Εθνικό Μετσόβιο Πολυτεχνείο
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Αναγνώριση Προτύπων
Ροή Σ: Σήματα, Έλεγχος, Ρομποτική
9ο Εξάμηνο

2η Εργαστηριακή Άσκηση:
Αναγνώριση φωνής με Κρυφά Μαρκοβιανά Μοντέλα και Αναδρομικά Νευρωνικά Δίκτυα

Χρήστος Δημόπουλος - 03117037
chrisdim1999@gmail.com

Δεκέμβριος, 2021

Contents

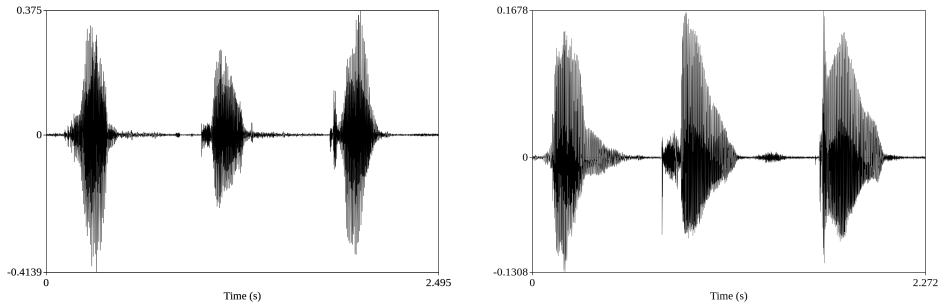
Θέμα Εργαστηριακής Άσκησης

Σκοπός της δεύτερης εργαστηριακής άσκησης είναι η υλοποίηση ενός συστήματος επεξεργασίας και αναγνώρισης φωνής, με εφαρμογή σε αναγνώριση μεμονωμένων λέξεων. Το πρώτο μέρος αποσκοπεί στην εξαγωγή κατάλληλων ακουστικών χαρακτηριστικών από φωνητικά δεδομένα, χρησιμοποιώντας τα κατάλληλα πακέτα python, καθώς και στην ανάλυση και απεικόνισή τους με σκοπό την κατανόηση και την εξαγωγή χρήσιμων πληροφοριών από αυτά. Τα εν λόγω χαρακτηριστικά είναι στην ουσία ένας αριθμός συντελεστών cepstrum που εξάγονται μετά από ανάλυση των σημάτων με μια ειδικά σχεδιασμένη συστοιχία φίλτρων (filterbank). Η συστοιχία αυτή είναι εμπνευσμένη από ψυχοακουστικές μελέτες.
Πιο συγκεκριμένα, το σύστημα που θα αναπτύξουμε αφορά σε αναγνώριση μεμονωμένων ψηφίων (isolated digits) στα Αγγλικά. Τα δεδομένα που θα χρησιμοποιήσουμε περιέχουν εκφωνήσεις 9 ψηφίων από 15 διαφορετικούς ομιλητές σε ξεχωριστά .wav αρχεία. Συνολικά διατίθενται 133 αρχεία, αφού 2 εκφωνήσεις θεωρήθηκαν προβληματικές και δεν έχουν συμπεριληφθεί. Τα ονόματα των αρχείων (π.χ. eight8.wav) υποδηλώνουν τόσο το ψηφίο που εκφωνείται (π.χ. eight), όσο και τον ομιλητή (οι ομιλητές είναι αριθμημένοι από 1-15). Οι εκφωνήσεις έχουν ηχογραφηθεί με συχνότητα δειγματοληψίας Fs = 16kHz και η διάρκειά τους διαφέρει.

Βήματα Προπαρασκευής

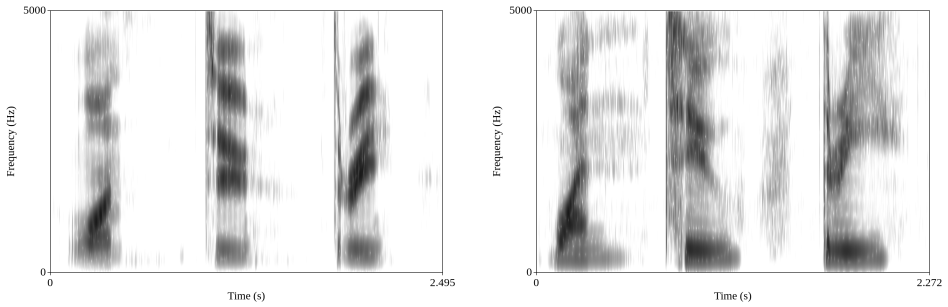
1. Ανάλυση αρχείων ήχου με Praat

Αρχικά, αναλύουμε τα αρχεία ήχου onetwothree1.wav και onetwothree8.wav με το πρόγραμμα Praat [**?**]. Τα αρχεία αυτά περιέχουν την πρόταση “one two three” από τους ομιλητές 1 και 8, οι οποίοι είναι άντρας και γυναίκα αντίστοιχα. Στα Σχήματα 1 και 2 φαίνονται οι κυματομορφές ήχου και τα spectograms αντίστοιχα για τα δύο αρχεία ήχου.



(a) (b)

Figure 1: Κυματομορφές ήχου των αρχείων (a) ’onetwothree1.wav’ και (b) ’onetwothree8.wav’.



(a) (b)

Figure 2: Spectograms των αρχείων (a) ’onetwothree1.wav’ και (b) ’onetwothree8.wav’.

Ακολούθως, εξάγουμε τη **μέση τιμή του pitch** για τα φωνήεντα 'α', 'ου' και 'ι' για τα 3 ψηφία και για κάθε ομιλητή και τα καταγράφουμε στον Πίνακα 1. Όπως ήταν αναμενόμενο, παρατηρούμε ότι στην εκφώνηση ψηφίων από γυναίκα το μέσο pitch των φωνηέντων είναι μεγαλύτερο από το αντίστοιχο στην εκφώνηση από άντρα. Συνεπώς, το pitch συνιστά ικανοποιητικό κριτήριο για την κατηγοριοποίηση των εκφωνήσεων με βάση το φύλο του ομιλητή. Ωστόσο το pitch δεν δύναται να κατηγοριοποίησει τις εκφωνήσεις με βάση το ψηφίο που εκφωνείται, καθώς στην περίπτωση αυτή παρατηρούμε παρεμφερείς τιμές.

.wav File	Gender	Digit	Vowel	Mean Pitch (Hz)
onetwothree1.wav	Male	1	α	133.88
onetwothree8.wav	Female	1	α	180.08
onetwothree1.wav	Male	2	ου	132.08
onetwothree8.wav	Female	2	ου	187.99
onetwothree1.wav	Male	3	ι	132.39
onetwothree8.wav	Female	3	ι	178.64

Table 1: Μέση Τιμή Pitch ανά φωνήεν ανά εκφωνητή.

