laboratories. Την ίδια χρονιά, η "The Illiac Suite" ήταν η πρώτη παρτιτούρα που συντέθηκε από υπολογιστή [18]. Ονομάστηκε έτσι από τον υπολογιστή ILLIAC I στο Πανεπιστήμιο του Illinois στο Urbana-Champaign (UIUC) στις Ηνωμένες Πολιτείες. Οι ανθρώποι «μετα-συνθέτες» ήταν οι Lejaren A. Hiller και Leonard M. Isaacson, και οι δύο μουσικοί επιστήμονες. Αυτό ήταν ένα πρώιμο παράδειγμα αλγοριθμικής σύνθεσης, χρησιμοποιώντας τα στοχαστικά μοντέλα (Μαρκοβιανές αλυσίδες) καθώς και κανόνες για το φιλτράρισμα των παραγόμενων αποτελεσμάτων σύμφωνα με επιθυμητές ιδιότητες.

Στον τομέα της σύνθεσης ήχου, ένα ορόσημο ήταν η απελευθέρωση το 1983 από την Yamaha του συνθέτη DX 7, ο οποίος βασίζεται σε ένα μοντέλο σύνθεσης με βάση τη διαμόρφωση της συχνότητας (FM). Την ίδια χρονιά, η MIDI6interface ξεκίνησε ως ένας τρόπος διαλειτουργικότητας διαφόρων λογισμικών και οργάνων (συμπεριλαμβανομένου του συνθέτη Yamaha DX7). Ένα άλλο ορόσημο ήταν η ανάπτυξη από τον Puckette στο IRCAM του περιβάλλοντος διαδραστικής επεξεργασίας πραγματικού χρόνου Max / MSP, που χρησιμοποιείται για σύνθεση μουσικής για real time εφαρμογές.

Σχετικά με την αλγοριθμική σύνθεση, στις αρχές της δεκαετίας του 1960 ο Ιωάννης Ξενάκης διερεύνησε την ιδέα της στοχαστικής σύνθεσης[190] , στη σύνθεσή του με τίτλο "Atr ees" το 1962. Σε μια άλλη προσέγγιση που ακολούθησε την αρχική κατεύθυνση της "The Iliac Suite", χρησιμοποιήθηκαν γραμματικές και κανόνες για να προσδιοριστεί το στυλ ενός συγκεκριμένου σώματος ή γενικότερα της θεωρίας της τονικής μουσικής. Ένα παράδειγμα είναι η παραγωγή, στη δεκαετία του 1980 από το λογισμικό σύνθεσης του Ebcio ̆glu's που ονομάζεται CHORAL, ενός τετραμερούς χορού στο ύφος του Johann Sebastian Bach, το οποίο συντέθηκε σύμφωνα με πάνω από 350 χειροποίητους κανόνες [18]. Στα τέλη της δεκαετίας του 1980, το σύστημα του David Cope, που ονομάζεται Experiments in Musical Intelligence (EMI), επέκτεινε την προσέγγιση αυτή με την ικανότητα να μαθαίνει από ένα σύνολο από παρτιτούρες ενός συνθέτη και αυτός με την σειρά του να μπορεί να δημιουργήσει τη δική του γραμματική και βάση δεδομένων.

Από τότε, οι αλγόριθμοί αυτόματης σύνθεσης μουσικής συνέχιζαν συνεχώς να εξελίσσονται δημιουργώντας τελικά προϊόντα για το ευρύ κοινό. Πλέον αντιπροσωπευτικό παράδειγμα τέτοιας εφαρμογής αποτελεί το 'The Garage Band' για τις πλατφόρμες τις Apple (υπολογιστές, κινητά και tablet) το οποίο θεωρείται πρόγονος του λογισμικού 'Cubase' που κυκλοφόρησε ο Steinberg το 1989.

Από τότε μέχρι σήμερα δεν έχει σταματήσει η συνεχής αναζήτηση αποδοτικών αλγορίθμων για αυτόματη σύνθεση μουσικής. Σήμερα για το πρόβλημα αυτό έχουν προταθεί διάφορες λύσεις με τις πιο αποδοτικές να χρησιμοποιούν συστήματα βαθιάς μηχανικής μάθησης.

1.2 Σκοπός

Σκοπός της παρούσας εργασίας είναι η μελέτη και η κατασκευή state-of-the-art συστημάτων για την αυτόματη σύνθεση μουσικής. Παράλληλα εξετάζεται η επίδραση διαφόρων υπερπαραμέτρων των συστημάτων αυτών καθώς η αλλαγές που αυτές επιφέρουν

στο τελικό αποτέλεσμα. Η αξιολόγηση των παραγόμενων κομματιών έγινε από χρήστες οι οποίοι χωριστήκαν σε ομάδες ανάλογα με τις μουσικές τους γνώσεις. Τελικός σκοπός της εργασίας είναι η παραγωγή ενός πραγματικού εργαλείου, για την χρήση από το ευρύ κοινό, για την δημιουργία και την συνέχιση μελωδιών, από **30 διαφορετικά μοντέλα**.

1.3. Δομή της Εργασίας

Η παρούσα διπλωματική εργασία είναι δομημένη σε 13 κεφάλαια. Στο Κεφάλαιο 2 παρουσιάζεται η ανάγκη αλλά και δυσκολίες του εξεταζόμενου προβλήματος.

Στο Κεφάλαιο 3 αναφέρεται στην συμβολική αναπαράσταση μουσικής ενώ παράλληλα εξηγείται η λειτουργία του πρωτοκόλλου Midi το οποίο αποτελεί έναν από τους πιο βασικούς τρόπους επικοινωνίας συμβολικών δεδομένων.

Στο Κεφάλαιο 4 παρουσιάζονται τα σύνολα δεδομένων που χρησιμοποιήθηκαν για την εκπαίδευση των αρχιτεκτονικών μαζί με ορισμένα βασικά χαρακτηριστικά τους.

Στο Κεφάλαιο 5 και 6 αναλύονται ορισμένα βασικά μοντέλα νευρωνικών δικτύων, τα οποία χρησιμοποιούνται ως υποσυστήματα των τελικών αρχιτεκτονικών που χρησιμοποιήθηκαν για την σύνθεση ενώ συγχρόνως αναλύονται και οι βασικές υπερπαράμετροί τους.

Το Κεφάλαιο 7 εστιάζει στην μέθοδο και στον τρόπο αξιολόγησης καθώς και στο εργαλείο που κατασκευάστηκε στα πλαίσια αυτής της διπλωματικής εργασίας για την αυτόματη σύνθεση ή συνέχιση μουσικής από κάθε χρήστη.

Στα Κεφάλαια 8,9,10, 11, 12 εξετάζεται η επίδραση κάθε μιας διαφορετικής υπερπαραμέτρου στο τελικό αποτέλεσμα. Αρχικά εξετάζονται τα αποτελέσματα των διαφορετικών αρχιτεκτονικών ανά μέγεθός τους, η επίδραση της τυχαιότητας στην πρόβλεψη των μοντέλων, της πόλωσης των δεδομένων και τελικά του οργάνου που αποτελεί το σύνολο εκπαίδευσης.

Τέλος στο Κεφάλαιο 13 αναλύονται τα εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση των παραπάνω ενώ παράλληλα παρατίθενται και τα απαραίτητα links για μελέτη και εξέλιξη του πηγαίου κώδικα και των μηχανισμών που κατασκευάστηκαν.

Κεφάλαιο 2 – Θεωρεία Μουσικής

2.1 Ορισμός Μουσικής

Η μουσική είναι μια μορφή τέχνης και πολιτιστικής δραστηριότητας η οποία εμφανίζεται με την μετάφραση των ήχων σε κάποια αφηρημένη μορφή. Οι γενικοί ορισμοί της μουσικής περιλαμβάνουν κοινά στοιχεία όπως η συχνότητα (η οποία ρυθμίζει τη μελωδία και την αρμονία), ο ρυθμός (και οι συναφείς του έννοιες όπως τέμπο), της δυναμικής καθώς και των ηχητικών ιδιοτήτων του μέσου αναπαραγωγής. Διαφορετικά στυλ ή είδη μουσικής μπορεί να τονίζουν, να υπογραμμίζουν ή να παραλείπουν ορισμένα από αυτά τα στοιχεία. Η μουσική εκτελείται από ένα ευρύ φάσμα μουσικών οργάνων με μερικές κατηγορίες αυτών να αποτελούν τα πνευστά, τα έγχορδα, τα κρουστά και άλλα ενώ η σύνθεση και η αναπαραγωγή μπορεί να έχει πολλούς διαφορετικούς σκοπούς όπως ψυχαγωγία και απόλαυση μέχρι και την τίμηση προσώπων ή θεών , προσευχή κ.α.

2.2 Μουσική πληροφορία

Ο άνθρωπος δύναται να αντιλαμβάνεται ήχους, με αισθητήριο όργανο το αυτί, οι οποίοι κυμαίνονται από 20Hz έως και 20kHz. Για κάποιο εύρος συχνοτήτων (μεταξύ 50Hz και 1kHz), το αυτί μπορεί να αντιλαμβάνεται ήχους με πολύ μεγαλύτερη ευαισθησία. Αν σε έναν ήχο ξεχωρίζει μια συχνότητα, δηλαδή έχει αισθητά μεγαλύτερη ένταση, τότε αυτή μεταφράζεται από τον άνθρωπο ως **τόνος**. Αν από την άλλη ένας ήχος επαναλαμβάνεται με συχνότητα περίπου από 0.5Hz έως και 3Hz τότε αυτή εκλαμβάνεται ως **ρυθμός**.

Ως τόνο μιας νότας στην σύγχρονη μουσική ορίζεται η βασική συχνότητά της. Μια ακολουθία τέτοιων νοτών δομημένες τον χρόνο συνθέτουν ουσιαστικά ένα μουσικό κομμάτι.

Οι νότες που σχηματίζουν ένα μουσικό κομμάτι δεν έχουν μόνο οξύτητα αλλά έχουν και χρονική διάρκεια δηλαδή άλλες νότες μπορεί να κρατούν περισσότερο και άλλες λιγότερο.

Η διάρκεια αυτή ονομάζεται αξία κάθε νότας και οι τιμές που μπορεί να λάβει μαζί με τον συμβολισμό τους στο πεντάγραμμο παρουσιάζονται στον παρακάτω πίνακα.

Όνομα Αξίας	Συμβολισμός
Ολόκληρο	
Μισό	
Τέταρτο	
Όγδοο	
Δέκατο έκτο	
Τριακοστό τέταρτο	

Εξηκοστό Τέταρτο	

Οι παραπάνω ποσότητες δεν δηλώνουν απόλυτες χρονικές διάρκειες αλλά ρυθμικές αναλογίες. Κάθε φθογγόσημο έχει την μισή αξία από το προηγούμενό του και την διπλάσια αξία από το επόμενό του. Δηλαδή αυτό σημαίνει ότι ένα ολόκληρο ισούται με δυο μισά, τέσσερα τέταρτα κ.ο.κ. Συνεπώς η χρονική διάρκεια κάθε νότας μπορεί να μετατραπεί σε διακριτή είτε συνεχής μεταβλητή.

Αξίζει να σημειωθεί ότι σε κάθε κομμάτι ορίζεται μια βασική μονάδα χρονισμού η οποία ονομάζεται χτύπος (beat). Κάθε χτύπος επαναλαμβάνεται έπειτα από σταθερό χρόνο και ουσιαστικά δίνει τον ρυθμό σε κάθε κομμάτι μιας και όλες οι χρονικές διάρκειες εξαρτώνται από την απόσταση αυτών. Η απόσταση αυτή μετριέται από τον αριθμό των χτύπων ανά λεπτό (beats per minute) και ουσιαστικά αν αυξηθεί ή μειωθεί ο αριθμός αυτός το τραγούδι γίνεται πιο γρήγορο ή αργό αντίστοιχα. Η χρονική διαφορά μεταξύ 2 beats (δηλαδή η χρονική διάρκεια ενός χτύπου) ισούται χρονικά με την διάρκεια μιας νότα αξίας ένα τέταρτο. Κατά αναλογία η χρονική διάρκεια μιας νότας με αξία ένα ολόκληρο ισούται με 4 beats. Για τον λόγο αυτόν η αλλαγή του beats per minute δεν επιφέρει καμία αλλαγή στην αξία των νοτών παρά μόνο αλλάζει την χρονική διάρκειά της. Για τον λόγο αυτόν και η παραπάνω διάρκεια δεν εκφράζει απόλυτο χρόνο.

2.3 Σύνθεση Μουσικής Σήμερα

2.3.1 Σύνθεση Μουσικής από τον Άνθρωπο

Λίγοι άνθρωποι έχουν την δυνατότητα να συνθέσουν νέα και ενδιαφέροντα κομμάτια. Κατά κύριο λόγο ο στόχος της σύνθεσης είναι η κατασκευή κομματιών τα οποία θα εγείρουν συναισθήματα και σκέψεις στους ανθρώπους που το ακούν. Για τον σκοπό αυτόν απαιτείται βαθιά γνώση και εξειδίκευση και για τον λόγο αυτόν υπάρχουν λίγοι μόνο πολύ διάσημοι συνθέτες στην ιστορία. Άνθρωποι με χρόνια εμπειρία στην μουσική και θεωρούν αδύνατον το να μπορέσουν να γράψουν ένα σωστά δομημένο και παράλληλα ενδιαφέρον κομμάτι.

2.3.2Συνθέση μουσικής από Υπολογιστή

Η δυσκολία συγγραφής μουσικών κομματιών έγκειται στο γεγονός ότι πίσω από τις ακολουθίες κρύβονται συνδέσεις και συναρτήσεις οι οποίες είναι πολύ δύσκολο να εντοπιστούν και να γίνουν κατανοητές. Για παράδειγμα υπάρχει πολύ μεγάλη πρόοδος τα τελευταία χρόνια στην αυτόματη σύνθεση κειμένου (καθώς και σε όλα τα υποπεδία του όπως μετάφραση, σύνθεση λόγου, συνομιλίες κ.α.) ενώ στην σύνθεση μουσικής τα αντίστοιχα μέσα φαίνονται να αποτυγχάνουν παταγωδώς. Αυτό κατά κύριο λόγω συμβαίνει επειδή η μουσική πληροφορία που κρύβεται πίσω από κάθε κομμάτι είναι πολύ πιο σύνθετη από την πληροφορία μεταξύ των λέξεων μιας πρότασης. Παράλληλα ένα κομμάτι θεωρείται ενδιαφέρον αν σε αυτό υπάρχουν έντονες αλλαγές συναισθημάτων οι οποίες επιτυγχάνονται με την συνεχή αλλαγή των ακολουθιών, κάτι που δεν συναντάται συχνά σε άλλες εφαρμογές όπως σε μια ομιλία για παράδειγμα. Αυτό δημιουργεί σοβαρό πρόβλημα στις μηχανικές μεθόδους σύνθεσης μιας και ο τρόπος με τον όποιον αυτές προσπαθούν να εξάγουν τις συναρτήσεις συσχέτισης μεταξύ των νοτών είναι προσπαθώντας «μάθουν» να προβλέπουν τις μεταγενέστερες ακολουθίες.

Πρώτο και απαραίτητο βήμα για την σύνθεση μουσικών κομματιών, αποτελεί η αποκωδικοποίηση της πληροφορίας εισόδου. Η αναπαράσταση ενός κομματιού μπορεί να έχει διάφορες μορφές όπως σε audio μορφή (κυματομορφή), σε εικόνα μέσω παρτιτούρας,

σε ακολουθιακή μορφή, σε αναπαράσταση με υψηλότερου επιπέδου χαρακτηριστικά, κ.α. Όπως εξηγείτε εκτενέστερα και παρακάτω κάθε μια μορφή εισόδου επηρεάζει σημαντικά το είδος και την μορφή της αρχιτεκτονικής η οποία δύναται να την αποκωδικοποιήσει. Τα τελευταία χρόνια έχουν κατασκευαστεί μοντέλα τα οποία προσπαθούν να συνθέσουν μουσικής σε όλες τις παραπάνω μορφές.

Για παράδειγμα το σύστημα Magenta της Google αποτελεί μια από τις καλύτερες έως τώρα αρχιτεκτονικές για την αυτόματη σύνθεση μουσικής πιάνο. Αξιοσημείωτο χαρακτηριστικό της, πέρα από τα αποτελέσματά της, είναι ότι η έξοδος παράγετε σε audio μορφή. Αυτό σημαίνει η ακολουθία εξόδου μπορεί να είναι μερικά χιλιάδες samples, πράγμα πολύ δύσκολο από τα παραδοσιακά συστήματα. Η αρχιτεκτονικής που χρησιμοποιείται εσωτερικά ονομάζεται Dilated Convolution Neural Networks η οποία ουσιαστικά είναι μια κλασσική CNN αρχιτεκτονική αλλά όπως δηλώνει και το όνομά της οι συνδέσεις μεταξύ των κόμβων είναι πολύ πιο αραιές.

Επίσης το BachBot είναι ένα σύστημα το οποίο παράγει μουσική σε ακολουθιακή μορφή (όπως αυτά που υλοποιήθηκαν στα πλαίσια αυτής της διπλωματικής εργασίας) και στόχος του είναι να παράγει μελωδίες οι οποίες είναι θα είναι δυσδιάκριτες από αυτές του διάσημου συνθέτη Bach. Οι δημιουργοί του ισχυρίζονται, έπειτα μια διαδικασία αξιολόγησης των κομματιών που συντέθηκαν, ανάλογη με αυτή που θα γίνει και στην συνέχεια, ότι οι χρήστες μπορούσαν να ξεχωρίσουν τα αληθινά από τα κατασκευασμένα από υπολογιστή κομμάτια μόνο κατά 1%. Οι καινοτομίες του συστήματος αυτού βρίσκονται στην αναπαράσταση των συγχορδιών και όχι στις εσωτερική αρχιτεκτονική, όπως με το σύστημα της Google.

Παρόλα αυτά οι πλειονότητα των αρχιτεκτονικών που έχουν κατασκευαστεί για να δώσουν λύση στο πρόβλημα της σύνθεσης έχουν γνωρίσει επιτυχία μόνο σε συγκεκριμένα tasks και δεν αποτελούν λύσεις για πιο γενικευμένα προβλήματα.

Αντίθετα στην παρούσα εργασία μελετώνται και κατασκευάζονται μοντέλα και αρχιτεκτονικές για την σύνθεση- συνέχιση μελωδιών ανεξαρτήτως δομής και στυλ. Για τον λόγο αυτό κατασκευάζονται διάφορες αρχιτεκτονικές ικανές να συνεχίσουν μια οποιαδήποτε δοσμένη αρχική μελωδία και στην συνέχεια μελετώνται τα αποτελέσματά τους. Παράλληλα εκτός των αρχιτεκτονικών αλλάζει και το σύνολο εκπαίδευσης τους αλλά και οι τιμές ορισμένων βασικών υπερπαραμέτρων, με σκοπό την μελέτη της επίδρασής τους στα τελικά αποτελέσματα.

Κεφάλαιο 3- Αναπαράσταση Δεδομένων

3.1 Τύποι Δεδομένων

Για την επεξεργασία ακουλουθιακών δεδομένων (όπως κείμενο ή μουσική) υπάρχουν τουλάχιστον τρείς τύποι και στάδια όπου η αναπαράσταση των δεδομένων πρέπει εξεταστεί:

- *Είσοδος κατά την διαδικασία Εκπαίδευσης (Training Input):* Η αναπαράσταση των δεδομένων που χρησιμοποιείται κατά την διαδικασία εκπαίδευσης του δικτύου.
- *Είσοδος για την πρόβλεψη (Test Input):* Η είσοδος στο δίκτυο για την παραγωγή δεομένων. Για παράδειγμα σε ένα σύστημα που παράγει την συνέχεια μιας μελωδίας ως αρχική είσοδος για την παραγωγή μπορεί να είναι η αρχική μελωδία ή ένα κομμάτι αυτής.
- *Παραγόμενη ακολουθία:* Ο στόχος παραγωγής.

Οι παραπάνω επιλογές καθορίζουν την μορφή και την αρχιτεκτονική του δικτύου. Για παράδειγμα για την παραγωγή μελωδιών στο [14] τόσο η είσοδος εκπαίδευσης όσο και η είσοδος για την πρόβλεψη καθώς και η παραγόμενη ακολουθία είναι μελωδίες η οποίες περιορίζονται σε μέγεθος από έναν αριθμό νοτών (μέγεθος ακολουθίας). Αντίθετα σε άλλα συστήματα με διαφορετικό στόχο, όπως στο [15], που παράγονται συνοδευτικές συγχορδίες σε μελωδίες, η είσοδος εκπαίδευσης καθώς και η είσοδος για την σύνθεση είναι μελωδίες ενώ η παραγόμενη έξοδος είναι ένα σύνολο συγχορδιών.

Σε όλα τα μοντέλα που κατασκευάστηκαν στα πλαίσια της εργασίας αυτής χρησιμοποιήθηκε η τεχνική που αναφέρεται στο [14], δηλαδή οι αναπαραστάσεις σε όλα τα στάδια είναι μονοφωνικές μελωδίες δηλαδή απλές ακολουθίες νοτών.

3.2 Αναπαράσταση Δεδομένων

Ανάλογα την αρχιτεκτονική του δικτύου που χρησιμοποιείται για την σύνθεση, η κωδικοποίηση των παραπάνω δεδομένων μπορεί να διαφέρει. Αυτό οφείλεται στο γεγονός ότι μια μελωδία έχει διάφορους τρόπους αναπαράστασης όπως σε παρτιτούρα (με μορφή εικόνας), σε ένα συνεχές σήμα (audio signal), σε συμβολική μορφή κ.α. Γενικότερα οι περισσότερες βαθιές αρχιτεκτονικές που ασχολούνται με την μουσική (είτε αφορά την σύνθεση είτε αλλά προβλήματα) χρησιμοποιούν συμβολικές αναπαραστάσεις. Η πλέον ευρέως χρησιμοποιούμενη συμβολική κωδικοποίηση είναι η μορφή Midi (Musical Instrument Digital Interface) η οποία περιγράφεται παρακάτω.

3.2.1 Κωδικοποίηση Midi

Η κωδικοποίηση Midi είναι ένα πρωτόκολλό που επιτρέπει την εύκολη επικοινωνία μεταξύ υπολογιστών και ηλεκτρονικών μουσικών οργάνων όπως πιάνο, keyboards, synthesizer και άλλων.

Ένα Midi αρχείο αποτελείται από διακριτά μηνύματα (Midi Events) τα όποια μπορεί να έχουν διάφορους προορισμούς. Κάθε ένα Midi αρχείο μπορεί να επικοινωνεί συγχρόνως με 16 ανεξάρτητα κανάλια (channels ή streams). Αυτή ιδιότητα του πρωτοκόλλου, δηλαδή η ταυτόχρονη διαχείριση 16 ανεξάρτητων συνομιλίων, επιτρέπει το ταίριασμα των σημαντικών πληροφοριών μεταξύ των προορισμών. Σε κάθε ένα κανάλι στέλνεται κάθε φορά ένα Midi Event το οποίο μπορεί να έχει διαφορετικό τύπο και προορισμό ανάλογα με την πληροφορία που φέρει.

Ένα Midi Event αποτελείται από ένα 8-bit status byte το οποίο δηλώνει τον τύπο του μηνύματος και ακολουθείται από άλλο ένα ή δυο byte δεδομένων (data bytes) τα οποία περιέχουν τις πληροφορίες σχετικά με το event αυτό. Υπάρχουν διάφοροί τύποι τέτοιων μηνυμάτων αλλά σε ένα γενικότερο πλαίσιο αφαίρεσης χωρίζονται σε Μηνύματα Καναλιού

(Channel Messages) και Μηνύματα Συστήματος (System Messages). Τα πρώτα είναι μηνύματα τα οποία απευθύνονται σε συγκεκριμένα κανάλια και ο αριθμός του καναλιού αυτού περιέχεται στο status byte. Αντίθετα τα Μηνύματα Συστήματος δεν αναφέρονται σε κάποιο κανάλι και συνεπώς το status byte τους δεν περιέχει αυτήν την πληροφορία.

Με την σειρά τους τα Channel Messages χωρίζονται σε Channel Voice Messages και Mode Messages. Τα Channel Voice Messages περιέχουν τις μουσικές πληροφορίες των κομματιών και ουσιαστικά αποτελούν το μεγαλύτερο μέρος των μηνυμάτων σε ένα Midi αρχείο. Σε αυτή την κατηγορία μηνυμάτων ανήκουν τα παρακάτω events:

- *Note On*
- *Note off*
- *Program Change*
- *Aftertouch*
- *Pitch Bend*
- *Control Change*
- *Bank Select*
- *RPN / NRPN*

Αντίθετα τα Channel Mode Messages επηρεάζουν τον τρόπο με τον οποίον ένα όργανο λήψης θα ανταποκρίνεται στα Voice Messages. Τα τελευταία χωρίζονται σε System Exclusive Messages για την μεταφορά μηνυμάτων σε κάθε ένα κατασκευαστή οργάνων μοναδικά και τα Real Time Messages τα οποία είναι υπεύθυνα για τον έλεγχο των Midi συσκευών.

Από τα παραπάνω είδη μηνυμάτων τα πραγματικά χρήσιμα για την αποκωδικοποίηση της μουσικής πληροφορίας αλλά και την ανακατασκευή αυτής είναι τα δύο πρώτα που ορίζουν το ποια νότα παίζεται κάθε φορά και με πόση δύναμη (ένταση) αυτή πατήθηκε. Η μορφή των μηνυμάτων αυτών παρουσιάζεται παρακάτω:

- *Note on*: Η ενεργοποίηση μιας νότας είναι ένα μήνυμα τύπου Note on όπου στο status byte του περιέχεται ο αριθμός του καναλιού του οργάνου, με πεδίο τιμών [0,15] και ακολουθείται από δυο data bytes που εκφράζουν την νότα που πατήθηκε (pitch), με πεδίο τιμών [0, 127] και την επιτάχυνση (velocity) αυτής, με πεδίο τιμών [0, 127]. Ως επιτάχυνση ορίζεται η δύναμη με την οποία πατιέται ένα πλήκτρο (μια νότα). Για παράδειγμα το μήνυμα <Note on, 0, 60, 50> σημαίνει την ενεργοποίηση μια νότας στο κανάλι 1, με pitch 60 (middle C) με επιτάχυνση 50.
- *Note off:* Όταν αφήνεται μια νότα στέλνεται ένα Note off μήνυμα το οποίο έχει επίσης την ίδια μορφή με ένα Note on μήνυμα. Σε αυτήν όμως την περίπτωση τα data bytes περιέχουν πληροφορία σχετικά με το ποια νότα αφήνεται (pitch) και με πόση δύναμη (velocity) αφήνεται αυτή. Στα περισσότερα πραγματικά συστήματα το velocity των note off νοτών αγνοείται. Για παράδειγμα το μήνυμα <Note off, 0, 60, 20> σημαίνει ότι στο κανάλι 1, σταματάει να παίζει η νότα 60 και αυτή αφήνεται με επιτάχυνση 20.

Κάθε συμβάν Midi ενσωματώνεται σε ένα γεγονός που περιέχει μια τιμή χρόνου delta η οποία περιλαμβάνει πληροφορίες χρονισμού για το event αυτό. Η τιμή delta αντιπροσωπεύει την χρονική στιγμή του event και μπορεί να αναπαριστά είτε τον αριθμό των χτύπων είτε τον πραγματικό χρόνο (σε δευτερόλεπτα) από την αρχή του κομματιού. Αξίζει να σημειωθεί ότι οι 2 παραπάνω αναπαραστάσεις είναι ισοδύναμες και μπορούμε γνωρίζοντας μερικά χαρακτηριστικά του κομματιού να τις εναλλάσσουμε. Συνεπώς ο χρόνος για τον οποίον μια νότα είναι ενεργή υπολογίζεται άμεσα ως η διαφορά της χρονικής στιγμής που τελειώνει μια νότα (Note off event) μείον την στιγμή που η αντίστοιχη νότα ενεργοποιείται (Note On event). Συνεπώς η τιμή του χρόνου είναι μια συνεχής μη αρνητική μεταβλητή και για τον λόγο αυτόν

δεν μπορεί να μοντελοποιηθεί όπως το Pitch και το Velocity (δεν αποτελεί ένα πρόβλημα ταξινόμησης γιατί δεν υπάρχουν κλάσεις ή αν υπάρξουν είναι υπερβολικά πολλές για να μπορέσουν να μοντελοποιηθούν αποδοτικά).

3.2.2 Πληροφορίες κάθε νότας

Κάθε νότα ενός Midi αρχείου απαρτίζεται 3 βασικές πληροφορίες την συχνότητα της (Pitch), την ταχύτητα που παίχθηκε (Velocity) καθώς και τον χρόνο που αυτή είναι ενεργή. Όπως έχει αναφερθεί και παραπάνω υπάρχουν και αλλά σημαντικά στοιχεία όπως ή πίεση που αυτή αφέθηκε κ.α. αλλά οι παραπάνω αποτελούν τα πλέον σημαντικά χαρακτηριστικά για την ανάλυση και την μοντελοποιήση των κομματιών. Από αυτά τα στοιχεία οι αρχιτεκτονικές που αναπτύχθηκαν στα πλαίσια της διπλωματικής αυτής χρησιμοποιούν μόνο τι δυο από τις τρείς παραπάνω παραμέτρους, δηλαδή το pitch και την χρονική διάρκεια, αγνοώντας τελείως την επιτάχυνση κάθε νότας. Η παραπάνω επιλογή έγινε για τον λόγο ότι η μοντελοποίηση της επιτάχυνσης δεν καθίσταται σύνθετο έργο μιας και αποτελεί μια διακριτή μεταβλητή με ένα πεπερασμένο σύνολο τιμών ενώ η απαλοιφή της δεν μειώνει κατά πολύ το περιεχόμενο κάθε κομματιού (όπως για παράδειγμα αν δεν λαμβανόταν υπόψιν το pitch).

Συνεπώς κάθε νότα που μελετάται, επεξεργάζεται ή παράγεται από κάποιο από τα μοντέλα μηχανικής μάθησης που κατασκευάστηκαν αποτελείται αποκλειστικά από την συχνότητα και την χρονική της διάρκεια. Η πρώτη παράμετρος, όπως και η επιτάχυνση, είναι αρκετά απλή στην μοντέλοποιήση της μιας και το πεδίο τιμών τις είναι [0, 126]. Από την άλλη η χρονική διάρκεια κάθε νότας είναι μια συνεχής μεταβλητή και η αναπαράστασή της είναι πιο απαιτητική. Έτσι για την αποδοτική μελέτη του χρόνου καθίσταται επιτακτική ανάγκη η μετατροπή του σε μια διακριτή μεταβλητή. Ο μετασχηματισμός αυτός είναι δυνατός χρησιμοποιώντας ορισμένους βασικούς μουσικούς κανόνες.

Αρχικά απαιτείται ο μετασχηματισμός του χρόνου από συνεχή σε διακριτή τιμή, μετατρέποντας δηλαδή τον χρόνο άφιξης των μηνυμάτων από δευτερόλεπτα σε χτύπους.

Ο παραπάνω μετασχηματισμός είναι αρκετά απλός μιας και είναι έμφυτος στην πλειονότητα των κωδικοποιήσεων των συστημάτων που υλοποιούν την διεπαφή Midi. Στα περισσότερα συστήματα που αναλαμβάνουν την εξαγωγή πληροφορίας από Midi κομμάτια ορίζεται μια βασική μονάδα χρονισμού τα ticks όπου σε κάθε ένα από αυτά μεταφέρεται ένα μήνυμα. Τα ticks εκφράζονται σε μονάδες ανά λεπτό, όπως και τα beats, αλλά οι μονάδες αυτές δεν ταυτίζονται, μιας και τα ticks βοηθούν απλά στην υλοποίηση της διεπαφής και δεν έχουν κάποια μουσική αξία. Έτσι η παραπάνω μετατροπή έγινε αυτόματα από τα συστήματα που χρησιμοποιήθηκαν για την εξαγωγή πληροφορίας από τα κομμάτια τους συνόλου εκπαίδευσης[17] (περισσότερα στο Κεφάλαιο 13 όπου αναλύεται η υλοποίηση).

Στην συνέχεια έχοντας χρονίσει το κομμάτι σε χτύπους η μετατροπή των διαρκειών σε διακριτές τιμές είναι αρκετά εύκολη αφού για κάθε κομμάτι Midi μαζί με άλλες υπερπαραμέτρους του (όπως το συνθέτης, το όργανο κ.α. ως meta μηνύματα) ορίζονται και οι αριθμοί των χτύπων ανά τέταρτο.

Έτσι για παράδειγμα αν έρθει ένα μήνυμα Note On στο tick= 4 και ένα Note Off στο tick= 8 τότε αυτή η νότα διαρκεί 4 κτύπους. Αν για το παραπάνω κομμάτι οι χτύποι ανά τέταρτό είναι ίσοι με 16 τότε η διάρκεια της παραπάνω νότας ισούται με 0.25 τέταρτα ή αλλιώς ένα δέκατο έκτο.

Με τον παραπάνω αλγόριθμο, για παράδειγμα, στο σύνολο δεδομένων του πιάνο υπάρχουν 87 διαφορετικές τιμές Pitch, 103 διαφορετικές τιμές Velocity ενώ μόλις 54 διαφορετικές τιμές χρόνου (ενώ αρχικά ήταν μια συνεχής μεταβλητή).

Παρόλα αυτά ο παραπάνω μετασχηματισμός έφερε και απώλεια κάποιας μουσικής πληροφορίας. Για παράδειγμα οι σύνθετες νότες (μια νότα η οποία αποτελείται δύο ή περισσότερες νότες με το ίδιο pitch) αναπαρίστανται ως μια ακολουθία πολλών νοτών με το

ίδιο Pitch και με διάρκειες ίσες με τις διάρκειες των βασικών νοτών. Αυτή η επιλογή δεν είναι η μοναδική που θα μπορούσε να γίνει. Για παράδειγμα θα μπορούσε να αυξηθεί ο διαφορετικός αριθμός των χρονικών διαρκειών και οι νότες αυτού του είδους να μείνουν ως έχουν. Η επιλογή αυτή στο σύνολο δεδομένων του πιάνο αυξάνει τον συνολικό αριθμό των διαφορετικών διαρκειών από 54 σε 498. Η αύξηση δεν είναι δραματικά μεγάλη αλλά τελικά επιλέχθηκε η πρώτη τεχνική όπου ο χρόνος έχει μικρότερο πεδίο τιμών και μικρότερη πόλωση (περισσότερα για την επιλογή αυτή εξηγούνται παρακάτω).

3.3 Εξισορρόπηση Δεδομένων

Στα πλαίσια της διπλωματικής αυτής κατασκευάστηκαν διαφορά μοντέλα και εκπαιδευτήκαν με dataset από διαφορετικά μουσικά όργανα. Η χρήση διαφορετικών μουσικών οργάνων για την εκπαίδευση πιστεύεται ότι θα αλλάξει την ποιότητα των παραγόμενων αποτελεσμάτων. Σε έναν παραλληλισμό με την παραγωγή κειμένου κάτι αντίστοιχό θα ήταν η αλλαγή του ύφους των κειμένων που χρησιμοποιείται για την εκπαίδευση (κωμικό, δραματικό, περιπέτεια). Συγκεκριμένα χρησιμοποιήθηκαν δεδομένα από **πιάνο** και από **κιθάρα** καθώς και **παραλλαγές τους αλλά και συνδυασμοί αυτών**.

Σε ένα κομμάτι ορισμένες νότες εμφανίζονται πολύ πιο συχνά από άλλες. Αυτό δημιουργεί σημαντικά προβλήματα στην διαδικασία εκπαίδευσης μιας και το δίκτυο πολώνεται στις πιο συνηθισμένες νότες και όπως είναι λογικό οι απαντήσεις του θα τείνουν να στρέφονται γύρω από αυτές. Για την επίλυση του παραπάνω προβλήματος δημιουργήθηκε ένα νέο σύνολο δεδομένων στο όποιο από κάθε νότα έχει αφαιρεθεί ένας τυχαίος αριθμός από 0 έως 12 από το Pitch, ενώ τα υπόλοιπα χαρακτηριστικά της παραμένουν τα ίδια. Η παραπάνω διαδικασία εφαρμόζεται σε όλα τα σύνολα δεδομένων.

Στα σύνολα αυτά πλέον έχουν μειωθεί κατά πολύ οι ανομοιομορφίες που υπήρχαν ως προς το πλήθος εμφάνισης των νοτών και για τον λόγο αυτό η παραπάνω διαδικασία λέγεται διαδικασία εξισορρόπησης. Τα σύνολα αυτά θα χρησιμοποιηθούν για την εκπαίδευση όλων των μοντέλων με σκοπό να μπορέσει για γίνει μια αξιολόγηση της επίδρασης της πόλωσης των δεδομένων στην εκπαίδευση των αρχιτεκτονικών. Παρακάτω παρουσιάζονται τα datasets που κατασκευάστηκαν καθώς και ορισμένα βασικά χαρακτηριστικά τους.

Κεφάλαιο 4 - Δεδομένα Εκπαίδευσης

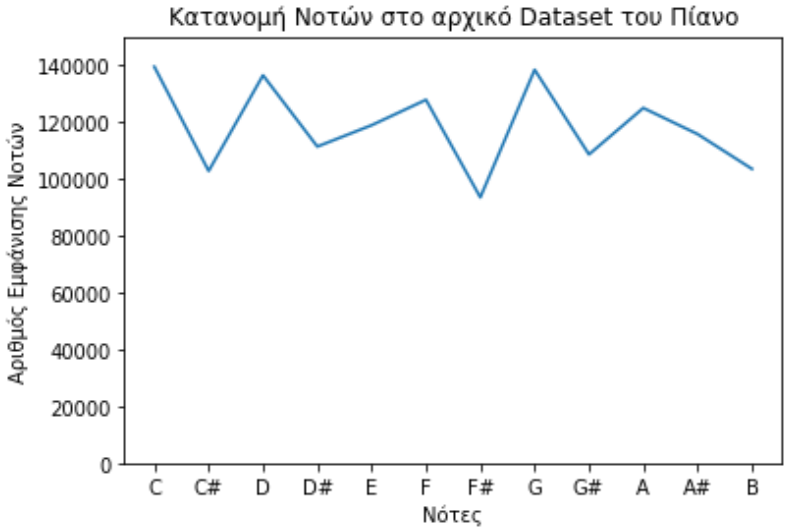
Η λειτουργία των νευρώνικων δικτύων στηρίζεται στο θεώρημα καθολικής προσέγγισης (universal approximator), το όποιο αποδεικνύει ότι ένα απλό στρώμα ενός πλήρους συνδεδεμένου δικτύου είναι ικανό να προσεγγίσει οποιαδήποτε συνάρτηση. Η συνάρτηση αυτή όπως είναι λογικό δεν υπάρχει σε αναλυτική μορφή, αλλά πηγάζει-προέρχεται από τα δεδομένα εκπαίδευσης. Έτσι τα δεδομένα αυτά πρέπει να μελετηθούν και να επιλεχθούν με μεγάλη προσοχή μιας και ο ρόλος τους είναι καθοριστικός για την ποιότητα και το είδος της εκπαίδευσης των αρχιτεκτονικών.

Για τον λόγο ένα μεγάλο μέρος της εργασίας επικεντρώνεται στα σύνολα αυτά με σκοπό να μελετηθεί εκτενέστερα ο ρόλος και η επίδρασή τους. Παρακάτω παρουσιάζονται αναλυτικά τα σύνολα εκπαίδευσης που χρησιμοποιήθηκαν μαζί με ορισμένα βασικά χαρακτηριστικά τους.