Στη συνέχεια, εξάγουμε τα **3 πρώτα formants**, δηλαδή τα τρία πρώτα τοπικά μέγιστα του φάσματος του ηχητικού σήματος, κάθε φωνήεντος για τους δύο ομιλητές. Συγκεντρώνοντας τις τιμές των formants στον Πίνακα 2, παρατηρούμε ότι αποτελούν μια καλύτερη μετρική για τη κατηγοριοποίηση ήχου με βάση το εκφωνούμενου ψηφίου. Αυτό συμβαίνει διότι, ίδια φωνήεντα εμφανίζουν formants σε πολύ κοντινές συχνότητες, ανεξαρτήτως το φύλο του εκφωνητή. Για τον λόγο αυτό, η υλοποίηση του ASR συστήματος μας θα στηριχθεί σε μεγάλο βαθμό στα τοπικά μέγιστα του φάσματος ήχου.

.wav File	Gender	Digit	Vowel	F1 (Hz)	F2 (Hz)	F3 (Hz)
onetwothree1.wav	Male	1	α	698.4	1119.25	2365.46
onetwothree8.wav	Female	1	α	646.50	1667.39	2900.75
onetwothree1.wav	Male	2	ου	351.74	1787.73	2423.83
onetwothree8.wav	Female	2	ου	378.64	1835.13	2741.23
onetwothree1.wav	Male	3	ι	388.98	1818.53	2229.29
onetwothree8.wav	Female	3	ι	341.78	2085.45	2760.96

Table 2: Τα 3 πρώτα formants ανά φωνήεν ανά εκφωνητή.

2. Δημιουργία Data Parser

Ως μέρος του preprocessing, κατασκευάζουμε μια συνάρτηση data parser, η οποία δέχεται ως είσοδο όλα τα αρχεία ήχου που δίνονται μέσα στο φάκελο digits/ και επιστρέφει τις εξής τρεις λίστες:

- **wav**: λίστα που περιλαμβάνει τα δεδομένα των σημάτων ήχου, όπως διαβάστηκαν από τη librosa.
- **speaker**: λίστα που περιλαμβάνει τον εκφωνητή κάθε αρχείου.
- **digit**: λίστα που περιλαμβάνει το ψηφίο που εκφωνείται.

3. Εξαγωγή Χαρακτηριστικών: MFCCs, Deltas, Delta-Deltas

Στη συνέχεια, χρησιμοποιώντας μήκος παραθύρου 25 ms και βήμα 10 ms, εξάγουμε 13 Mel-Frequency Cepstral Coefficients (**MFCCs**) ανά αρχείο, καθώς και την πρώτη και δεύτερη τοπική παράγωγο των χαρακτηριστικών, τις λεγόμενες **deltas** και **delta-deltas**.

4. Δημιουργία Ιστογραμμάτων και Σύγκριση MFCCs - MFSCs

Έχοντας συγκεντρώσει τα ηχητικά χαρακτηριστικά για κάθε αρχείο, αναπαριστούμε τα ιστογράμματα του 1ου και του 2ου MFCC των ψηφίων **n1 = 3** και **n2 = 7** για όλες τους τις εκφωνήσεις (Σχήμα 4). Παρατηρώντας τα, καταλήγουμε στο συμπέρασμα ότι η απόκλιση των ιστογραμμάτων ανάμεσα στα ψηφία 3 και 7 δεν είναι σημαντική, τουλάχιστον για τους πρώτους 2 MFCCs. Επομένως, θα χρειαστεί να χρησιμοποιήσουμε και τους 13 συντελεστές στο ASR σύστημα που χτίζουμε, ώστε να συμπεριλαβουμε τις υψίσυχνες λεπτομέρειες των μεγαλύτερων συντελεστών ως ειδοποιούς διαφορές των ψηφίων που εκφωνούνται.

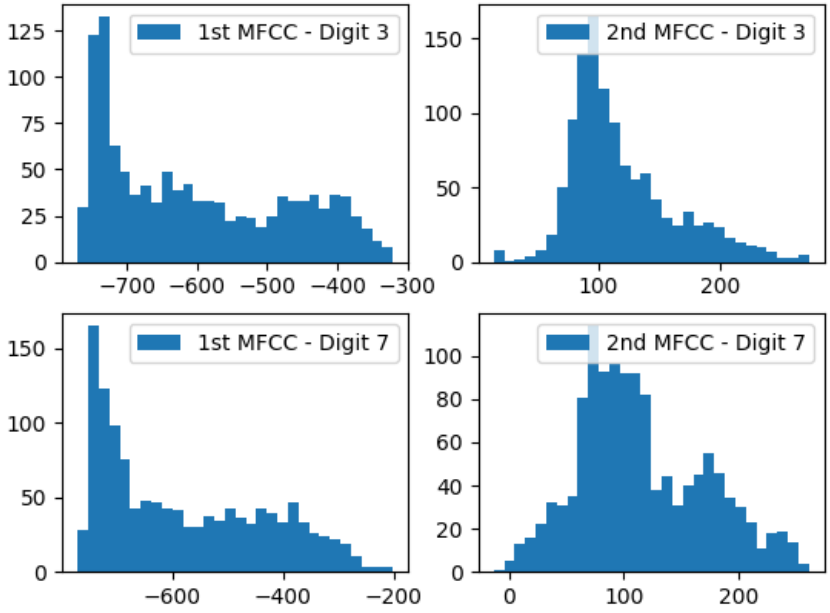
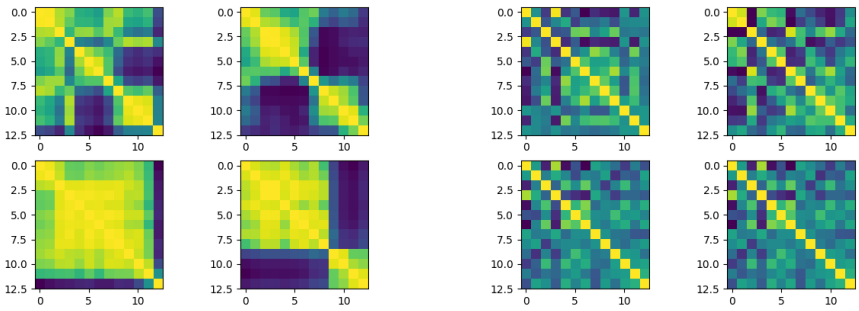


Figure 3: Ιστογράμματα των δύο πρώτων MFCCs για τα ψηφία 3 και 7.

.

Σε δεύτερο στάδιο, εξάγουμε για 2 εκφωνήσεις των ψηφίων 3 και 7 από δύο διαφορετικούς ομιλητές τα **Mel Filterbank Spectral Coefficients (MFSCs)**, δηλαδή τα χαρακτηριστικά που εξάγονται αφού εφαρμοστεί η συστοιχία φίλτρων της κλίμακας Mel πάνω στο φάσμα του σήματος φωνής, αλλά χωρίς να εφαρμοστεί στο τέλος ο μετασχηματισμός DCT. Ομοίως με πριν εξάγουμε 13 το πλήθος συντελεστές. Αναπαριστώντας γραφικά τη συχέτιση των MFSCs και των MFCCs (Σχήμα 4) για την κάθε εκφώνηση, παρατηρούμε ότι **οι MFCCs είναι σημαντικά λιγότερο correlated σε σχέση με τους MFSCs**, γι'αυτό και προτιμώνται στο πρόβλημα αναγνώρισης ψηφίων από σήματα φωνής.



(a) (b)

Figure 4: Συσχέτιση των 13 συντελεστών (α) MFSCs, (β) MFCCs.

5. Εξαγωγή Μοναδικού Διανύσματος Χαρακτηριστικών

Στο συγκεκριμένο βήμα, προκειμένου να διαμορφώσουμε το feature space του προβλήματος ASR, καλούμαστε να εξάγουμε ένα μοναδικό διάνυσμα χαρακτηριστικών για κάθε εκφώνηση. Για τον σκοπό αυτό, ορίζουμε ως feature vector τη συνένωση μέσης τιμής και τυπικής απόκλισης των MFCCs, deltas και delta-deltas για όλα τα παράθυρα της εκφώνησης:

$$X_i = [\mu_{MFCCs}|\sigma_{MFCCs}|\mu_{Deltas}|\sigma_{Deltas}|\mu_{Delta-Deltas}|\sigma_{Delta-Deltas}]^T \quad (1)$$

Αναπαριστώντας στο επίπεδο τις 2 πρώτες διαστάσεις των διανυσμάτων αυτών (Σχήμα 5), παρατηρεί κανείς ότι **οι περιοχές απόφασεις για τα 9 ψηφία δεν είναι εύκολα διακρίσιμες**. Κάτι τέτοιο είναι αναμενόμενο, καθώς η τακτική να κρατήσουμε μόνο τις δύο πρώτες διαστάσεις των διανυσμάτων χαρακτηριστικών είναι αρκετά αφελής από μόνη της.

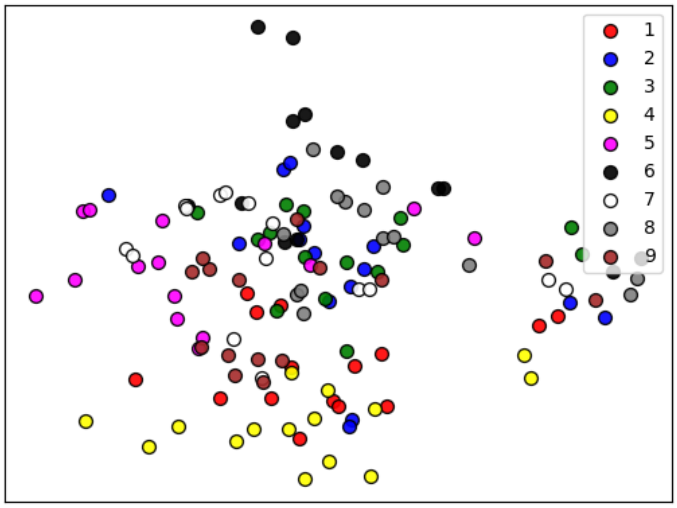


Figure 5: Scatter Plot των δειγμάτων στο επίπεδο κρατώντας μόνο τις πρώτες 2 διαστάσεις.

.

6. Τεχνικές Μείωσης Διαστατικότητας (PCA)

Δεδομένου ότι η διάσταση των feature vectors που δημιουργήσαμε στο προηγούμενο βήμα είναι αρκετά μεγάλη, καταφεύγουμε σε τεχνικές μείωσης διαστατικότητας με **Principal Component Analysis**. Μειώνοντας τις διαστάσεις των διανυσμάτων σε 2 και 3 (Σχήμα 6 και 7 αντίστοιχα), παρατηρούμε ότι οι περιοχές απόφασης για τα 9 ψηφία είναι σίγουρα πιο ευδιάκριτες σε σχέση με πριν, ωστόσο και πάλι δεν είναι σαφώς καθορισμένες. Με άλλα λόγια, το σφάλμα κατηγοριοποίησης βελτιώνεται, αλλά όχι σε ικανοποιητικό βαθμό.

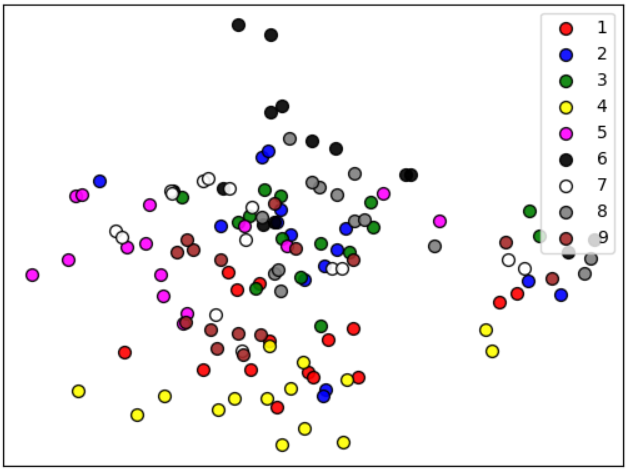


Figure 6: Scatter Plot των δειγμάτων στο επίπεδο μετά από 2D-PCA.

.