4.1 Δεδομένα από Πιάνο

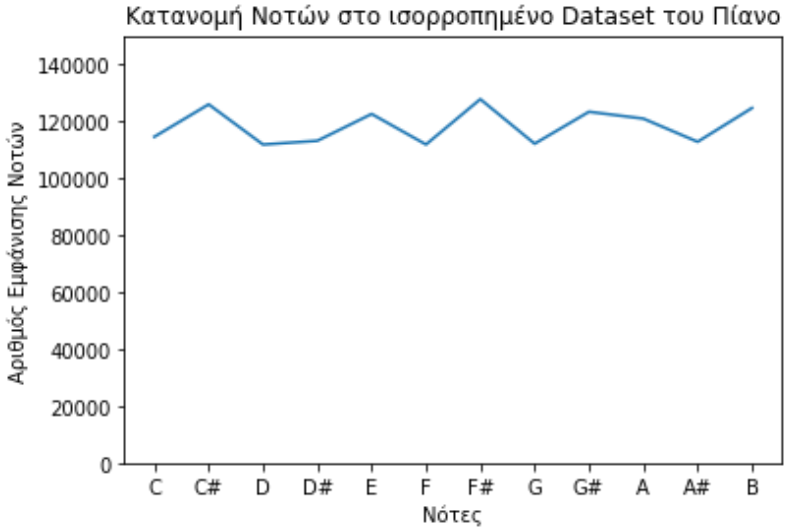
Σε αυτό το dataset υπάρχουν 623 κομμάτια από κλασσικού πιάνου τα οποία δημιουργούν μια ακολουθία 1.425.671 νοτών. Κάθε κομμάτι μεταχειρίζεται ως ακολουθία νοτών και λόγω του ότι τα κομμάτια είναι πολύ λίγα σε σχέση με των ρυθμό εμφάνισης των νοτών δεν χρησιμοποιήθηκαν κάποιες ειδικές τιμές για την εναλλαγή των κομματιών, μιας και λόγω του bias που θα δημιουργούσαν αυτές θα δυσκολεύαν την εκπαίδευση. Για να γίνει καλύτερα κατανοητό στο σύνολο αυτό η μέση τιμή εμφάνισης κάθε pitch είναι 13.172,87 σημαντικά μεγαλύτερη του 623 (που είναι οι εναλλαγές). Επίσης σκοπός των μοντέλων δεν ήταν η παραγωγή μελωδιών συγκεκριμένου μήκους (όπως σε ένα σύστημα μετάφρασης όπου για κάθε είσοδο πρέπει να παραχθεί συγκεκριμένος αριθμός λέξεων) αλλά σκοπός ήταν να δοκιμαστούν τα συστήματα στην παραγωγή μελωδιών αυθαίρετου μήκους και ιδιαίτερα να εξεταστεί η συμπεριφορά τους σε μεγάλες ακολουθίες εισόδου και εξόδου. Αυτός ήταν και ο λόγος που κατά την διαίρεση των κομμάτων σε ακολουθίες δεν χρησιμοποιήθηκαν κατάλληλες τιμές για την έναρξη και για την λήξη των ακολουθιών, όπως συνηθίζεται σε άλλες προσεγγίσεις. Συνεπώς σε κάθε σύνολο (εκπαίδευσης, επικύρωσης και δοκιμής) υπάρχουν απλά νότες στην σειρά, χωρίς να παραβάλλεται κάτι αναμεσά τους και πρακτικά μπορούν όλες να ξεκινήσουν να αναπαράγονται σαν να ήταν ένα ενιαίο κομμάτι. Δοκιμάστηκαν και άλλες τεχνικές (ειδικές τιμές για αρχή κομματιού, για αρχή και λήξη ακολουθιών κ.α.) αλλά αυτή επέδειξε τα καλύτερα αποτελέσματα.

Σε αυτό τα dataset υπάρχουν 87 διαφορετικές τιμές Pitch, 103 διαφορετικές τιμές Velocity ενώ μόλις 54 διαφορετικές τιμές χρόνου. Στο παρακάτω διάγραμμα παρουσιάζεται ο αριθμός εμφάνισης κάθε νότας στο σύνολο δεδομένων του πιάνο.



Από το παραπάνω διάγραμμα φαίνεται η κατανομή των νοτών είναι αρκετά ανομοιόμορφη. Η πιο συνηθισμένη νότα στο παραπάνω σύνολο είναι η **C** όπου εμφανίζεται 141.016 ενώ αυτή που εμφανίζεται τις λιγότερες φορές είναι η **F#** και εμφανίζεται 92.642 (εύρος 48.374 φορές). Αυτό αποτελεί σημαντικό πρόβλημα για την εκπαίδευση των μοντέλων, όπως αναφέραμε και παραπάνω λόγω της μεγάλης πόλωσης (bias) που υπάρχει στις πιο συχνές νότες. Πρακτικά το πρόβλημα αυτό παρουσιάζεται όταν σαν είσοδος στο δίκτυο δίνονται λιγότερα συνηθισμένες νότες όπου αυτά μπορούσαν να συνεχίσουν την μελωδία δυσκολότερα από ότι άλλες φορές και κατέληγαν πολύ σύντομα να ανακυκλώνουν την έξοδό τους (παραπάνω στην αξιολόγηση των μοντέλων). Το πρόβλημα αυτό ήταν πιο αισθητό στα μοντέλα που είχαν εκπαιδευτεί με τα δεδομένα κιθάρας.

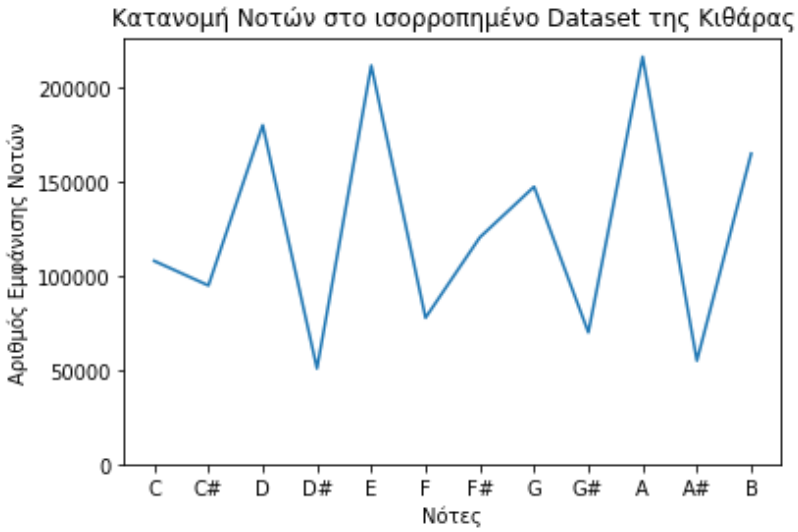
Από την άλλη στο ισορροπημένο dataset υπάρχουν 93 διαφορετικές τιμές pitch ενώ το Velocity και ο χρόνος προφανώς παραμένουν ο ίδια (διότι δεν επηρεάζονται από την διαδικασία εξισορρόπησης).



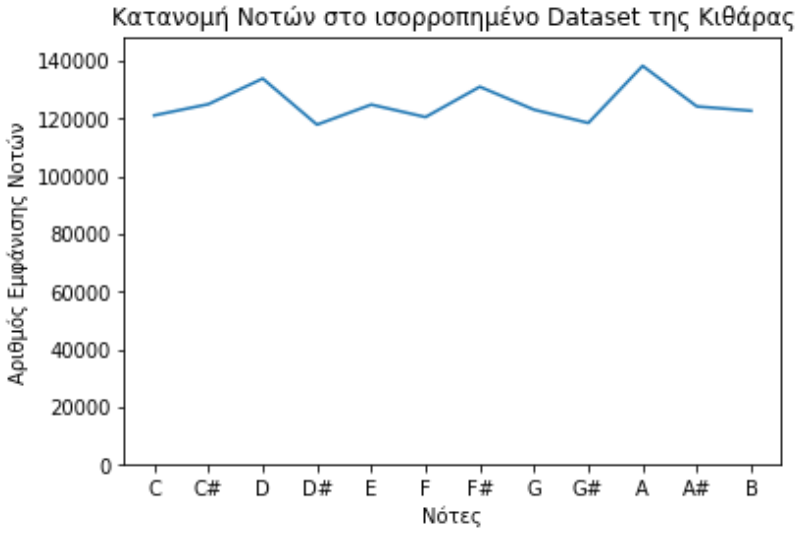
Όπως φαίνεται και από το παραπάνω διάγραμμα τα δεδομένα αυτά είναι πιο ισορροπημένα καθώς η πιο συχνή νότα είναι η **F#** η οποία εμφανίζεται 124.849 ενώ η πιο ασυνήθιστη είναι η **B** όπου εμφανίζεται 114.941 φορές (η διαφορά μειώθηκε σε 9.908), **ενώ η διασπορά του ρυθμού εμφανίσεων κάθε νότας υπό-τριπλασιάστηκε**.

4.2 Δεδομένα από κιθάρα

Σε αυτό το σύνολο δεδομένων υπάρχουν 711 κομμάτια τα όποια δημιουργούν μια ακολουθία 1.499.904 νοτών (το μήκος των ακολουθιών δεν διαφέρει ιδιαίτερα για να υπάρχει ομοιομορφία στην εκπαίδευση των μοντέλων). Σε αυτό υπάρχουν 68 διαφορετικές τιμές Pitch, 124 διαφορετικές τιμές Velocity και 178 διαφορετικές τιμές χρόνου. Όπως και με το σύνολο δεδομένων του πιάνου έτσι και σε αυτό δεν χρησιμοποιήθηκαν κάποιες ειδικές τιμές νοτών για την έναρξη ή την λήξη κομματιών ή ακολουθιών. Στο παρακάτω διάγραμμα παρουσιάζεται ο αριθμός εμφάνισης κάθε νότας στο σύνολο δεδομένων της κιθάρας.



Στο dataset αυτό όπως φαίνεται και από το παραπάνω διάγραμμα το πρόβλημα της ανομοιομορφίας των δεδομένων είναι πολύ μεγαλύτερο από ότι σε αυτό του πιάνο. Εδώ η πιο χρησιμοποιούμενη νότα είναι η **A** η οποία εμφανίζεται 217.460 φορές ενώ η ελάχιστα συνηθισμένη νότα είναι η **D#** η οποία εμφανίζεται 50.188 (διαφορά 167.272 φορές). Στο ισορροπημένο dataset η πιο συχνή νότα είναι η **A** η οποία εμφανίζεται 137.179 φορές ενώ η ελάχιστα συνηθισμένη είναι η **D#** η οποία εμφανίζεται 127.813 φορές (η διαφορά μειώθηκε σε 9.366), **ενώ η διασπορά του ρυθμού εμφανίσεων υπό-δεκαπλασιάστηκε**. Στο παρακάτω διάγραμμα παρουσιάζεται ο αριθμός εμφανίσεων των νοτών στο ισορροπημένο dataset της κιθάρας.



Όπως φαίνεται και από το παραπάνω διάγραμμα το σύνολο αυτών των δεδομένων είναι πλέον πολύ πιο ισορροπημένο. Όπως και με το dataset του πιάνου τα μοντέλα εκπαιδευτήκαν και με το αρχικό αλλά και με το ισορροπημένο dataset για να μπορεί να γίνει αξιολόγηση των αποτελεσμάτων της ανομοιομορφίας αυτής στην σύνθεση των κομματιών.

4.3 Κοινά Δεδομένα

Σε αυτό το σύνολο δεδομένων υπάρχουν 1.334 κομμάτια (623 συν 711) τα όποια αποτελούν συνδυασμό (ένωση) των δυο παραπάνω (αρχικών) συνόλων.

Όπως έχει αναφερθεί και προηγουμένως κάθε κομμάτι αποτελεί μια ακολουθία από νότες και συνεπώς δεν θα είχε κανένα νόημα απλά να τυχαία τα δύο παραπάνω σύνολα μεταξύ τους. Αντί αυτού αναμείχθηκαν τα τραγούδια μεταξύ τους και στην συνέχεια έγινε η μετατροπή των τραγουδιών σε μια ενιαία ακολουθία νοτών (που ουσιαστικά είναι το κοινό σύνολο δεδομένων). Με αυτόν τον τρόπο το δίκτυο διαβάζει τυχαία κάθε φορά κομμάτια από διαφορετικά όργανα. Η ακολουθία αυτού του συνόλου αποτελείται από 2.922.496 νότες. Αυτές περιέχουν 87 διαφορετικές τιμές Pitch και 204 διαφορετικές τιμές αξιών (διακριτές τιμές χρόνου). Όπως και στα παραπάνω σύνολα δεδομένων δεν χρησιμοποιήθηκαν κάποιες ειδικές τιμές για την έναρξη ή την λήξη τραγουδιών ή ακολουθιών για τους ίδιους λόγους που δεν χρησιμοποιήθηκαν και στα ξεχωριστά σύνολα δεδομένων.

4.4 Προ-επεξεργασία Δεδομένων

Όπως έχει αναφερθεί και προηγουμένως από κάθε Midi αρχείο εξάγεται μια ακολουθία νοτών η οποίες αν παιχθούν ακολουθιακά η μια πίσω από την άλλη θα ακουστεί το αρχικό κομμάτι (αγνοούνται άλλες παράμετροι όπως κανάλια κ.α. μιας και οι μελωδίες των δεδομένων είναι μονοφωνικές). Κάθε μια από τις νότες αυτές απαρτίζεται από 3 διαφορετικές μεταβλητές το Pitch της νότας, το Velocity καθώς και τον χρόνο. Παρόλα αυτά τα μοντέλα εκπαιδευτήκαν στο να διαβάζουν και να προβλέπουν μόνο τις 2 από αυτές το pitch και τον χρόνο, μιας και μόνο με αυτές τις παραμέτρους δίνεται μια αρκετά καλή εικόνα της μουσικής πληροφορίας και ένας άνθρωπός (απλός χρήστης ή μουσικός) μπορεί να έχει μια αρχική εικόνα της ποιότητας του κομματιού.

Συγκεκριμένα εξετάστηκαν 2 διαφορετικές προσεγγίσεις στην κωδικοποίηση αυτών των παραμέτρων. Η πρώτη ήταν το Pitch και ο χρόνος να κωδικοποιηθούν ξεχωριστά και έπειτα να γίνει η συνένωση των κωδικοποιήσεων τους. Για παράδειγμα στο αρχικό dataset του πιάνου αν μια νότα είχε Pitch 62 και η αξία της ήταν ¼ ($7^{η}$ διαφορετική τιμή) τότε η είσοδος στο δίκτυο θα ήταν η παρακάτω:

Τύπος	Pitch								Αξία					
Θέση	1	2	…	61	62	63	…	87	1	…	7	8	…	54
Είσοδος	0	0	0	0	1	0	0	0	0	0	1	0	0	0

Δηλαδή η είσοδος αποτελείται από 87 + 54 = 141 διαφορετικές τιμές (για το αρχικό σύνολο δεδομένων του πιάνου) με 139 τιμές 0 και δύο 1 (μια για την θέση του Pitch και μια για την θέση της αξία). Πρακτικά η αναπαράσταση αυτή δηλώνει ότι σε κάθε μοντέλο πρέπει να υπάρχουν επίπεδα τα όποια ασχολούνται αποκλειστικά με το Pitch κάθε νότας και αντίστοιχα επίπεδα που ασχολούνται αποκλειστικά με την αξία της. Τέτοια μπορεί να είναι κάποιο στρώμα Embedding που αναλαμβάνει την κωδικοποίηση κάθε μιας παραμέτρου ή κάποιο τελικό στρώμα Dense που αναλαμβάνει την αποκωδικοποίηση την πρόβλεψης.

Η κωδικοποίηση αυτή δεν επιλέχθηκε τελικά λόγω του μεγάλου bias στις τιμές των μεταβλητών (κυρίως στα επίπεδα που αναφέρθηκαν παραπάνω). Συγκεκριμένα στο αρχικό σύνολο του πιάνου το **57.38%** των νοτών έχουν αξία 0. Αντίστοιχο πρόβλημα υπάρχει και στο pitch, όπου η νότα με pitch 62 εμφανίζεται με συχνότητα **52.37%**. Αυτό σημαίνει ότι πάνω από τις μισές νότες έχουν το ίδιο pitch και την ίδια αξία. Αξίζει να σημειωθεί ότι τα

ποσοστά αυτά προέρχονται **623 διαφορετικά** κομμάτια. Το πρόβλημα αυτό είναι πολύ εντονότερο αν μελετηθούν τα ποσοστά των νοτών που είναι κοινά σε κάθε κομμάτι ή σε τμήματα των κομματιών, π.χ. σε κάθε 20 ή 30 νότες. Εκεί φαίνεται ότι αν για παράδειγμα σπάσουμε την αρχική ακολουθία (των 1.500.000 νοτών) σε κομμάτια μήκους **30 νοτών** τότε τα κομμάτια όπου έχουν πάνω από το **80%** του pitch τους ίδια (δηλαδή πάνω από τις 24 στις 30 νότες είναι ίδιες) είναι πάνω από **7000** ενώ υπάρχουν **3125** κομμάτια όπου **και οι 30 νότες έχουν ακριβώς το ίδιο pitch**. Στο πεδίο του χρόνου επειδή οι διαφορετικές τιμές είναι λιγότερες το πρόβλημα είναι πιο αισθητό. Συγκεκριμένα με το ίδιο μήκος ακολουθιών τα κομμάτια των οποίων οι νότες έχουν πάνω από το **70%** των αξιών τους ίδια είναι **227.497 (15.91%)** ενώ πάνω από **90%** είναι **11.597** και τελικά σε **9145** τέτοια κομμάτια και οι **30 νότες έχουν ακριβώς την ίδια αξία**. Όπως είναι λογικό όσο μειώνεται το μήκος των ακολουθιών που μελετώνται τόσο το παραπάνω πρόβλημα μεγαλώνει. Παρακάτω παρουσιάζεται ένας συνοπτικός πίνακας με τις παραπάνω πληροφορίες ανά σύνολο δεδομένων και μήκος ακολουθίας.

	Pitch				Αξία			
	20%<	>70%	>80%	=100%	20%<	>70%	>80%	=100%
Πιάνο	12.584	10.741	6.243	2.863	4	228.732	49.392	11.488
Κιθάρα	5.144	32.460	17.149	9.878	3.089	175.930	40.314	610

Seq_length = 20

	Pitch				Αξία			
	20%<	>70%	>80%	=100%	20%<	>70%	>80%	=100%
Πιάνο	59.443	9.971	5.372	2.040	0	227.497	48.479	9.145
Κιθάρα	37.338	25.337	12.064	5.890	4.897	173.229	39.082	460

Seq_length = 30

Οι παραπάνω πληροφορίες είναι καίριας σημασίας διότι ένα νευρωνικό δίκτυο δεν εξετάζει κάθε φορά ολόκληρο το σύνολο δεδομένων ούτε ολόκληρα τραγούδια αλλά κομμάτια αυτών μήκους 20 ή 30 νοτών (μεγαλύτερές ακολουθίες αυξάνουν το βάθος του δικτύου και συνεπώς αυξάνουν κατά πολύ τον χρόνο αλλά και την ποιότητα εκπαίδευσης βλ. παρακάτω). Συνεπώς η πραγματική εικόνα της εισόδου που βλέπουν τα δίκτυα είναι αυτή που παρουσιάστηκε στους παραπάνω πίνακες στην οποία τα δεδομένα κατανέμονται πολύ ανομοιόμορφα.

Αυτή η ασσυμετρία όπως έχει αναφερθεί και παραπάνω δυσκολεύει κατά πολύ την εκμάθηση. Οι πρώτες απόπειρες εκπαίδευσης των μοντέλων με αυτή την κωδικοποίηση της εισόδου δεν έδωσαν καθόλου καλά αποτελέσματα μιας και η μάθηση γινόταν παρά πολύ γρήγορά αλλά κατά την πρόβλεψη, όπως αναμενόταν, δεν έφεραν κανένα ουσιαστικό-ενδιαφέρον αποτέλεσμα (παρήγαγαν συνεχώς την ίδια έξοδο).

Για να αποφευχθεί το παραπάνω πρόβλημα χρησιμοποιήθηκε διαφορετική κωδικοποίηση της εισόδου. Σε αυτήν το pitch και η αξία κάθε νότας συνενώνονται πριν την τελική κωδικοποίηση σχηματίζοντας μια νέα δομή όπου κάθε τέτοια απαρτίζεται από τις δυο αυτές παραμέτρους. Για παράδειγμα η νότα με Pitch 62 και αξία ¼ είναι πλέον η νότα 956 όπου κωδικοποιείται ως εξής:

Θέση	0	1	2	…	955	956	957	…	2048
Είσοδος	0	0	0	0	0	1	0	0	0

Δηλαδή σε αυτήν την περίπτωση με την θέση της νότας εννοείται και το pitch αλλά και η αξία της. Η αλλαγή αυτή έδειξε μείωση της ανομοιομορφίας της εισόδου και το δίκτυο πλέον «βλέπει» μια πιο ομοιόμορφη κατανομή. Αυτό οφείλεται στο γεγονός ότι η θέση της νότας αλλάζει κάθε φορά είτε αν αλλάξει το Pitch είτε αν αλλάξει η αξία της. Παρακάτω

παρουσιάζονται οι ανάλογοι πίνακες που παρουσιάστηκαν και προηγουμένως, με το ποσοστό των κοινών τιμών στα κομμάτια των εισόδου για μήκος ακολουθιών 20 και 30, για τα αρχικά σύνολα δεδομένων των δυο διαφορετικών οργάνων.

	20%<	>70%	>80%	=100%
Πιάνο	23.449	8.669	3.585	22
Κιθάρα	27.707	13.556	3.608	79

Seq_length = 20

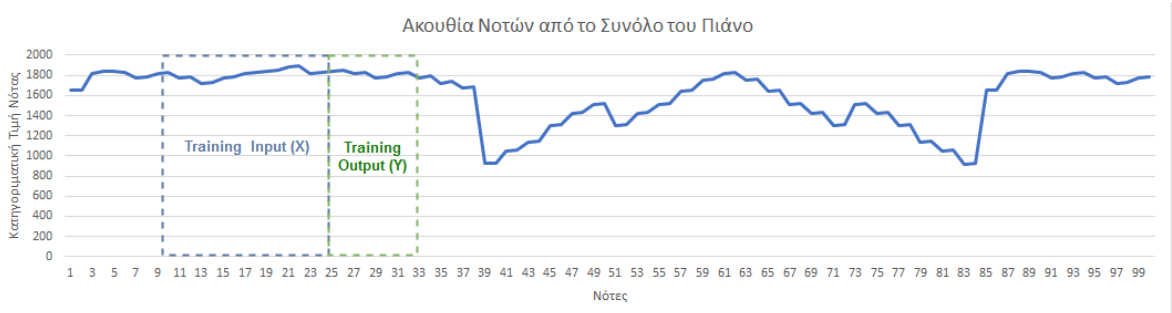
	20%<	>70%	>80%	=100%
Πιάνο	80.269	8.744	3.360	0
Κιθάρα	84.828	12.181	3.100	40

Seq_length = 30

Η κωδικοποίηση αυτή έδειξε πολύ καλύτερα αποτελέσματα σε σχέση με την προηγούμενη και η τελική εκπαίδευση των μοντέλων έγινε με τις εισόδους όπως περιγράφηκαν παραπάνω. Ο συνολικός αριθμός των πιθανών τιμών είσοδού δεν αυξήθηκε σημαντικά. Πλέον στο αρχικό σύνολο δεδομένων του πιάνο υπάρχουν 2060 διαφορετικές τιμές, στης κιθάρας 2440 ενώ στο κοινό dataset (το οποίο είναι σχεδόν διπλάσιο σε μέγεθος) υπάρχουν 3.326. Στα ισορροπημένα σύνολα νοτών η εικόνα δεν διαφέρει σημαντικά όπου για το πιάνο έχουμε 2182 διαφορετικές τιμές ενώ για την κιθάρα 2764.

4.5 Δομή Συνόλων Εκπαίδευσης

Κάθε σύνολο εκπαίδευσης ουσιαστικά αποτελείται από έναν αριθμό κομματιών σε μορφή Midi. Αυτά μετασχηματίζονται με τις μεθόδους που αναφέρονται παραπάνω και τελικά δημιουργείται μια ενιαία ακολουθία νοτών για όλα τις μελωδίες του συνόλου. Σκοπός των μοντέλων που κατασκευάστηκαν είναι δοσμένης μια αρχικής ακολουθίας να συντεθεί η συνεχεία της. Για τον λόγο αυτόν τα δεδομένα του συνόλου εκπαίδευσης ουσιαστικά κατασκευάζονται από 2 κινούμενα παράθυρα, ένα της εισόδου και ένα της επιθυμητής εξόδου, όπως φαίνεται στο παρακάτω σχήμα.



Τα παράθυρα, σε κάθε βήμα του αλγορίθμου κατάρτισης, μετακινούνται κατά μια θέση προς τα δεξιά σχηματίζοντας τελικά ένα σύνολο το οποίο αποτελείται από n βήματα, όσες και οι νότες του αρχικού συνόλου εκπαίδευσης. Δηλαδή για παράδειγμα για το πιάνο, σχηματίζεται μια ακολουθία μήκους 1.425.671 και από αυτήν δημιουργούνται 1.425.671 δείγματα εισόδου και εξόδου όπου το κάθε ένα αποτελείται από ακολουθίες μεγέθους n_{in}, n_{out}, όπως φαίνεται και στο παραπάνω διάγραμμα.

Οι ποσότητες n_{in}, n_{out} αποτελούν υπερπαράμετρους του δικτύου και πρακτικά εκφράζουν το μέγεθος της ακολουθίας εισόδου και εξόδου του δικτύου αντίστοιχα. Η τιμή τους εξαρτάται από διάφορα χαρακτηριστικά των μοντέλων όπως την αρχιτεκτονική, τον

αριθμό των εκπαιδεύσιμων παραμέτρων, το σύνολο εκπαίδευσης κ.α. και η τιμές τους επηρεάζουν σημαντικά την ποιότητα και το είδος της παραγόμενων αποτελεσμάτων.

Κεφάλαιο 5– Κομμάτια Δικτύου

5.1 Πλήρες Συνδεδεμένο Επίπεδο (Fully Connected Layer)

Πλήρες συνδεδεμένο (Fully Connected Layers) ονομάζεται ένα επίπεδο του οποίου όλοι οι νευρώνες συνδέονται όλους τους νευρώνες του προηγούμενου επιπέδου. Η εξίσωση που το περιγράφει είναι:

$$y = W * x + b$$

Όπου W, b είναι εκπαιδεύσιμες παράμετροι του δικτύου. Ο συνολικός αριθμός παραμέτρων που χρησιμοποιούνται είναι το άθροισμα των μεγεθών των πινάκων W, b, δηλαδή:

$$\#παραμέτρων = n_{in} * n_{out} + n_{out}$$

Όπου n_{in} το μέγεθος εισόδου και n_{out} το σύνολο των νευρώνων του επιπέδου, δηλαδή το μέγεθος της εξόδου.

5.2 Embedding

Γενικότερα η είσοδος κατηγορηματικών τιμών σε ένα νευρωνικό δίκτυο είναι ένα διάνυσμα με μηδενικά και ένα μοναδικό 1 στην θέση της αντίστοιχης τιμής (one hot encoded). Δηλαδή αν έχουμε 4 πιθανές τιμές εισόδου το διάνυσμα εισόδου για την είσοδο με τιμή 3 είναι το [0, 0, 1, 0]. Η αναπαράσταση αυτή είναι πολύ ακριβή μιας και για εισόδους με πολλές πιθανές τιμές (> 2000 όπως στην περίπτωση που εξετάζεται) το διάνυσμα εισόδου θα έχει μόνο μια θέση με 1 και 2000 μηδενικά. Για την μείωση της διαστατικότητας εισόδου χρησιμοποιείται η τεχνική Embedding. Ένα Embedding Layer είναι ουσιαστικά ένα πίνακας που για κάθε τιμή εισόδου παράγει ένα διάνυσμα δεκαδικών αριθμών με συγκεκριμένου μεγέθους. Συνεπώς αν οι πιθανές είσοδοι είναι *n_inp* και το επιθυμητό μέγεθος του διανύσματος εισόδου είναι *emb_size* τότε ο Embedding matrix θα είναι ένας πίνακας μεγέθους *(emb_size, n_inp)* όπου για κάθε μια τιμή εισόδου θα παράγει ένα διάνυσμα *(1, n_inp)* που βρίσκεται στην αντίστοιχη γραμμή. Συνεπώς ο αριθμός των παραμέτρων του επιπέδου αυτού είναι:

$$\#παραμέτρων = emb\_size * n\_inp$$

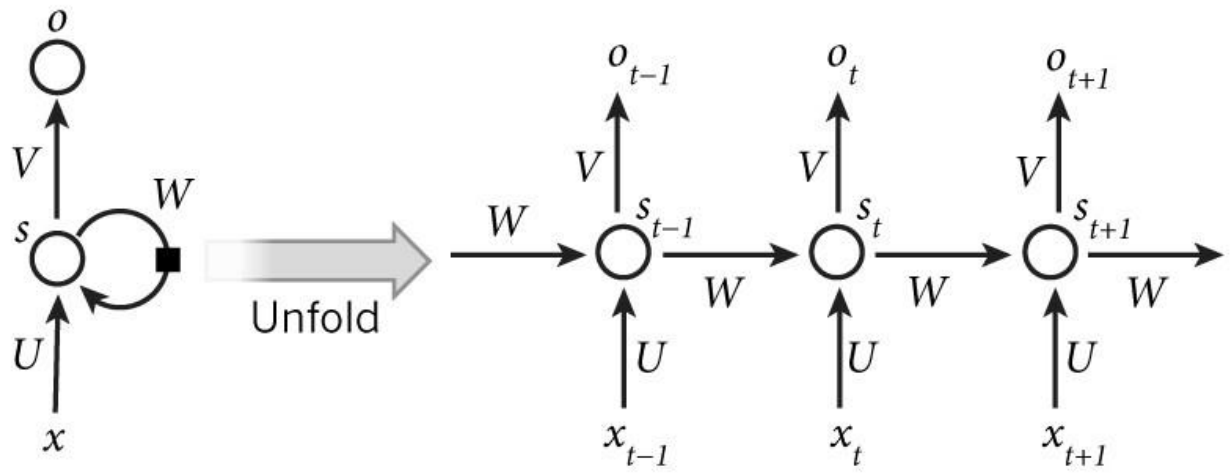
Οι τιμές του πίνακα αυτού εκπαιδεύονται όπως οι παράμετροι ενός κανονικού δικτύου (back propagation).

Η χρήση της τεχνικής αυτής στην περίπτωση την σύνθεσης μουσικής έγινε για την μείωση της διαστατικότητας της εισόδου ώστε η είσοδος να είναι πιο διαχειρίσιμη από το υπόλοιπο δίκτυο. Ο πίνακας αυτός επίσης μπορεί να δώσει και μια γραφική απεικόνιση των δεδομένων. Για έναν πίνακα που έχει εκπαιδευτεί με κείμενο συχνά φαίνεται ότι το αποτέλεσμα λέξεων με κοινό νόημα να έχουν πολύ μικρότερη απόσταση από λέξεις χωρίς σχέση μεταξύ τους. Έτσι αν απεικονιστούν σε δυο άξονες αυτές οι τιμές θα σχηματιστούν διάφορες γειτονίες με λέξεις. Η παραπάνω ιδιότητα είναι πολύ χρήσιμη μιας μπορεί να δώσει πολύ χρήσιμα αποτελέσματα σε σχέση με την εκπαίδευση του δικτύου.

5.3 Αρχιτεκτονική Απλού Αναδρομικού Δικτύου (RNN-LSTM)

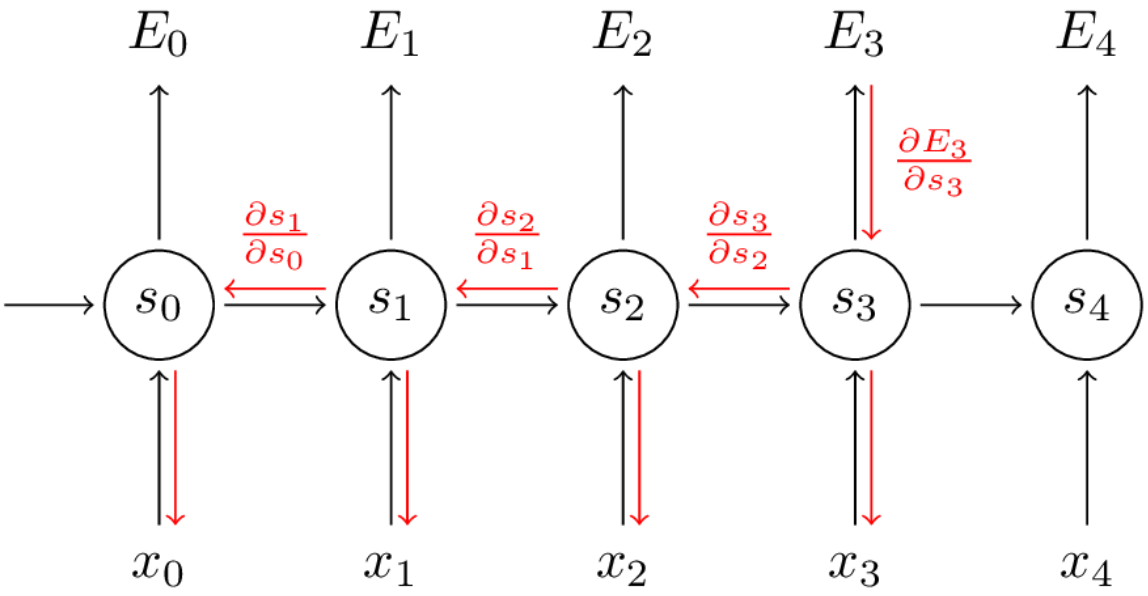
Τα αναδρομικά δίκτυα χρησιμοποιούνται για την επίλυση προβλημάτων όπου υπάρχει χρονική εξάρτηση μεταξύ των δεδομένων. Για παράδειγμα ένα πρόβλημα στο οποίο υπάρχουν τέτοιου είδους εξαρτήσεις είναι στην επεξεργασία κείμενου ή μουσικής. Στα προβλήματα αυτά η έξοδος κάθε στιγμή εξαρτάται από τα δεδομένα που έχουν περάσει μέχρι τώρα (προηγούμενες λέξεις ή νότες) και το δίκτυο πρέπει με κάποιον τρόπο να τα ‘θυμάται’.

Σχηματικά ένα αναδρομικό νευρωνικό δίκτυο παρουσιάζεται παρακάτω[10] όπου φαίνεται η είσοδος η έξοδος και το βασικό κελί:



Από το πρώτο σχήμα φαίνεται και ο λόγος για τον όποιον το δίκτυο ονομάζεται αναδρομικό. Ουσιαστικά σε κάθε βήμα η έξοδος υπολογίζεται όπως ακριβώς οι προηγούμενες αποθηκεύοντάς την συγχρόνως την είσοδο στην 'μνήμη' του. Στο δεξί σχήμα φαίνεται η λειτουργία του δικτύου για κάθε βήμα μιας ακολουθίας (ξεδιπλωμένο ουσιαστικά στον χρόνο). Ως s_t ορίζουμε την εσωτερική κατάσταση του δικτύου (hidden state). Πρακτικά αυτή είναι η μνήμη του δικτύου και υπολογίζεται από την είσοδο x_t και την μέχρι τώρα πληροφορία που έχει αποθηκευτεί ως εξής: $s_t = f(Ux_t + Ws_{t-1})$, όπου f είναι κάποια activation function ενώ η έξοδος του δικτύου δίνεται από την σχέση $o_t = g(Vs_t)$ όπου g συνήθως η softmax. Οι πράξεις αυτές ορίζουν την λειτουργία και την αρχιτεκτονική του βασικού κελιού (cell) ενός αναδρομικού δικτύου. Οι W, U, V είναι εκπαιδευσίμες παράμετροι του δικτύου και καθορίζουν την ροή και την αποθήκευση πληροφορίας σε αυτό. Αξίζει να σημειωθεί ότι για κάθε βήμα μιας ακολουθίας οι παράμετροι W,V,U είναι οι ίδιες (μοιραζόμενες παράμετροι). Θεωρητικά ένα αναδρομικό δίκτυο μπορεί να αποθηκεύσει και να ανταποκριθεί σε αυθαίρετα μεγάλες ακολουθίες εισόδου. Στην πραγματικότητα όμως κάτι τέτοιο δεν ισχύει και πρακτικά το δίκτυο 'θυμάται' πληροφορία μόνο μερικών προγενέστερων βημάτων[10] . Συνεπώς κατά την επεξεργασία μιας παραγράφου ή ενός μουσικού κομματιού το παραπάνω δίκτυο θα θυμάται μόνο τις τελευταίες λέξεις ή νότες αντίστοιχα.

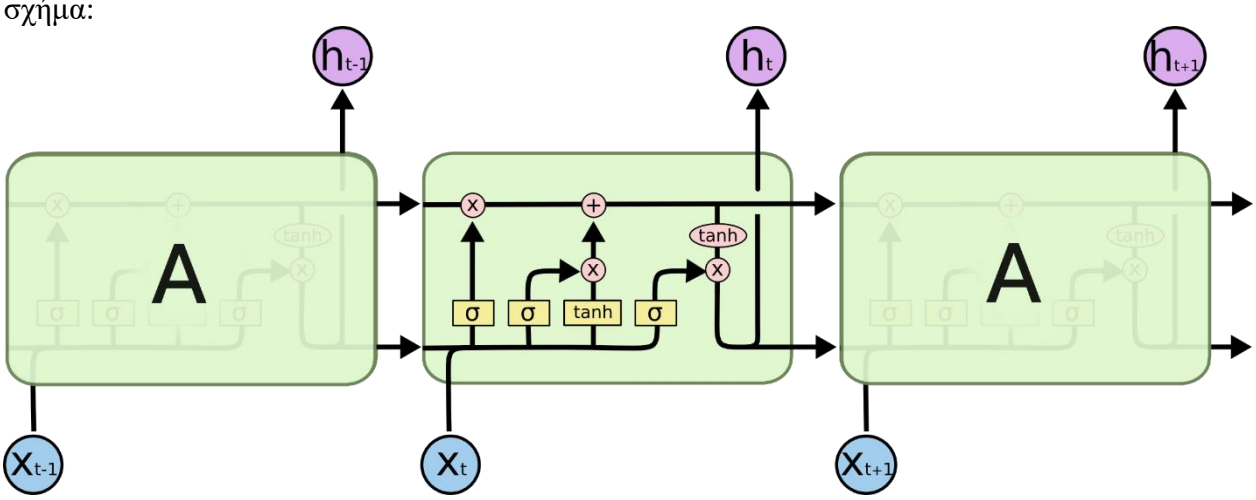
Το παραπάνω πρόβλημα πηγάζει από την μέθοδο εκπαίδευσης τέτοιων αρχιτεκτονικών, όπου χρησιμοποιείται ο αλγόριθμος back propagation through time (BPTT) και ουσιαστικά η διόρθωση των παραμέτρων γίνεται στο ξεδιπλωμένο δίκτυο με τις παράγωγους να υπολογίζονται όπως στο παρακάτω διάγραμμα[8].



Συνεπώς αν η ακολουθία εισόδου είναι μεγάλη (δηλαδή το ξεδιπλωμένο δίκτυο είναι πολύ βαθύ) τότε θα εμφανίζονται πάλι προβλήματα Vanishing και Exploding Gradient, δηλαδή το σφάλμα που μεταβιβάζεται προς τα προγενέστερα βήματα θα τείνει στο 0 ή στο ∞. Αυτό δυσκολεύει ή ακόμα καθιστά αδύνατο στο δίκτυο να εκπαιδευτεί[12].