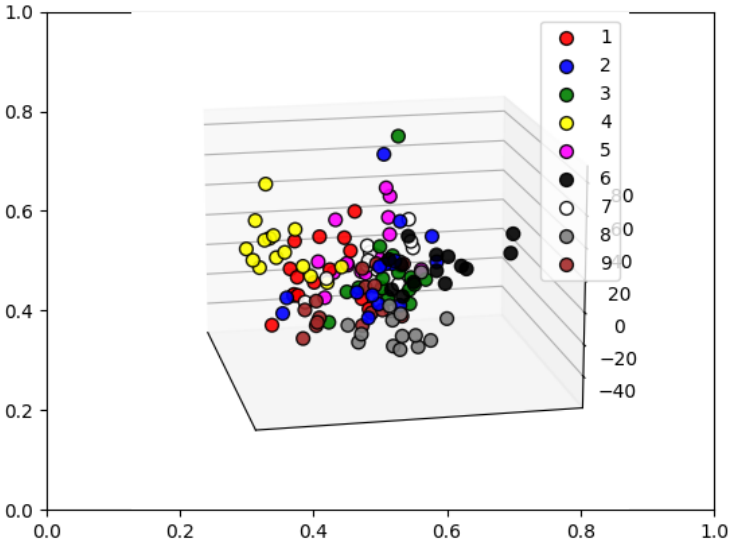


Figure 7: Scatter Plot των δειγμάτων στον χώρο μετά από 3D-PCA.

.

Η επίδραση των τεχνικών PCA γίνεται καλύτερα αντιληπτή αν αναλύσουμε το ποσοστό της αρχικής διασποράς που διατηρούν οι συνιστώσεις που προέκυψαν (Πίνακας 3). Παρατηρούμε ότι διατηρώντας 2 principal components εξηγούμε κατά μέγιστο το 70.66% της διασποράς των αρχικών δεδομένων, κάτι το οπόιο δεν είναι ιδιαίτερο ικανοποιητικό. Η κατάσταση βελτιώνεται αν διατηρήσουμε 3 διαστάσεις με PCA, οπόταν αγγίζουμε ποσοστά διασποράς της τάξεως του 80%. Παρόλα αυτά, εφαρμόζοντας τεχνικές PCA για τόσο μικρές διαστάσεις χάνεται ένα αισθητό ποσοστό πληροφορίας, γεγονός που θα δυσχεραίνει στη συνέχεια το ASR σύστημα μας με φαινόμενα overfitting.

PCA Dimensionality	1st Component Variance Ratio	2nd Component Variance Ratio	3rd Component Variance Ratio	Total Variance Ratio
2	58.8%	11.86%	-	70.66%
3	58.8%	11.86%	10.83%	81.49%

Table 3: PCA Variance Ratio per Component.

7. Train-Test Διαχωρισμός και Ταξινόμηση

Ως επόμενο βήμα, διαχωρίζουμε τα συνολικά δεδομένα σε train-test με αναλογία 70%-30% και προβαίνουμε στην κατηγοριοποίηση τους με supervised learning ταξινομητές. Για τον σκοπό αυτό, κανονικοποιούμε τα δεδομένα, ώστε να λαμβάνουν τιμές στο διάστημα [0,1].
Ακολούθως, χρησιμοποιούμε ταξινομητές επιβλεπόμενης μάθησης στο πρόβλημα αναγνώρισης ψηφίων με είσοδο τα μοναδικά διανύσματα χαρακτηριστικών που δημιουργήσαμε. Στον πίνακα 4 φαίνονται τα ποσοστά ευστοχίας του κάθε ταξινομητή πάνω στο test set. Παρατηρούμε ότι τις καλύτερες επιδόσεις λαμβάνουμε με τον Naive Bayes, για χαμηλό smooth parameter ώστε να μην εκφυλλίζεται σε Ταξινομητή Ευκλείδιας Απόστασης, ενώ σχετικά καλά αποτελέσματα δίνει και ο ταξινομητής KNN. Αντιθέτως, οι ταξινομητές LR, Euclidean Distance και SVM φαίνεται να αποτυγχάνουν παταγωδώς.

Classifier	Test Accuracy (%)
Custom NB - No Unit Variance (smooth = 1e-5)	47.5
Custom NB - Unit Variance	5
sklearn NB (smooth = 1e-9)	40
SVM - Linear Kernel	5
SVM - Polynomial Kernel	5
SVM - rbf kernel	5
KNN - 3 neighbors	40
KNN - 5 neighbors	50
Logistic Regression	12.5

Table 4: Test Accuracy per Classifier.

Σημείωση: Προκειμένου ο Naive Bayes ταξινομητής να μπορεί να ταξινομήσει το εν λόγω dataset που δημιουργήσαμε, χρειάστηκε να κάνουμε κάποιες προσαρμογές στον κώδικα *lib.py* του 1ου εργαστηρίου. Οι αλλαγές αυτές οφείλονται στο γεγονός, ότι δεν έχουμε 10 κλάσεις στο classification problem, αλλά 9, καθώς δεν εκφωνείται σε καμία περίπτωση το ψηφίο μηδέν.

Ως πιθανή βελτίωση του συστήματος, ενσωματώνουμε στα διανύσματα χαρακτηριστικών του dataset το **zero crossing rate** κάθε σήματος ήχου. Ειδικότερα, περιοριζόμαστε στα 50 παράθυρα για κάθε δείγμα, ώστε να έχουμε διανύσματα ίδιας διάστασης. Επαναλαμβάνοντας την ταξινόμηση πάνω στα νέα feature vectors μεγαλύτερης διαστατικότητας (Πίνακας 5), παρατηρούμε ότι τα αποτελέσματα ευστοχίας στο test set κατά βάση **βελτιώνονται**, με τη μεγαλύτερη αύξηση να εμφανίζεται στην περίπτωση του Naive Bayes. Παρόλα αυτά, κάποιοι ταξινομητές, όπως ο SVM και ο LR, εξακολουθούν να δίνουν απογοητευτικά αποτελέσματα.

Classifier	Test Accuracy (%)
Custom NB - No Unit Variance (smooth = 1e-5)	65
Custom NB - Unit Variance	12.5
sklearn NB (smooth = 1e-9)	**67.5**
SVM - Linear Kernel	12.5
SVM - Polynomial Kernel	12.5
SVM - rbf kernel	12.5
KNN - 3 neighbors	47.5
KNN - 5 neighbors	47.5
Logistic Regression	15

Table 5: Test Accuracy per Classifier (with Zero Crossing Rate).

Ένας πιθανός λόγος για τον οποίο ο ταξινομητής SVM αποτυγχάνει μπορεί να είναι το γεγονός ότι το dataset που δημιουργήσαμε είναι imbalanced, και αποτελείται από πάρα πολλά δεδομένα. Ο Logistic Regression από την άλλη, αποτυγχάνει καθώς υποθέτει πως το πρόβλημα είναι γραμμικά διαχωρίσιμο, ενώ γνωρίζουμε ότι δεν ενδείκνυται τόσο για multiclass problems, στα αποδίδει πολύ περισσότερο ένας Naive Bayes ταξινομητής. Τέλος, η βελτίωση της ευστοχίας του NB με την προσθήκη των zero-crossing rates ίσως οφείλεται στο γεγονός ότι κάνουμε mapping τα διανύσματα εισόδου σε χώρους υψηλότερης διάστασης, και έτσι τα δείγματα γίνονται περισσότερο disentangled μέσω των Ευκλείδιων αποστάσεων.
Στη γενικότερη περίπτωση, βέβαια, η παραπάνω προσέγγιση για την ταξινόμηση ηχητικών σημάτων δεν ενδείκνυται. Αυτό συμβαίνει διότι, συνενώνοντας ακουστικά χαρακτηριστικά ως μοναδικά διανύσματα - features, **δεν εκμεταλλευόμαστε τη χρονική αλληλουχία** που διέπει τα ακουστικά σήματα, και έτσι το ASR σύστημα θα υπολειτουργεί.

8. Πρόβλεψη Συνημιτόνων με RNNs/LSTMs

Ως τελευταίο προπαρασκευαστικό βήμα, εξοικειωνόμαστε με τα Αναδρομικά Νευρωνικά Δίκτυα και τις παραλλαγές τους σε ένα πρόβλημα κατηγοριοποίησης ημιτονοειδών συναρτήσεων. Πιο συγκεκριμένα, δημιουργούμε ακολουθίες 10 σημείων ενός ημιτόνου και ενός συνημιτόνου με συχνότητα **f = 40 Hz**. Σκοπός είναι η πρόβλεψη του συνημιτόνου, με δεδομένη την ακολουθία του ημιτόνου. Επιλέγουμε σταθερή και μικρή απόσταση ανάμεσα στα διαδοχικά σημεία **Δx = 0.001**.
Αρχικά, δημιουγούμε το dataset του άνωθι προβλήματος με 1000 ζεύγη ημιτόνων - συνημιτόνων. Οι ημιτονοειδές αυτές συναρτήσεις έχουν ως πλάτος A μια τυχαία μεταβλητή ομοιόμορφα κατανεμημένη στο διάστημα [0,10] και σημείο εκκίνησης t0, το οποίο είναι επίσης TM ομοίομρφα κατανεμημένη στο διάστημα [0,T] sec. Στο Σχήμα 8 φαίνονται τα πρώτα οκτώ ζεύγη ημιτόνων-συνημιτόνων του dataset που δημιουργούμε.

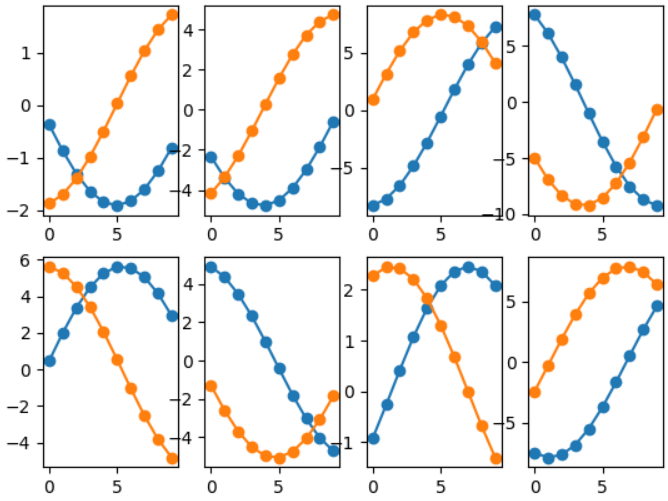
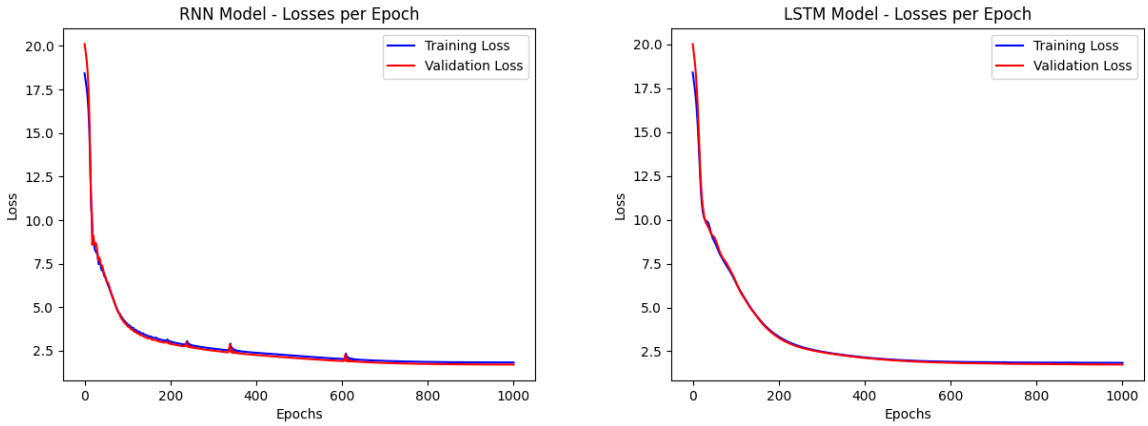


Figure 8: Τα 8 πρώτα τυχαία δείγματα ημιτόνων (μπλε) και των ground truth συνημιτόνων τους (πορτοκαλί).

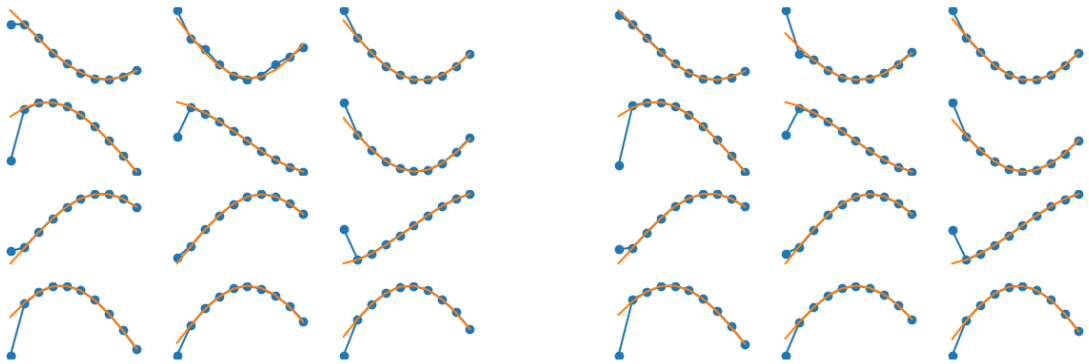
.