Με τον καιρό έχουν προταθεί διάφορα μοντέλα για την επίλυση των παραπάνω προβλημάτων. Οι διαφορές των αρχιτεκτονικών αυτών βρίσκονται στην εσωτερική οργάνωση και λειτουργία των πράξεων που γίνονται σε κάθε βήμα μιας ακολουθίας (δηλαδή της αρχιτεκτονικής κάθε κελιού). Η πλέον διαδεδομένη αρχιτεκτονική κελίου για αναδρομικά νευρωνικά δίκτυα είναι το Σύστημα Μακράς Βραχυπρόθεσμης Μνήμης – LSTM [13].

Όπως δηλώνει και το όνομα το σύστημα LSTM έχει την δυνατότητα να 'θυμάται' μεγάλες ακολουθίες εισόδου, μιας και κατά την εκπαίδευση του δεν υποφέρει από τα παραπάνω προβλήματα. Η αρχιτεκτονική ενός κελιού LSTM παρουσιάζεται στο παρακάτω σχήμα:



Η λειτουργία ενός συστήματος μακράς βραχυπρόθεσμης μνήμης στηρίζεται σε 3 πύλες την πύλη διαγραφής (forget gate), την πύλη ανανέωσης (update gate) και την πύλη

εξόδου (result gate). Οι εξισώσεις ενός LSTM κελίου μαζί με το μέγεθος των διανυσμάτων του παρουσιάζονται παρακάτω:

Όνομα	Εξίσωση	Μέγεθος
Είσοδος	$X = X_t \mid H_{t-1}$	p + lstm_size
Forget Gate	$f = \sigma\left(X.W_f + b_f\right)$	lstm_size
Update Gate	$u = \sigma\left(X.W_u + b_u\right)$	lstm_size
Result Gate	$r = \sigma\left(X.W_r + b_r\right)$	lstm_size
Input	$X' = \tanh(X.W_c + b_c)$	lstm_size
Νέα C	$C_t = f * C_{t-1} + u * X'$	lstm_size
Νεα H	$H_t = r * \tanh(C_t)$	lstm_size

Όπου:

- σ η sigmoid function
- n: μέγεθος κελιού και είναι ανάλογο με την μνήμη που έχει το δίκτυο
- p: το μέγεθος του διανύσματος εισόδου
- lstm_size: το μέγεθος του κελιού

Ουσιαστικά η τιμή κάθε πύλης υπολογίζεται από ένα διαφορετικό νευρωνικό δίκτυο ενός επιπέδου. Ο προσδιορισμός πύλη πηγάζει από το γεγονός ότι αυτές έχουν πεδίο τιμών [0, 1] (λόγω της sigmoid) και πολλαπλασιάζονται ανά στοιχείο (elementwise). Έτσι αυτές ελέγχουν την ροή των δεδομένων από το ένα βήμα της ακολουθίας στο επόμενο. Η νέα κατάσταση C_t είναι αυτά που θα ξεχάσει το δίκτυο, δηλαδή το αποτέλεσμα της πύλης διαγραφής επί την προηγούμενη C_{t-1}, συν αυτά που θέλει να ανανεώσει το δίκτυο, δηλαδή το αποτέλεσμα της πύλης ανανέωσης επί την καινούργια είσοδο. Το αποτέλεσμα h_t του κελίου θα είναι το γινόμενο ανά στοιχείο της πύλης αποτελέσματος επί την νέα κατάσταση C_t, προσθέτοντας μια μη- γραμμικότητα. Συνεπώς ο συνολικός αριθμός παραμέτρων ενός LSTM κελίου είναι:

$$\#παραμέτρων = 4 * (p + lstm\_size + 1) * lstm\_size$$

Το 4 πηγάζει από το γεγονός ότι εσωτερικά βρίσκονται 4 Fully Connected Layers και το +1 από τον πίνακα βαρών των επιπέδων αυτών.

Τέλος επειδή μπορεί η προβλεπόμενη έξοδος να μην έχει το ίδιο μέγεθος με το μέγεθος των διανυσμάτων εξόδου συνηθίζεται να προστίθεται ένα Fully Connected Layer στο τέλος του κελίου ώστε να γίνει ο μετασχηματισμός αυτός.

Σε συνέχεια της παραπάνω αρχιτεκτονικής έχουν προταθεί και άλλες βελτιστοποιήσεις του παραπάνω μοντέλου οι οποίες χρησιμοποιούν μικρότερο πλήθος παραμέτρων (όπως GRU), αλλά όλες στηρίζονται στην ίδια βασική λειτουργία.

Τέλος να σημειωθεί ότι αυτές οι βασικές μονάδες μπορούν να στοιβαχθούν η μια πάνω στην άλλη (stacked) δημιουργώντας ένα πιο βαθύ νευρωνικό δίκτυο.

Κεφάλαιο 6 – Επιλογή Υπερπαραμέτρων

Για την εκπαίδευση των μοντέλων τα δεδομένα κάθε ενός dataset χωριστήκαν το κάθε ένα σε 3 διαφορετικά κομμάτια, ως είθισται:

- Το σύνολο εκπαίδευσης (train set) το οποίο χρησιμοποιείται για την εκπαίδευση του εκάστοτε μοντέλου. Αποτελεί το 60- με 70% του συνόλου των δεδομένων.
- Το σύνολο ανάπτυξης (development set) το οποίο περιέχει δεδομένα από την ίδια κατανομή με το σύνολο εκπαίδευσης και χρησιμοποιείται για την αξιολόγηση του μοντέλου καθώς και για την διασπορά των δεδομένων.
- Σύνολο δοκιμής (test set), το οποίο περιέχει δεδομένα από διαφορετική κατανομή από το σύνολο εκπαίδευσης και χρησιμοποιείται για την τελική αξιολόγηση των μοντέλων καθώς και της πόλωσης των δεδομένων.

Ο χωρισμός των δεδομένων σε train και dev sets γίνεται τυχαία πριν από κάθε εκτέλεση. Από την άλλη το σύνολο δοκιμής (test set) είναι το ίδιο σε όλες τις εκτελέσεις των αλγορίθμων εκπαίδευσης των μοντέλων. Η επιλογή αυτή έγινε για να μπορέσει να υπάρχει μια κοινή βάση σε όλα τα μοντέλα και να μπορέσει να γίνει μια δίκαιη σύγκριση μεταξύ τους.

Τον πιο σημαντικό ρόλο στην ταχύτητα αλλά και την ποιότητα εκπαίδευσης διαδραματίζουν και οι επιλογές των υπερπαραμέτρων των μοντέλων. Παρακάτω παρουσιάζονται μερικές από αυτές αλλά και τα κριτήρια με τα οποία έγιναν οι επιλογές τους.

6.1 Συναρτήσεις Ενεργοποίησης (Activation Functions)

Ως συνάρτηση ενεργοποίησης ορίζεται μια μη-γραμμική συνάρτηση η οποία παραβάλλεται μεταξύ επιπέδων και ουσιαστικά τα διαχωρίζει μεταξύ τους. Ο ρόλος της συνάρτησης αυτής είναι καθοριστικός μιας και αυτή ορίζουν το βάθος των μοντέλων. Για παράδειγμα σε ένα απλό Feed Forward δίκτυο με εκατό Hidden Layer (δηλαδή βάθους 100) αν δεν παραβάλλεται κάποια μη γραμμικότητα μεταξύ των επιπέδων τότε το μοντέλο αυτό είναι ακριβώς ισοδύναμο με ένα δίκτυο ενός Hidden Layer με ανάλογο αριθμό βαρών (χάνεται η έννοια του βάθους) μιας και πλέον η έξοδος του μπορεί να γραφεί ως συνδυασμός γραμμικών εισόδων.

Υπάρχουν διάφορες επιλογές μεταξύ των συναρτήσεων και οι πιο διαδεδομένες παρουσιάζονται παρακάτω:

- Sigmoid Activation (σ): Η συνάρτηση αυτή έχει αρκετά καλές ιδιότητες όπως ότι είναι μη- γραμμική, διαφορίσιμη καθώς και το πεδίο τιμών τις είναι (0,1) και έτσι μπορούν να εκφραστούν τα αποτελέσματα της ως πιθανότητες. Παρόλα αυτά έχει το μειονέκτημα ότι η μέγιστή τιμή της παραγώγου της sigmoid $\left(\frac{d\sigma}{dx}\right)$ είναι πολύ μικρή, μόλις 0.25. Αυτό δημιουργεί πρόβλημα κατά την εκπαίδευση βαθιών αρχιτεκτονικών, μιας και στον back propagation περνιέται μόνο ένα μικρό μέρος του σφάλματος στα πίσω επίπεδα, με αποτέλεσμα το δίκτυο τα πρώτα επίπεδά των δικτύων να εκπαιδεύονται πολύ πιο αργά (Vanishing Gradient) .
- Tanh: Η συνάρτηση αυτή είναι επίσης μια μη- γραμμική και διαφορίσιμη συνάρτηση. Σε αντίθεση με την sigmoid όμως το πεδίο τιμών της είναι στο (-1, 1), το οποίο δεν είναι υποσύνολό του πεδίου τιμών των πιθανοτήτων. Παρόλα αυτά ή μέγιστή τιμή της παραγώγου της είναι 1 με αποτέλεσμα να μην παρουσιάζονται τα προβλήματα της πρώτης.
- ReLU (Rectified Linear Unit): Το πεδίο τιμών της ReLU είναι το (0, +∞) και επίσης είναι μη- διαφορίσιμη στο 0 (γενικά υπάρχει λύση για το πρόβλημα αυτό). Το μεγάλο θετικό αυτής της οικογένειας συναρτήσεων (ReLU, PReLU, RPeLu) είναι ότι η τιμή

της παραγώγου για τιμές > 0 είναι πάντα 1, δηλαδή κάθε φορά περνιέται το μέγιστο δυνατό σφάλμα στα προηγούμενα επίπεδα κατά την εκτέλεση του back propagation.

- Softmax: Η συνάρτησή αυτή χρησιμοποιείται σαν activation function στο τελευταίο επίπεδο των δικτύων που λύνουν προβλήματα ταξινόμησης (Logistic Regression). Η εξίσωση αυτή παρουσιάζεται παρακάτω:

$$softmax = \frac{\exp(y^j)}{\sum_{j=1}^{N} \exp(y^j)}$$

Όπου N είναι το πλήθος των πιθανών προβλέψεων των εξόδων, δηλαδή αν έχουμε 4 πιθανές εξόδους (συνεπώς κατανομή 4 παραμέτρων) τότε το N=4.

Δηλαδή η έξοδος της softmax εκφράζει την πιθανότητα κάθε μιας από τις εξόδους ενός επιπέδου να είναι η πραγματική. Οπότε το άθροισμα όλων των εξόδων ενός επιπέδου ισούται με 1 και συνεπώς η έξοδος της softmax είναι μια κατανομή N μεταβλητών.

Για τους παραπάνω λόγους ως activation function στο τελευταίο επίπεδο χρησιμοποιείται η softmax, ενώ σε άλλα επίπεδα όπως στο εσωτερικό του LSTM χρησιμοποιείται η sigmoid και η tanh. Τέλος ανάμεσα των πλήρως συνδεδεμένων επιπέδων χρησιμοποιείται η ReLU. Η παραπάνω επιλογές είναι και οι πιο συνήθης στην βιβλιογραφία.

6.2 Συναρτήσεις Κόστους

Η συνάρτηση κόστους στο πλαίσιο των νευρωνικών δικτύων είναι η συνάρτηση που επιθυμούμε να ελαχιστοποιήσουμε και ουσιαστικά εκφράζει την απόσταση των προβλεπόμενων τιμών από τις πραγματικές.

Σε όλα τα μοντέλα ως συνάρτηση κόστους χρησιμοποιήθηκε η κατηγορηματική διασταυρούμενη εντροπία (categorical cross entropy). Η συνάρτηση αυτή όπως αναφέρθηκε και προηγουμένως υπολογίζει την διαφορά – απόσταση μεταξύ της πραγματικής και της προβλεπόμενης εξόδου. Πρακτικά έστω ότι η πραγματική έξοδος έχει την κατανομή y ενώ η έξοδος έπειτα από κάποια πρόβλεψη έχει την κατανομή $\hat{y}$. Το αποτέλεσμα της categorical cross entropy είναι:

$$H(y, \hat{y}) = -\frac{1}{N} \sum_{n=1}^{N} [y_n \log(\widehat{y_n}) + (1 - y_n) \log(1 - \widehat{y_n})]$$

Όπου N είναι το πλήθος των πιθανών προβλέψεων των εξόδων. Συνεπώς κατά την εκπαίδευση ενός δικτύου με αυτήν την συνάρτηση κόστους προσπαθούμε να κάνουμε την προβλεπόμενη κατανομή κάθε στιγμή να ταυτίζεται με την πραγματική. Σημαντικό για την λειτουργία των παραπάνω είναι η προβλεπόμενη έξοδος να είναι πραγματικά μια κατανομή δηλαδή κάθε έξοδος να αποτελείτε από N αριθμός με πεδίο ορισμού το [0, 1] όπου θα αθροίζουν στο 1. Συνεπώς από τα παραπάνω η μοναδική συνάρτηση ενεργοποίησης που πληροί της προϋποθέσεις αυτές είναι η softmax (και η sigmoid έχει κατάλληλο πεδίο τιμών αλλά τα αποτελέσματα της δεν αποτελούν κατανομή μιας και δεν αθροίζουν στο 1). Η σύνθεση αυτή χρησιμοποιείται σχεδόν πάντα στην βιβλιογραφία σε προβλήματα ταξινόμησης.

6.3 Ρυθμός Μάθησης (Learning Rate)

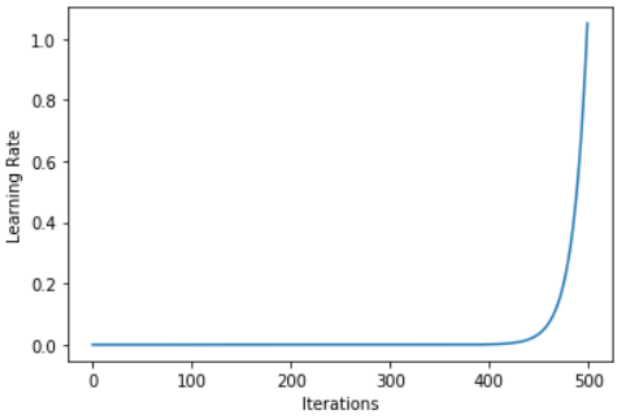
Επίσης σημαντική παράμετρος ενός δικτύου είναι ο ρυθμός μάθησης ο οποίος καθορίζει πόσο μεγάλες θα είναι οι αλλαγές που θα γίνονται στα βάρη του δικτύου σε κάθε βήμα εκπαίδευσης. Ακόμα και σε προσαρμοστικούς βελτιστοποιητές (όπως είναι ο Adam και

Rmsprop) η αρχικοποίηση του ρυθμού μάθησης παίζει μεγάλο ρόλο στην ταχύτητά εκπαίδευσης.

Αν ο ρυθμός μάθησης είναι πολύ μικρός τότε η διαδικασία της εκπαίδευσής είναι μεν πιο αξιόπιστη, αλλά πολύ πιο αργή μιας και τα βήματα μέχρι να οδηγηθεί, η συνάρτηση κόστους, σε κάποιο ελάχιστο είναι πολύ μικρά. Αντίθετα αν ο ρυθμός μάθησης είναι πολύ μεγάλος τότε μπορεί η διαδικασία να μην συγκλίνει. Αυτό μπορεί να συμβεί αν οι αλλαγές στις παραμέτρους είναι τόσο μεγάλες που να υπερβαίνουν το ελάχιστο με αποτέλεσμα να χειροτερεύει συνεχώς το σφάλμα. Συνεπώς πρέπει να υπάρξει μια κατάλληλη ρύθμιση του ρυθμού εκπαίδευσης κάπου ανάμεσα των τιμώ αυτών ώστε ο αλγόριθμός να συγκλίνει σε κάποιο λογικό χρόνο. Η συνήθης πρακτική για την εύρεση της κατάλληλης τιμής αυτής της παραμέτρου είναι η δοκιμή μεταξύ διαφόρων τιμών και η χειροκίνητη αλλαγή τους με βάση τους παραπάνω κανόνες.

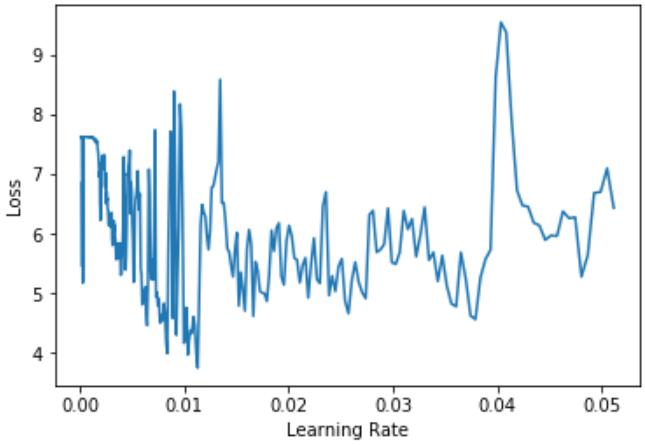
Στα δίκτυα που κατασκευάστηκαν στα πλαίσια της διπλωματικής αυτής ο ρυθμός εκπαίδευσης επιλέχθηκε με βάση έναν αυτόματο τρόπο[1], ο οποίος και αναλύεται παρακάτω.

Αρχικά η εκπαίδευση ξεκινά με έναν πολύ μικρό ρυθμό εκπαίδευσης ο οποίος αυξάνεται εκθετικά μετά από κάθε mini-batch όπως φαίνεται στο παρακάτω διάγραμμα.



Αρχικά μιας και ο ρυθμός μάθησης είναι πολύ μικρός το σφάλμα σε κάθε midi-batch θα παραμένει σταθερό (μαθαίνει πολύ αργά το δίκτυο). Στην συνέχεια καθώς ο ρυθμός αυξάνεται το σφάλμα θα αρχίσει να μειώνεται όλο και πιο γρήγορα μέχρι το σημείο όπου ο ρυθμός θα μεγαλώσει τόσο που το σφάλμα απλά θα ταλαντώνεται και θα αυξάνεται συνεχώς. Η ιδανική συνεπώς τιμή του ρυθμού μάθησης είναι αυτή για την οποία το σφάλμα μειώνεται με τον μεγαλύτερο δυνατό ρυθμό. Αυτή η τιμή θα κάνει το δίκτυο να μαθαίνει όσο τον δυνατόν γρηγορότερα χωρίς ταυτόχρονα να το περιορίζει.

Ενδεικτικά παρακάτω παρουσιάζεται η καμπύλη του σφάλματος σε σχέση με τον ρυθμό μάθησης για την αρχιτεκτονική του Autoncoder με lstm_cells = 512, seq_length_in = 30, seq_length_out = 30, batch_size = 64.



Για το παραπάνω μοντέλο ο ιδανικός ρυθμός μάθησης που υπολόγισε ο αλγόριθμ0ς είναι ο 0.00972. Στην περιοχή αυτή όπως φαίνεται και από την παραπάνω γραφική παράσταση ο ρυθμός μείωσης του σφάλματος φαίνεται να είναι ο μεγαλύτερος που επετεύχθη. Με αντίστοιχο τρόπο υπολογίστηκαν και οι ρυθμοί μάθησης για τα υπόλοιπα μοντέλα που εκπαιδευτήκαν στο πλαίσιο αυτής της διπλωματικής.

6.4 Βελτιστοποιήτες (Optimizers)

Η επιλογή του κατάλληλου βελτιστοποιήτη είναι μια πολύ σημαντική υπερπαράμετρος η οποίο επηρεάζει σημαντικά τον χρόνο εκπαίδευσης των δικτύων. Η επιλογή αυτού εξαρτάται περισσότερο από το είδος και την δομή του δικτύου και λιγότερο από τα δεδομένα εκπαίδευσης. Για τον λόγο αυτόν δεν έχουν χρησιμοποιηθεί οι ίδιοι βελτιστοποιήτες για τις 3 διαφορετικές αρχιτεκτονικές που έχουν κατασκευαστεί (Βαθύ αναδρομικό δίκτυο, Κωδικοποιητή- Αποκωδικοποιητή, Κωδικοποιητή- Αποκωδικοποιητή με Συγκέντρωση). Η πρώτη έχει εκπαιδευτεί με βάση τον αλγόριθμό Rmsprop ενώ οι υπόλοιπες χρησιμοποιώντας τον Adam. Αυτή η επιλογή είναι και η πιο συνήθης επιλογή στην βιβλιογραφία, μιας και για τα συγκεκριμένα δίκτυα έχουν επιδείξει ταχύτατα αποτελέσματα εκπαίδευσης.

Γενικότερα η πιο κλασσική και η πιο ευρέως δοκιμασμένη τεχνική εκπαίδευσης νευρωτικών δικτύων είναι η μέθοδος κατάβασης δυναμικού (Stochastic Gradient Decent ή SGD). Όλοι οι αλγόριθμοί εκπαίδευσης έχουν κοινή αρχή λειτουργιάς με αυτή του SGD και για αυτό αναφέρεται συνοπτικά ο τρόπος λειτουργίας του.

Ο SGD στηρίζεται στην ανανέωση όλων των παραμέτρων ανάλογα με την τιμή των παραγώγων πρώτης τάξης της συνάρτησης κόστους ως προς τις παραμέτρους του δικτύου την δεδομένη χρονική στιγμή. Η αλλαγή των τιμών αυτή γίνεται προς την κατεύθυνση της αρνητικής κλίσης με σκοπό ουσιαστικά την ελαχιστοποίηση της συνάρτησης κόστους. Παρόλα αυτά το γεγονός ότι η μέθοδος αυτή είναι πρώτης τάξης σημαίνει ότι λαμβάνεται υπόψιν η κλίση της συνάρτησης κόστους αλλά όχι η καμπυλότητά της με αποτέλεσμα να μην προσαρμόζεται αυτόματα ο ρυθμός μάθησης αλλά να παραμένει ορισμένος από την αρχή, ανεξάρτητα την τρέχουσα κατάσταση.

Και οι δύο μέθοδοι που έχουν χρησιμοποιηθεί είναι δεύτερης τάξης και για αυτό ανήκουν στην κατηγορία των προσαρμοστικών αλγορίθμων μάθησης.

RMSPROP: Ο αλγόριθμός αυτός όπως αναφέρθηκε και προηγούμενος είναι ένας προσαρμοστικός αλγόριθμος μάθησης και συνεπώς δεν χρειάζεται προσαρμογή ο ρυθμός μάθησης. Επίσης κατά την λειτουργία του κατασκευάζεται ένας ξεχωριστός ρυθμός μάθησης για κάθε μια παράμετρο του δικτύου ξεχωριστά. Οι εξισώσεις ανανέωσης των βαρών για κάθε μια παράμετρο w^j παρουσιάζονται παρακάτω:

$$v_t = \rho\, v_{t-1} + (1-\rho) g_t^2$$
$$\Delta w_t^j = -\frac{\eta}{\sqrt{v_t + \varepsilon}}\, g_t$$
$$w_{t+1} = w_t + \Delta w_t$$

όπου:

- η: η αρχική τιμή του ρυθμού μάθησης
- v_t: Ο εκθετικός μέσος όρος των τετραγώνων της παραγώγου
- g_t: Η παράγωγος την χρονική στιγμή t σε σχέση με την παράμετρο w^j του δικτύου.
- ρ: υπερπαράμετρος του αλγορίθμου (προτεινόμενη τιμή 0.9)

Γενικά η αυτή η μέθοδος συνιστάτε για αναδρομικά δίκτυα (RNN) όπως το πρώτο μοντέλο που χρησιμοποιήθηκε[20, 21].

ADAM: η μέθοδος εκπαίδευσης Adam είναι άλλη μια προσαρμοστική μέθοδος η όποια υπολογίζει και αυτή έναν διαφορετικό ρυθμό μάθησης για κάθε μια παράμετρο του δικτύου. Οι εξισώσεις ανανέωσης κάθε μιας παραμέτρου w^j παρουσιάζονται παρακάτω:

$$v_t = \beta_1\, v_{t-1} - (1-\beta_1) g_t$$
$$s_t = \beta_2 s_{t-1} - (1-\beta_2) g_t^2$$
$$\Delta w_t^j = -\eta\, \frac{v_t}{\sqrt{s_t + \varepsilon}}\, g_t$$
$$w_{t+1}^j = w_t^j + \Delta w_t^j$$

Όπου:

- η: η αρχική τιμή του ρυθμού μάθησης
- g_t: η παράγωγος την χρονική στιγμή ως προς την παράμετρο w^j
- β^1, β^2: υπερπαραμέτροι αλγορίθμου, όπως γ στον RMSPROP
- v^t: ο εκθετικός μέσος όρος της παραγώγου
- s_t: ο εκθετικός μέσος όρος των τετραγώνων των παραγώγων

Ο αλγόριθμος αυτός έχει χρησιμοποιηθεί σε πολλές διαφορετικές αρχιτεκτονικές με εξαιρετικά αποτελέσματα [2].

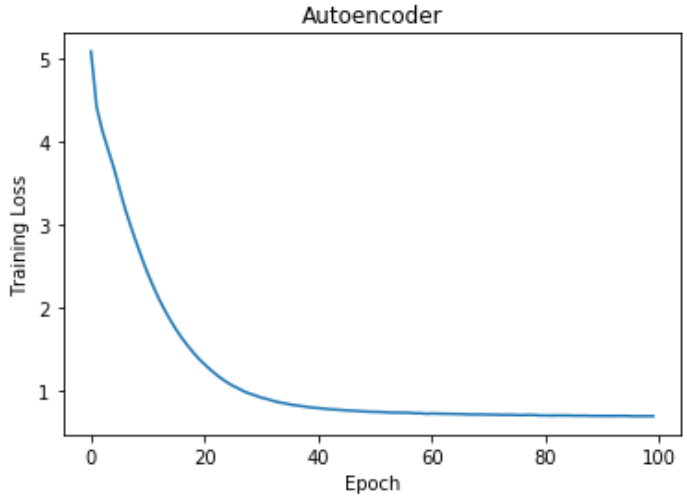
Γενικά στα μοντέλα χρησιμοποιήθηκαν και άλλοι βελτιστοποιήτες αλλά οι συγκεκριμένοι έδειξαν τα καλύτερα αποτελέσματα.

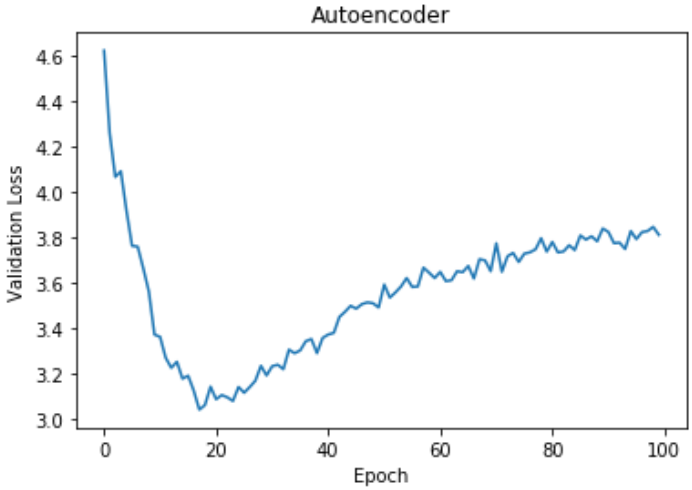
6.5 Συστηματοποιήση – Regularization

Η συνεχής αύξηση της υπολογιστικής δύναμης δίνει την δυνατότητα εκπαίδευσης όλο και πιο σύνθετών μοντέλων, με μεγαλύτερο αριθμό παραμέτρων. Αυτό έχει σαν αποτέλεσμα για την επίλυση προβλημάτων να χρησιμοποιούνται πιο μεγάλες αρχιτεκτονικές από αυτές που είναι απαραίτητες. Παρόλα αυτά στα πιο σύνθετα και πολύπλοκα δίκτυα εμφανίζονται συχνά το φαινόμενο της υπερεκαπαίδευσης (overfittng).

Το πρόβλημα αυτό ουσιαστικά πηγάζει από το γεγονός ότι το μοντέλο εκπαιδεύτηκε τόσο πολύ που πλέον έχει απομνημονεύσει τα δεδομένα εκπαίδευσής και έχει πολύ μικρό σφάλμα για αυτά. Πρακτικά όμως δεν έχει γενικεύσει σωστά με αποτέλεσμα το σφάλμα, στο σύνολο δοκιμής να είναι σημαντικά μεγάλο (ουσιαστικά το δίκτυο δεν έχει μάθει να λύνει το πρόβλημα).

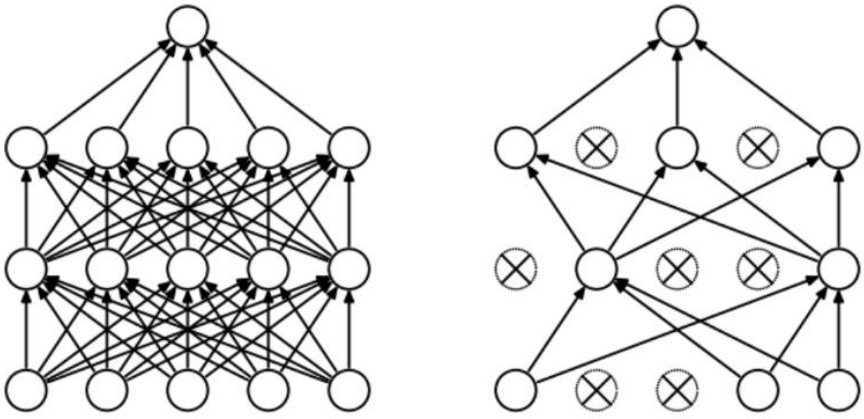
Πρακτικά το πρόβλημα του overfitting γίνεται ευκολά κατανοητό από το παρακάτω παράδειγμα όπου εκπαιδεύεται έναν απλό Autoencoder με μέγεθος κελιού 512, με το 1/10 του συνόλου εκπαίδευσης του πιάνο.





Εκ πρώτης όψεις φαίνεται ότι το σφάλμα του δικτύου συνεχώς μειώνεται άλλα στην πραγματικότητα το validation loss από ένα σημείο και έπειτα αυξάνεται συνεχώς. Αυτή είναι μια κλασσική εικόνα ενός δικτύου το οποίο έχει υπερεκπαιδευτεί.

Ο τρόπος που αντιμετωπίζεται το overfitting είναι η συστηματοποιήση (regularization). Υπάρχουν πολλές τεχνικές συστηματοποιήσης αλλά στα πλαίσια της διπλωματικής χρησιμοποιήθηκε η μέθοδος Dropout. Η τεχνική αυτή ουσιαστικά βοηθά στην μείωση της εξάρτησης μεταξύ των νευρώνων[4]. Κατά την διαδικασία εκπαίδευσης (training phase) για κάθε νευρώνα κάθε επιπέδου (με Dropout) και σε κάθε επανάληψη αγνοούνται, δηλαδή μηδενίζονται, οι έξοδοι ορισμένων κόμβων (ποσοστού p), ενώ ταυτόχρονα κλιμακώνονται οι έξοδοι των υπολοίπων. Πρακτικά αν στην έξοδο ενός επιπέδου εφαρμοστεί Dropout με ποσοστό 0.5 τότε σε κάθε βήμα εκπαίδευσης θα επιλέγονται τυχαία και θα μηδενίζονται οι έξοδοι των μισών κόμβων ενώ τον άλλων μισών θα διπλασιάζονται. Η κλιμάκωση των εξόδων γίνεται για να διατηρείτε σταθερή η κατανομή των εξόδων σε κάθε βήμα. Με αυτόν τον τρόπο ουσιαστικά δεν εκπαιδεύεται μόνο ένα δίκτυο αλλά πολλά δίκτυα μαζί μιας και σε κάθε βήμα εκπαίδευσης αλλάζουν οι εσωτερικές συνδέσεις και συνεπώς η διαρρύθμισή του. Αυτό επίσης μειώνει όπως είναι λογικό και την εξάρτηση των τιμών που έχει ο κάθε νευρώνας από τους υπολοίπους στο ίδιο επίπεδο. Παρακάτω παρουσιάζεται μια εικόνα ενός Feed Forward νευρωνικού δικτύου σε ένα βήμα εκπαίδευσης όπου ανάμεσα από κάθε επίπεδο εφαρμόζεται η παραπάνω τεχνική.



(a) Standard Neural Net (b) After applying dropout.

Srivastava, Nitish, et al. ”Dropout: a simple way to prevent neural networks from overfitting”, JMLR 2014

Τέλος κατά την πρόβλεψη χρησιμοποιούμε όλες τις εξόδους των νευρώνων απλά της μειώνουμε κάθε φορά κατά τον παράγοντα p (δηλαδή στο παραπάνω παράδειγμα τις μειώνουμε κατά 2).

6.6 Τυχαιότητα πρόβλεψης

Όλα τα νευρωνικά δίκτυα (και όλα τα κομμάτια τους) που περιγράφονται αποτελούν ντετερμινιστικές μηχανές. Αυτό σημαίνει ότι αν μια αρχική μελωδία περάσει από ένα δίκτυο δυο φορές τότε και τις δυο φορές θα παράγει ακριβώς το ίδιο αποτέλεσμα. Αυτό αποτελεί πρόβλημα μιας και ειδικά στην αυτόματη σύνθεση μουσικής το ιδανικό θα ήταν κάθε φορά το δίκτυο να παρήγαγε ένα πρωτότυπο αποτέλεσμα.

Επίσης αν έξοδος του δικτύου μια φορά είναι η ίδια με την είσοδο τότε το δίκτυο θα «κολλήσει» σε αυτή και δεν θα μπορέσει ποτέ να παράγει κάτι διαφορετικό, όσος χρόνος και να του δοθεί. Οπότε η ντετερμινιστική φύση των μοντέλων ενισχύει και τα φαινόμενα αναδίπλωσης της εξόδου.

Για την αποφυγή των παραπάνω προβλημάτων χρησιμοποιείται μια μη – ντετερμινιστική τεχνική για την επιλογή της επόμενης νότας.

Πρακτικά αντί η συνάρτηση softmax να υπολογίζεται με βάση την έξοδο του Dense Layer, υπολογίζεται με μια νέα εκδοχή της στην όποια τις έχει γίνει κάποιο scale όπως παρακάτω:

$$new\_logits = \frac{logits}{temp}$$

Στην συνέχεια τελική έξοδος επιλέγεται τυχαία με βάση κάποια κατανομή (αντί αυτής με την μεγαλύτερη πιθανότητα). Στην συγκεκριμένη περίπτωση έχει χρησιμοποιηθεί η πολυωνυμική κατανομή.

Η υπερπαράμετρος $temp$ ονομάζεται θερμοκρασία και η τιμή της επηρεάζει την τυχαιότητα της εξόδου. Αρχικά αν το $\text{temp} = 1$ τότε δεν γίνεται κανένα scale της εξόδου του dense επιπέδου και συνεπώς η επιλογή γίνεται με βάση τις πραγματικές πιθανότητες, αφού το αποτέλεσμα της συνάρτησης softmax είναι ακριβώς το ίδιο. Παρόλα αυτά αν $\text{temp} < 1$ (π.χ. 0.6) τότε το αποτέλεσμα των logits είναι πιο μεγάλο και έτσι τα παραγόμενα αποτελέσματά είναι πιο «σίγουρα», δηλαδή πιο κοντά σε αυτά που θα παρήγαγε ντετερμινιστικά. Αυτό οφείλεται στο γεγονός ότι ο υπολογισμός της softmax σε μεγαλύτερες τιμές εντείνει της διαφορές. Αντίθετα ο υπολογισμός της σε μικρότερες τιμές, δηλαδή με $\text{temp} > 1$, παράγει μια πιο ομαλή κατανομή πιθανότητας με αποτέλεσμα μεγαλύτερη ποικιλία στην έξοδο αλλά παράλληλα και περισσότερα λάθη.

Με την χρήση της παραπάνω τεχνικής η ίδια είσοδος δεν μπορεί να παράγει την ίδια έξοδο λόγω της μη- ντετερμινιστικής επιλογής της κάθε φορά, όποτε λύνεται το φαινόμενο της αναδίπλωσης που περιεγράφηκε παραπάνω. Παράλληλα κάνει την έξοδο πιο ενδιαφέρουσα μιας και η τυχαία αυτή επιλογή μπορεί να οδηγήσει το δίκτυο στην παραγωγή πιο πρωτότυπων κομματιών. Παρόλα αυτά μπορεί μια κακή επιλογή να χαλάσει ολόκληρη την παραγόμενη μελωδία και οι χρήστες να καταλάβουν λόγω αυτής ότι η μελωδία δεν είναι πραγματική.

6.7 Αρχικοποίηση Παραμέτρων

Πριν από την εκπαίδευση κάθε μοντέλου πρέπει να γίνει η αρχικοποίηση των παραμέτρων του. Η αρχικοποίηση αυτή επηρεάζει την απόδοση και την ταχύτητα σύγκλισης του δικτύου. Υπάρχουν διάφορες τεχνικές για την αρχικοποίηση των παραμέτρων ανάλογα τον τύπο του δικτύου, τις activation function κάθε επιπέδου των αριθμό των παραμέτρων κ.α.

Γενικότερα οι παράμετροι του δικτύου αρχικοποιούνται με τυχαίες μεταβλητές της κανονικής κατανομής (με mean = 0, stdv = 1). Η αρχικοποιήση αυτή όμως μπορεί να δυσκολέψει την εκμάθηση λόγω των προβλημάτων Exploding ή Vanishing Gradient. Για παράδειγμα έστω ένα επίπεδο με N νευρώνες εισόδου που συνδέονται σε έναν κόμβο εξόδου. Για απλότητα όλες οι είσοδοι είναι 1 ενώ όλα τα βάρη έχουν αρχικοποιηθεί με την κανονική κατανομή με mean = 0 και stdv= 1. ($N(0,1)$). Τότε το αποτέλεσμα της άθροισης στον κόμβο εξόδου θα είναι το ίδιο με το άθροισμα των τιμών της κανονικής κατανομής (όλες οι είσοδοι ίσες με 1), δηλαδή θα ακολουθεί και αυτή την κανονική κατανομή με μέση τιμή mean = 0 και τυπική απόκλιση = $\sqrt{N}$. Αυτό σημαίνει ότι αν N = 512 τότε το άθροισμα στον νευρώνα εξόδου θα ακολουθεί την κανονική κατανομή με διακύμανση 22.62. Αυτή η μεγάλη τιμή της διακύμανσης δημιουργεί πρόβλημα μιας και αν ως activation function έχει χρησιμοποιηθεί η sigmoid στον νευρώνα εξόδου, τότε τις περισσότερες φορές το αποτέλεσμα του δικτύου θα είναι 0 ή 1 (λόγω του κορεσμού της συνάρτησης ενεργοποίησης).

Για να μην δημιουργείται το παραπάνω πρόβλημα το αρχικό βάρος κάθε παραμέτρου ενός δικτύου επιλέγεται όπως προηγουμένως τυχαία από τη κανονική κατανομή $N(0,1)$ και έπειτα διαιρείται με $\sqrt{\frac{k}{size_{in}-size_{out}}}$, όπου k είναι κάποιο heuristic, ενώ $size_{in}$, $size_{out}$ είναι το μέγεθος εισόδου και εξόδου αντίστοιχα. Η τεχνική αυτή ονομάζεται αρχικοποιήση Xavier[5,6,7] (ή glorot) και αυτή χρησιμοποιήθηκε σε όλα τα τελικά μοντέλα που εκπαιδευτήκαν. Το heuristic k επιλέχθηκε ίσο με 6 η όποια είναι και η προτεινόμενη τιμή [8], για αναδρομικά μοντέλα, όπως αυτά που κατασκευάστηκαν.

6.7 Αποφυγή μη- κυρτότητας

Η διαδικασία εκπαίδευσης ενός νευρωνικού δικτύου γίνεται με σκοπό να βρεθεί η ελάχιστή τιμή μιας συνάρτησης απώλειας $L_X(W)$, όπου το W αντιπροσωπεύει την μήτρα (ή τις μήτρες) βάρους μεταξύ των νευρώνων, ενώ το X αντιπροσωπεύει το σύνολό των δεδομένων κατάρτισης όπως αυτά περιεγράφηκαν παραπάνω. Αξίζει να σημειωθεί ότι η ελαχιστοποίηση της συνάρτησης L η οποία καθορίζει τις τιμές των βαρών W (αναζητούνται τα βάρη W που ελαχιστοποιούν την L) γίνεται μόνο πάνω στο σύνολο εκπαίδευσης X, το οποίο παραμένει σταθερό.

Όπως είναι λογικό αν ένα νευρωνικό δίκτυο έχει P εκπαιδεύσιμες παραμέτρους τότε η συνάρτηση κόστους L είναι μια επιφάνεια διαστάσεων P + 1. Για να γίνει κατανοητό αν το νευρωνικό δίκτυο είχε 2 παραμέτρους προς εκπαίδευση τότε η συνάρτηση κόστους θα ήταν μια επιφάνεια στον τρισδιάστατο χώρο. Η επιφάνεια αυτή θα παραμένει η ίδια όσο είναι σταθερό το σύνολο εκπαίδευσης.

Για αυτόν τον λόγο κατά την εκπαίδευση δημιουργείται το πρόβλημα της μη-κυρτότητας Αυτό σημαίνει ότι επιφάνεια $L_x(W)$ θα έχει πολυάριθμα τοπικά ελάχιστα και συνεπώς οι αλγόριθμοι κατάρτισης θα τείνουν να κολλούν σε αυτά ενώ μπορεί σε μια πολύ κοντινή απόσταση να υπάρχει μια καλύτερη ή και η βέλτιστη λύση.

Για την αποφυγή του παραπάνω προβλήματος χρησιμοποιούνται 2 τεχνικές παράλληλα. Αρχικά η εκπαίδευση του συνόλου δεν γίνεται με ολόκληρο το σύνολο (διαδικασία που θα ήταν και πολύ χρονοβόρα λόγω έλλειψης παραλληλισμού) αλλά σε μικρότερες παρτίδες (mini- batch) οι οποίες ανακατεύονται μεταξύ τους. Με το σπάσιμο και την ανακατάταξη του συνόλου δεδομένων το Χ αλλάζει σε κάθε επανάληψη είναι αρκετά πιθανό ότι δεν θα πραγματοποιηθούν δυο βήματα εκπαίδευσης σε ολόκληρη την ακολουθία επαναλήψεων και εποχών με ακριβώς το ίδιο σύνολο εκπαίδευσης.

Έστω ότι ο αλγόριθμος εκπαίδευσης έχει κολλήσει σε ένα τοπικό ελάχιστό κατά την επανάληψη i. Το σημείο αυτό ανήκει στην καμπύλη της συνάρτησης $L_{x_i}(W)$. Στην επόμενη όμως επανάληψη (i + 1) η καμπύλη αυτή αλλάζει μιας και αλλάζει και η συνάρτηση κόστους η οποία πλέον είναι η $L_{x_{i+1}}$ και αυτό έχει ως αποτέλεσμα ο αλγόριθμος εκπαίδευσης να αναπηδήσει από το σημείο που είχε κολλήσει μιας και το σημείο αυτό δεν θα ανήκει πλέον στην καμπύλη της $L_{x_{i+1}}$.

Αξίζει να σημειωθεί ότι η παραπάνω λύση δουλεύει και μόνο με την χρήση των μικρότερων παρτίδων (χωρίς δηλαδή την ανακατάταξη τους) αλλά μεγαλώνει η πιθανότητα να βρεθούν ίδιες ακολουθίες στο σύνολο εκπαίδευσης. Παρόλα αυτά η ανακατάταξη των δειγμάτων του συνόλου εκπαίδευσης δεν έχει καμία επίδραση όταν σε κάθε βήμα χρησιμοποιείται ολόκληρο το σύνολο εκπαίδευσης (δηλαδή χωρίς το σπάσιμο σε μικρότερες παρτίδες). Για όλα τα μοντέλα που εκπαιδευτήκαν στα πλαίσια αυτής της διπλωματικής χρησιμοποιήθηκαν και οι 2 τεχνικές.

Η επιλογή του μεγέθους κάθε mini-batch είναι επίσης μια σημαντική υπερπαράμετρος που πρέπει να ληφθεί υπόψιν πριν την εκπαίδευση. Η τιμή της επηρεάζεται από πολλά διαφορετικές παραμέτρους αλλά η πλέον σημαντική είναι η υπολογιστική δύναμη καθώς και ο βαθμός παραλληλίας που μπορεί να επιτευχθεί στο σύστημα που αναλαμβάνει την εκπαίδευση. Σε όλα τα μοντέλα χρησιμοποιήθηκαν 2048 δείγματα ανά mini-batch.

Κεφάλαιο 7 – Διαδικασία Αξιολόγησης

7.1 Δείγματα Μοντέλων

Συνολικά στο πλαίσιο της διπλωματικής αυτής έχουν κατασκευαστεί τα μοντέλα που παρουσιάζονται στον παρακάτω πίνακα.

Τύπος Μοντέλου	Cell Size	Dataset
Feed Forward LSTM	**256**	Piano
		Piano Balanced
		Guitar
		Guitar Balanced
	512	Piano
		Piano Balanced
		Guitar
		Guitar Balanced
		Mixed
		Mixed Balanced
Autoencoder	**256**	Piano
		Piano Balanced
		Guitar
		Guitar Balanced
	512	Piano
		Piano Balanced
		Guitar
		Guitar Balanced
		Mixed
		Mixed Balanced
Autoencoder with Attention Layer	**256**	Piano
		Piano Balanced
		Guitar
		Guitar Balanced
	512	Piano
		Piano Balanced
		Guitar
		Guitar Balanced
		Mixed
		Mixed Balanced

Συνεπώς έχουν κατασκευαστεί τα εξής 6 διαφορετικά dataset:

1. Πιάνο
2. Ισορροπημένο σύνολο Πιάνο
3. Κιθάρας
4. Ισορροπημένο σύνολο Κιθάρας
5. Κοινό – που ουσιαστικά αποτελεί την ένωση των αρχικών συνόλων Πιάνου και Κιθάρας
6. Κοινό των ισορροπημένων συνόλων Πιάνου και Κιθάρας

Με τα τέσσερα πρώτα έχουν εκπαιδευτεί όλες οι παραπάνω αρχιτεκτονικές δημιουργώντας 24 διαφορετικά μοντέλα. Παράλληλα οι τρεις από τις αρχιτεκτονικές αυτές

(αυτές με μέγεθος κελίου 512) έχουν εκπαιδευτεί και με τα κοινά σύνολα δεδομένων (ισορροπημένα και μη), δημιουργώντας έξι επιπλέον μοντέλα τα όποια είναι ικανά να συνθέσουν την συνέχεια μιας δοσμένης αρχικής μελωδίας. Συνεπώς στα πλαίσια της διπλωματικής αυτής κατασκευάστηκαν συνολικά 30 διαφορετικά μοντέλα.

Για την αξιολόγηση τους έχουν επιλεχθεί τυχαία 12 μελωδίες από κάθε dataset και έχουν περάσει από όλα τα μοντέλα δημιουργώντας τελικά 360, κατασκευασμένα από υπολογιστή κομμάτια. Στο σύνολο αυτών προστέθηκαν και τα πραγματικά κομμάτια, δηλαδή η αρχική μελωδία με την πραγματική της συνέχεια, αυξάνοντας συνεπώς τον συνολικό αριθμό των μελωδιών προς αξιολόγηση στα 432 (360 κατασκευασμένα από υπολογιστή και 12*3= 76 πραγματικά). Επίσης στα παραπάνω προστέθηκαν και 182 κομμάτια τα όποια είχαν συντεθεί από τις παραπάνω αρχιτεκτονικές αλλά με την χρήση της τυχαίας πρόβλεψης, με σκοπό να μελετηθεί η επίδρασης της στην ποιότητα των παραγόμενων αποτελεσμάτων.

Αξίζει να σημειωθεί ότι για όλα τα μοντέλα υπήρχαν ορισμένες αρχικές μελωδίες οι οποίες παρήγαγαν πολύ καλύτερα αποτελέσματα από αυτά που τα αντιπροσωπεύουν στο σύνολο των κομματιών προς αξιολόγηση αλλά για λόγους ισότιμης και δικαίας αξιολόγησης των δικτύων οι αρχικές μελωδίες ήταν τυχαίες και κοινές για όλα τα μοντέλα.

Τέλος η αναζήτηση χρηστών έγινε αποκλειστικά και μόνο μέσω μέσων κοινωνικής δικτύωσης (Social Media) και δεν αξιοποιήθηκαν πλατφόρμες όπως το Amazon MTurk όπως γίνεται σε αντίστοιχες μελέτες[18].

7.2 Μέθοδος συλλογής αποτελεσμάτων

7.2.1 Περιγραφή Παιχνιδιού - Σελίδας Αξιολόγησης

Λόγου του μεγάλου αριθμού των διαφορετικών μοντέλων που κατασκευάστηκαν η αξιολόγηση όλων των κομματιών από χρήστες καθίσταται αρκετά δύσκολή και χρονοβόρα. Για την επίλυση του παραπάνω προβλήματος κατασκευάστηκε μια online σελίδα για την εκτίμηση ενός μικρού συνόλου κομματιών κάθε φορά, με την μορφή ενός παιχνιδιού.

Στην σελίδα που κατασκευάστηκε αρχικά εξηγείται στην χρήστη ο σκοπός του παιχνιδιού και στην συνέχεια η βασική λειτουργικότητά της. Όπως έχει αναφερθεί και παραπάνω τα κομμάτια που παράγονται είναι σε ακολουθιακή μορφή (πρωτόκολλο Midi) και συνεπώς μετατροπή τους σε audio εξαρτάται πολύ από εξωγενείς παράγοντες όπως η χροιά του οργάνου που χρησιμοποιήθηκε, τα προγράμματα αναπαραγωγής που υποστηρίζουν οι browser κ.α. Αυτό σημαίνει ότι το ίδιο κομμάτι ακούγεται καλύτερα αν αυτό αναπαραχθεί μέσω ενός διαφορετικού προγράμματος, εξειδικευμένου στην ανακατασκευή κομματιών Midi, από ότι ακούγεται στην σελίδα αξιολόγησης, μέσω του browser. Για αυτό πριν ο χρήστης αρχίσει να αξιολογεί τα κομμάτια περνάει από μια μικρή εκπαίδευση του πως θα ακούγονταν διάσημα κομμάτια (όπως το έργο του Beethoven Fur Elise) μέσω της σελίδας αυτής. Αυτό γίνεται με σκοπό οι χρήστες να απαντούν στα ερωτήματα χωρίς να επηρεάζονται από παράγοντες που δεν εξαρτώνται ή παράγονται από τα συστήματα σύνθεσης. Στο τέλος της εκπαίδευσης αυτής ο χρήστης καλείται να δηλώσει αν έχει γνώσεις μουσικής ή όχι όπου και εκεί γίνεται ένας διαχωρισμός τους με σκοπό την καλύτερη εκτίμηση των αποτελεσμάτων.

Στην συνέχεια οδηγείται στην κεντρική σελίδα όπου κάθε φορά αναπαράγεται μια διαφορετική μελωδία και αφού ο χρήστης την ακούσει υποχρεωτικά ένα ποσοστό καλείται να απαντήσει στις εξής ερωτήσεις:

- Ποιος πιστεύει ότι σύνθεσε αυτή την μελωδία όπου καλείται να επιλέξει μεταξύ ανθρώπου και υπολογιστή.
- Πόσο του αρέσει από το 1 έως το 5
- Καθώς και πόσο ενδιαφέρουσα την βρήκε από το 1 μέχρι το 5.

Όταν απαντηθούν τα παραπάνω ερωτήματα τότε αυτές οι απαντήσεις που έδωσε αποθηκεύονται στην βάση δεδομένων της σελίδας και στην συνέχεια απαντάται στον χρήστη αν ή πρόβλεψή του, για το ποιος είναι ο συνθέτης του κομματιού, είναι σωστή η λάθος, εμφανίζοντάς του κατάλληλο μήνυμα. Η διαδικασία αυτή συνεχίζεται για συνολικά 20 κομμάτια και στο τέλος της διαδικασίας εμφανίζεται το σκορ του, δηλαδή πόσα από τα 20 κομμάτια προέβλεψε σωστά.

Αξίζει να σημειωθεί ότι ο χρόνος που πρέπει να μεσολαβήσει για να μπορέσεις ο χρήστης να δώσεις της απαντήσεις του στην σελίδα είναι απαραίτητος για την εξασφάλιση ότι έχει ακουστεί ένα σημαντικό μέρος της μουσικής πληροφορίας κάθε κομματιού αλλά συγχρόνως και για την μείωση της επίδρασης πιθανών αυτόματων αξιολογήσεων.

7.2.2 Μέθοδος επιλογής κομματιών προς αξιολόγηση

Όπως αναφέρθηκε και παραπάνω στο σύνολο των κομματιών προς αξιολόγηση υπάρχουν πολλά περισσότερα κομμάτια κατασκευασμένα από υπολογιστή από ότι πραγματικά. Συνεπώς αν τα κομμάτια επιλεγόντουσαν τυχαία, δηλαδή με την μέθοδο της απλής τυχαίας δειγματοληψίας, τότε το μεγαλύτερο ποσοστό των κομματιών σε κάθε test θα ήταν κατασκευασμένα από υπολογιστή (π.χ. από τις 20 μόνο μια ή δυο πραγματικές μελωδίες). Αυτή η παρατήρηση μπορεί να οδηγούσε τους χρήστες (μιας και βλέπουν το πραγματικό αποτέλεσμα κάθε φορά) να ψηφίζουν ότι όλα τα κομμάτια είναι κατασκευασμένα από υπολογιστή (για να αυξήσουν το σκορ τους) και συνεπώς τα τελικά αποτελέσματα να μην ανταποκρίνονται στην πραγματικότητα. Σε ανάλογα προβλήματα, δηλαδή σε προβλήματα όπου υπάρχει ανομοιογένεια στα δείγματα ενός πληθυσμού, προτείνεται η **κατά στρώματα τυχαία δειγματοληψία (cluster sampling)**. Στη μέθοδο αυτή ο πληθυσμός των δειγμάτων χωρίζεται σε στρώματα με κοινά χαρακτηριστικά, στην περίπτωση που μελετάται σε πραγματικά και κατασκευασμένα από κάποιο μοντέλο κομμάτια και στην συνέχεια επιλέγονται τυχαία δείγματα από κάθε ένα στρώμα με βάση κάποιο ποσοστό εμφάνισης των στρωμάτων στο τελικό δείγμα. Έτσι κάθε φορά αρχικά επιλέγετε τυχαία ο αριθμός n των πραγματικών μελωδιών που θα υπάρχουν στο σύνολο αξιολόγησης (ποσοστό εμφάνισης στρωμάτων) και στην συνέχεια επιλέγονται επίσης τυχαία n πραγματικά και 20 – n κατασκευασμένα από υπολογιστή κομμάτια. Αξίζει να σημειωθεί ότι σε κάθε ένα τέτοιο test όλα τα «ψεύτικά» κομμάτια έχουν συντεθεί αυστηρά από μια αρχιτεκτονική. Αυτό γίνεται με σκοπό να μπορέσει να εξαχθεί για κάθε ένα δίκτυο το ποσοστό με το οποίο οι χρήστες μπορούν να ξεχωρίσουν με επιτυχία ένα πραγματικό από ένα ψεύτικο κομμάτι. Με αυτό τελικά θα μπορέσει να γίνει μέτρηση της επίδοσης κάθε ενός μοντέλου καθώς και αξιολόγηση της επίδρασης κάθε μιας από τις υπερπαραμέτρους του.

Αξίζει να σημειωθεί ότι το ποσοστό των πραγματικών ως προς το συνολικό αριθμό κομματιών σε κάθε test επιλέγεται τυχαία έτσι ώστε ο χρήστης να μην μπορεί να εξάγει κανένα συμπέρασμα σχετικά με την διάταξη του συνόλου αξιολόγησης. Για παράδειγμα να το ποσοστό εμφάνισης των πραγματικών κομματιών είναι 50% τότε έστω και στο τελευταίο κομμάτι θα μπορούσε να προβλεφθεί η σωστή απάντηση χωρίς καμία γνώση του τραγουδιού (μιας και ο χρηστής γνωρίζει τελικά τον συνθέτη κάθε κομματιού). Η τεχνική που χρησιμοποιήθηκε όμως δεν αφήνει να υπάρχει καμία τέτοια αξιολόγηση.

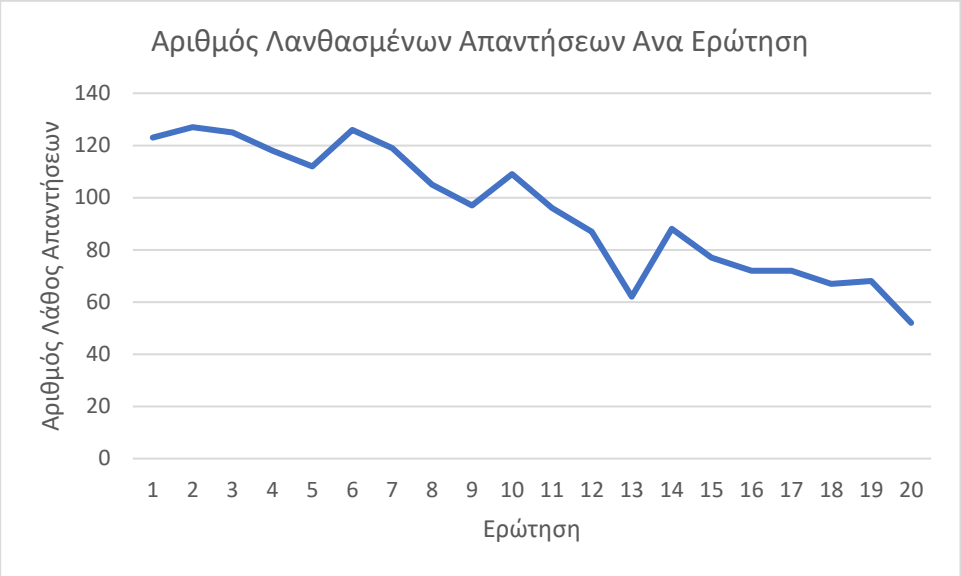
Το link της σελίδας αυτής είναι: http://geofila.pythonanywhere.com/vote.

7.2.3 Αρνητικά της Μεθόδου Αξιολόγησης

Παρόλα τα θετικά του τρόπου με τον οποίον εξετάζονται κομμάτια η συγκεκριμένη σελίδα, λόγω της παιχνιδιοποιήσης, μπορεί να φέρει μερικώς αλλοιωμένα αποτελέσματα. Αυτό οφείλεται στο γεγονός ότι ο χρήστης εκπαιδεύεται κατά την διάρκεια της διαδικασίας, μιας και λαμβάνει ανάδραση για το αν απάντησε σωστά ή όχι.

Το παραπάνω συμπέρασμα εξάχθηκε αρχικά από σχόλια χρηστών οι οποίοι δηλώσαν για παράδειγμα ότι «τα πραγματικά κομμάτια έχουν παραπάνω συναίσθημα ή παραπάνω χρώμα» κ.α. Δηλαδή οι χρήστες είναι σε θέση να εξάγουν συμπεράσματα για την δομή και την φύση των κομματιών κατά την διαδικασία αξιολόγησης των τους.

Η παραπάνω άποψη επιβεβαιώνεται και από το παρακάτω διάγραμμα όπου φαίνονται οι λάθος απαντήσεις που έχουν δώσει οι χρήστες που ολοκλήρωσαν το test ανά ερώτηση.



Από το παραπάνω διάγραμμα φαίνεται ότι πολλοί περισσότεροι χρήστες έδωσαν λάθος απαντήσεις στις πρώτες ερωτήσεις από ότι στις τελευταίες. Συνεπώς λόγω αυτού το αναμένεται ότι στο σύνολο των tests οι χρήστες θα βρίσκουν πιο εύκολα τον σωστό συνθέτη, από αν δεν λαμβάναν την εν λόγω ανάδραση.

7.4 Σελίδα Αυτόματης σύνθεσης κομματιών

Πέρα από την αξιολόγηση των κομματιών στην πλατφόρμα που κατασκευάστηκε υπάρχει και μια σελίδα για την αυτόματη σύνθεση κομματιών. Σε αυτήν με 4 απλά βήματά ο χρήστης μπορεί να κατασκευάσει το μοντέλο που επιθυμεί και στην συνέχεια να παράγει μια ακολουθία Midi την οποία στην συνέχεια μπορεί να την κατεβάσει, να την επεξεργαστεί και να την μελετήσει. Τα απαιτούμενα βήματα είναι τα παρακάτω:

1. Επιλογή αρχιτεκτονικής, μεταξύ των FFLSTM, του Autoencoder και του Autoencoder με συγκέντρωση
2. Επιλογή του με μεγέθους της μνήμης του δικτύου μεταξύ 256 και 512.
3. Επιλογή του αρχικού συνόλου εκπαίδευσης του δικτύου με επιλογές μεταξύ πιάνο, κιθάρας και κοινών δεδομένων
4. Επιλογή αρχικής μελωδίας με επιλογές μεταξύ πιάνο και κιθάρας.

Στην περίπτωση που αναλύεται παραπάνω τα παραγόμενα κομμάτια έχουν ως αρχικές μελωδίες τυχαίες ακολουθίες από το σύνολο πρόβλεψης και συνεπώς ο χρήστης δεν μπορεί να επέμβει στην παραγωγή παρά μονό να μελετήσει το αποτέλεσμα. Παρόλα αυτά την σελίδα υπάρχει και η επιλογή *Advanced Settings* όπου δίνει τον δυνατότητα στον χρήστη να επέμβει περισσότερο στην σύνθεση. Σε αυτές τις επιλογές ο χρήστης μπορεί να ανεβάσει την δική του αρχική μελωδία αλλά συγχρόνως να αλλάξει και άλλες ρυθμίσεις όπως το αν υπάρχει

τυχαιοποίηση της εξόδου, την παράμετρο temperature, το μέγεθός της ακολουθίας εξόδου κ.α.

Η παραπάνω σελίδα κατασκευάστηκε με σκοπό οι συνθέτες να παίρνουν έμπνευση από τις μελωδίες που παράγονται από τα μοντέλα που κατασκευάστηκαν στα πλαίσια της διπλωματικής αυτής. Για τον λόγο αυτό στην σελίδα υπάρχουν εκτενείς αναφορές στο πως κάθε ρύθμιση αλλάζει το αποτέλεσμα με σκοπό να μην απαιτούνται εξειδικευμένες γνώσεις από τους χρήστες για την χρησιμοποίησή της. Το link της σελίδας αυτής είναι: http://geofila.pythonanywhere.com/

Επίσης η σελίδα μπορεί να χρησιμοποιηθεί και για την συνέχεια της αξιολόγησης των κομματιών από άλλους έμπειρους χρήστες. Με λίγες δοκιμές μπορεί ευκολά κάποιος να καταλάβει πρακτικά τις διαφορές των αρχιτεκτονικών καθώς και τα προτερήματα και τα μειονεκτήματα κάθε επιλογής.

7.3 Σύνολο Αξιολογήσεων

Συνολικά η διαδικασία αξιολόγησης διήρκησε 100 ημέρες από 12 Μαρτίου μέχρι και 20 Ιουνίου και συνολικά συλλέχτηκαν αξιολογήσεις από 1152 άτομα εκ των όποιων οι 583 ήταν απλοί χρήστες ενώ οι 569 δήλωσαν ότι είχαν μουσικές γνώσεις. Για όλα τα κομμάτια (και πραγματικά και κατασκευασμένα από υπολογιστή) έχουν συλλεχθεί 2613 αξιολογήσεις από μουσικούς και 2734 αξιολογήσεις από απλούς χρήστες. Εξ αυτών για τις ψήφους από τους απλούς χρήστες οι 1519 αφορούν τα πραγματικά κομμάτια ενώ οι 1094 τα κατασκευασμένα από υπολογιστή. Αντίστοιχα από τις συνολικές αξιολογήσεις των μουσικών οι 1717 αφορούν κομμάτια που έχουν συντεθεί από υπολογιστή ενώ οι 1050 αφορούν πραγματικά κομμάτια.

Η διαφορά αυτή μεταξύ του αριθμού των αξιολογήσεων των πραγματικών κομματιών και των κατασκευασμένων από υπολογιστή έγκειται από την διαφορά που στο πλήθος τους μιας και τα πραγματικά κομμάτια στο σύνολο των κομματιών προς αξιολόγηση είναι πολύ λιγότερα. Παρόλα αυτά λόγω της μεθόδου επιλογής των κομματιών προς αξιολόγηση που χρησιμοποιήθηκε τελικά η διαφορά αυτή είναι πολύ μικρότερη από την διαφορά που θα υπήρχε αν απλά τα κομμάτια επιλεγόντουσαν τυχαία.

Κεφάλαιο 8 – Επίδραση Αρχιτεκτονικής

Στα πλαίσια αυτής της διπλωματικής εργασίας, όπως έχει αναφερθεί και παραπάνω, ερευνώνται οι παρακάτω αρχιτεκτονικές για την αυτόματη σύνθεση μελωδιών:

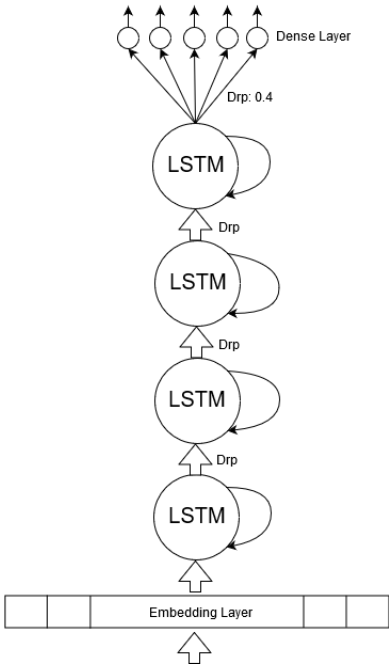
- Αρχιτεκτονική απλού αναδρομικού δικτύου (RNN)
- Αρχιτεκτονική Κωδικοποιητή- Αποκωδικοποιητή (Encoder-Decoder ή Autoencoder)
- Αρχιτεκτονική Κωδικοποιητή- Αποκωδικοποιητή με χρήση Στρώματος Συγκέντρωσης (Attention Layer)

Στα παραπάνω αλλάζει ο αριθμός του cells size καθώς και ο αλγόριθμος πρόβλεψης με σκοπό με μελετηθεί η επίδραση τους και ο ρόλος τους. Επίσης οι παραπάνω αρχιτεκτονικές εκπαιδεύονται με datasets από διαφορετικά μουσικά όργανα, παράγοντας τελικά έναν αριθμό διαφορετικών μοντέλων τα οποία τελικά αξιολογούνται.

8.1 Αρχιτεκτονική Αναδρομικού Δικτύου

Σε πρώτη φάση για την αυτόματη σύνθεση μουσικής χρησιμοποιήθηκε ένα απλό αναδρομικό δίκτυο Μακράς Βραχυπρόθεσμης Μνήμης.

Συγκεκριμένα η δομή του μοντέλου που κατασκευάστηκε παρουσιάζεται στο παρακάτω σχήμα:



Κάθε βήμα της ακολουθίας εισόδου περνά από έναν Embedding Matrix όπου παράγει ένα διάνυσμα δεκαδικών τιμών το όποιο αποτελεί την είσοδο σε ένα αναδρομικό δίκτυο 4 επιπέδων. Στην συνέχεια το αποτέλεσμα από το τελευταίο LSTM περνά από ένα Fully Connected Layer με συνάρτηση ενεργοποίησης την softmax (με ή χωρίς τυχαιότητα), έτσι ώστε η έξοδος κάθε βήματος να είναι μια κατανομή πιθανοτήτων, με μέγεθος ίσο με τις πιθανές διαφορετικές τιμές της εξόδου. Αυτό σημαίνει ότι διάνυσμα εξόδου του δικτύου κάθε φορά εκφράζει την πιθανότητα κάθε μια νότας να είναι η επόμενη. Αξίζει να σημειωθεί ότι ανάμεσα από κάθε επίπεδο της παραπάνω αρχιτεκτονικής υπάρχει ένα επίπεδο Dropout ώστε να αποφευχθεί το Overfitting, όπως εξηγείται αναλυτικότερα παραπάνω.

Το δίκτυο αυτό εκπαιδεύτηκε με σκοπό να προβλέπει μια νότα κάθε φορά. Πρακτικά σε κάθε βήμα του δίνεται μια ακολουθία εισόδου (μεγέθους n_{in}) και εξετάζεται η έξοδος του δικτύου μόνο για την επόμενη νότα ($n_{out} = 1$), δηλαδή το αν προέβλεψε σωστά την νότα n_{in} +

1, ενώ όλες οι ενδιάμεσες έξοδοι του απορρίπτονται. Μαθηματικά η ποσότητα αυτή εκφράζεται από την παρακάτω σχέση:

$$P\left(note_{n_{in}+1} \middle| note_1, note_2, \dots, note_{n_{in}}\right)$$

Έτσι το δίκτυο είναι εκπαιδευμένο στο να υπολογίζει την πιθανότητα της νότας n_{in} + 1 με δεδομένο τις προηγούμενες n_{in} νότες. Κατά την πρόβλεψη ακολουθείται η ίδια ακριβώς διαδικασία με την μόνη διαφορά ότι η νότα που προβλέπεται κάθε φορά ενώνεται με τις προηγούμενες και πλέον το δίκτυο καλείται να προβλέψει την νότα $note_{n_{in}+2}$ με είσοδο την ακολουθία $note_2, note_3, \dots, note_{n_{in}+1}$ (το μέγεθος της ακολουθίας εισόδου παραμένει πάντα σταθερό).

Το δίκτυο αυτό εκπαιδεύτηκε με 2 διαφορετικές τιμές μεγέθους κελιού (cell_size), 256 και 512. Αξίζει να σημειωθεί ότι δοκιμάστηκαν και άλλες τιμές αλλά τα αποτελέσματα των παραπάνω είναι τα πλέον αντιπροσωπευτικά μιας και για τιμές μικρότερες του 256 το δίκτυο δεν μπορούσε να μάθει κάτι αξιόλογο ενώ για τιμές μεγαλύτερες του 512 η εκπαίδευση αργούσε πολύ και τα αποτελέσματα δεν διέφεραν αισθητά (το σφάλμα του δικτύου διέφερε ελάχιστα).

Επίσης δοκιμάστηκαν και διαφορετικές τιμές της παραμέτρους n_{in}. Αξίζει να σημειωθεί ότι για την παραπάνω αρχιτεκτονική το μέγεθος της ακολουθίας εισόδου δεν αλλάζει τον αριθμό των παραμέτρων ή την δομή του μοντέλου. Το μόνο που αλλάζει είναι ο χρόνος (και η ποιότητα) εκπαίδευσης μιας και το ξεδιπλωμένο δίκτυο είναι πολύ πιο βαθύ με αποτέλεσμα να χρειάζεται περισσότερο χρόνο ο αλγόριθμος εκπαίδευσης.

8.1.1 Μέγεθος κελίου 256

Οι υπεπαράμετροι του δικτύου αυτού εμφανίζονται οι παρακάτω:

- Embedding Size = 50
- Lstm_size = 256
- Drp = 0.4
- Το Fully Connected Layer έχει μόνο ένα επίπεδο και η έξοδος του είναι όσες και οι διαφορετικές πιθανές τιμές εξόδου (οι οποίες εξαρτώνται από το σύνολο εκπαίδευσης).

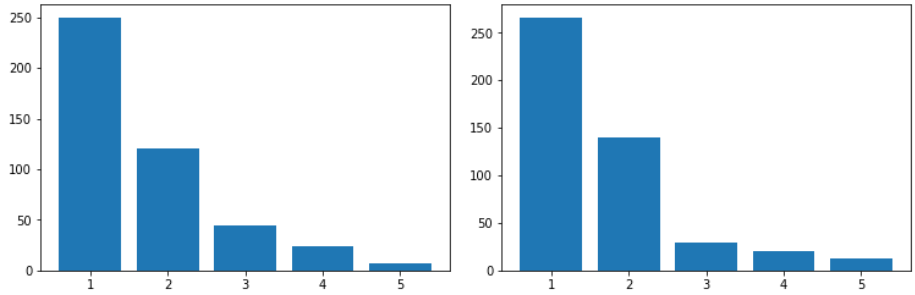
Συνολικά από αυτήν την αρχιτεκτονική κατασκευάστηκαν 4 διαφορετικά μοντέλα (ένα για κάθε dataset εκτός από τα κοινά).

Αξιολόγηση Αρχιτεκτονικής

Αρχικά θα γίνει μια πρώτη αξιολόγηση της αρχιτεκτονικής συνολικά ανεξάρτητα του συνόλου εκπαίδευσης και των υπολοίπων παραμέτρων.

Για την αρχιτεκτονική αυτή συλλέχθηκαν 911 αξιολογήσεις συνολικά εκ των οποίων οι 462 είναι από χρήστες χωρίς μουσικές γνώσεις ενώ οι υπόλοιπες 449 από χρηστές οι οποίοι δήλωσαν ότι έχουν μουσικές γνώσεις. Από τις αξιολογήσεις αυτές φάνηκε ότι όλοι οι χρήστες μπορούσαν με ευκολία να ξεχωρίσουν τον πραγματικό συνθέτη των κομματιών μιας και είχαν ποσοστό επιτυχών προβλέψεων **92.64%** και **93.38%** για κάθε μια κατηγορία χρηστών αντίστοιχα.

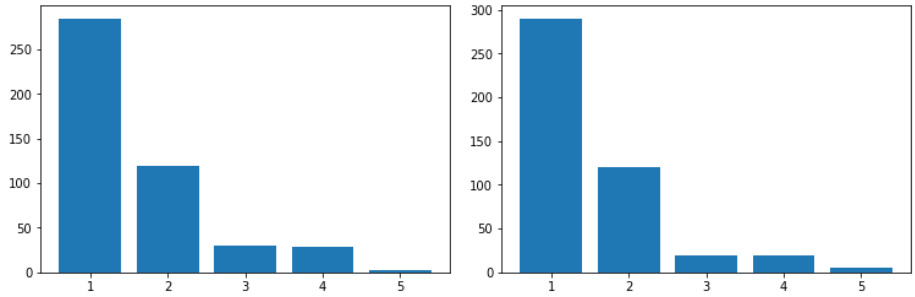
Παρακάτω παρουσιάζονται τα διαγράμματα που παράχθηκαν από τις παραπάνω αξιολογήσεις για την αρέσκεια και το ενδιαφέρον των παραγόμενων κομματιών.



(α) (β)

Από τα παραπάνω φαίνεται ότι τα παραγόμενα κομμάτια δεν άρεσαν ιδιαίτερα στους χρήστες μιας και ο μέσος όρος αρεσκείας τους ήταν **1.51** στα 5 ενώ του ενδιαφέροντος τους ήταν **1.56** στα 5.

Παρακάτω παρουσιάζονται τα αντίστοιχα διαγράμματα για τους χρήστες με μουσικές γνώσεις.



(α) (β)

Από τα παραπάνω φαίνεται ότι τα παραγόμενα κομμάτια δεν άρεσαν ιδιαίτερα στους χρήστες μιας και ο μέσος όρος αρεσκείας τους ήταν **1.64** στα 5 ενώ του ενδιαφέροντος τους ήταν **1.7** στα 5.

8.1.2 Μέγεθος κελίου 512

Στο μοντέλο οι υπεπαράμετροι του δικτύου αυτού εμφανίζονται οι παρακάτω:

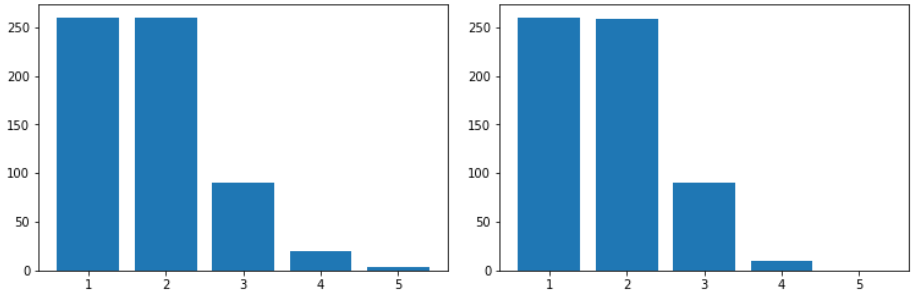
- Embedding Size = 50
- Lstm_size = 512
- Drp = 0.4
- Το Fully Connected Layer έχει μόνο ένα επίπεδο και η έξοδος του είναι όσες και οι διαφορετικές πιθανές τιμές εξόδου (οι οποίες εξαρτώνται από το σύνολο εκπαίδευσης).

Αξιολόγηση Αρχιτεκτονικής

Όπως και προηγουμένως θα γίνει μια πρώτη αξιολόγηση της αρχιτεκτονικής συνολικά ανεξάρτητα του συνόλου εκπαίδευσης και των υπολοίπων παραμέτρων.

Για την αρχιτεκτονική αυτή συλλέχθηκαν 1340 αξιολογήσεις συνολικά εκ των οποίων οι 695 είναι από χρήστες χωρίς μουσικές γνώσεις ενώ οι υπόλοιπες 645 από χρηστές οι οποίοι δήλωσαν ότι έχουν μουσικές γνώσεις. Από τις αξιολογήσεις αυτές φάνηκε ότι οι χρήστες μπορούσαν με ευκολία να ξεχωρίσουν τον πραγματικό συνθέτη των κομματιών μιας και είχαν ποσοστό επιτυχών προβλέψεων **75.64%** και **84.07%** για κάθε μια κατηγορία χρηστών αντίστοιχα.

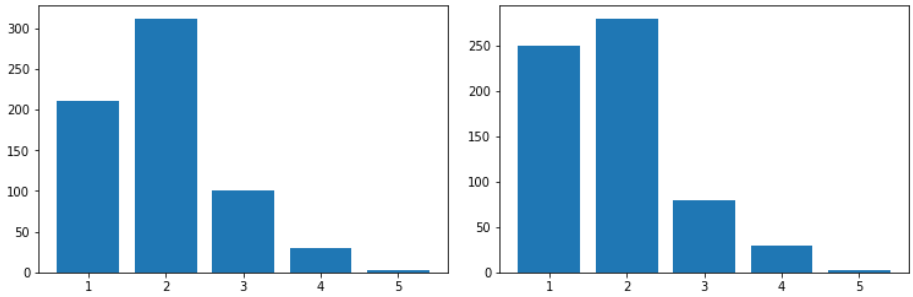
Παρακάτω παρουσιάζονται τα διαγράμματα που παράχθηκαν από τις παραπάνω αξιολογήσεις για την αρέσκεια και το ενδιαφέρον των παραγόμενων κομματιών.



(α) (β)

Από τα παραπάνω φαίνεται ότι τα παραγόμενα κομμάτια δεν άρεσαν ιδιαίτερα στους χρήστες μιας και ο μέσος όρος αρεσκείας τους ήταν **1.94** στα 5 ενώ του ενδιαφέροντος τους ήταν **1.96** στα 5.

Παρακάτω παρουσιάζονται τα αντίστοιχα διαγράμματα για τους χρήστες με μουσικές γνώσεις.



(α) (β)

Από τα παραπάνω φαίνεται ότι τα παραγόμενα κομμάτια δεν άρεσαν ιδιαίτερα στους χρήστες μιας και ο μέσος όρος αρεσκείας τους ήταν **2.07** στα 5 ενώ του ενδιαφέροντος τους ήταν **2.1** στα 5.