Αφού διαχωρίσουμε το σύνολο ζευγών ημιτονοειδών συναρτήσεων σε train και test set με αναλογία 70/30, εκπαιδεύουμε ένα **Αναδρομικό Νευρωνικό Δίκτυο (Recurrent Neural Network – RNN)**, το οποίο θα δέχεται ως είσοδο τις ακολουθίες του ημιτόνου και προβλέπει τις αντίστοιχες ακολουθίες συνημιτόνου. Ως συνάρτηση κόστους χρησιμοποίουμε το **Mean Square Error**, η βελτιστοποίηση γίνεται με **Adam optimizer** ρυθμού μάθησης 0.001, ενώ η εκπαίδευση του δικτύου πραγματοποιείται για **1000 εποχές**.
Για τον ίδιο σκοπό, εκπαιδεύουμε και ένα **LSTM** αναδρομικό νευρωνικό δίκτυο, με τις ίδιες τιμές υπερπαραμέτρων. Στο Σχήμα 9 φαίνεται η εξέλιξη του train και validation Loss ανά τις 1000 εποχές εκπαίδευσης, ενώ στο Σχήμα 10 φαίνονται τα εκτιμώμενα συνημίτονα των πρώτων 12 συναρτήσεων σε κοινό διάγραμμα με το Ground Truth, τόσο για το απλό RNN όσο και για το LSTM δίκτυο.
Παρατηρούμε ότι και τα δύο δίκτυα εκπαιδεύονται επαρκώς, δίχως προβλήματα overfitting, καθώς καταφέρνουν να προβλέψουν με τεράστια ακρίβεια τα σημεία των συνημιτόνων του dataset. Μοναδική παρέκκλιση αποτελεί προφανώς το πρώτο σημείο κάθε συνημιτόνου, γεγονός αναμενόμενο, καθώς τα Αναδρομικά Δίκτυα δεν έχουν στην μνήμη τους καμία πληροφορία στο στάδιο της 'χρονικής' αρχικοποίησης.



(a) (b)

Figure 9: Train και Test Loss ανά τις 1000 εποχές εκπαίδευσης για το (α) RNN, (β) LSTM μοντέλο.



(a) (b)

Figure 10: Εκτιμώμενα συνημίτονα των πρώτων 12 συναρτήσεων με το (α) RNN, (β) LSTM μοντέλο.

Πλέον δεν χρησιμοποιείται σχεδόν καθόλου το απλό RNN αλλά παραλλαγές του, οι οποίες επιχειρούν να ξεπεράσουν ορισμένα προβλήματα, με σημαντικότερο αυτό του vanishing gradient κατά την διάρκεια της εκπαίδευσης του δικτύου [**?**]. Η πιο δημοφιλής παραλλαγή είναι το δίκτυο Long Short-Term Memory (LSTM) [**?**].
Χρησιμοποιεί έναν εξεζητημένο μηχανισμό, ο οποίος του επιτρέπει να ξεπεράσει το πρόβλημα του RNN, σχετικά με την αναγνώριση απομακρυσμένων εξαρτήσεων. Το LSTM έχει δύο βασικές διαφορές από το απλό RNN:

- **Δεν εφαρμόζει συνάρτηση ενεργοποίησης στις αναδρομικές συνδέσεις**. Αυτό σημαίνει ότι οι ενημερώσεις θα είναι γραμμικές. Ετσι εγγυάται ότι τα σφάλματα (gradients), δεν θα εξαφανίζονται από την επαναληπτική εφαρμογή των ενημερώσεων (backpropagation through-time). Συνεπώς εξασφαλίζει τη ροή της πληροφορίας στο δίκτυο.
- **Μηχανισμός με θύρες**. Ο μηχανισμός αυτός εισάγει θύρες, οι οποίες ρυθμίζουν το πόσο θα ενημερώνεται κάθε διάνυσμα του δικτύου (εσωτερική κατάσταση, έξοδος κλπ.). Με αυτό τον τρόπο το δίκτυο αφομοιώνει και διατηρεί τις πιο σημαντικές πληροφορίες καλύτερα.

Βήματα Κυρίως Μέρους

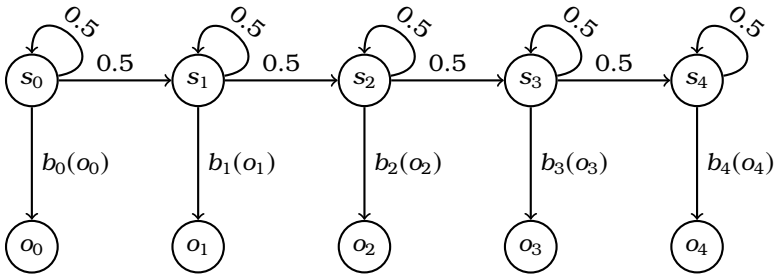
Για το κυρίως μέρος της άσκησης θα χρησιμοποιηθεί ένα μεγαλύτερο σετ δεδομένων, το **Free Spoken Digit Dataset (FSDD)** [**?**]. Το εν λόγω dataset αποτελείται από ηχογραφήσεις εκφωνήσεων ψηφίων σε .wav αρχεία με συχνότητα δειγματοληψίας 8kHz. Ειδικότερα, περιλαμβάνει 3,000 εκφωνήσεις (50 για κάθε ψηφίο ανά ομιλητή), οι οποίες εκφέρονται από 6 διαφορετικούς ομιλητές. Επιπλέον, οι ηχογραφήσεις είναι προσαρμοσμένες, ώστε να μην επικρατεί σιωπή στην αρχή και στο τέλος τους.
Η κατηγοριοποίηση των αρχείων ήχου με βάση το εκφωρούμενο ψηφίο, σε πρώτο στάδιο, θα πραγματοποιηθεί με χρήση GMM-HMMs και έπειτα με χρήση Αναδρομικών Νευρωνικών Δικτύων (RNNs).

9. Δημιουργία των train, validation και test sets.

Σε πρώτο στάδιο, προετοιμάζουμε διαβάζουμε τα δεδομένα του προβλήματος με χρήση του βοηθητικού κώδικα *parser()*, ο οποίος πραγματοποιεί **feature extraction με 6 MFCCs** ανά frame και διαχωρίζει το σύνολο των δεδομένων σε train και test. Ακολούθως, διαχωρίζουμε τα train δεδομένα σε training και validation set με ποσοστό 80%-20%, με τρόπο ώστε να διατηρηθεί ίδιος ο αριθμός των διαφορετικών ψηφίων σε κάθε set (**stratified split**). Έπειτα κανονικοποιούμε όλα τα δεδομένα με χρήση ενός **Standard Scaler**, βάσει της μέσης τιμής και τυπικής απόκλισης του train set και μόνο.

10. Αναγνώριση ψηφίων με GMM-HMMs.