8.1.3 Συνολική αξιολόγηση Αρχιτεκτονικής

Τελικά τα αποτελέσματά που παρήγαγε η αρχιτεκτονική αυτή ήταν πολύ φτωχά μιας και ελάχιστες φορές κατάφερνε να συνεχίσει ένα κομμάτι ενώ τις περισσότερες φορές απλώς αναδίπλωνε την έξοδό του. Η μέγιστη ακολουθία (χωρίς αναδίπλωση) που μπόρεσε να παράγει περιείχε λιγότερες από 100 νότες και αυτό συνέβη μόνο ελάχιστες φορές. Επίσης η προσθήκη τυχαιότητας στην επιλογή της εξόδου, όπως αυτή περιγράφεται παραπάνω δεν έφερε κανένα αποτέλεσμα μιας και οι παραγόμενες νότες φαινόταν σαν να ήταν τελείως τυχαίες (η αιτία καθώς και περισσότερες πληροφορίες για τον παραπάνω φαινόμενο εξηγείται στο Κεφ. 9).

Τέλος ο χρόνος για την παραγωγή των αποτελεσμάτων ήταν αρκετά μεγάλος καθώς για την παραγωγή 100 νοτών απαιτείται κατά μέσω όρο 8 sec σε έναν συμβατικό υπολογιστή. Αυτό σημαίνει ότι μια τέτοια αρχιτεκτονική (ακόμα και αν είχε αποφευχθεί η αναδίπλωση της εξόδου) δεν ενδείκνυται για διάφορες εφαρμογές της αυτόματης σύνθεσης όπως μια εφαρμογή συνεχής παραγωγής (π.χ. ένα συνεχές ράδιο) μιας και η παραγόμενη μελωδία θα παιζόταν πιο γρήγορα από τον χρόνο που θα έπαιρνε στο σύστημα για να κατασκευάσει την συνέχειά της, δημιουργώντας συνεχώς διακοπές στην ροή.

Παρόλα αυτά η συγκεκριμένη αρχιτεκτονική είναι πολύ εύκολη στην κατανόηση και ο χρόνος κατασκευής και αποσφαλμάτωσης είναι ιδιαίτερα μικρός.

8.2 Αρχιτεκτονική Κωδικοποιητή- Αποκωδικοποιητή

Η αρχιτεκτονική Κωδικοποιητή- Αποκωδικοποιητή είναι ένας τύπος αναδρομικού δικτύου η όποια αποτελείται από τον:

- Κωδικοποιητή (Encoder): ο οποίος είναι υπεύθυνος για την κωδικοποίηση της εισόδου σε ένα διάνυσμα σταθερού μεγέθους (context vector).
- Αποκωδικοποιητή (Decoder): ο οποίος είναι υπεύθυνος για την δημιουργία της εξόδου με βάση την κωδικοποιημένη αναπαράσταση της εισόδου, που του παρέχεται από τον encoder.

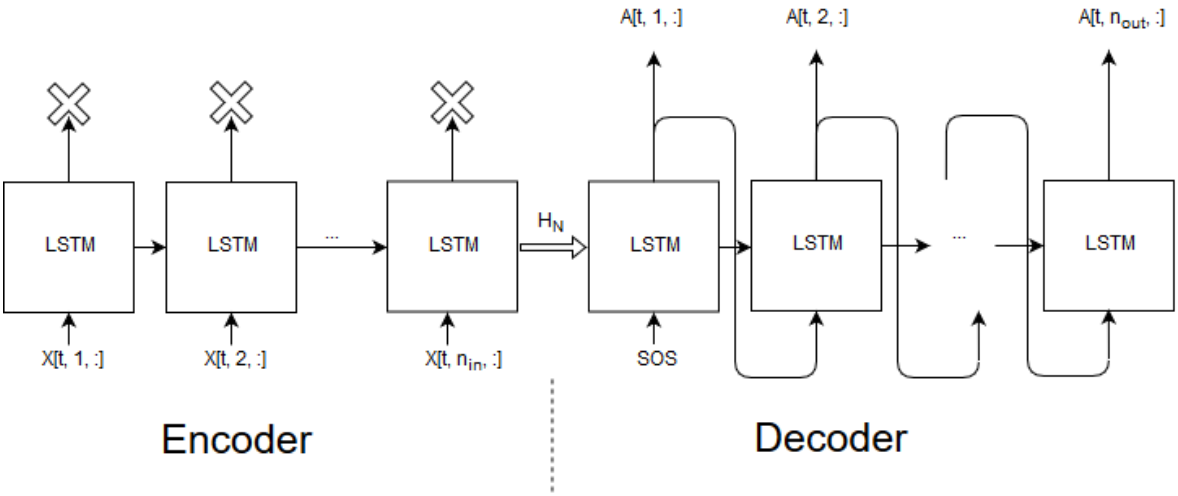
Δηλαδή η αρχιτεκτονική αυτή επιδιώκει να μάθει μια συμπιεσμένη αναπαράσταση την εισόδου και στην συνέχεια με βάση αυτήν να κατασκευάσει κάποια έξοδο.

Πρακτικά ο encoder και ο decoder αποτελούνται από ένα Feed Forward LSTM ο καθένας. Η είσοδος κάθε ενός Feed Forward LSTM περνά από ένα Embedding Layer, όπως και στην προηγούμενη αρχιτεκτονική, με σκοπό να γίνει μια αρχική κωδικοποίησή της. Για την παραγωγή της τελικής εξόδου, στην έξοδο του decoder τοποθετείται ένα πλήρως συνδεδεμένο δίκτυο ενώ η έξοδος του encoder απορρίπτεται.

Αξίζει να σημειωθεί ότι κατά την διάρκεια της εκπαίδευσης και της πρόβλεψης η λειτουργία και η οργάνωση της αρχιτεκτονικής διαφέρει. Στην συνέχεια παρουσιάζονται οι αρχιτεκτονικές κατά τις 2 παραπάνω φάσεις.

LSTM Encoder-Decoder κατά την εκπαίδευση

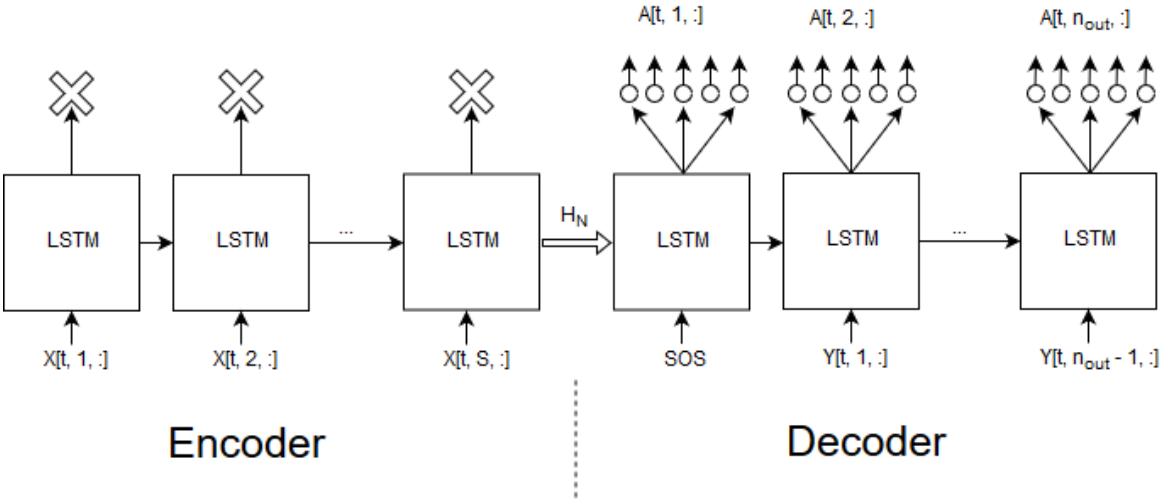
Η διαφορά της εκπαίδευσης από την πρόβλεψη πηγάζει από το γεγονός ότι κατά την εκπαίδευση είναι γνωστή η επιθυμητή έξοδος. Έστω X[t, n_{in}, N] η είσοδος στον encoder (η οποία πρακτικά αποτελεί την αρχική μελωδία και την όποια καλείται να συνεχίσει) και A[t, n_{in}, N] η έξοδος του decoder (η οποία είναι η συνέχεια που παρήγαγε το δίκτυο για την αρχική μελωδία X), όπου t είναι η θέση ενός τυχαίου δείγματος από το σύνολο δοκιμής, n_{in} και n_{out} το μέγεθος της ακολουθίας εισόδου και εξόδου αντίστοιχα και N η διαστατικότητά τους. Παρακάτω παρουσιάζεται η αρχιτεκτονική κατά την διαδικασία πρόβλεψης.



Όπως αναφέρθηκε και προηγουμένως η ακολουθία εισόδου αρχικά περνά από τον encoder όπου κωδικοποιείται. Στην συνέχεια αυτή η κωδικοποίηση περνάει στην εσωτερική κατάσταση του decoder ενώ ως είσοδος του δίνεται ένας χαρακτήρας εκκίνησης (Start Of Sequence). Συνεπώς με την ήδη υπάρχουσα γνώση που υπάρχει στην εσωτερική κατάσταση του και με τον χαρακτήρα εκκίνησης ο αποκωδικοποιητής παράγει έναν χαρακτήρα εξόδου. Στο επόμενο βήμα σαν είσοδο στον decoder δίνεται ο χαρακτήρας που προβλέφθηκε προηγουμένως με σκοπό την παραγωγή του επόμενου χαρακτήρα της ακολουθίας εξόδου. Η διαδικασία αυτή συνεχίζεται μέχρις ότου είτε να τελειώσει το μέγεθος της ακολουθίας εξόδου είτε να φτάσει σε κάποιον χαρακτήρα που σηματοδοτεί την λήξη της σύνθεσης (End Of Sequence).

LSTM Encoder-Decoder κατά την εκπαίδευση- Teacher Forcing

Το κύριο χαρακτηριστικό της διαδικασίας εκπαίδευσης είναι ότι για κάθε ακολουθία εισόδου είναι γνωστή η επιθυμητή έξοδος. Πρακτικά έστω ότι η είσοδος του autoencoder σε ένα βήμα εκτέλεσης είναι μια ακολουθία εισόδου X[B, n_{in}, N] και η έξοδος Y[B, n_{out}, N] όπου B είναι το μέγεθος του batch, n_{in} και n_{out} το μέγεθος της ακολουθίας εισόδου και εξόδου αντίστοιχα και N η διαστατικότητά τους. Έτσι η αρχιτεκτονική κατά την εκπαίδευση του autoencoder παρουσιάζεται στο παρακάτω σχήμα.



Η διαδικασία παραμένει ίδια με την μόνη διαφορά πλέον ότι σε κάθε βήμα του decoder δεν δίνετε ως είσοδος το αποτέλεσμα του προηγούμενου βήματος αλλά η επιθυμητή έξοδος σαν να την είχε παράγει. . Αυτός ο τρόπος εκπαίδευσης ονομάζεται Teacher Forcing και αποτελεί έναν από τους πιο διαδεδομένους και γρήγορους τρόπους εκπαίδευσης τέτοιων αρχιτεκτονικών. Παρόλα αυτά η παραπάνω μέθοδος έχει και ορισμένα σημαντικά μειονεκτήματα η ανάλυση όμως των οποίων δεν αποτελεί θέμα προς μελέτη στα πλαίσια της παρούσας εργασίας.

8.2.1 Μέγεθος κελίου 256

Στο μοντέλο οι υπεπαράμετροι του δικτύου αυτού εμφανίζονται οι παρακάτω:

- Embedding Size = 50
- Lstm_size = 256
- Drp = 0.4
- Το Fully Connected Layer έχει μόνο ένα επίπεδο και η έξοδος του είναι όσες και οι διαφορετικές πιθανές τιμές εξόδου (οι οποίες εξαρτώνται από το σύνολο εκπαίδευσης).

Συνολικά από αυτήν την αρχιτεκτονική παράχθηκαν 4 διαφορετικά μοντέλα (ένα για κάθε dataset εκτός των κοινών).

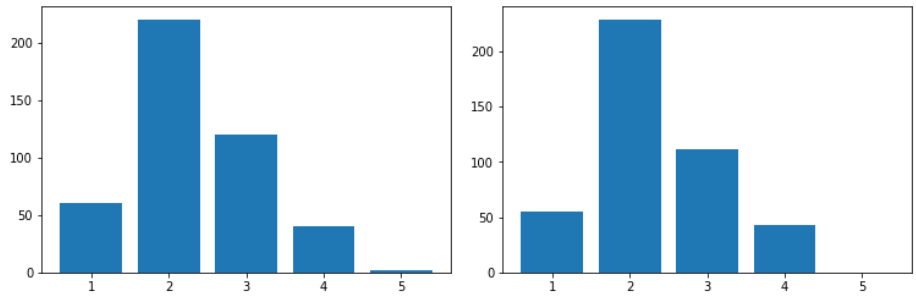
Αξιολόγηση Αρχιτεκτονικής

Αρχικά όπως και με το Απλό Αναδρομικό Δίκτυο θα γίνει μια πρώτη αξιολόγηση της αρχιτεκτονικής συνολικά ανεξάρτητα του συνόλου εκπαίδευσης.

Για την αρχιτεκτονική αυτή συλλέχθηκαν 883 αξιολογήσεις συνολικά εκ των οποίων οι 439 είναι από χρήστες χωρίς μουσικές γνώσεις ενώ οι υπόλοιπες 444 από χρηστές οι οποίοι δήλωσαν ότι έχουν μουσικές γνώσεις. Από τις αξιολογήσεις αυτές φάνηκε η ανωτερότητά της αρχιτεκτονικής αυτής σε σχέση με την προηγούμενη μιας και το ποσοστό επιτυχών

προβλέψεων για κάθε μια κατηγορία χρηστών είναι πλέον **60.78%** και **65.62%**, το όποιο έχει πλησιάσει σημαντικά στο ποσοστό της τυχαίας πρόβλεψης. Πρακτικά το παραπάνω σημαίνει ότι οι χρήστες δυσκολευόντουσαν αρκετά να εντοπίσουν τον πραγματικό συνθέτη κάθε κομματιού.

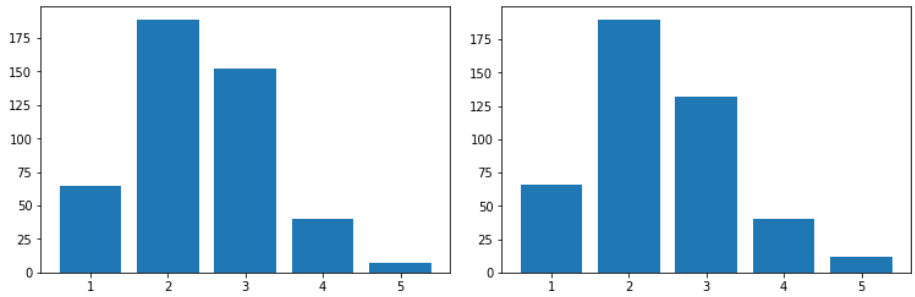
Παρακάτω παρουσιάζονται και τα διαγράμματα που παράχθηκαν από τις ίδιες αξιολογήσεις για την αρέσκεια και το ενδιαφέρον των παραγόμενων κομματιών, της αρχιτεκτονικής αυτής.



(α) (β)

Από τα παραπάνω φαίνεται ότι τα παραγόμενα κομμάτια δεν άρεσαν ιδιαίτερα στους χρήστες μιας και ο μέσος όρος αρεσκείας τους ήταν **2.62** στα 5 ενώ του ενδιαφέροντος τους ήταν **2.59** στα 5.

Παρακάτω παρουσιάζονται τα αντίστοιχα διαγράμματα για τους χρήστες με μουσικές γνώσεις.



(α) (β)

Από τα παραπάνω φαίνεται ότι τα παραγόμενα κομμάτια δεν άρεσαν ιδιαίτερα στους χρήστες μιας και ο μέσος όρος αρεσκείας τους ήταν **2.47** στα 5 ενώ του ενδιαφέροντος τους ήταν **2.46** στα 5.

8.2.1 Μέγεθος κελίου 512

Στο μοντέλο οι υπεπαράμετροι του δικτύου αυτού εμφανίζονται οι παρακάτω:

- Embedding Size = 50
- Lstm_size = 512
- Drp = 0.4
- To Fully Connected Layer έχει μόνο ένα επίπεδο και η έξοδος του είναι όσες και οι διαφορετικές πιθανές τιμές εξόδου (οι οποίες εξαρτώνται από το σύνολο εκπαίδευσης).

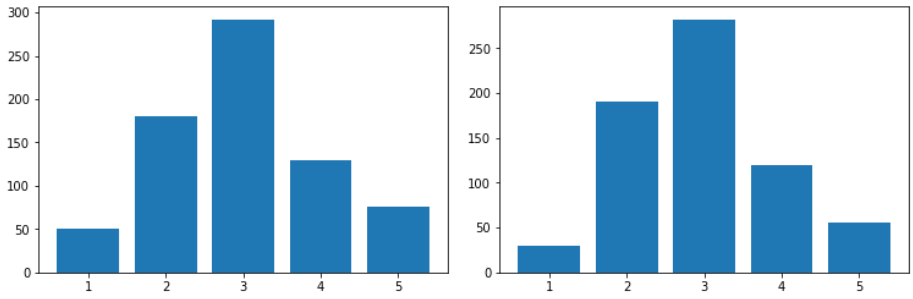
Συνολικά από αυτήν την αρχιτεκτονική παράχθηκαν 6 διαφορετικά μοντέλα (ένα για κάθε dataset).

Αξιολόγηση Αρχιτεκτονικής

Για την αρχιτεκτονική αυτή συλλέχθηκαν 1273 αξιολογήσεις συνολικά εκ των οποίων οι 656 είναι από χρήστες χωρίς μουσικές γνώσεις ενώ οι υπόλοιπες 617 από χρηστές οι οποίοι

δήλωσαν ότι έχουν μουσικές γνώσεις. Οι αξιολογήσεις αυτές έδειξαν ότι οι χρήστες δυσκολευόντουσαν ακόμα περισσότερο στο να ξεχωρίσουν τον πραγματικό συνθέτη κάθε κομματιού μιας και το ποσοστό επιτυχών προβλέψεων για κάθε μια κατηγορία χρηστών είναι πλέον **56.28%** και **59.48% (**αντίστοιχα για χρήστες χωρίς και με μουσικές γνώσεις).

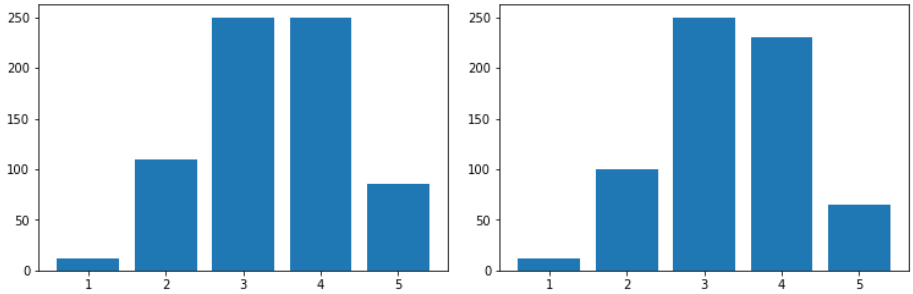
Παρακάτω παρουσιάζονται και τα διαγράμματα που παράχθηκαν από τις ίδιες αξιολογήσεις για την αρέσκεια και το ενδιαφέρον των παραγόμενων κομματιών, της αρχιτεκτονικής αυτής.



(α) (β)

Ο μέσος όρος αρεσκείας τους για τα κομμάτια της αρχιτεκτονικής αυτής ήταν **3.0** στα 5 ενώ του ενδιαφέροντος τους ήταν **3.012** στα 5.

Παρακάτω παρουσιάζονται τα αντίστοιχα διαγράμματα για τους χρήστες με μουσικές γνώσεις.



(α) (β)

Από τα παραπάνω φαίνεται ότι τα παραγόμενα κομμάτια δεν άρεσαν ιδιαίτερα στους χρήστες μιας και ο μέσος όρος αρεσκείας τους ήταν **3.05** στα 5 ενώ του ενδιαφέροντος τους ήταν **3.028** στα 5.

8.2.3 Συνολική Αξιολόγηση Αρχιτεκτονικής

Τα αποτελέσματα που παρήγαγε η αρχιτεκτονική αυτή είναι πολύ καλύτερα από ότι προηγουμένως. Αρχικά για όλες σχεδόν τις αρχικές μελωδίες το δίκτυο μπορούσε να συνεχίσει αποδοτικά παράγοντας σχεδόν πάντα πάνω από 400 νότες ενώ σπάνια έπεφτε σε αναδιπλώσεις (ιδιαίτερα για την αρχιτεκτονική με lstm_size = 512). Αξιοσημείωτο παραμένει και το γεγονός ότι το δίκτυο μπορούσε να ανακάμψει από καταστάσεις αναδιπλώσεις, δηλαδή ενώ η έξοδος για 120 νότες φαινόταν η ίδια στην συνέχεια μπορούσε να αλλάξει χωρίς την προσθήκη τυχαιότητας. Αυτό έκανε τα κομμάτια της αρχιτεκτονικής αυτής να δείχνουν ότι έχουν μια δομή κάνοντας συγχρόνως τους χρήστες να τα περνούν για πραγματικά.

Επίσης σημαντικό είναι το γεγονός ότι με την προσθήκη της τεχνικής της θερμοκρασίας τα αποτελέσματα δεν έχαναν την ποιότητα τους και το δίκτυο μπορούσε να παράγει ακόμα μεγαλύτερες ακολουθίες εξόδου αυξάνοντας συγχρόνως και το ενδιαφέρον τους.

Τέλος ο χρόνος για την παραγωγή των αποτελεσμάτων είναι πολύ μικρότερος σε σχέση με προηγουμένως και πλέον για την παραγωγή 100 νοτών απαιτείται περίπου 1 sec σε

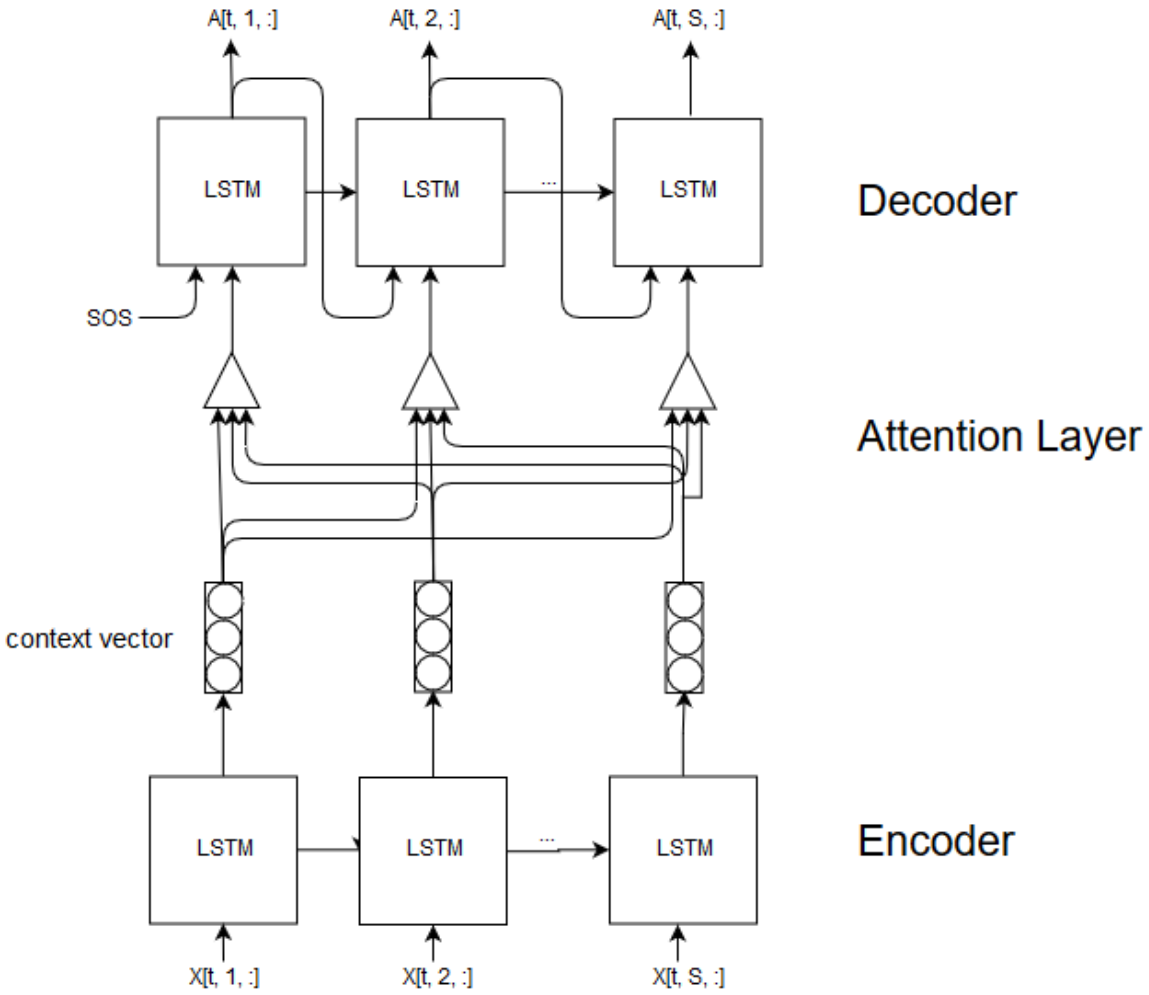
έναν συμβατικό υπολογιστή. Η μείωση αυτή καθιστά την αρχιτεκτονική πολύ πιο εποικοδομητική μιας και αυξάνεται σημαντικά το πεδίο εφαρμογών της.

8.3 Αρχιτεκτονική Κωδικοποιητή- Αποκωδικοποιητή με Συγκέντρωση

Το πρόβλημα της απλής αρχιτεκτονικής Κωδικοποιητή- Αποκωδικοποιητή είναι ότι παράγει πολύ φτωχά αποτελέσματα όταν καλείται να μεταχειριστεί μεγάλες ακολουθίες εισόδου ή εξόδου. Το πρόβλημα αυτό πηγάζει από το γεγονός ότι το context vector είναι σταθερού μεγέθους. Έτσι αν για παράδειγμα μια είσοδος είναι αρκετά μεγάλη υπάρχει η περίπτωση η εσωτερική κατάσταση του Encoder να είναι αδύνατο να χωρέσει όλη αυτή την ποσότητα πληροφορίας. Αυτό έχει ως αποτέλεσμα ο decoder είτε να μην έχει ολόκληρη την πληροφορία εισόδου είτε αυτή να είναι αλλοιωμένη.

Το παραπάνω πρόβλημα έρχεται να λύσει ο μηχανισμός της συγκέντρωσης. Η μέθοδος αρχικά παρέχει μια πιο πλούσια αναπαράσταση της εισόδου στον decoder ενώ παράλληλα του προσφέρει και έναν μηχανισμό εκμάθησης, για το που να εστιάζει την προσοχή του. Αξίζει να σημειωθεί ότι η τεχνική αυτή δεν αλλάζει την αρχική δομή του απλού Κωδικοποιητή- Αποκωδικοποιητή απλά προσθέτει ένα επίπεδο ανάμεσα τους.

Στα μοντέλα που κατασκευάστηκαν στα πλαίσια της διπλωματικής αυτής χρησιμοποιήθηκε ο μηχανισμός συγκέντρωσης του Bahdanau[16]. Όπως και στον απλό autoencoder έτσι και εδώ η οργάνωση του αλλάζει κατά την διαδικασία εκπαίδευσης και πρόβλεψης, με τον ίδιο ακριβώς τρόπο. Στο παρακάτω σχήμα φαίνεται η δομή της αρχιτεκτονικής αυτής κατά την διάρκεια της πρόβλεψης.



Όπως φαίνεται και στο παραπάνω σχήμα ανάμεσα στον κωδικοποιητή και τον αποκωδικοποιητή έχει προστεθεί ένα στρώμα attention το οποίο ουσιαστικά είναι ένα dense layer όπως αυτό παρουσιάζεται στο κεφάλαιο 5 Ουσιαστικά ο decoder για να παράγει την έξοδο της θέσης t λαμβάνει ως είσοδο την κωδικοποιημένη ακολουθία του encode h = (h_1, h_2, …, h_s), την προηγούμενη κατάσταση s_t -1 (η οποία βρίσκεται εσωτερικά στον lstm του decoder) καθώς επίσης και το προηγούμενο χαρακτήρα εξόδου y_{t-1} (κατά την διαδικασία πρόβλεψης ενώ κατά την εκπαίδευση τον επιθυμητό χαρακτήρα εισόδου, ακριβώς όπως και χωρίς το attention layer).

Σε κάθε βήμα πρόβλεψης αρχικά υπολογίζονται οι πιθανότητες $a_{j,t}$. Αυτές δηλώνουν κατά πόσο ο χαρακτήρας εισόδου j επηρεάζει την πρόβλεψη που θα κάνει ο decoder για το βήμα εξόδου t. Οι εξισώσεις που υπολογίζουν αυτήν την ποσότητα παρουσιάζονται παρακάτω, όπου οι μεταβλητές με κεφαλαίο γράμμα αποτελούν μήτρες προς εκπαίδευση.

$$e_{j,t} = V_a \tanh(W_a s_{t-1} + U_a h_j)$$

$$a_{j,t} = \frac{\exp(e_{j,t})}{\sum_{k=1}^{S} \exp(e_{k,t})}$$

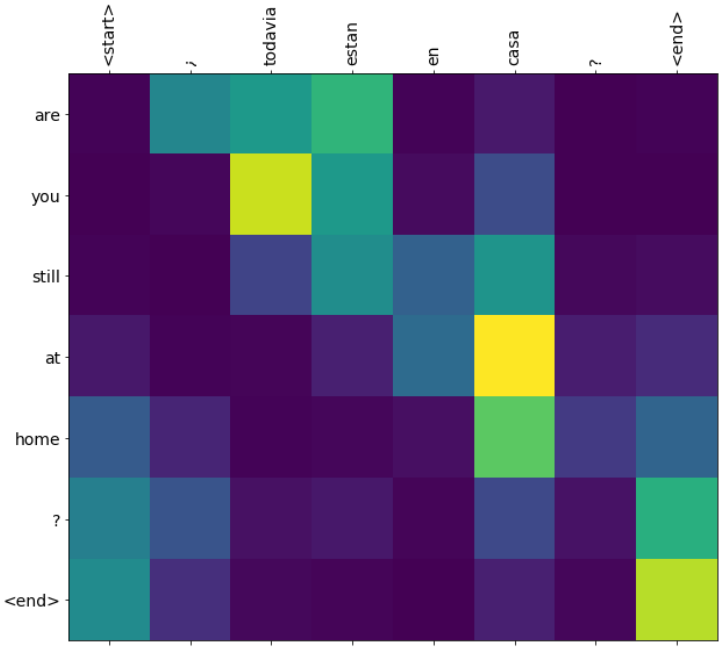
Στην πρώτη εξίσωση γίνεται ο υπολογισμός του Feed-Forward νευρωνικού δικτύου ενώ στην δεύτερη, υπολογίζεται η πιθανότητα που παράγει η συνάρτηση softmax. Στην συνέχεια με βάση τις τιμές αυτές υπολογίζεται το context vector όπως παρακάτω:

$$c_t = \sum_{k=1}^{S} a_{k,t} h_k$$

Ουσιαστικά το context vector που υπολογίζεται από το μηχανισμό της συγκέντρωσης είναι το άθροισμα του γινομένου της εξάρτησης της εξόδου της θέση t από τον χαρακτήρα εισόδου της θέσης j επί την κωδικοποιημένη αναπαράσταση της που παράχθηκε από τον encoder για την είσοδο αυτή. Με βάση τώρα αυτό το context vector γίνονται όλοι οι υπολογισμοί στον decoder (από τις εξισώσεις του lstm ακριβώς όπως και χωρίς το attention layer).

Εκτός των άλλων η πιθανότητές $a_{j,t}$ μπορούν να απεικονιστούν σε ένα δισδιάστατο διάγραμμα και έτσι μπορούν να εξαχθούν πολύ χρήσιμα συμπεράσματα τόσο της ποιότητας εκπαίδευσης όσο και του ιδίου του πεδίου εφαρμογής. Ο μηχανισμός αυτός φαίνεται καλύτερα σε ένα παράδειγμα ενός συστήματος μετάφρασης όπου γίνεται η μετατροπή της πρότασης "todavia estan en casa?" η όποια στα ελληνικά σημαίνει "Είσαι ακόμα στο σπίτι?".

Στο παράδειγμα αυτό η λέξη "σπίτι'' εξαρτάται πολύ περισσότερο από την λέξη "casa'' με την όποια έχουν ακριβώς την ίδια σημασία από την λέξη "todavia'' η οποία σημαίνει "ακόμα''. Συνεπώς η πιθανότητα $a_{4,4}$ πρέπει να είναι πολύ μεγαλύτερη από την $a_{4,1}$. Παρακάτω παρουσιάζεται ένα διάγραμμα που παρουσιάζει τις πιθανότητες που παράχθηκαν για την μετάφραση της παραπάνω πρότασης στα Αγγλικά από ένα σύστημα μετάφρασης της Google, όπου και φαίνεται το φαινόμενο που αναλύθηκε παραπάνω.



Συνεπώς αν στο παραπάνω διάγραμμα οι συσχετίσεις δεν είχαν κάποια λογική σχέση μεταξύ τους τότε ευκολά συμπεραίνεται ότι η μοντελοποιήση του προβλήματος δεν έχει γίνει σωστά και τα αποτελέσματα δεν θα ήταν κοντά στην επιθυμητά.

8.3.1 Μέγεθος Κελιού 256

Στο μοντέλο οι υπεπαράμετροι του δικτύου αυτού εμφανίζονται οι παρακάτω:

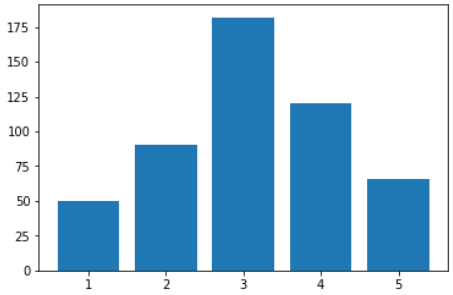
- Embedding Size = 50
- Lstm_size = 256
- Drp = 0.4
- Το Fully Connected Layer έχει μόνο ένα επίπεδο και η έξοδος του είναι όσες και οι διαφορετικές πιθανές τιμές εξόδου (οι οποίες εξαρτώνται από το σύνολο εκπαίδευσης).

Συνολικά από αυτήν την αρχιτεκτονική παράχθηκαν 4 διαφορετικά μοντέλα (ένα για κάθε dataset εκτός των κοινών δεδομένων).

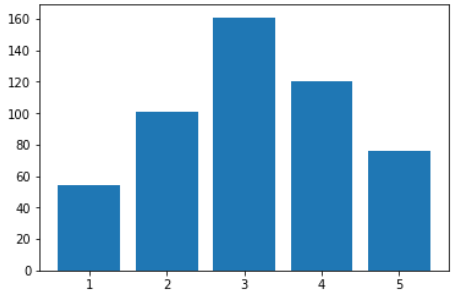
Αξιολόγηση Αρχιτεκτονικής

Για την αρχιτεκτονική αυτή συλλέχθηκαν 940 αξιολογήσεις συνολικά εκ των οποίων οι 482 είναι από χρήστες χωρίς μουσικές γνώσεις ενώ οι υπόλοιπες 458 από χρηστές οι οποίοι δήλωσαν ότι έχουν μουσικές γνώσεις. Το ποσοστό επιτυχών προβλέψεων για κάθε μια κατηγορία χρηστών για τα test με κομμάτια του μοντέλου αυτού είναι **58.66%** και **59.48%** **(**αντίστοιχα για χρήστες χωρίς και με μουσικές γνώσεις). Τα ποσοστά αυτά είναι παρόμοια με αυτά του απλού Κωδικοποιητή- Αποκωδικοποιητή με μέγεθος κελιού 512. Έτσι μπορούμε να πούμε ότι η συγκέντρωση εξισορρόπησε την επίδραση της μείωση του μεγέθους του κελίου, με τελικό αποτέλεσμα την μείωση του μεγέθους της αρχιτεκτονικής.

Παρακάτω παρουσιάζονται και τα διαγράμματα που παράχθηκαν από τις ίδιες αξιολογήσεις για την αρέσκεια και το ενδιαφέρον των παραγόμενων κομματιών, της αρχιτεκτονικής αυτής.



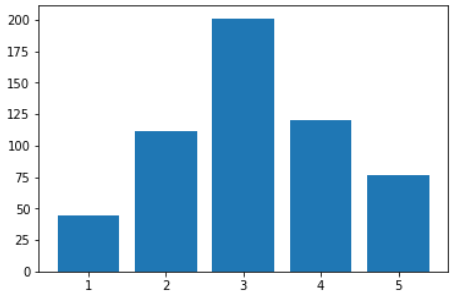
(α)



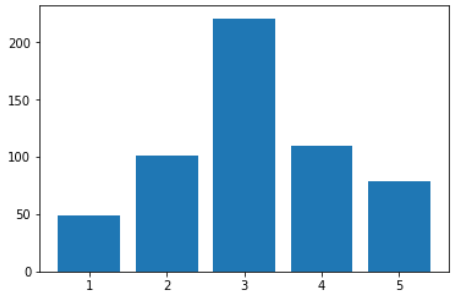
(β)

Ο μέσος όρος αρεσκείας των απλών χρηστών για τα κομμάτια της παραπάνω αρχιτεκτονικής ήταν **2.97** στα 5 ενώ του ενδιαφέροντος τους ήταν **3.37** στα 5.

Παρακάτω παρουσιάζονται τα αντίστοιχα διαγράμματα για τους χρήστες με μουσικές γνώσεις.



(α)



(β)

Ο μέσος όρος αρεσκείας των μουσικών για τα κομμάτια της παραπάνω αρχιτεκτονικής ήταν **2.89** στα 5 ενώ του ενδιαφέροντος τους ήταν **3.31** στα 5.

8.3.2 Μέγεθος Κελιού 512

Στο μοντέλο οι υπεπαράμετροι του δικτύου αυτού εμφανίζονται οι παρακάτω:

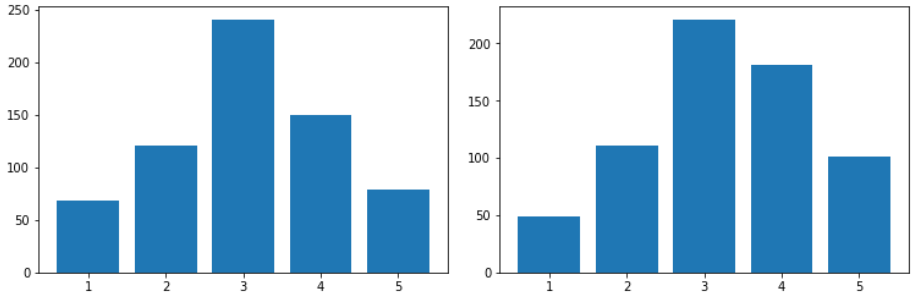
- Embedding Size = 50
- Lstm_size = 512
- Drp = 0.4
- Το Fully Connected Layer έχει μόνο ένα επίπεδο και η έξοδος του είναι όσες και οι διαφορετικές πιθανές τιμές εξόδου (οι οποίες εξαρτώνται από το σύνολο εκπαίδευσης).

Συνολικά από αυτήν την αρχιτεκτονική παράχθηκαν 6 διαφορετικά μοντέλα (ένα για κάθε dataset).