Ως πρώτη προσέγγιση, κατηγοριοποιούμε τα άνωθι δεδομένα με χρήση GMM-HMMs (Gaussian Mixture Models – Hidden Markov Models). Για κάθε ένα εκ των 10 ψηφίων - κλάσεων, αρχικοποιούμε ένα μοντέλο GMM-HMM της μορφής left-right. Συγκεκριμένα, αν $A = \{a_{ij}\}$ είναι ο πίνακας μεταβάσεων του μοντέλου, τότε $a_{ij} = 0$ για $j < i$, ενώ οι αρχικές πιθανότητες των καταστάσεων είναι: $\pi_i = \{0 \text{ για } i \neq 1 \text{ και } 1 \text{ για } i = 1\}$. Επιπλέον επιτρέπονται μεταβάσεις μόνο μεταξύ διαδοχικών καταστάσεων, δηλαδή υπάρχει ο περιορισμός $a_{ij} = 0$ για $j > i + 1$. Η πιθανότητα για κάθε υπαρκτή μετάβαση ορίζεται σε 1/2.



Παράδειγμα γράφου μεταβάσεων για ένα GMM-HMM 5 καταστάσεων.

Ένα διάνυσμα ακουστικών χαρακτηριστικών, όπως αυτό που εξάγεται από την επεξεργασία ενός πλαισίου φωνής, αποτελεί μια πιθανή παρατήρηση σε κάποια κατάσταση. Λόγω του ότι είναι επιτρεπτές συνεχείς μεταβολές τέτοιων παρατηρήσεων, η πιθανότητα τους μοντελοποιείται με ένα μίγμα Γκαουσιανών κατανομών (GMM).

11. Αλγόριθμος Εκπαίδευσης Baum–Welch

Έχοντας αρχικοποίησει ένα GMM-HMM για κάθε ένα από τα 10 ψηφία, προβαίνουμε στην εκπαίδευση τους με τον **αλγόριθμο Baum-Welch**, ο οποίος υπάγεται στην κατηγορία των Expectation-Maximization αλγορίθμων []. Για τον σκοπό αυτό, διαχωρίζουμε το train set με βάση τα διαθέσιμα δεδομένα ανά ψηφίο, ώστε να εκπαιδεύσουμε καθένα από τα 10 μοντέλα ξεχωριστά. Ο αλγόριθμος εφαρμόζεται για καθορισμένο πλήθος επαναλήψεων N_{iter} ή εως να υπάρξει σύγκλιση. Η σύγκλιση ελέγχεται μέσω της μεταβολής του αλγορίθμου της πιθανοφάνειας (Log Likelihood, πιθανότητα των δεδομένων με γνωστό μοντέλο).
Ως κατώφλι σύγκλισης επιλέγουμε την default τιμή 10^{-9}, ενώ για την υπερπαράμετρο των μέγιστων επαναλήψεων

N_{iter} πειραματιζόμαστε με τις τιμές 5,10,15,20,30. Τέλος, χρησιμοποιούμε από 1 έως 4 καταστάσεις HMM και από 1 έως 5 Γκαουσιανές κατανομές.

12. Testing - Inference Time

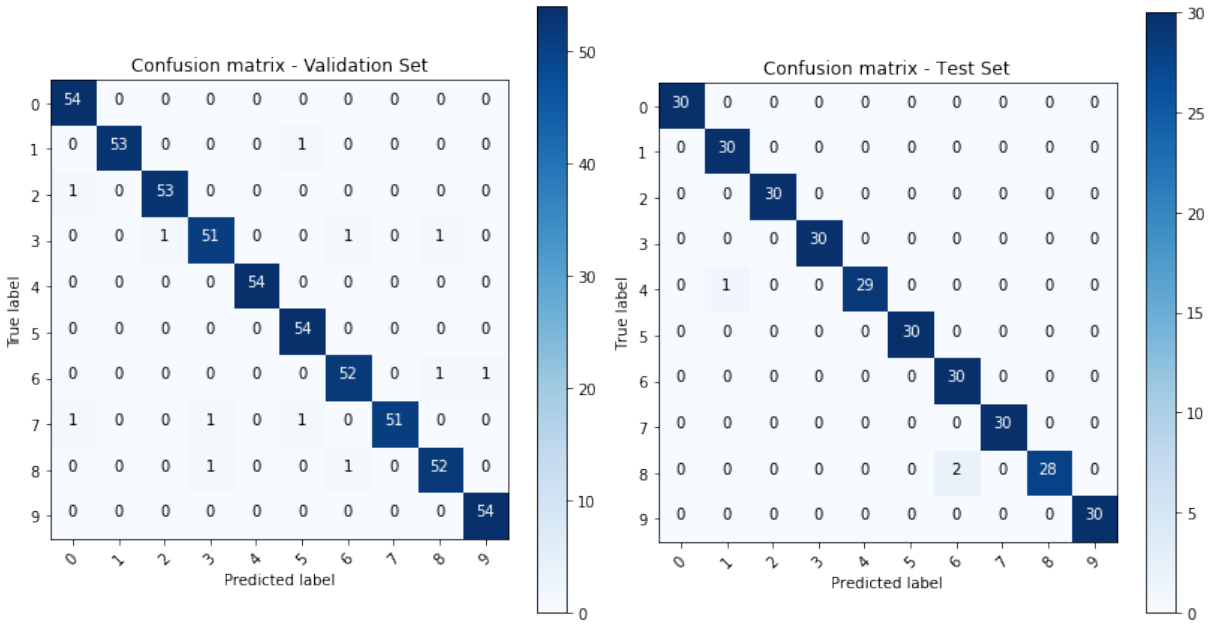
Έχοντας ολοκληρώσει τη διαδικασία της εκπαίδευσης, καταλήγουμε στις εκτιμήσεις των παραμέτρων των 10 μοντέλων (δηλαδή ένα μοντέλο για κάθε ψηφίο). Στη συνέχεια, υπολογίζεται ο λογάριθμος της πιθανοφάνειας (log likelihood) για κάθε εκφώνηση η οποία ανήκει στο σύνολο των δεδομένων για αναγνώριση. Το μοντέλο το οποίο δίνει τη μέγιστη πιθανοφάνεια είναι και το αποτέλεσμα της αναγνώρισης για τη συγκεκριμένη εκφώνηση (**Maximum Log-Likeihood Estimator**).
Σε πρώτο στάδιο, πραγματοποιούμε την άνωθι διαδικασία πάνω στο validation set, πειραματιζόμενοι με τις τιμές των υπερπαραμέτρων, όπως αυτές ορίστηκαν παραπάνω. Τα μοντέλα που δίνουν το βέλτιστο ποσοστό ευστοχίας στο validation set προκύπτουν από τον κάτωθι συνδυασμό υπερπαραμέτρων:

- Number of Hidden States = 4
- Number of Gaussian Mixtures = 5
- Maximum Number of Iterations = 30

Η διαδικασία του hyperparameter tuning με βάση το validation set ενδείκνυται, καθώς αποφεύγονται φαινόμενα **over-fitting**. Δηλαδη, οι βέλτιστες υπερπαράμετροι επιλέγονται από ένα εύρος πιθανών τιμών, ώστε το προς εκπαίδευση μοντέλο να μην εξειδικεύεται αποκλειστικά πάνω στο train set, αλλά να μπορεί να γενικεύει και σε δεδομένα που δεν έχει δει (**generalization ability**). Μόλις οι βέλτιστες τιμές υπερπαραμέτρων έχουν προσδιοριστεί, η επίδοση του μοντέλου μπορεί πλέον να ελεγχθεί πάνω στο test set.

13. Ποσοστό Ευστοχίας και Confusion Matrices

Για τις παραπάνω τιμές υπερπαραμέτρων, το **ποσοστό ευστοχίας** στο validation set είναι **97.78%**, ενώ στο test set αγγίζει το **99%**. Παρατηρώντας τους confusion matrices (Σχήμα 11), φαίνεται ότι τα βέλτιστα GMM-HMMs κατηγοριοποιούν εσφαλμένα 12 ψηφία στο validation set και μόλις 3 ψηφία στο test set.



(a) (b)

Figure 11: Confusion Matrices των βέλτιστων GMM-HMMs στο (a) Validation Set, (b) Train Set.

14. Αναδρομικά Νευρωνικά Δίκτυα

Σε δεύτερο στάδιο, καλούμαστε να αναπτύξουμε ένα ASR σύστημα αναγνώρισης ψηφίων με χρήση Αναδρομικών Νευρωνικών Δικτύων. Για τον σκοπό αυτό, θα επιτελέσουμε μια σειρά από πειράματα με διαφορετικά μοντέλα κάθε φορά, εκπαιδεύοντας τα πάνω στο train set και ρυθμίζοντας τις υπερπαραμέτρους πάνω στο validation set. Σε κάθε περίπτωση, επιλέγουμε **Batch Size = 64, 100 εποχές, κριτήριο κόστους Cross Entropy και Adam Optimizer με ρυθμό μάθησης 0,001**.

Ως πρώτο πείραμα, εκπαιδεύουμε ένα απλό LSTM αναδρομικό δίκτυο με ένα κρυφό επίπεδο 64 νευρώνων. Απεικονίζοντας το train και validation loss ανά τις 100 εποχές (Σχήμα 12a), παρατηρούμε ότι υπάρχουν μικρά 'peaks', ενώ το μοντέλο τείνει να υπερεκπαιδευτεί και να χάσει την ικανότητα γενίκευσής του. Αυτό φαίνεται από το γεγονός ότι το train loss διαρκώς μειώνεται, σε αντίθεση με το validation loss που φαίνεται να σταθεροποιείται στην τιμή 0.08 (εγκλωβισμός σε τοπικό ελάχιστο).

Ως δεύτερο πείραμα, προσθέτουμε στο LSTM μοντέλο **Dropout** και **L2-Regularization**. Εφαρμόζοντας Dropout, το δίκτυο αγνοεί την επίδραση ορισμένων νευρώνων κατά τη διάρκεια της εκπαίδευσης του - τόσο κατά τη φάση forward οσο και για το back-propagation- οι οποίοι επιλέγονται τυχαία με πιθανότητα p. Ο λόγος που συμβαίνει αυτό, είναι για να αποφευχθεί το over-fitting. Ειδικότερα, ένα Fully Connecter Layer καταλαμβάνει τις περισσότερες παραμέτρους του δικτύου, με αποτέλεσμα οι νευρώνες να αναπτύσσουν αλληλοεξαρτήσεις μεταξύ τους κατά τη φάση της εκπαίδευσης. Κάτι τετοιο, μπορεί να περιορίσει τις δυνατότες κάθε νευρώνα ως μονάδα και να οδήγησει σε φαινόμενα over-fitting [**?**]. Έπειτα, η τεχνική του L2-Regularization (Ridge Regression) εφαρμόζει στο κριτήριο κόστους έναν επιπλέον τετραγωνικό όρο, ο οποίος λειτουργεί ως penalty προς τα βάρη του δικτύου που λαμβάνουν πολύ υψηλές τιμές. Η τεχνική αυτή συμβάλλει, επίσης, στην αποφυγή φαινομένων over-fitting [**?**]. Στην προκειμένη περίπτωση, εφαρμόζουμε στο LSTM δίκτυο dropout με πιθανότητα 50% και Ridge Regression με weight decay = 10^{-5}.

Ακολούθως, εφοδιάζουμε το μοντέλο μας με τεχνικές **Early Stopping**, ώστε να περιορίσουμε ακόμα περισσότερο φαινόμενα over-fitting. Ειδικότερα, τερματίζουμε τη διαδικασία εκπαίδευσης, εφόσον παρατηρηθούν πέντε διαδοχικές εποχές κατά τις οποίες το validation loss δεν μειώνεται σε σχέση με την εως τώρα βέλτιστη τιμή του. Με άλλα λόγια, η τεχνική Early Stopping εκπαιδεύει το μοντέλο μέχρι και το σημείο αυτό που το validation loss τείνει να αυξηθεί, λόγω overfitting και κατ'επέκταση generalization error [**?**]. Όπως φαίνεται και στο Σχήμα 12c η διαδικασία εκπαίδευσης τερματίζεται περίπου μετά τις 35 εποχές, αποτρέποντας το validation loss να λάβει υψηλότερες τιμές.

Ως τελικό πείραμα, μετατρέπουμε το εως τώρα μοντέλο μας σε **Αμφίδρομο Αναδρομικό Δίκτυο (bidirectional RNN ή BiRNN)**. Ένα τέτοιο δίκτυο, αποτελείται από τον συνδυασμό δύο διαφορετικών RNN, όπου το κάθε ένα επεξεργάζεται την ακολουθία εισόδου με διαφορετική φορά. Το κίνητρο αυτής της τεχνικής, είναι η δημιουργία μίας σύνοψης του ηχητικού σήματος και από τις δύο κατευθύνσεις, ώστε να σχηματιστεί μία καλύτερη αναπαράσταση, προβλέποντας κατά κάποια έννοια το μέλλον. Ετσι έχουμε