Αξιολόγηση Αρχιτεκτονικής

Για την αρχιτεκτονική αυτή συλλέχθηκαν 1216 αξιολογήσεις συνολικά εκ των οποίων οι 654 είναι από χρήστες χωρίς μουσικές γνώσεις ενώ οι υπόλοιπες 562 από χρηστές οι οποίοι δήλωσαν ότι έχουν μουσικές γνώσεις. Το ποσοστό επιτυχών προβλέψεων για κάθε μια κατηγορία χρηστών είναι **53.07 %** και **53.75% (**αντίστοιχα για χρήστες χωρίς και με μουσικές γνώσεις).

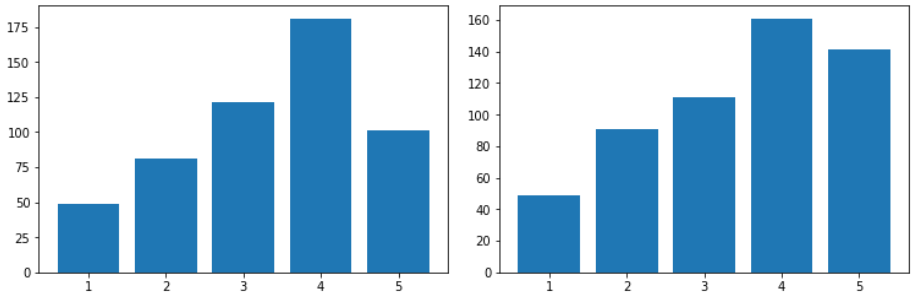
Παρακάτω παρουσιάζονται και τα διαγράμματα που παράχθηκαν από τις ίδιες αξιολογήσεις για την αρέσκεια και το ενδιαφέρον των παραγόμενων κομματιών, της αρχιτεκτονικής αυτής.



(α) (β)

Από τα παραπάνω φαίνεται ότι τα παραγόμενα κομμάτια δεν άρεσαν ιδιαίτερα στους χρήστες μιας και ο μέσος όρος αρεσκείας τους ήταν **3.21** στα 5 ενώ του ενδιαφέροντος τους ήταν **3.9** στα 5.

Παρακάτω παρουσιάζονται τα αντίστοιχα διαγράμματα για τους χρήστες με μουσικές γνώσεις.



(α) (β)

Από τα παραπάνω φαίνεται ότι τα παραγόμενα κομμάτια δεν άρεσαν ιδιαίτερα στους χρήστες μιας και ο μέσος όρος αρεσκείας τους ήταν **3.38** στα 5 ενώ του ενδιαφέροντος τους ήταν **3.98** στα 5.

8.3.2 Συνολική Αξιολόγηση Αρχιτεκτονικής

Το δίκτυο αυτό κατά την λειτουργία του δεν διέφερε σημαντικά από τον απλό autoencoder ο οποίος αναλύθηκε προηγουμένως. Ένα σημαντικό πλεονέκτημα σε σχέση με πριν είναι ότι αυξήθηκε πολύ ο μέσος όρος του μήκους της ακολουθίας που το δίκτυο μπορούσε να κωδικοποιήσει αποδοτικά αλλά και να παράγει. Στην περίπτωση αυτή το δίκτυο για όλες σχεδόν τις αρχικές μελωδίες, μήκους από 30 έως 150 νότες, μπορούσε να παράγει αποδοτικά πάνω από 600-800 νότες. Παράλληλα η αύξηση του μεγέθους της ακολουθίας εισόδου αποτελεί σημαντικό πλεονέκτημα της αρχιτεκτονικής αυτής μιας και έχοντας μεγαλύτερη γνώση της εισόδου το δίκτυο μπορεί να παράγει αποτελέσματα πολύ πιο κοντά στα πραγματικά. Αυτό παρατηρήθηκε αρκετές φορές όπου στην ακολουθία εξόδου υπήρχαν κομμάτια μελωδίας που αντιστοιχίζοντας ακριβώς σε υποσύνολα της αρχικής εισόδου. Επίσης για την αρχιτεκτονική αυτή είχαν μειωθεί κατά πολύ οι μελωδίες οι οποίες την δυσκόλευαν και γενικότερα ανεξάρτητα του αρχικού κομματιού η απόκριση του δικτύου ήταν σχεδόν πάντα το ίδιο αποτελεσματική κάνοντας το δίκτυο να φαίνεται πολύ πιο εύρωστο.

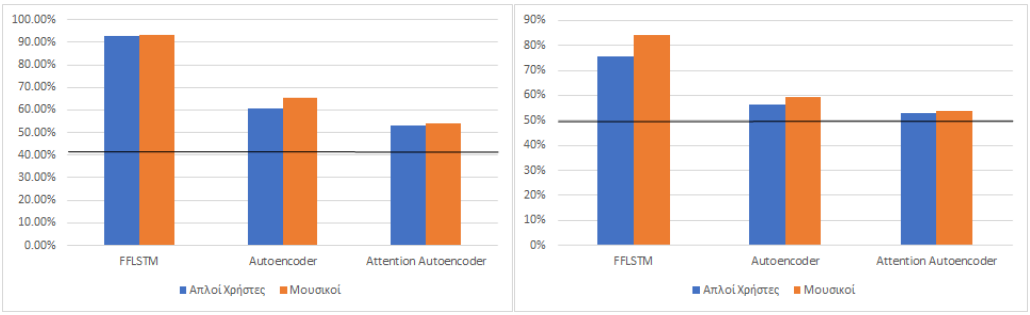
Παρόλα αυτά ένα αρνητικό της παραπάνω αρχιτεκτονική είναι ότι πάντα έπεφτε σε αναδιπλώσεις και μάλιστα πολύ σύντομα. Το φαινόμενο αυτό παρατηρήθηκε για όλα τα σύνολα εκπαίδευσης, μήκους της ακολουθίας εισόδου καθώς και μεγέθους του lstm_size. Για την επίλυση του προβλήματος αυτού χρησιμοποιήθηκε η μη- ντετερμινιστική τεχνική για την επιλογή της εξόδου. Η χρήση όμως της τεχνικής αυτής όπως αναφέρθηκε και προηγουμένως μπορεί να οδηγήσει το δίκτυο σε λανθασμένες επιλογές με αποτέλεσμα διάφορες νότες να ακούγονται ότι δεν εντάσσονται στο κομμάτι με τελικό αποτέλεσμα ο χρήστης να μπορεί να καταλάβει ότι το κομμάτι είναι κατασκευασμένο από υπολογιστή. Παρόλα αυτά αυτό μπορεί

να συμβάλει και στη αύξηση του ενδιαφέροντος των κομματιών (όπως παρατηρείται από την παραπάνω ανάλυση των αποτελεσμάτων). Συνεπώς όλα τα κομμάτια της αρχιτεκτονικής αυτής είναι με την χρήση της μη- ντετερμινιστικής πρόβλεψης.

Τέλος ο χρόνος για την σύνθεση αλλά και την εκπαίδευση είναι παρόμοιος με τον απλό autoencoder και συνεπώς τα συμπεράσματα που προκύπτουν είναι ανάλογα.

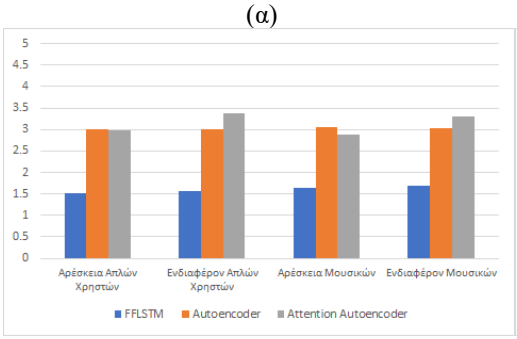
8.4 Αξιολόγηση Αρχιτεκτονικών

Παρακάτω παρουσιάζονται τα βασικά συγκριτικά διαγράμματα των παραπάνω αρχιτεκτονικών για μεγέθη κελίων 256 και 512.

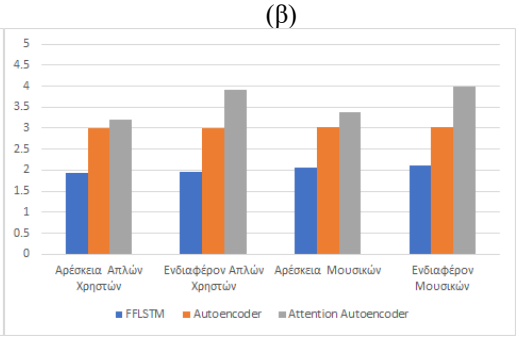


(α)

(β)



(γ)



(δ)

Από τα παραπάνω φαίνεται η ανωτερότητα των αρχιτεκτονικών Κωδικοποιητή-Αποκωδικοποιητή σε σχέση με αυτή του απλού αναδρομικού δικτύου. Επίσης σημαντική βελτίωση των αποτελεσμάτων παρατηρήθηκε και με την προσθήκη του στρώματος συγκέντρωσης. Η βελτίωση αυτή εντοπίστηκε σε όλους τους τομείς προς εξέταση αφού υπήρξε μείωση του ποσοστού επιτυχία των χρηστών ενώ παράλληλα αύξηση του μέσου όρου της αρεσκείας και του ενδιαφέροντος.

Κεφάλαιο 9 – Επίδραση Μεγέθους Κελίου (lstm_size)

Το μέγεθος του κελίου πρακτικά δηλώνει και το μέγεθός του δικτύου ο οποίος εκφράζεται ως ο συνολικός αριθμός των εκπαιδεύσιμων παράμετρών. Το συμπέρασμα αυτό συνεπάγεται από το γεγονός ότι μια αύξηση της τιμής του κελίου θα αυξήσει κατά πολύ τον συνολικό αριθμό των παραμέτρων αυτών. Παρακάτω γίνεται μια ανάλυση της αύξησης του μεγέθους του δικτύου και του κόστους που αυτή επιφέρει στα διάφορα μοντέλα. Αξίζει να σημειωθούν ότι η πρόβλεψη των μοντέλων έγινε σε ένα μέσο λάπτοπ ενώ ο η εκπαίδευση έγινε σε μια κάρτα γραφικών Nvidia K80 (μέσω της πλατφόρμας Google Collab), όποτε και οι αντίστοιχοι χρόνοι αναφέρονται στο παραπάνω hardware.

Παράλληλα με τις απαιτήσεις κάθε μιας αρχιτεκτονικής παρουσιάζονται και τα αποτελέσματα των αξιολογήσεις που συλλέχθηκαν από τους χρήστες. Έτσι μπορεί να γίνει μια συνολική μελέτη του κόστους που επέφερε η αύξηση του δικτύου σε σχέση με την αλλαγή των αποτελεσμάτων.

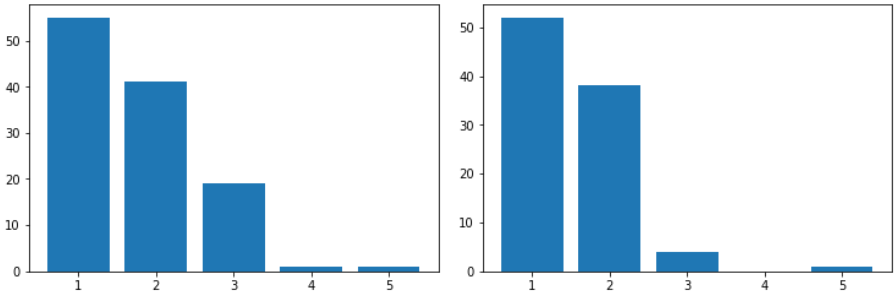
Αξίζει να σημειωθεί παρακάτω παρουσιάζονται μόνο οι αξιολογήσεις για τα μοντέλα τα οποία έχουν εκπαιδευτεί με δεδομένα από πιάνο. Η επιλογή αυτή έγινε μιας και τα αποτελέσματα ανά σύνολο εκπαίδευσης δεν διαφέρουν ιδιαίτερα και έτσι τα τελικά συμπεράσματα που θα εξαγόντουσαν αν είχαν προστεθεί και τα άλλα μοντέλα θα ήταν τα ανάλογα.

9.1 Επίδραση Μεγέθους Κελίου για Απλό Αναδρομικό Δίκτυο

Για την λειτουργία του απλού αναδρομικού δικτύου με cell_size = **256** και σύνολό εκπαίδευσης το αρχικό σύνολο του πιάνο απαιτούνται:

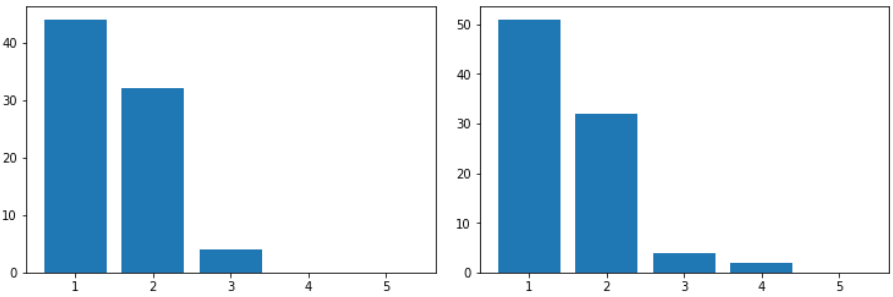
Επίπεδο	Μέγεθος Εισόδου	Μέγεθος εξόδου	Αριθμός Εκπαιδεύσιμων Παραμέτρων
Embedding Layer	1	50	102.950
1ο στρώμα FFLSTM	50	256	314.368
2ο στρώμα FFLSTM	256	256	525.312
3ο στρώμα FFLSTM	256	256	525.312
4ο στρώμα FFLSTM	256	256	525.312
Dense	256	2059	529.163
Σύνολο	-	-	**2.522.417**

Το ποσοστό επιτυχίας εύρεσης του σωστού συνθέτη των χρηστών για το παραπάνω μοντέλο είναι **89.16%** για απλούς χρήστες και **91.81%** για τους χρήστες οι οποίοι δήλωσαν ότι έχουν μουσικές γνώσεις. Ενώ τα αποτελέσματα για τις ερωτήσεις της αρεσκείας και του ενδιαφέροντος των κομματιών της αρχιτεκτονικής αυτής για κάθε μια από τις παραπάνω κατηγορίες παρουσιάζονται παρακάτω



(α) (β)

Ενώ παρακάτω παρουσιάζονται τα αντίστοιχα διαγράμματα για τους χρήστες που δήλωσαν ότι έχουν μουσικές γνώσεις.

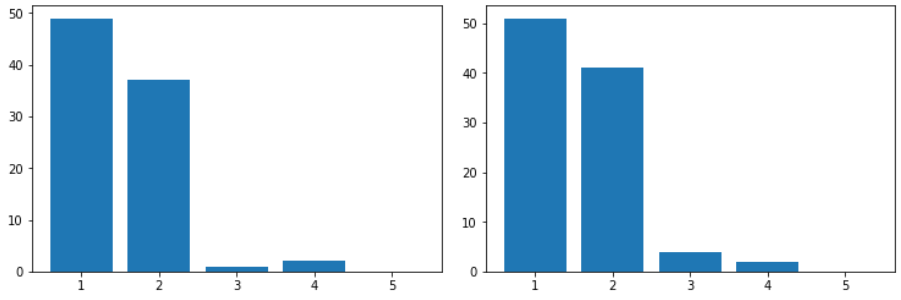


(α) (β)

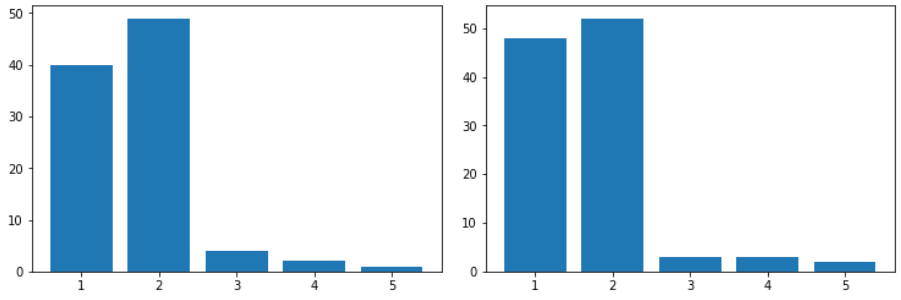
Κατά αντιστοιχία για την λειτουργία του απλού αναδρομικού δικτύου με cell_size = **512** απαιτούνται:

Επίπεδο	**Μέγεθος Εισόδου**	**Μέγεθος εξόδου**	**Αριθμός Εκπαιδεύσιμων Παραμέτρων**
Embedding Layer	1	50	102.950
1ο στρώμα FFLSTM	50	512	1.153.024
2ο στρώμα FFLSTM	512	512	2.099.200
3ο στρώμα FFLSTM	512	512	2.099.200
4ο στρώμα FFLSTM	512	512	2.099.200
Dense	512	2059	1.056.267
Σύνολο	-	-	**8.609.841**

Το ποσοστό επιτυχίας εύρεσης του σωστού συνθέτη των χρηστών για το παραπάνω μοντέλο είναι **76.07%** για απλούς χρήστες και **79.58%** για τους χρήστες οι οποίοι δήλωσαν ότι έχουν μουσικές γνώσεις. Τα αποτελέσματα που συγκεντρώθηκαν για το μοντέλο αυτό παρουσιάζονται παρακάτω, για τους χρήστες με και χωρίς μουσικές γνώσεις.



(α) (β)



(γ) (δ)

Παρακάτω παρουσιάζεται ένας συγκεντρωτικός πίνακας με τα βασικά μετρικά χαρακτηριστικά των αρχιτεκτονικών αυτών.

Μέγεθος Κελιού	Αριθμός Εκπαιδεύσιμων Παραμέτρων	Χρόνος Πρόβλεψης 100 νοτών (δεπτ.)	Χρόνος Εκπαίδευσης μιας Εποχής (δεπτ.)	Μήκος Προβλεπόμενης Ακολουθίας
256	2.522.417	2 - 3	300-400	0-20
512	8.609.841	8 - 9	1000-1100	0-80

Μέγεθος Κελιού	Ποσοστό Λάθους		Μέσος Όρος Αρεσκείας		Μέσος Όρος Ενδιαφέροντος	
	Απλοί Χρήστες	Μουσικοί	Απλοί Χρήστες	Μουσικοί	Απλοί Χρήστες	Μουσικοί
256	89.16%	91.81%	1.4	1.3	1.59	1.41
512	76.07%	79.58%	1.99	1.93	2.17	1.89

Όπως αναφέρθηκε και προηγουμένως και όπως φαίνεται και από τα παραπάνω διαγράμματα, καμία από τις δύο αυτές αρχιτεκτονικές δεν έδωσε καλά αποτελέσματα. Δηλαδή παρόλο που ο συνολικός αριθμός των εκπαιδεύσιμων παραμέτρων αυξήθηκε κατά 341% το τελικό αποτέλεσμα δεν είχε σχεδόν καμία διαφορά. Η παραπάνω αύξηση όμως δυσκολεύει πολύ την εκπαίδευση μιας και πλέον απαιτούνται πολύ παραπάνω δεδομένα με συνέπεια την μεγάλη αύξηση του χρόνου εκπαίδευσης και πρόβλεψης καθιστώντας την αρχιτεκτονική αυτή αδύνατο να ανταποκριθεί σε διάφορες εφαρμογές αυτόματης σύνθεσης μουσικής (όπως την συνεχή παραγωγή).

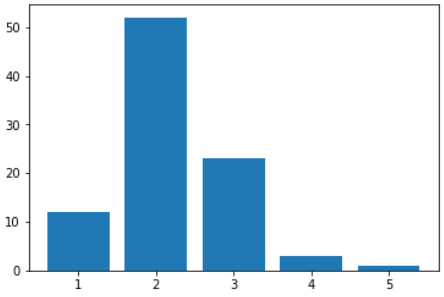
Αξίζει να σημειωθεί ότι και για τις 2 αρχιτεκτονικές οι μέγιστες τιμές του μήκους των ακολουθιών εξόδου, που αναφέρονται στους πίνακες, παρατηρήθηκαν πολύ σπάνια και σε πολύ λίγες αρχικές μελωδίες. Ο γενικός κανόνας ήταν το δίκτυο να παρήγαγε έναν μικρό αριθμό νοτών (περίπου 10 νότες) και στην συνέχεια να τις επαναλάμβανε συνεχώς. Παρόλα αυτά στην μεγαλύτερη αρχιτεκτονική οι παραγόμενες νότες φαινόντουσαν να έχουν κάποια μεγαλύτερη σχέση με την αρχική μελωδία σε αντίθεση με την πιο μικρή όπου ως επί το πλείστων οι παραγωγές φαινόντουσαν τυχαίες.

9.2 Επίδραση Μεγέθους Κελίου για την Αρχιτεκτονική Κωδικοποιητή-Αποκωδικοποιητή

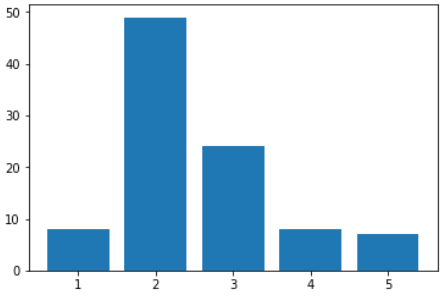
Η αρχιτεκτονική Κωδικοποιητή- Αποκωδικοποιητή με cell_size = **256** και με σύνολο εκπαίδευσης το πιάνο χρησιμοποιεί:

Επίπεδο		Μέγεθος Εισόδου	Μέγεθος Εξόδου	Αριθμός Εκπαιδεύσιμων Μεταβλητών
Encoder	**Embedding**	1	50	102.950
	Lstm	50	256	314.368
Decoder	**Embedding**	1	50	102.950
	Lstm	50	256	314.368
	Dense	256	2059	529.420
Σύνολο		-	-	**1.364.056**

Το ποσοστό επιτυχίας εύρεσης του σωστού συνθέτη των χρηστών για το παραπάνω μοντέλο είναι **62.08%** για απλούς χρήστες και **63.74%** για τους χρήστες οι οποίοι δήλωσαν ότι έχουν μουσικές γνώσεις. Επίσης τα αποτελέσματα της αρχιτεκτονικής αυτής με σύνολο εκπαίδευσης το αρχικό σύνολο του πιάνο παρουσιάζονται παρακάτω, για τους χρήστες χωρίς μουσικές γνώσεις.

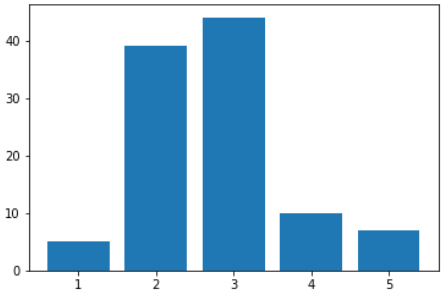


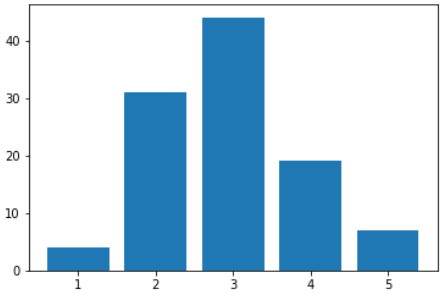
(α)



(β)

Παρακάτω παρουσιάζονται τα αποτελέσματα του παραπάνω μοντέλου για τους χρήστες που δήλωσαν ότι έχουν μουσικές γνώσεις.

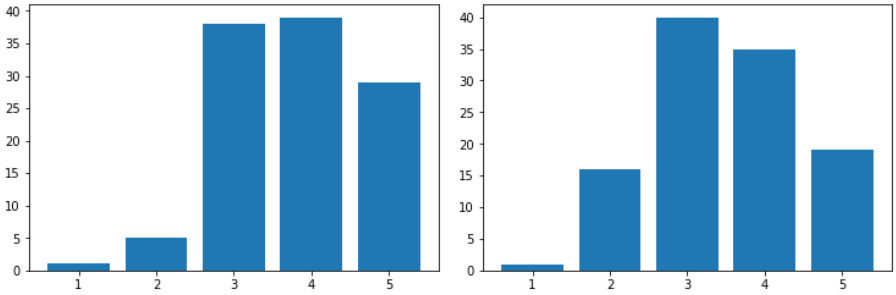




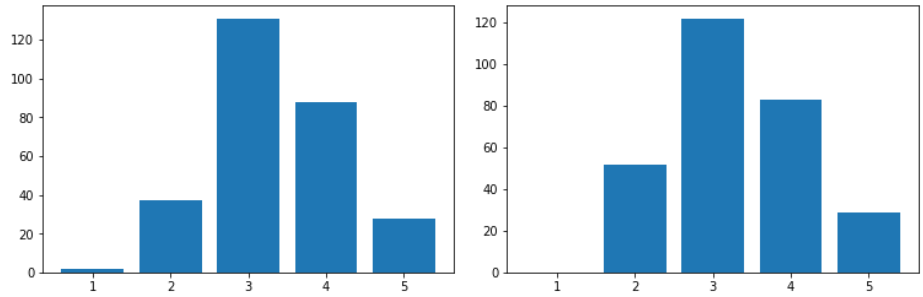
Ενώ για η ίδια αρχιτεκτονική με το ίδιο σύνολο εκπαίδευσης αλλά με cell_size=**512** απαιτεί:

Επίπεδο		**Μέγεθος Εισόδου**	**Μέγεθος Εξόδου**	**Αριθμός Εκπαιδεύσιμων Μεταβλητών**
Encoder	**Embedding**	1	50	102.950
	Lstm	50	512	1.153.024
Decoder	**Embedding**	1	50	102.950
	Lstm	50	512	1.153.024
	Dense	512	2059	1.056.267
Σύνολο		-	-	**3.568.215**

Το ποσοστό επιτυχίας εύρεσης του σωστού συνθέτη των χρηστών για το παραπάνω μοντέλο είναι **56.45%** για απλούς χρήστες και **58.5%** για τους χρήστες οι οποίοι δήλωσαν ότι έχουν μουσικές γνώσεις. Οι αξιολογήσεις που συλλέχθηκαν για αυτό το μοντέλο, από τους χρήστες που δήλωσαν ότι δε έχουν ιδιαίτερες μουσικές γνώσεις παρουσιάζεται παρακάτω.



Και παρακάτω παρουσιάζεται τα αντίστοιχα διαγράμματα αλλά για τους χρήστες που δήλωσαν ότι έχουν μουσικές γνώσεις.



(α) (β)

Παρακάτω παρουσιάζεται ένας συγκεντρωτικός πίνακας με τα χαρακτηριστικά των αρχιτεκτονικών αυτών για να μπορέσει να γίνει πιο ευκολά η αξιολόγηση της επίδρασης του μεγέθους του κελίου.

Μέγεθος Κελιού	**Αριθμός Εκπαιδεύσιμων Παραμέτρων**	**Χρόνος Πρόβλεψης 100 νοτών (δεπτ.)**	**Χρόνος Εκπαίδευσης μιας Εποχής (δεπτ.)**	**Μήκος Προβλεπόμενης Ακολουθίας**
256	1.364.056	1 - 1.5	150-180	200-300
512	3.568.215	1.5 - 2	700-800	300-400

Μέγεθος Κελιού	**Ποσοστό Λάθους**		**Μέσος Όρος Αρεσκείας**		**Μέσος Όρος Ενδιαφέροντος**	
	Απλοί Χρήστες	**Μουσικοί**	**Απλοί Χρήστες**	**Μουσικοί**	**Απλοί Χρήστες**	**Μουσικοί**
256	62.08%	63.74%	2.68	3.15	3.15	2.99
512	56.45%	58.50%	3.05	3.36	3.45	3.31

Από τα παραπάνω φαίνεται ότι οι χρονικές διαφορές των μοντέλων κατά την εκπαίδευση και την πρόβλεψη δεν είναι σημαντικά μεγάλες. Ιδιαιτέρα κατά την πρόβλεψη, όπου εκεί πραγματικά κρίνεται η ταχύτητα του μοντέλου, οι χρόνοι είναι σχεδόν οι ίδιοι. Παρόλα αυτά υπήρξε σημαντική βελτίωση των αποτελεσμάτων, με την μεγαλύτερη αρχιτεκτονική να επιδεικνύει αρκετά βελτιωμένα αποτελέσματά. Συγκεκριμένα όπως φαίνεται και παραπάνω το ποσοστό λάθους των χρηστών (έμπειρών) έπεσε κάτω από το 60% ενώ παράλληλα ανάλογη αύξηση επιτευχθεί και στους μέσους όρους της αρεσκείας και του ενδιαφέροντος των κομματιών.

Όπως είναι λογικό η αύξηση των συνολικών εκπαιδεύσιμων παραμέτρων καθιστά επιτακτική ανάγκη την χρήση μεγαλύτερου συνόλου δεδομένων για την εκπαίδευση. Η αύξηση όμως αυτή δεν αποτελεί ιδιαίτερο πρόβλημα στο συγκεκριμένο πρόβλημα μιας και υπάρχουν ελευθέρα εκατομμύρια κομμάτια μουσικής, από όλα τα είδη και με όλα τα όργανα.

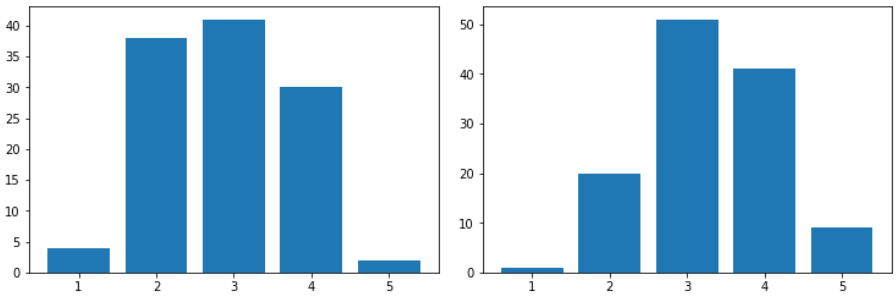
Επίσης αυτά δεν απαιτούν κάποια μη-αυτόματη προεπεξεργασία (όπως κατηγοριοποίηση των δειγμάτων που απαιτείται για αλλά προβλήματα ταξινόμησης κ.α.) και συνεπώς η δημιουργία ενός μεγαλύτερου συνόλου εκπαίδευσης δεν αποτελεί δύσκολο έργο. Συνεπώς η αύξηση του μοντέλου φαίνεται να αξίζει για την αρχιτεκτονική Κωδικοποιητή - Αποκωδικοποιητή.

9.3 Επίδραση Μεγέθους Κελίου για την Αρχιτεκτονική Κωδικοποιητή-Αποκωδικοποιητή με Συγκέντρωση

Η αρχιτεκτονική Κωδικοποιητή- Αποκωδικοποιητή με Συγκέντρωση, με cell_size = **256** και με σύνολο εκπαίδευσης το αρχικό σύνολο του πιάνο χρησιμοποιεί:

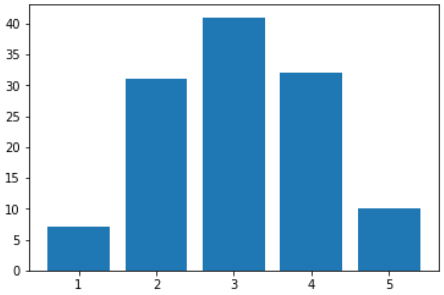
Επίπεδο		**Μέγεθος Εισόδου**	**Μέγεθος Εξόδου**	**Αριθμός Εκπαιδεύσιμων Μεταβλητών**
Encoder	**Embedding**	1	50	102.950
	Lstm	50	256	314.368
Attention Layer	**V_a**	256	256	65792
	W_a	256	256	65792
	U_a	256	1	257
Decoder	**Embedding**	1	50	102.950
	Lstm	50	256	314.368
	Dense	256	2059	529.420
Σύνολο		-	-	**1.495.897**

Το ποσοστό επιτυχίας εύρεσης του σωστού συνθέτη των χρηστών για το παραπάνω μοντέλο είναι **57.49%** για απλούς χρήστες και **58.63%** για τους χρήστες οι οποίοι δήλωσαν ότι έχουν μουσικές γνώσεις. Οι αξιολογήσεις που συλλέχθηκαν για αυτό το μοντέλο, από τους χρήστες που δήλωσαν ότι δε έχουν μουσικές γνώσεις παρουσιάζεται παρακάτω.

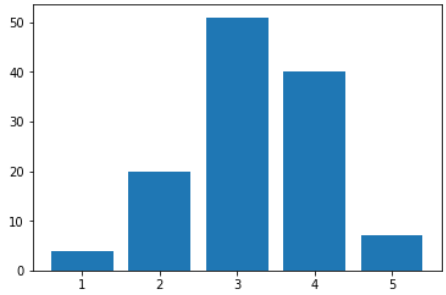


(α) (β)

Ενώ παρακάτω παρουσιάζονται οι αξιολογήσεις για το ιδιο μοντέλο, από τους χρήστες που δήλωσαν ότι έχουν μουσικές γνώσεις.



(α)

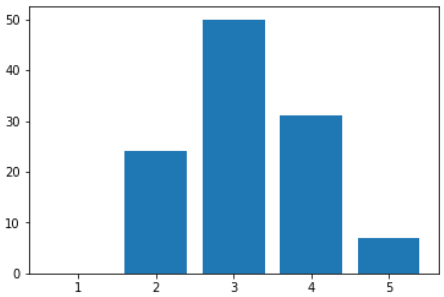


(β)

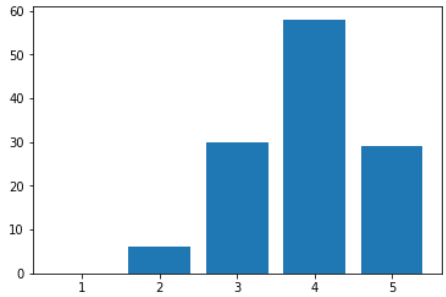
Για την ίδια αρχιτεκτονική, με το ίδιο σύνολο εκπαίδευσης αλλά με lstm_size = **512** το ποσοστό επιτυχίας εύρεσης του σωστού συνθέτη ήταν **53.18%** για απλούς χρήστες και **54.44%** για τους χρήστες οι οποίοι δήλωσαν ότι έχουν μουσικές γνώσεις, ενώ ο αριθμός των παραμέτρων που απαιτούνται για την λειτουργία της παρουσιάζονται στο παρακάτω πίνακα:

Επίπεδο		**Μέγεθος Εισόδου**	**Μέγεθος Εξόδου**	**Αριθμός Εκπαιδεύσιμων Μεταβλητών**
Encoder	**Embedding**	1	50	102.950
	Lstm	50	512	1.153.024
Attention Layer	**V_a**	512	512	262.656
	W_a	512	512	262.656
	U_a	512	1	513
Decoder	**Embedding**	1	50	102.950
	Lstm	50	512	1.153.024
	Dense	512	2059	1.056.267
Σύνολο		-	-	**4.094.040**

Οι αξιολογήσεις που συλλέχθηκαν για αυτό το μοντέλο, από τους χρήστες που δήλωσαν ότι δε έχουν μουσικές γνώσεις παρουσιάζεται παρακάτω.

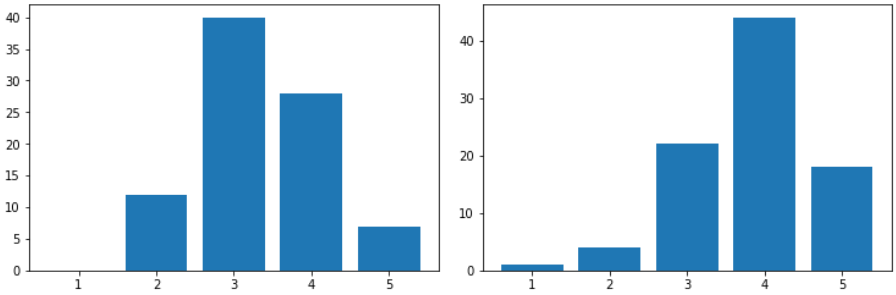


(α)



(β)

Ενώ παρακάτω παρουσιάζονται οι αξιολογήσεις του παραπάνω μοντέλου αλλά για τους χρήστες με μουσικές γνώσεις.



(α) (β)

Παρακάτω παρουσιάζεται ένας συγκεντρωτικός πίνακας με τα χαρακτηριστικά των αρχιτεκτονικών αυτών για να μπορέσει να γίνει πιο ευκολά η αξιολόγηση της επίδρασης του μεγέθους της μνήμης του Lstm.

Μέγεθος Κελιού	**Αριθμός Εκπαιδεύσιμων Παραμέτρων**	**Χρόνος Πρόβλεψης 100 νοτών (δεπτ.)**	**Χρόνος Εκπαίδευσης μιας Εποχής (δεπτ.)**	**Μήκος Προβλεπόμενης Ακολουθίας**
256	1.495.897	1 - 1.6	50-180	400-500
512	4.094.040	1.6 - 2	700-800	600-700

Μέγεθος Κελιού	**Ποσοστό Λάθους**		**Μέσος Όρος Αρεσκείας**		**Μέσος Όρος Ενδιαφέροντος**	
	Απλοί Χρήστες	**Μουσικοί**	**Απλοί Χρήστες**	**Μουσικοί**	**Απλοί Χρήστες**	**Μουσικοί**
256	57.49%	58.43%	3.021	3.07	3.47	3.27
512	53.18%	54.44%	3.28	3.14	3.63	3.85

Όπως και στον απλό autoencoder έτσι και τώρα η αύξηση του δικτύου δεν επέφερε σημαντικές αλλαγές στον χρόνο εκπαίδευσης και πρόβλεψης ενώ παράλληλα αύξησε αισθητά την αισθητική των παραγόμενων αποτελεσμάτων, όπως υποδεικνύουν οι αξιολογήσεις οι οποίες συλλέχθηκαν. Συγκεκριμένα οι απλοί χρήστες είχαν ποσοστό επιτυχίας περίπου 53%, δηλαδή μόνο 3% καλύτερα από την τυχαία πρόβλεψη. Για να ερμηνευθεί η στατιστική σημασία του παραπάνω αποτελέσματος διεξάχθηκε ένα One- Tailed Binomial test (117 επιτυχίες από συνολικά 220 αξιολογήσεις) με δείκτη σημαντικότητας $\alpha = 0.05$. Από το παραπάνω φάνηκε η πιθανότητα επιτυχούς διαχωρισμού των κομματιών μεγαλύτερη από 53% έχει p-value ίση με $0.19 > 0.05$. Συνεπώς από το παραπάνω φαίνεται ότι δεν υπάρχουν αρκετές αποδείξεις (για $\alpha = 0.05$) για να γίνει αποδεκτός ο ισχυρισμός ότι το ποσοστό διαχωρισμού μεταξύ των πραγματικών και των κομματιών που έχουν συντεθεί από την παραπάνω αρχιτεκτονική διαφέρει από την τυχαία επιλογή. Αυτό πρακτικά σημαίνει ότι οι χρήστες δεν ήταν σε θέση να διαχωρίσουν τα πραγματικά από τα παραγόμενα από το συγκεκριμένο δίκτυο κομμάτια. Παρακάτω παρουσιάζονται οι τιμές των p-value για το αντίστοιχο test για τα δυο μεγέθη της αρχιτεκτονικής αυτής για κάθε μια ομάδα χρηστών.

Μέγεθος Κελίου	**Απλοί Χρήστες**	**Μουσικοί**
256	0.022	0.006
512	0.19	0.13

Από τον παραπάνω πίνακα φαίνεται ότι μόνο η μεγαλύτερη αρχιτεκτονική ήταν σε θέση να παράγει κομμάτια τα όποια δεν μπορούσαν να διαχωριστούν από οποιαδήποτε κατηγορία χρηστών (με $\alpha = 0.05$).

Η βελτίωση που επιτεύχθηκε όπως και προηγούμενος φαίνεται να αξίζει μιας και η συνολική αύξηση του μοντέλου καθώς και των δεδομένων που απαιτούνται για την εκπαίδευση δεν είναι ιδιαίτερα μεγάλη ή απαιτητική.

Κεφάλαιο 10- Επίδραση Μη-Ντετερμινιστικής Επιλογής της Πρόβλεψης

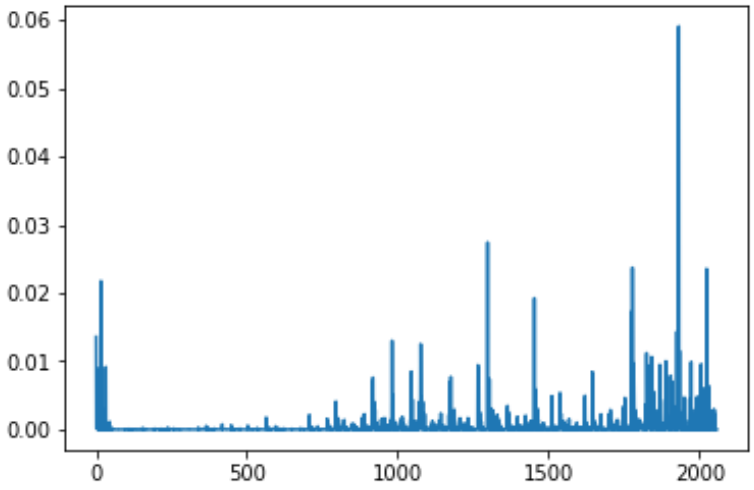
Όπως έχει αναφερθεί και προηγουμένως συχνό πρόβλημα των αναδρομικών αρχιτεκτονικών είναι ότι συχνά μπορεί να αναδιπλώνουν την έξοδό τους, παράγοντας από ένα σημείο και μετά μια μονότονη μελωδία. Μια από τις βασικές αιτίες του φαινομένου αυτού είναι ότι η πρόβλεψη των δικτύων γίνεται με ντετερμινιστικό τρόπο. Έτσι η μείωση των φαινόμενων αυτών μπορεί να επιτευχθεί με την προσθήκη μιας μεθόδου τυχαίας επιλογής της εξόδου, όπως αυτή εξηγείται στο κεφάλαιο 5. Παρακάτω παρουσιάζονται τα αποτελέσματα που επιφέρει αυτή η αλλαγή στα παραγόμενα κομμάτια.

10.1 Επίδραση στο Απλό Αναδρομικό Δίκτυο

Για την αρχιτεκτονική αυτή (ανεξαρτήτως του μεγέθους της, ή του συνόλου εκπαίδευσης) τα αποτελέσματα της τυχαικρατικής μεθόδου επιλογής δεν βελτιώσαν καθόλου την έξοδο, αντίθετα την χειροτέρεψαν κατά πολύ, μιας και οι παραγόμενες νότες φαινόντουσαν σαν να επιλεγόντουσαν τυχαία.

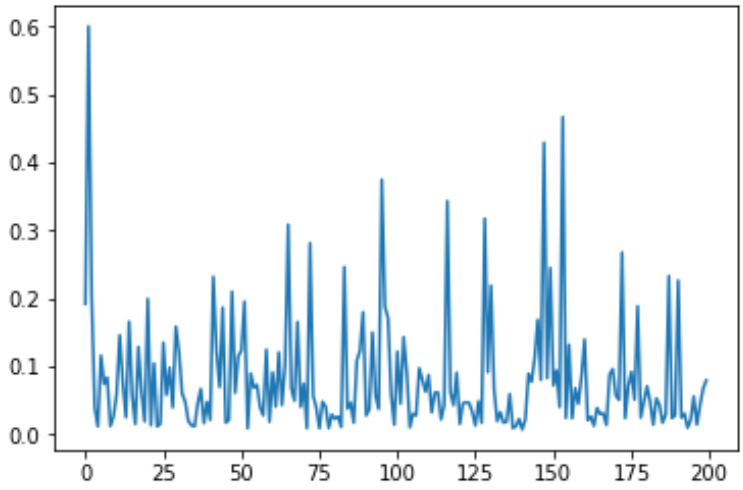
Το φαινόμενο αυτό οφείλεται στο γεγονός ότι η πιθανότητα εξόδου είναι τις περισσότερες φορές κατανέμεται ομοιόμορφα μεταξύ αρκετών νοτών.

Αυτό πρακτικά σημαίνει ότι το σύστημα δεν είναι καθόλου ‘σίγουρο’ για το ποια νότα θα είναι η επόμενη. Παρακάτω παρουσιάζεται μια ενδεικτική κατανομή που παράγει η συνάρτηση softmax που βρίσκεται στο τελευταίο επίπεδο του αναδρομικού δικτύου με seq_length = 512, για μια τυχαία νότα εξόδου.



Από το παραπάνω διάγραμμα βλέπουμε ότι δεν υπάρχει κάποια ισχυρή πιθανότητα για την επομένη επιλογή καθώς η πιθανότητα εξόδου είναι περίπου ισσομοιρασμένη. Αυτό έχει σαν αποτέλεσμα το δίκτυο να κάνει πολλές λανθασμένες επιλογές.

Το παραπάνω φαινόμενο γίνεται αντιληπτό και από το γεγονός ότι η πιθανότητα της πιο σίγουρης εξόδου είναι κάθε φορά πάρα πολύ μικρή. Παρακάτω παρουσιάζεται το διάγραμμα των μεγίστων τιμών της εξόδου της συνάρτησης softmax για την παραγωγή 200 νοτών από το ίδιο μοντέλο και για το ίδιο κομμάτι.



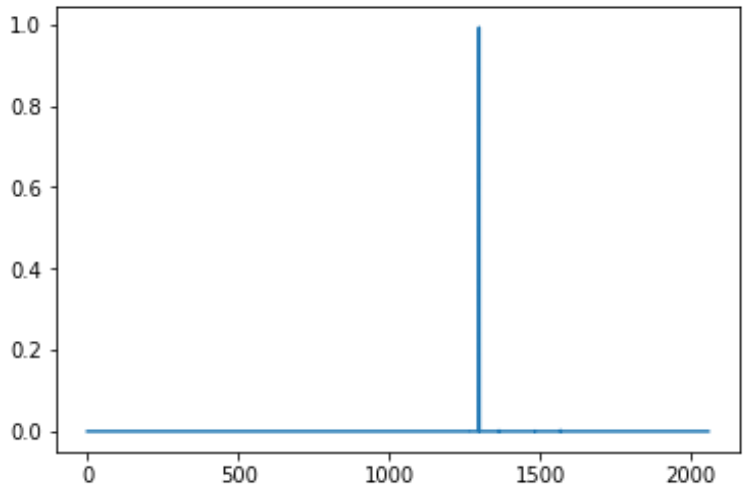
Η μέση τιμή της παραπάνω κατανομής είναι 0.0815. Αυτό σημαίνει ότι ο μέσος όρος του πόσο σίγουρο είναι το δίκτυο αυτό για της εξόδους του είναι μικρότερο από 10%, ενώ η πιο βέβαιη από όλες έχει μόλις πιθανότητα 30%. Το πρόβλημα αυτό είναι εντονότερο στο πιο μικρό δίκτυο, δηλαδή στο απλό αναδρομικό δίκτυο με seq_length= 256.

Για τους λόγους αυτούς δεν χρησιμοποιήθηκαν κομμάτια από την παραπάνω αρχιτεκτονική με την μέθοδο της τυχαικρατικής επιλογής, μιας και τα αποτελέσματα ήταν πολύ κακά.

10.2 Επίδραση στον Κωδικοποιητή - Αποκωδικοποιητή

Η αρχιτεκτονική αυτή από την άλλη επέδειξε πολύ καλύτερα αποτελέσματα με την προσθήκη της μη- ντετερμινιστική συνάρτηση softmax. Με την προσθήκη της τεχνικής αυτής τα αποτελέσματα δεν έχασαν την ποιότητα τους με το δίκτυο πλέον να μπορεί να παράγει ακόμα μεγαλύτερες ακολουθίες εξόδου. Το αποτέλεσμα αυτό είναι αναμενόμενο αν παρατηρηθεί η κατανομή της εξόδου που παράγει η νετερμινιστική συνάρτηση softmax.

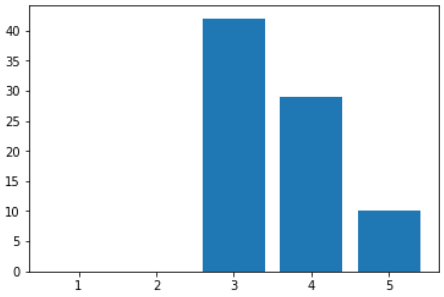
Παρακάτω παρουσιάζεται μια ενδεικτική κατανομή της εξόδου της συνάρτησης αυτής για την παραγωγή μιας νότας από την αρχιτεκτονική αυτή με lstm_size = 512 (πρόκειται για την ίδια αρχική μελωδία και νότα με το αντίστοιχο διάγραμμα του fflstm).



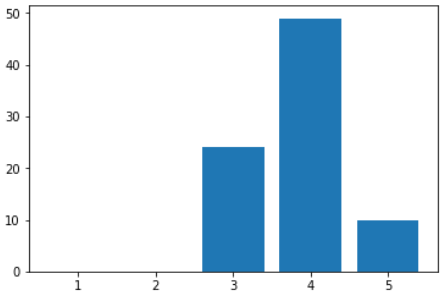
Από το παραπάνω διάγραμμα φαίνεται ότι το δίκτυο είναι πολύ πιο σίγουρο για το ποια θα είναι η επόμενη νότα σε σχέση με την αρχιτεκτονική του fflstm.

Στο παραπάνω κομμάτι για παράδειγμά όλες οι παραγόμενες νότες έχουν πιθανότητα μεγαλύτερη του 50% ενώ περισσότερες από τις 170 έχουν μεγαλύτερη του 80%.

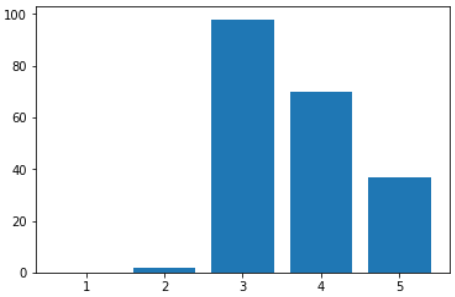
Για να μπορέσει να γίνει μια μελέτη της επίδρασης της τυχαιοκρατικής επιλογής της εξόδου στα τελικά αποτελέσματα, στο σύνολο των αξιολογήσεων έχουν προστεθεί και τα αντίστοιχα κομμάτια που αυτή παρήγαγε. Έτσι το ποσοστό επιτυχών προβλέψεων για τα κομμάτια του Autoencoder, με σύνολο εκπαίδευσης το αρχικό σύνολο του πιάνο και μέγεθος κελίου ίσο με 512 είναι **62.5%** και **65.625%** για τους χρήστες χωρίς και με μουσικές γνώσεις αντίστοιχα. Παρακάτω παρουσιάζονται και τα αντίστοιχα διαγράμματα για τις άλλες δυο εξεταζόμενες παραμέτρου για κάθε μια από τις παραπάνω κατηγορίες χρηστών.



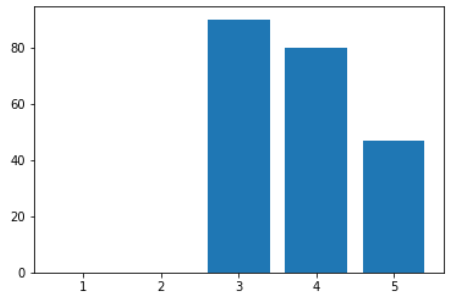
(α)



(β)



(α)



(β)

Αξίζει να σημειωθεί ότι οι αξιολογήσεις για τα κομμάτια που παράχθηκαν με το αντίστοιχο ντετερμινιστικό μοντέλο έχουν παρουσιαστεί στο κεφάλαιο 8. Παρακάτω παρουσιάζεται ένας συγκριτικός πίνακας των αποτελεσμάτων για τις 2 αυτές τεχνικές.

Τεχνική	**Ποσοστό Λάθους Πρόβλεψης**		**Μέσος Όρος Αρεσκείας**		**Μέσος Όρος Ενδιαφέροντος**		**Μήκος Ακολουθίας Εξόδου**
	Απλοί Χρήστες	**Μουσικοί**	**Απλοί Χρήστες**	**Μουσικοί**	**Απλοί Χρήστες**	**Μουσικοί**	
Ντετερμινιστική	53.18%	54.44%	3.05	3.36	3.45	3.31	300-400
Μη-Ντετερμινιστική	58.13%	60.00%	3.68	3.68	3.83	3.80	500-600

Από τον παραπάνω πίνακα παρατηρείται ότι ο μέσος όρος αρεσκείας αλλά και ο μέσος όρος του ενδιαφέροντος είναι σημαντικά αυξημένοι. Η αύξηση αυτή ήταν αναμενόμενη μιας και η προσθήκη μιας τυχαίας επιλογής της εξόδου οδηγεί το δίκτυο στην εξερεύνηση πρωτότυπων και διαφορετικών μελωδιών.

Ένα επίσης σημαντικό χαρακτηριστικό της μη- ντετερμινιστικής softmax είναι ότι το μοντέλο πλέον μπορεί να συνεχίσει επαρκώς ένα κομμάτι για πολύ μεγαλύτερο αριθμό νοτών. Συγκεκριμένα το παραπάνω μοντέλο μπορεί να παράγει αξιόπιστα περίπου στις 300-400 νότες ενώ με την προσθήκη της τυχαιότητας ο αριθμός αυτός υπερδιπλασιάζεται αυξάνοντας τον συνολικό αριθμό των παραγόμενων ακολουθιών σε 700 - 800. Ο αριθμός

αυτός είναι σημαντικά αυξημένος και αποτελεί ένα πολύ σημαντικό πλεονέκτημα της τεχνικής αυτής.

Παρόλα αυτά μέσα στις διαφορετικές μπορεί να εμπεριέχονται και λανθασμένες επιλογές, δηλαδή νότες οι οποίες δεν θα έπρεπε να βρίσκονται μέσα στο κομμάτι. Αυτές ουσιαστικά μπορεί να υποδεικνύουν στους χρήστες ότι το τραγούδι που ακούει έχει συντεθεί από υπολογιστή και δεν αποτελεί κάποια πραγματική μελωδία, μιας και κανένας συνθέτης δεν θα έκανε την συγκεκριμένη επιλογή. Συνεπώς αυτό εξηγεί το γεγονός ότι οι χρήστες μπορούσαν να ξεχωρίσουν με μεγαλύτερη ευκολία τον συνθέτη των κομματιών που παράχθηκαν τυχαία. Το παραπάνω φαινόμενο ενισχύεται και από την μέθοδο αξιολόγησης η οποία όπως αναφέρεται και στο κεφάλαιο 7 βοηθάει σημαντικά τους χρήστες, εκπαιδεύοντάς τους. Για παράδειγμα αν στο πρώτο δείγμα για αξιολόγηση παρουσιαστεί ένα κομμάτι με μια λάθος επιλογή τότε ο χρήστης στην συνέχεια αν ακούσει κάποια ανάλογη μελωδία θα μπορέσει ευκολά να επιλέξει τον σωστό συνθέτη, μειώνοντας και άλλο το ποσοστό λάθους των χρηστών για την συγκεκριμένη μέθοδο. Παρακάτω παρουσιάζονται τα αποτελέσματα του One-Tailed Binomial test με δείκτη σημαντικότητα $a = 0.05$ για το παραπάνω μοντέλο για κάθε μια από τις τεχνικές πρόβλεψης και για κάθε κατηγορία χρηστών.

Μέθοδος Πρόβλεψης	Απλοί Χρήστες	Μουσικοί
Ντετερμινιστική	0.051	0.19
Τυχαία	0.00009	0.00004

Τελικά η επιλογή της συνάρτησης softmax εξαρτάται σημαντικά από τη φύση του προβλήματος που πρέπει να λυθεί, και δεν υπάρχει κάποιο αντικειμενικό κριτήριο που καθιστά κάποια ξεκάθαρα ανώτερη από την άλλη. Πρακτικά όταν ο σκοπός του συστήματος είναι η παραγωγή μεγάλων μελωδιών ή μικρότερών ακολουθιών αλλά με πολύ γρήγορο ρυθμό συνίσταται η επιλογή της τυχαιοκραρικής μεθόδου.

10.3 Επίδραση στον Κωδικοποιητή - Αποκωδικοποιητή με Συγκέντρωση

Η αρχιτεκτονική αυτή δεν έδειξε καθόλου καλά αποτελέσματα με την ντεντερμινιστική τεχνική, μιας και έπεφτε συνεχώς σε αναδιπλώσεις. Παρόλα αυτά με την προσθήκη της τυχαιότητας τα αποτελέσματα βελτιώθηκαν πάρα πολύ και για αυτό η χρήση της (στο συγκεκριμένο πρόβλημα) καθίσταται αναγκαία. Για τον λόγο αυτό όλα τα κομμάτια του συνόλου αξιολόγησης από αυτήν την αρχιτεκτονική αφορούν μόνο την τυχαικρατική μέθοδο. Το μοντέλο αυτό μπορεί να παράγει με ευκολία περισσότερες από 600 νότες χωρίς να υπάρχει απώλεια της ποιότητας. Αξίζει να σημειωθεί ότι τα παραπάνω αποτελέσματα δεν εξαρτώνται από το είδος της αρχικής μελωδίας όπως στις προηγούμενες αρχιτεκτονικές. Πρακτικά αυτό σημαίνει ότι τα παραπάνω μοντέλα μπορούσαν να συνεχίζουν ποιοτικά σχεδόν κάθε μελωδία εισόδου, πράγμα που κάνει το μοντέλο να δείχνει πιο εύρωστο.

Επίσης σημαντικό χαρακτηριστικό είναι τα σφάλματα κατά την τυχαία επιλογή ήταν πολύ λιγότερα σε σχέση με τον Autoencoder. Αυτό οφείλεται στο γεγονός ότι το μοντέλο αυτό είναι πολύ πιο ‘σίγουρο’ για τις επιλογές του αφού οι πιθανές έξοδοι έχουν πολύ πιο ισχυρή πιθανότητα. Έτσι στα συνολικά αποτελέσματα για την αρχιτεκτονική αυτή φαίνεται ότι πολύ λιγότεροι χρήστες κατάλαβαν ότι τα κομμάτια αυτά ήταν κατασκευασμένα από υπολογιστή. Επίσης όπως είναι αναμενόμενο ο μέσος όρος της αρεσκείας και ιδιαίτερα ο μέσος όρος του ενδιαφέροντος είναι σημαντικά μεγαλύτεροι.

Κεφάλαιο 11- Επίδραση Πόλωσης των Δεδομένων Εκπαίδευσης

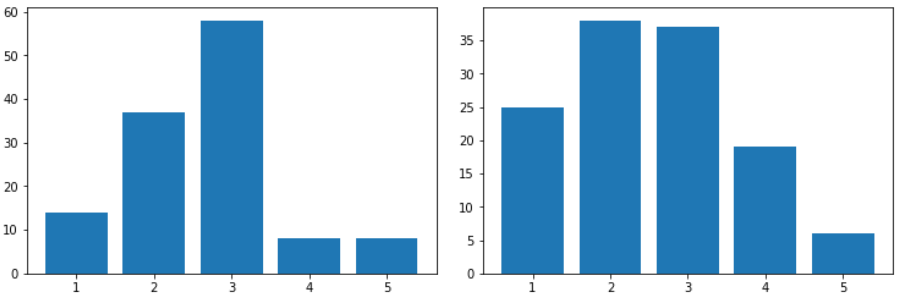
Όπως έχει αναφερθεί και παραπάνω στο αρχικό σύνολο εκπαίδευσης έχει γίνει ένας μετασχηματισμός με σκοπό την μείωση της πόλωσης των δεδομένων. Έτσι για κάθε αρχικό dataset (πιάνο, κιθάρας και κοινό) έχει κατασκευαστεί και ένα αντίστοιχο εξισορροπημένο. Τα δίκτυα εκπαιδευτήκαν με όλα τα σύνολα δεδομένων με σκοπό να μελετηθεί η επίδραση του bias στα παραγόμενα αποτελέσματα. Για την σύγκριση χρησιμοποιήθηκε μόνο το σύνολο του πιάνου μιας και είναι αρκετά αντιπροσωπευτικό των συνολικών αποτελεσμάτων.

Αξίζει να σημειωθεί ότι η ισορρόπηση των δεδομένων δεν απαιτεί την αύξηση του συνολικού αριθμού των εκπαιδεύσιμων παραμέτρων, ούτε την αλλαγή του χρόνου εκπαίδευσης και πρόβλεψης. Έτσι η αλλαγή αυτή δεν απαιτεί περισσότερους πόρους από το σύστημα και για αυτό η παρακάτω σύγκριση εστιάζεται αποκλειστικά στα αποτελέσματα από τους χρήστες.

11.1- Επίδραση στο Απλό Αναδρομικό Δίκτυο

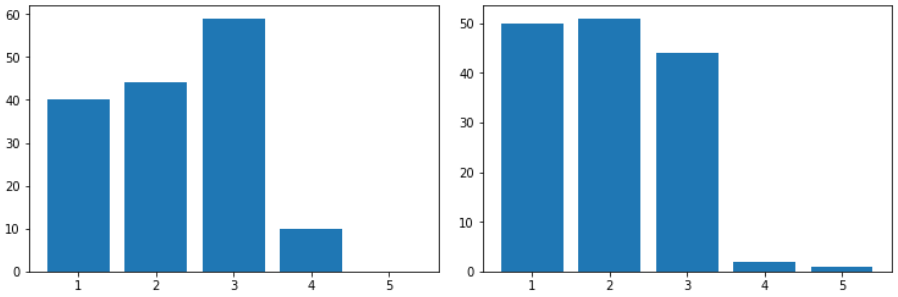
Στο απλό αναδρομικό δίκτυο υπήρξε αισθητή βελτίωση των αποτελεσμάτων με την χρήση του ισορροπημένου συνόλου εκπαίδευσης. Ιδιαίτερα το δίκτυο με lstm_cell=512 μπορεί να παράγει πλέον ακολουθίες των 100-150 νοτών. Υπενθυμίζεται ότι τα μοντέλα τα όποια έχουν εκπαιδευτεί με τα αρχικά σύνολα δεδομένων μπορούν να παράγουν έως το πολύ 50-80 νότες πρώτου αρχίσουν να αναδιπλώνουν την έξοδό τους. Συνεπώς η αύξηση των νοτών είναι πολύ σημαντική μιας και κάνει το δίκτυο λειτουργικό και το μόνο που απαιτεί είναι μια μικρή προεπεξεργασία των κομματιών προτού δοθούν για εκπαίδευση.

Έτσι το ποσοστό επιτυχών προβλέψεων για τα κομμάτια του απλού αναδρομικού δικτύου, με σύνολο εκπαίδευσης το ισορροπημένο σύνολο του πιάνο και μέγεθος κελίου ίσο με 512 είναι **75.63%** και **78.75%** για τους χρήστες χωρίς και με μουσικές γνώσεις αντίστοιχα. Παρακάτω παρουσιάζονται και τα αντίστοιχα διαγράμματα για τους μέσους όρους αρεσκείας και ενδιαφέροντος για όλες τις κατηγορίες χρηστών.



(α)

(β)



(γ)

(δ)

Παρακάτω παρουσιάζεται ένα συγκριτικός πίνακας μεταξύ των μοντέλων τα όποια εκπαιδευτήκαν με το αρχικό και με το ισορροπημένο σύνολο δεδομένων.

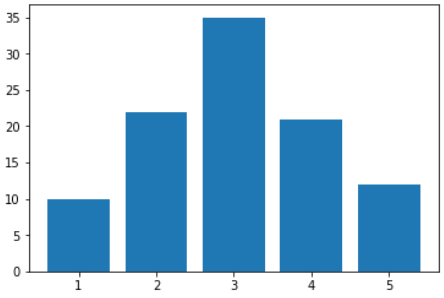
Σύνολο Εκπαίδευσης	Ποσοστό Λάθους		Μέσος όρος Αρεσκείας		Μέσος όρος Ενδιαφέροντος		Μήκος Προβλεπόμενης Ακολουθίας (νότες)
	Απλοί Χρηστές	Μουσικοί	Απλοί Χρήστες	Μουσικοί	Απλοί Χρήστες	Μουσικοί	
Αρχικό	75.64%	84.07%	1.99	1.93	2.17	1.89	0-80
Ισορροπημένο	75.63%	78.75%	2.2	2.21	1.88	1.9	100-150

Από τον παραπάνω πίνακα φαίνεται ότι και πάλι οι χρήστες μπορούσαν να ξεχωρίσουν τον συνθέτη σχεδόν με την ίδια ευκολία με το αρχικό σύνολο δεδομένων. Το ίδιο περίπου ισχύει και για το μέσο όρο της αρεσκείας και του ενδιαφέροντος όπου εκεί σημειώθηκε μια πολύ μικρή βελτίωση για τα μοντέλα που εκπαιδευτήκαν με το ισορροπημένο σύνολο. Όπως αναφέραμε και προηγουμένως και φαίνεται και από τον πίνακα, η ουσιαστική διαφορά των μοντέλων έγκειται στον αριθμό των νοτών που μπορεί να παράγει το δίκτυο ως έξοδο.

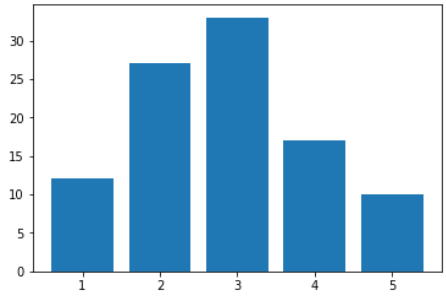
Ανάλογα αποτελέσματα ισχύουν και για την αρχιτεκτονική με lstm_size=256.

11.2- Επίδραση στην αρχιτεκτονική Κωδικοποιητή- Αποκωδικοποιητή

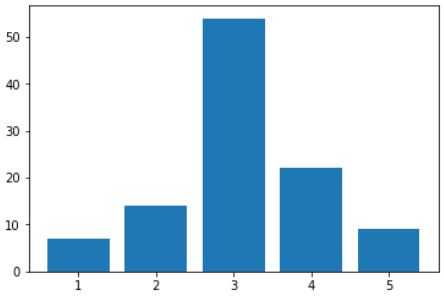
Το ποσοστό επιτυχούς πρόβλεψης των χρηστών για τα test της αρχιτεκτονικής Κωδικοποιητή- Αποκωδικοποιητή με μέγεθος κελίου 256 και σύνολο εκπαίδευσης το ισορροπημένο σύνολο του πιάνο ήταν **57.72%** για τους απλούς χρήστες και **59.54%** για τους χρήστες που δήλωσαν ότι έχουν μουσικές γνώσεις. Παρακάτω παρουσιάζονται οι αξιολογήσεις της αρεσκείας και του ενδιαφέροντος για τα κομμάτια της ίδιας αρχιτεκτονικής, για κάθε κατηγορία χρηστών.



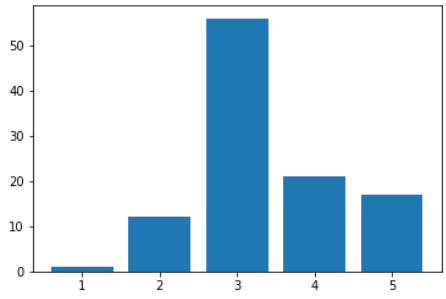
(α)



(β)



(γ)



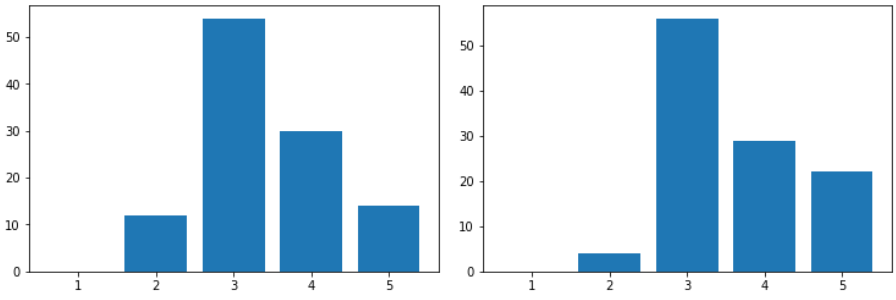
(δ)

Στην συνέχεια παρουσιάζεται ένας συγκριτικός πίνακας των αποτελεσμάτων για την συγκεκριμένη αρχιτεκτονική για τα μοντέλα που εκπαιδευτήκαν με το αρχικό και με το ισορροπημένο σύνολο του πιάνο.

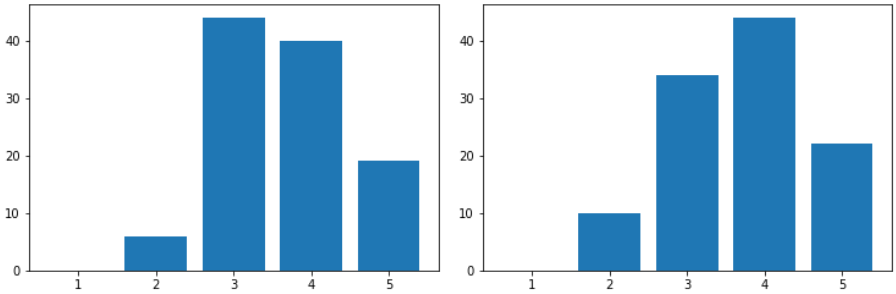
Σύνολο Εκπαίδευσης	Ποσοστό Λάθους		Μέσος όρος Αρεσκείας		Μέσος όρος Ενδιαφέροντος		Μήκος Προβλεπόμενης Ακολουθίας (νότες)
	Απλοί Χρηστές	Μουσικοί	Απλοί Χρήστες	Μουσικοί	Απλοί Χρήστες	Μουσικοί	
Αρχικό	62.08%	63.74%	2.88	3.15	3.15	2.99	200- 300

Ισορροπημένο	57.72%	59.54%	3.2	3.35	3.21	3.23	300-350

Το ποσοστό επιτυχούς πρόβλεψης των χρηστών για τα test της αρχιτεκτονικής Κωδικοποιητή- Αποκωδικοποιητή με μέγεθος κελίου 512 με το ίδιο σύνολο εκπαίδευσης ήταν **55.9%** για τους απλούς χρήστες και **56.81%** για τους χρήστες που δήλωσαν ότι έχουν μουσικές γνώσεις. Παρακάτω παρουσιάζονται οι αξιολογήσεις της αρεσκείας και του ενδιαφέροντος για τα κομμάτια της ίδιας αρχιτεκτονικής, για κάθε κατηγορία χρηστών.



(α) (β)



(γ) (δ)

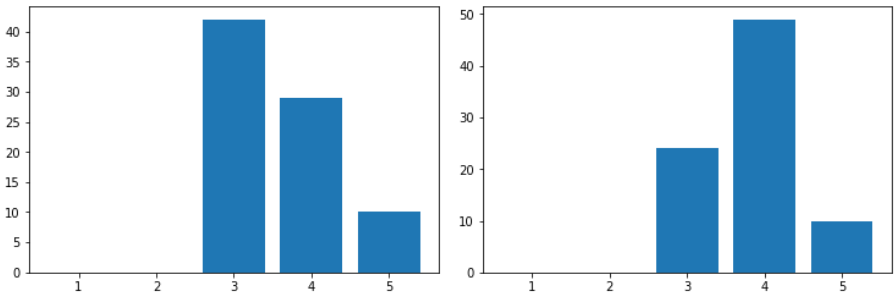
Παρακάτω παρουσιάζεται ένας συγκριτικός πίνακας μεταξύ των αρχιτεκτονικών που εκπαιδευτήκαν με το αρχικό και με το ισορροπημένο σύνολο δεδομένων.

Σύνολο Εκπαίδευσης	**Ποσοστό Λάθους**		**Μέσος όρος Αρεσκείας**		**Μέσος όρος Ενδιαφέροντος**		**Μήκος Προβλεπόμενης Ακολουθίας (νότες)**
	Απλοί Χρηστές	**Μουσικοί**	**Απλοί Χρήστες**	**Μουσικοί**	**Απλοί Χρήστες**	**Μουσικοί**	
Αρχικό	56.45%	58.50%	3.05	3.36	3.45	3.31	300-400
Ισορροπημένο	55.90%	56.81%	3.51	3.41	3.62	3.79	500-600

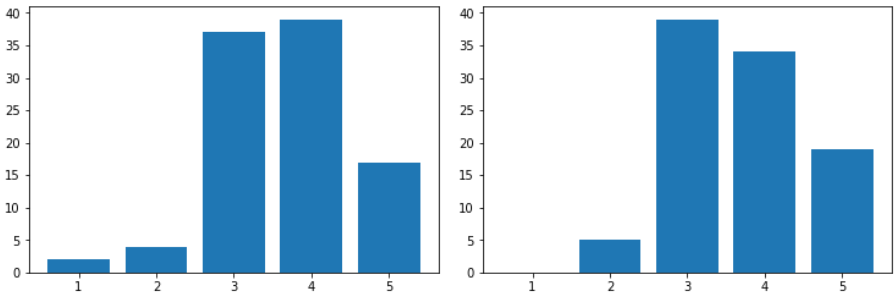
Από τους παραπάνω πίνακες φαίνεται ότι η διαφορές των μοντέλων στις αξιολογήσεις δεν είναι σημαντικά διαφορετικές. Η ουσιαστική διαφορά έγκειται ξανά στον μήκος των παραγόμενων ακολουθιών εξόδου, όπου τα μοντέλα που έχουν εκπαιδευτεί με το ισορροπημένο σύνολο μπορούν να παράγουν σημαντικά μεγαλύτερες ακολουθίες. Επίσης οι καταστάσεις αναδίπλωσης της εξόδου ήταν πολύ πιο σπάνιες και ουσιαστικά το μήκος της παραγόμενης ακολουθίας παρατηρήθηκε ότι ήταν ανεξάρτητο της αρχικής μελωδίας. Για παράδειγμα στην αρχιτεκτονική με cell_size = 256 η οποία είχε εκπαιδευτεί με αρχικό σύνολο του πιάνου, ήταν συχνό φαινόμενο να παράγονται πολύ λίγες νότες πριν το δίκτυο αρχίζει να αναδιπλώνει την έξοδο του, όταν η αρχική μελωδία ήταν από άλλο όργανό ή από διαφορετικό στυλ. Αυτό το φαινόμενο μειώθηκε κατά πολύ όταν η ίδια ακριβώς αρχιτεκτονική εκπαιδευόταν με τα ισορροπημένα σύνολα δεδομένων.

Επίσης αυτό που παρατηρήθηκε κατά την διάρκεια εκπαίδευσης είναι ότι τα παραπάνω μοντέλα συγκλίναν ταχύτερα και έτσι η διαδικασία της κατάρτισης ήταν αρκετά πιο γρήγορη.

Τέλος πολύ καλά αποτελέσματα έδειξαν και τα μοντέλα που εκπαιδεύτηκαν με τα ισορροπημένα σύνολα δεδομένων και τα οποία προέβλεπαν την έξοδο με μη- ντετερμινιστικό τρόπο. Το ποσοστό επιτυχούς πρόβλεψης των χρηστών για τα test της αρχιτεκτονικής αυτής ήταν **56.97%** για τους απλούς χρήστες και **60.50%** για τους χρήστες που δήλωσαν ότι έχουν μουσικές γνώσεις. Παρακάτω παρουσιάζονται οι αξιολογήσεις της αρεσκείας και του ενδιαφέροντος για τα κομμάτια της ίδιας αρχιτεκτονικής, για κάθε κατηγορία χρηστών.



(α) (β)



(γ) (δ)

Παρακάτω παρουσιάζεται ένας συγκριτικός πίνακας για την αρχιτεκτονική του Κωδικοποιητή- Αποκωδικοποιητή με lstm_size = 512 και την τυχαιοκρατική επιλογή της εξόδου με dataset το αρχικό και το ισορροπημένο σύνολο του πιάνο.

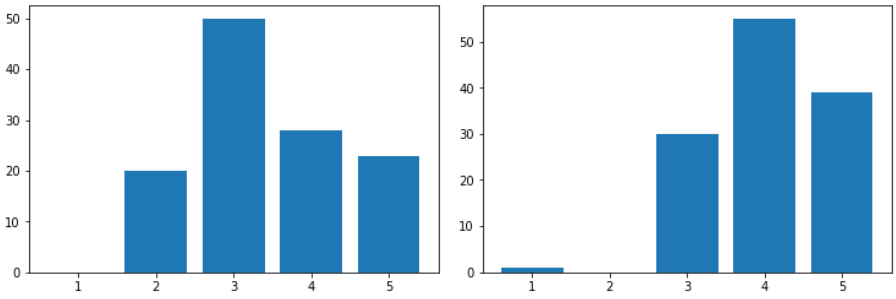
Σύνολο Εκπαίδευσης	**Ποσοστό Λάθους**		**Μέσος όρος Αρεσκείας**		**Μέσος όρος Ενδιαφέροντος**		**Μήκος Προβλεπόμενης Ακολουθίας (νότες)**
	Απλοί Χρηστές	**Μουσικοί**	**Απλοί Χρήστες**	**Μουσικοί**	**Απλοί Χρήστες**	**Μουσικοί**	
Αρχικό	62.50%	65.63%	3.68	3.68	3.83	3.8	500-600
Ισορροπημένο	57.97%	62.5%	3.88	3.75	3.95	3.78	600-700

Από τον παραπάνω πίνακα φαίνεται ότι η αρχιτεκτονική που εκπαιδεύτηκε με το ισορροπημένο σύνολο εκπαίδευσης επιδεικνύει καλύτερα αποτελέσματα σε όλες τις παραμέτρους που εξετάστηκαν, ενώ συγχρόνως μπορεί να παράγει μεγαλύτερες ακολουθίες εξόδου. Παρόλα αυτά και πάλι φαίνεται ότι λόγω της τυχαιοκρατικής επιλογής της πρόβλεψης οι χρήστες είναι σε θέση να ξεχωρίσουν ευκολότερα τον συνθέτη του κάθε κομματιού, όπως εξηγείται αναλυτικότερα και παραπάνω. Όμως στα μοντέλα χωρίς πόλωση το ποσοστό λάθους των χρηστών είναι σημαντικά μεγαλύτερο. Αυτό οφείλεται στο γεγονός ότι τα λάθη της πρόβλεψης, όπως παρατηρήθηκε κατά την σύνθεση των κομματιών, είτε ήταν λιγότερα είτε ήταν πιο μικρά, δηλαδή δεν διέφεραν πολύ από το υπόλοιπο κομμάτι (δεν ήταν τόσο αισθητά).

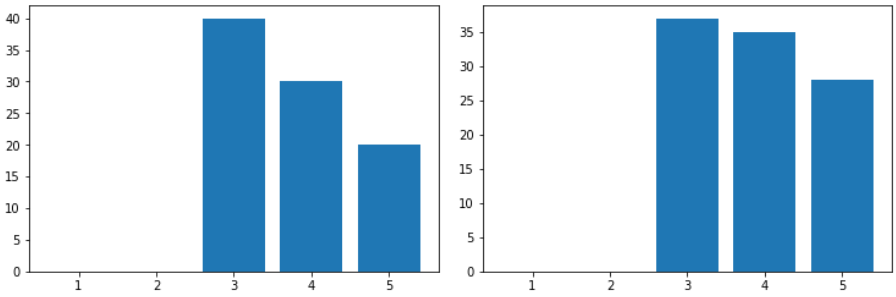
Συνεπώς συνολικά η αρχιτεκτονική Κωδικοποιητή- Αποκωδικοποιητή χωρίς πόλωση έδειξε σημαντικά καλύτερα αποτελέσματα και για τις 2 μεθόδους πρόβλεψης της εξόδου, αυξάνοντας τον συνολικό αριθμό των παραγόμενων νοτών αλλά και των υπόλοιπών παραμέτρων που μελετήθηκαν κατά την διαδικασία της αξιολόγησης.

11.3 Επίδραση στην Αρχιτεκτονική Κωδικοποιητή – Αποκωδικοποιητή με Συγκέντρωση

Το ποσοστό επιτυχούς πρόβλεψης των χρηστών για τα test της αρχιτεκτονικής Κωδικοποιητή- Αποκωδικοποιητή με Συγκέντρωση, με μέγεθος κελίου 512 και σύνολο εκπαίδευσης το ισορροπημένο σύνολο του πιάνο ήταν **52.91%** για τους απλούς χρήστες και **54.00%** για τους χρήστες που δήλωσαν ότι έχουν μουσικές γνώσεις. Παρακάτω παρουσιάζονται οι αξιολογήσεις της αρεσκείας και του ενδιαφέροντος για τα κομμάτια της ίδιας αρχιτεκτονικής, για κάθε κατηγορία χρηστών.



(α) (β)



(γ) (δ)

Παρακάτω παρουσιάζεται ένας συγκριτικός πίνακας για την αρχιτεκτονική του Κωδικοποιητή- Αποκωδικοποιητή με Συγκέντρωση με lstm_size = 512 και την τυχαιοκρατική επιλογή της εξόδου με dataset το αρχικό και το ισορροπημένο σύνολο του πιάνο.

Σύνολο Εκπαίδευσης	Ποσοστό Λάθους		Μέσος όρος Αρεσκείας		Μέσος όρος Ενδιαφέροντος		Μήκος Προβλεπόμενης Ακολουθίας (νότες)
	Απλοί Χρηστές	Μουσικοί	Απλοί Χρήστες	Μουσικοί	Απλοί Χρήστες	Μουσικοί	
Αρχικό	53.18%	54.44%	3.28	3.14	3.63	3.85	600-700
Ισορροπημένο	52.91%	54.00%	3.4	3.69	4.1	3.9	700-1000

Από τον παραπάνω συγκριτικό πίνακα φαίνεται ότι το ποσοστό εύρεσης του συνθέτη από τους χρήστες είναι περίπου το ίδιο. Επίσης διακρίνεται ότι τα κομμάτια της αρχιτεκτονικής που έχει εκπαιδευτεί με το ισορροπημένο σύνολο εκπαίδευσης έχουν σημαντικά αυξημένους μέσους όρους αρεσκείας και ενδιαφέροντος.

Παρόλα αυτά πέρα από τα παραπάνω ποσοτικά χαρακτηριστικά παρατηρήθηκαν και ορισμένες ποιοτικές διαφορές μεταξύ των δυο συγκρινόμενων μοντέλων. Αρχικά τα δίκτυα χωρίς πόλωση ήταν ικανά να παράγουν ακολουθίες πολύ μεγαλύτερες ακολουθίες εξόδου, ενώ παράλληλα μειώθηκαν κατά πολύ τα φαινόμενα αναδίπλωσης της εξόδου, κάνοντας το

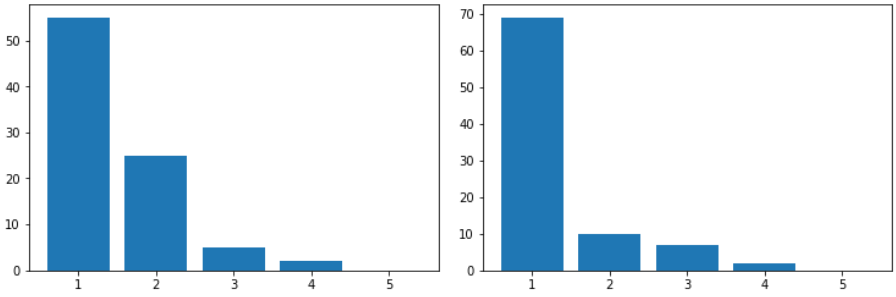
δίκτυο πιο αξιόπιστο. Επίσης αυξήθηκε σημαντικά και ο ελάχιστος αριθμός των νοτών που μπορούσε να παράγει κάθε φορά το δίκτυο. Τέλος η διαδικασία εκπαίδευσης των μοντέλων χωρίς πόλωση είναι ταχύτερη λόγω της γρηγορότερης σύγκλισης του αλγορίθμου εκπαίδευσης.

Κεφάλαιο 12 - Επίδραση Είδος Συνόλου Εκπαίδευσης.

Όπως αναφέρεται και παραπάνω τα μοντέλα που κατασκευάζονται εκπαιδεύονται εκτός των άλλων και με δεδομένα από διαφορετικά μουσικά όργανα. Παρακάτω παρουσιάζονται τα αποτελέσματα των που πέτυχε κάθε μια διαφορετική αρχιτεκτονική ανά σύνολο δεδομένων εκπαίδευσης.

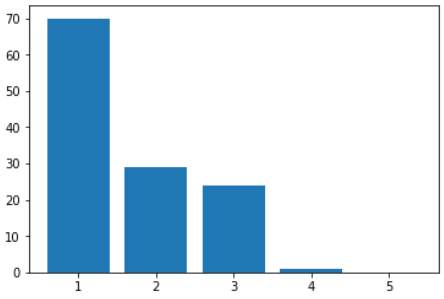
12.1 Επίδραση στο Απλό Αναδρομικό Δίκτυο

Το ποσοστό επιτυχούς του συνθέτη από τους χρηστών για τα tests των κομματιών του απλού αναδρομικού δικτύου με μέγεθος κελίου 256 και σύνολο εκπαίδευσης το αρχικό σύνολο της κιθάρας ήταν **92%** για τους απλούς χρήστες και **93.21%** για τους χρήστες που δήλωσαν ότι έχουν μουσικές γνώσεις. Παρακάτω παρουσιάζονται οι αξιολογήσεις της αρεσκείας και του ενδιαφέροντος για τα κομμάτια της ίδιας αρχιτεκτονικής, για κάθε κατηγορία χρηστών.

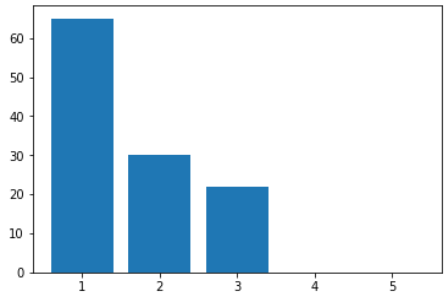


(α)

(β)



(γ)



(δ)

Όσο αναφορά τα κοινά δεδομένα το δίκτυο δεν κατάφερε να ανταποκριθεί μιας και το σφάλμα του μετά την διαδικασία εκπαίδευσης ήταν σημαντικά υψηλό. Για τον λόγο αυτόν η αρχιτεκτονική αυτή (με lstm_size = 256) δεν χρησιμοποιήθηκε με σύνολο εκπαίδευσης το κοινό σύνολο δεδομένων.

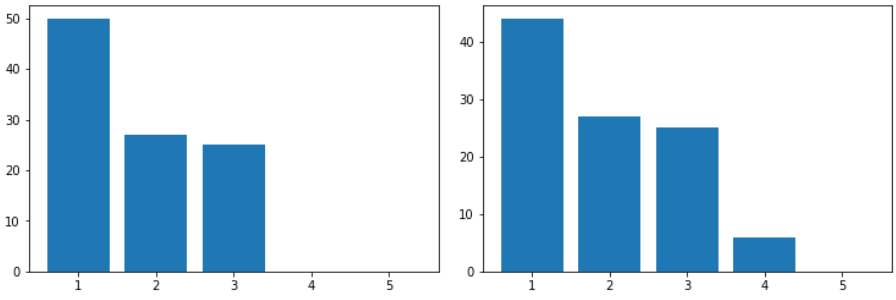
Στην συνέχεια παρουσιάζεται ένας συνοπτικός πίνακας με τα αποτελέσματα που παρουσιάζονται στα παραπάνω διαγράμματα.

Σύνολο Εκπαίδευσης	Ποσοστό Λάθους		Μέσος Όρος Αρεσκείας		Μέσος όρος Ενδιαφέροντος	
	Απλοί Χρήστες	Μουσικοί	Απλοί Χρήστες	Μουσικοί	Απλοί Χρήστες	Μουσικοί
Πιάνο	92.64%	93.38%	1.51	1.64	1.56	1.7
Κιθάρα	92.00%	93.21%	1.47	1.6	1.29	1.57

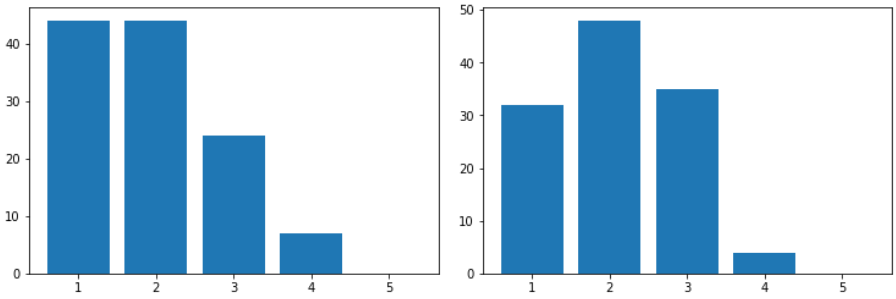
Από τον παραπάνω πίνακα φαίνεται ότι τα αποτελέσματα δεν διαφέρουν σημαντικά για τα 2 παραπάνω όργανα. Τόσο το μήκος της ακολουθίας εξόδου όσο και η δυνατότητα συνέχισης των κομματιών ήταν ανάλογη και για τα 2 μοντέλα. Έτσι η αλλαγή του στυλ στο

σύνολο εκπαίδευσης δεν έφερε ούτε κάποια ποιοτική ούτε και κάποια ποσοτική αλλαγή στα παραγόμενα αποτελέσματα.

Παρακάτω γίνεται η αντίστοιχη μελέτη για το απλό αναδρομικό δίκτυο με μέγεθος κελιού ίσο με 512. Το ποσοστό επιτυχής εύρεσης του συνθέτη για τα tests των κομματιών που έχουν παραχθεί από την παραπάνω αρχιτεκτονική με σύνολο εκπαίδευσης το αρχικό σύνολο του πιάνο ήταν **77.5%** και **85.41%** για τους χρήστες χωρίς και με μουσικές γνώσεις αντίστοιχα. Στην συνέχεια παρουσιάζονται οι αξιολογήσεις της αρεσκείας και του ενδιαφέροντος για τα κομμάτια της ίδιας αρχιτεκτονικής, για κάθε κατηγορία χρηστών.

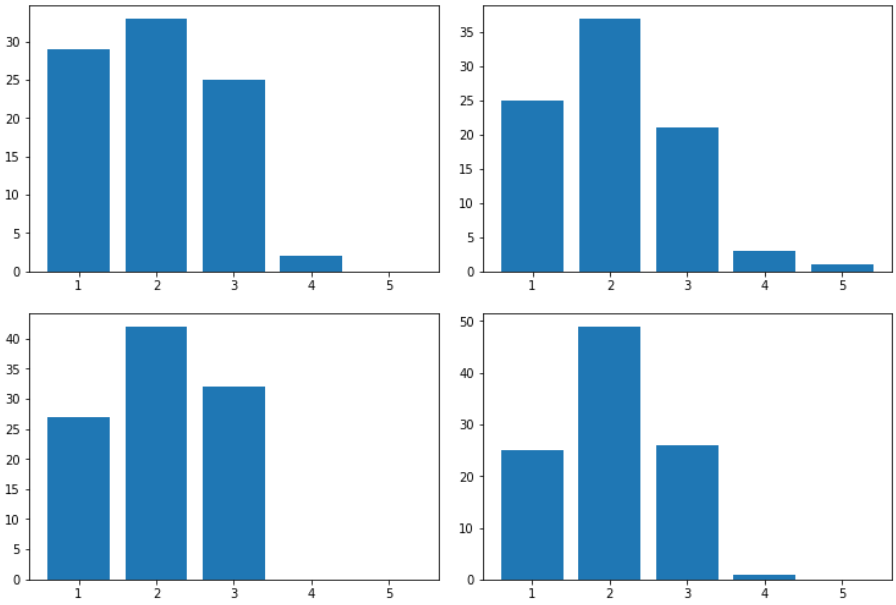


(α) (β)



(γ) (δ)

Τέλος τα αντίστοιχα ποσοστά επιτυχίας των χρηστών για το δίκτυο που εκπαιδεύτηκε με το κοινό σύνολο δεδομένων ήταν **83.33%** και **81.81%** ενώ παρακάτω παρουσιάζονται τα αποτελέσματα της αρεσκείας και του ενδιαφέροντος για κάθε μια κατηγορία χρηστών.



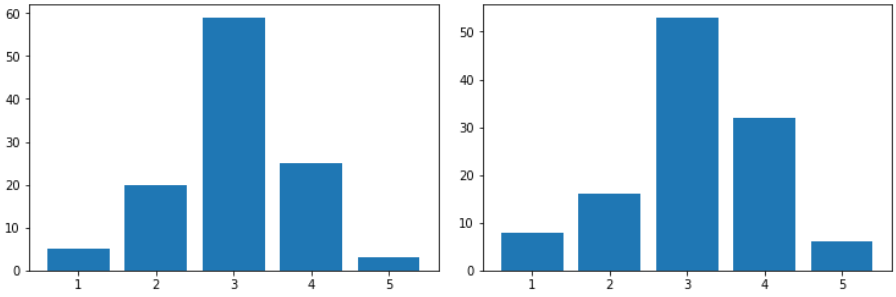
Σύνολο Εκπαίδευσης	**Ποσοστό Λάθους**	**Μέσος Όρος Αρεσκείας**	**Μέσος όρος Ενδιαφέροντος**

	Απλοί Χρήστες	Μουσικοί	Απλοί Χρήστες	Μουσικοί	Απλοί Χρήστες	Μουσικοί
Πιάνο	76%	84.07%	1.99	2.17	1.93	1.89
Κιθάρα	77.50%	85.41%	1.75	1.94	1.93	2.09
Κοινό	83.33%	81.81%	2	2.04	2.57	2.02

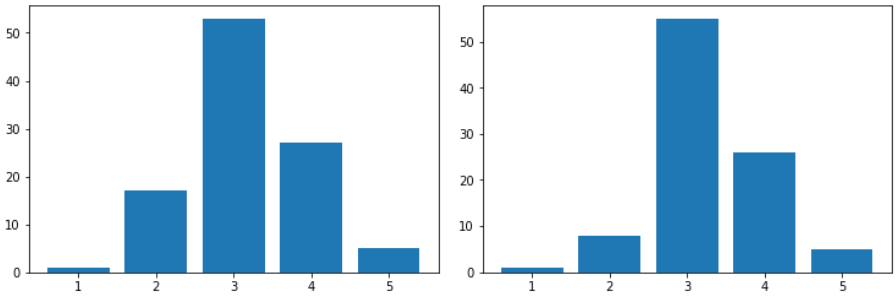
Από τον παραπάνω συγκριτικό πίνακα παρατηρείται ότι δεν υπάρχουν μεγάλες διαφορές μεταξύ των μοντέλων που έχουν εκπαιδευτεί με τα σύνολα του πιάνου και τις κιθάρας, δηλαδή μεταξύ αρχιτεκτονικών που έχουν εκπαιδευτεί με διαφορετικά στυλ. Παρόλα αυτά δεν υπήρξε επίσης καμία αξιοσημείωτη διαφορά για την αρχιτεκτονική με σύνολο εκπαίδευσης κοινό σύνολο δεδομένων. Η παραπάνω παρατήρηση έχει ιδιαίτερη αξία αφού το κοινό σύνολο δεδομένων είναι σχεδόν διπλάσιο σε μέγεθός πράγμα που σημαίνει ότι τα αποτελέσματα αναμένονταν να ήταν καλύτερα για το μοντέλο αυτό.

12.2 Επίδραση στην Αρχιτεκτονική Κωδικοποιητή- Αποκωδικοποιητή

Το ποσοστό επιτυχούς του συνθέτη από τους χρηστών για τα tests των κομματιών της αρχιτεκτονικής Κωδικοποιητή- Αποκωδικοποιητή με μέγεθος κελίου 256 και σύνολο εκπαίδευσης το αρχικό σύνολο της κιθάρας ήταν **64.09%** για τους απλούς χρήστες και **65.45%** για τους χρήστες που δήλωσαν ότι έχουν μουσικές γνώσεις. Παρακάτω παρουσιάζονται οι αξιολογήσεις της αρεσκείας και του ενδιαφέροντος για τα κομμάτια της ίδιας αρχιτεκτονικής, για κάθε κατηγορία χρηστών.



(α) (β)



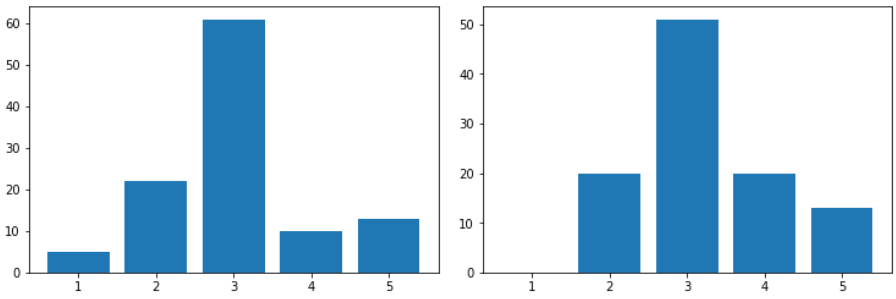
(γ) (δ)

Παρακάτω παρουσιάζονται ένας συγκριτικός πίνακας των αποτελεσμάτων για τα 2 παραπάνω μοντέλα.

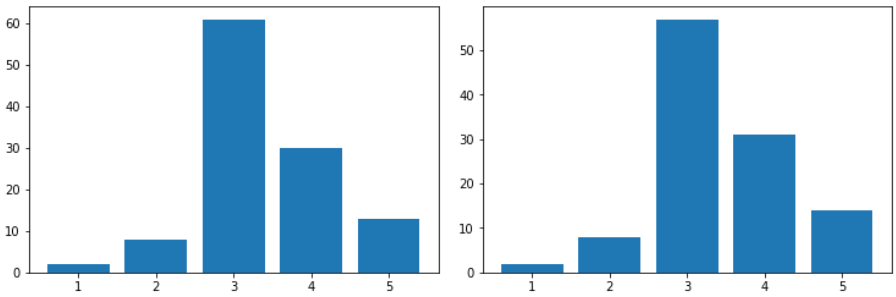
Σύνολο Εκπαίδευσης	**Ποσοστό Λάθους**		**Μέσος Όρος Αρεσκείας**		**Μέσος όρος Ενδιαφέροντος**	
	Απλοί Χρήστες	**Μουσικοί**	**Απλοί Χρήστες**	**Μουσικοί**	**Απλοί Χρήστες**	**Μουσικοί**
Πιάνο	62.08%	63.74%	2.68	3.15	3.15	2.99
Κιθάρα	64.09%	65.45%	3.21	2.93	3.1	3.12

Τα αποτελέσματα όπως και για την αρχιτεκτονική του απλού αναδρομικού δικτύου είναι ανάλογα και για τα 2 μοντέλα.

Το ποσοστό επιτυχούς του συνθέτη από τους χρηστών για τα tests των κομματιών της αρχιτεκτονικής Κωδικοποιητή- Αποκωδικοποιητή με μέγεθος κελίου 512 και σύνολο εκπαίδευσης το αρχικό σύνολο της **κιθάρας** ήταν **57.09%** για τους απλούς χρήστες και **58.63%** για τους χρήστες που δήλωσαν ότι έχουν μουσικές γνώσεις. Παρακάτω παρουσιάζονται οι αξιολογήσεις της αρεσκείας και του ενδιαφέροντος για τα κομμάτια της ίδιας αρχιτεκτονικής, για κάθε κατηγορία χρηστών.

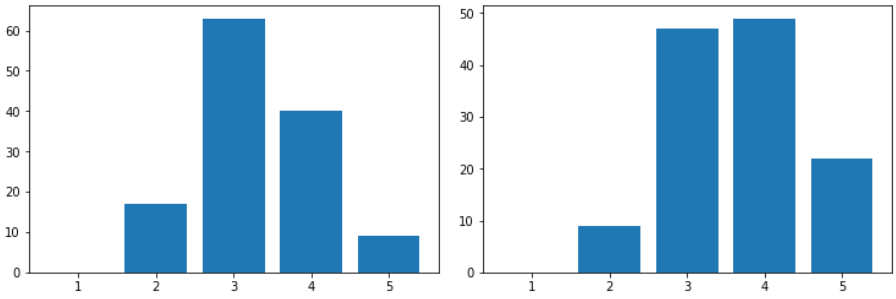


(α) (β)

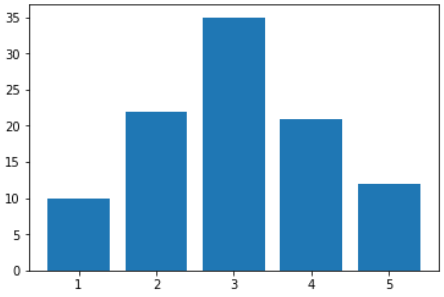


(γ) (δ)

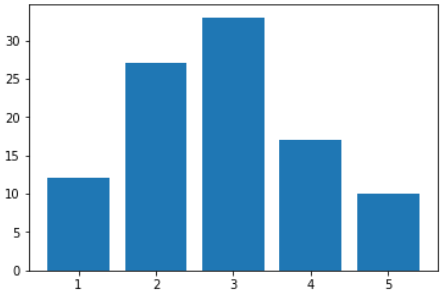
Το ποσοστό επιτυχούς του συνθέτη από τους χρηστών για τα tests των κομματιών της αρχιτεκτονικής Κωδικοποιητή- Αποκωδικοποιητή με μέγεθος κελίου 512 και σύνολο εκπαίδευσης το αρχικό σύνολο της **κοινό** σύνολο δεδομένων ήταν **53.84%** για τους απλούς χρήστες και **54.54%** για τους χρήστες που δήλωσαν ότι έχουν μουσικές γνώσεις. Παρακάτω παρουσιάζονται οι αξιολογήσεις της αρεσκείας και του ενδιαφέροντος για τα κομμάτια της ίδιας αρχιτεκτονικής, για κάθε κατηγορία χρηστών.



(α) (β)



(γ)

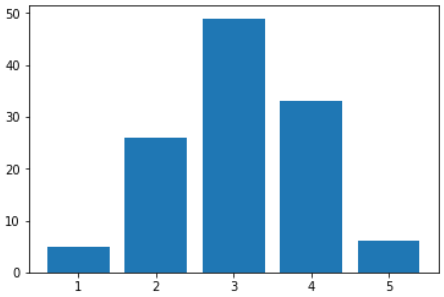


(δ)

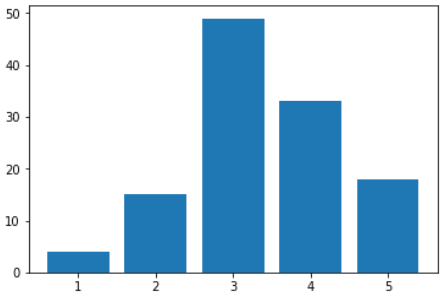
Σύνολο Εκπαίδευσης	Ποσοστό Λάθους		Μέσος Όρος Αρεσκείας		Μέσος όρος Ενδιαφέροντος	
	Απλοί Χρήστες	Μουσικοί	Απλοί Χρήστες	Μουσικοί	Απλοί Χρήστες	Μουσικοί
Πιάνο	56.45%	58.50%	3.05	3.36	3.45	3.31
Κιθάρα	57.27%	58.63%	3.03	3.22	3.25	3.27
Κοινά	53.84%	54.54%	3.31	3.44	3.61	3.53

12.3 Επίδραση στην Αρχιτεκτονική Κωδικοποιητή- Αποκωδικοποιητή με Συγκέντρωση

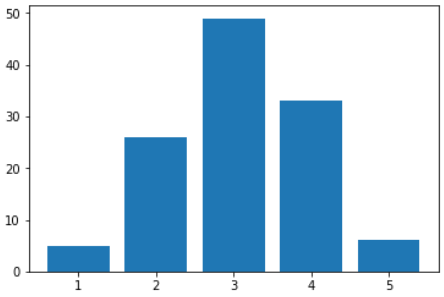
Το ποσοστό επιτυχούς του συνθέτη από τους χρηστών για τα tests των κομματιών της αρχιτεκτονικής Κωδικοποιητή- Αποκωδικοποιητή με Συγκέντρωση, με μέγεθος κελίου 256 και σύνολο εκπαίδευσης το αρχικό σύνολο της **κιθάρας** ήταν **57.91%** για τους απλούς χρήστες και **59.90%** για τους χρήστες που δήλωσαν ότι έχουν μουσικές γνώσεις. Παρακάτω παρουσιάζονται οι αξιολογήσεις της αρεσκείας και του ενδιαφέροντος για τα κομμάτια της ίδιας αρχιτεκτονικής, για κάθε κατηγορία χρηστών.



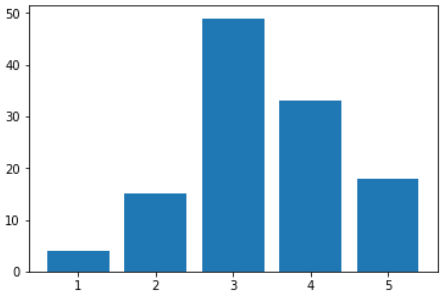
(α)



(β)



(γ)



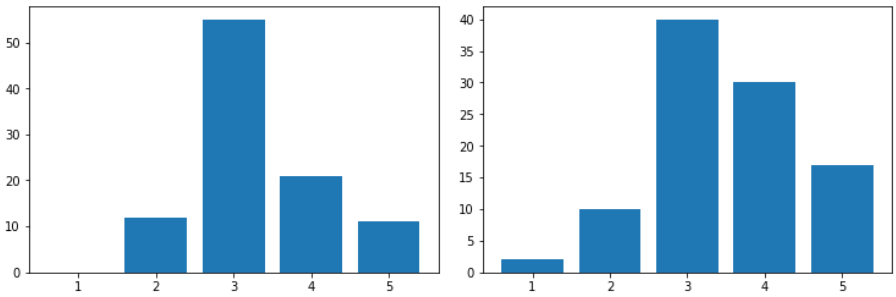
(δ)

Παρακάτω παρουσιάζονται ένας συγκριτικός πίνακας των αποτελεσμάτων για τα 2 παραπάνω μοντέλα.

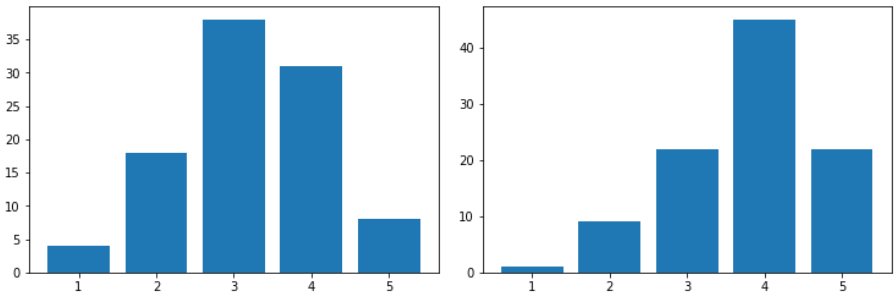
Σύνολο Εκπαίδευσης	Ποσοστό Λάθους		Μέσος Όρος Αρεσκείας		Μέσος όρος Ενδιαφέροντος	
	Απλοί Χρήστες	Μουσικοί	Απλοί Χρήστες	Μουσικοί	Απλοί Χρήστες	Μουσικοί
Πιάνο	57.49%	58.63%	3.021	3.07	3.47	3.27
Κιθάρα	57.91%	59.90%	3	3.1	3.39	3.45

Όπως και με τις προηγούμενες αρχιτεκτονικές έτσι και για αυτήν δεν υπήρξαν μεγάλες διαφορές στις αξιολογήσεις των χρηστών για τα δυο διαφορετικά datasets.

Το ποσοστό επιτυχούς του συνθέτη από τους χρηστών για τα tests των κομματιών της αρχιτεκτονικής Κωδικοποιητή- Αποκωδικοποιητή με Συγκέντρωση, με μέγεθος κελίου 512 και σύνολο εκπαίδευσης το αρχικό σύνολο της **κιθάρας** ήταν **53.5%** για τους απλούς χρήστες και **55.50%** για τους χρήστες που δήλωσαν ότι έχουν μουσικές γνώσεις. Παρακάτω παρουσιάζονται οι αξιολογήσεις της αρεσκείας και του ενδιαφέροντος για τα κομμάτια της ίδιας αρχιτεκτονικής, για κάθε κατηγορία χρηστών.

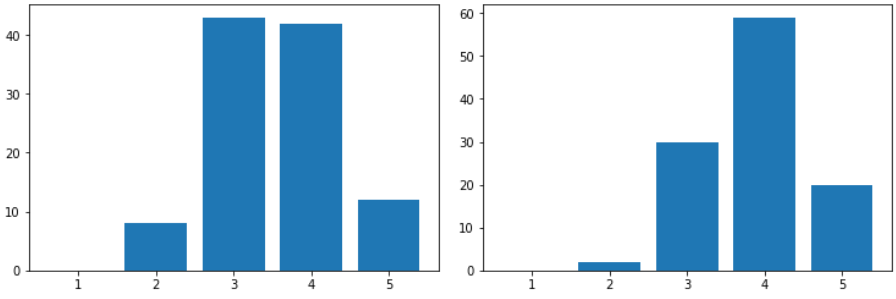


(α) (β)

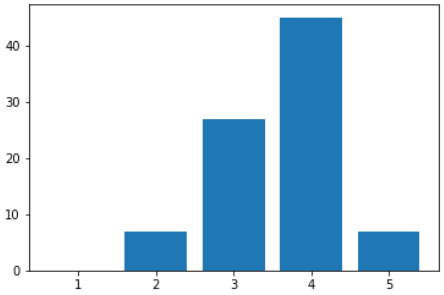


(γ) (δ)

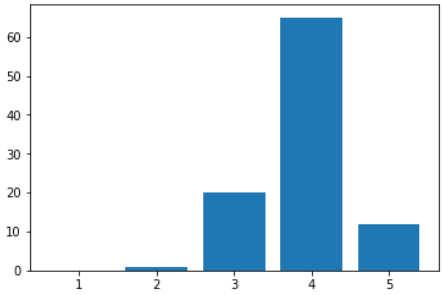
Το ποσοστό επιτυχούς του συνθέτη από τους χρηστών για τα tests των κομματιών της αρχιτεκτονικής Κωδικοποιητή- Αποκωδικοποιητή με Συγκέντρωση, με μέγεθος κελίου 512 και σύνολο εκπαίδευσης κοινό σύνολο των αρχικών δεδομένων ήταν **52.5%** για τους απλούς χρήστες και **53.50%** για τους χρήστες που δήλωσαν ότι έχουν μουσικές γνώσεις. Παρακάτω παρουσιάζονται οι αξιολογήσεις της αρεσκείας και του ενδιαφέροντος για τα κομμάτια της ίδιας αρχιτεκτονικής, για κάθε κατηγορία χρηστών.



(α) (β)



(α)



(β)

Παρακάτω παρουσιάζεται ένας συγκριτικός πίνακας των αξιολογήσεων που συλλέχθηκαν για τα μοντέλα με τα διαφορετικά σύνολα εκπαίδευσης.

Σύνολο Εκπαίδευσης	**Ποσοστό Λάθους**		**Μέσος Όρος Αρεσκείας**		**Μέσος όρος Ενδιαφέροντος**	
	Απλοί Χρήστες	**Μουσικοί**	**Απλοί Χρήστες**	**Μουσικοί**	**Απλοί Χρήστες**	**Μουσικοί**
Πιάνο	53.18%	54.44%	3.28	3.14	3.63	3.85
Κιθάρα	53.50%	55.50%	3.4	3.18	3.52	3.86
Κοινά	52.50%	53.50%	3.56	3.6	3.96	3.92

Συμπεράσματα

Από τις αξιολογήσεις των χρηστών φαίνεται ότι μεταξύ των μοντέλων που εκπαιδευτήκαν με το σύνολο του πιάνο και τις κιθάρας δεν παρατηρούνται ιδιαιτέρες διαφορές, ούτε κάποια παράμετρος η οποία να αυξήθηκε σημαντικά.

Παρόλα αυτά τα μοντέλα που εκπαιδευτήκαν με το κοινό σύνολο δεδομένων επέδειξαν πολύ καλύτερη συμπεριφορά. Αυτό κατά κύριο λόγο οφείλεται στο γεγονός ότι το σύνολο αυτό ήταν διπλάσιο σε μέγεθος από τα άλλα.

Κεφάλαιο 13- Υλοποίηση

Όλα τα πειράματα και τα σενάρια (scripts) για την συλλογή των δεδομένων, την προ επεξεργασία τους, την κατασκευή την εκπαίδευση και την πρόβλεψη των μοντέλων, αλλά και την εξαγωγή των αποτελεσμάτων υλοποιήθηκαν στην γλώσσα προγραμματισμού Python.

Συγκεκριμένα για την προεπεξεργασία των δεδομένων χρησιμοποιήθηκε το πακέτο music21 το όποιο αποτελεί ένα εργαλείο για την ανάλυση μουσικών κομματιών το οποίο έχει κατασκευαστεί από το MIT. Το παραπάνω εργαλείο παράγει διάφορες συναρτήσεις και αντικείμενα για την ταχεία αναπαραγωγή, ανάλυση και δημιουργία κομματιών σε ακολουθιακή μορφή (Midi).

Συγκεκριμένα για την κατασκευή και την εκπαίδευση των μοντέλων χρησιμοποιήθηκε το framework που έχει αναπτύξει η Google ειδικά για την μηχανική μάθηση με νευρωνικά δίκτυα, το Tensorflow[E]. Για τα απλό αναδρομικό δίκτυο καθώς και για τον απλό Autoencoder, χρησιμοποιήθηκε το framework Keras[E] το όποια στηρίζεται στο Tensorflow και κάνει την συγγραφή συνθέτων αρχιτεκτονικών πιο εύκολη.

Τα παραπάνω εργαλεία χρησιμοποιούνται ευρέως στον χώρο της μηχανικής μάθησης παρέχοντας βοηθητικές συναρτήσεις για την γρήγορη εκπαίδευση, τον υπολογισμό μετρικών καθώς και την επεξεργασίας μεγάλου όγκου δεδομένων με την χρήση καρτών γραφικών (GPU's), αν αυτές υπάρχουν. Έτσι ο χρηστής δεν σπαταλά χρόνο στην σχεδίαση και την συγγραφή κώδικά για την εκμετάλλευση των πόρων του συστήματος του, μιας και την δουλεία αυτή την αναλαμβάνουν τα παραπάνω Frameworks. Συγκεκριμένα το βιβλιοθήκη Tensorflow χρησιμοποιεί την διεπαφή Cuda [E], η όποια έχει αναπτυχθεί από την κατασκευάστρια καρτών γραφικών NVIDIA και έτσι μπορεί να τρέχει το μοντέλο σε μια ή περισσότερες GPU, αντί της CPU, μειώνοντας τελικά κατά πολύ τον χρόνο εκτέλεσης. Η βελτίωση της ταχύτητας έρχεται ως αποτέλεσμα της εκτέλεσης παράλληλου κώδικα που προσφέρεται από τις κάρτες γραφικών, ανάγκη η οποία συναντάται πολύ έντονα στα νευρωνικά δίκτυα.

Για την σελίδα αξιολόγησης χρησιμοποιήθηκε το framework για web development Flask. Αυτό είναι υπεύθυνο για την διαχείριση των requests, την λειτουργεία του backend στον server κ.α. Το backend ουσιαστικά είναι πανομοιότυπο με τα μοντέλα όπως αυτά περιεγράφηκαν παραπάνω, ενώ για το frontend χρησιμοποιήθηκε το framework Bootstrap 3.3. Για την αναπαραγωγή των Midi κομματιών στον browser χρησιμοποιήθηκε η βιβλιοθήκη midiJs, ενώ για την αποθήκευση και την διαχείριση των αποτελεσμάτων χρησιμοποιήθηκαν διάφορες βιβλιοθήκες της python καθώς και το εργαλείο Excel.

Τέλος αξίζει να σημειωθεί ότι όλος πηγαίος κώδικας όλων των μοντέλων καθώς και των συναρτήσεων προεπεξεργασία, ανάλυσης, εκπαίδευσης και σύνθεσης είναι διαθέσιμος στην σελίδα https://github.com/geofila/Music-Generation ενώ και ο κώδικας της σελίδας αξιολόγησης και σύνθεσης βρίσκεται στο παρακάτω link: https://github.com/geofila/site.

Βιβλιογραφία

[1] Cyclical Learning Rates for Training Neural Networks, Leslie N. SmithU.S. Naval Research Laboratory. Link:
https://arxiv.org/pdf/1506.01186.pdf

[2] Adam: A Method for Stochastic Optimization, Diederik P. Kingma University of Amsterdam, Jimmy Lei Ba University of Toronto. Link:
https://arxiv.org/pdf/1412.6980v8.pdf

[3] Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever,Ruslan Salakhutdinov. Lin
http://jmlr.org/papers/volume15/srivastava14a.old/srivastava14a.pdf

[5] Understanding the difficulty of training deep feedforward neural networks, Xavier Glorot, Yoshua Bengio. Link:
http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf

[6] Keras: How to use Glorot Intializer. Link:
https://keras.io/initializers/#glorot_uniform

[7] Deep Learning Best Practices (1) — Weight Initialization, Neerja Doshi in USF-Data Science. Link:
https://medium.com/usf-msds/deep-learning-best-practices-1-weight-initialization-14e5c0295b94

[8] Tips for Training Recurrent Neural Networks, Danijar Hafner. Link:
https://danijar.com/tips-for-training-recurrent-neural-networks/

[9]Long Short-Term Memory, Sepp Hochreiter University at Munchen, Jurgen Schmidhuber. Link:
https://www.researchgate.net/publication/13853244_Long_Short-term_Memory

[10] Recurrent Neural Networks Tutorial, Part 1 – Introduction to RNNs. Link:
http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/

[11] Recurrent Neural Networks Tutorial, Part 3 – Backpropagation Through Time and Vanishing Gradients. Link:
http://www.wildml.com/2015/10/recurrent-neural-networks-tutorial-part-3-backpropagation-through-time-and-vanishing-gradients/

[12] On the difficulty of training recurrent neural networks, Razvan Universit'e de Montr´ea, Tomas Mikolov, Yoshua Bengio Universit´e de Montr´ea. Link:
http://proceedings.mlr.press/v28/pascanu13.pdf

[13] Long Short-Term Memory, Sepp Hochreiter University at Munchen, Jurgen Schmidhuber. Link:
http://www.bioinf.jku.at/publications/older/2604.pdf

[14] Music transcription modelling and composition using deep learning, Bob L. Sturm, Jo˜ao Felipe Santos, Oded Ben-Tal, Iryna Korshunova. Link:
https://arxiv.org/pdf/1604.08723.pdf

[15] DeepBach: a Steerable Model for Bach Chorales Generation, Ga¨etan Hadjeres, Francois Pachet, Frank Nielsen. Link:
https://arxiv.org/abs/1612.01010

[16] Neural machine translation by jointly learning to align and translate, Dzmitry Bahdanau Jacobs University Bremen, Germany, Kyung Hyun Cho, Yoshua Bengio Universite de Montreal. Link:
https://arxiv.org/pdf/1409.0473.pdf

[17] MIDI Music Data Extraction using Music21 and Word2Vec on Kaggle. Link:
https://towardsdatascience.com/midi-music-data-extraction-using-music21-and-word2vec-on-kaggle-cb383261cd4e

[18] Donya Quick.Kulitta: A Framework for Automated Music Composition. PhD thesis, Yale University, 2014.

[19] Keras Usage of Optimizers, Link:
https://keras.io/optimizers

[20] Overview of mini-batch gradient descent, Geoffrey Hinton, Nitish Srivastava, Kevin Swersky, Link:
http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf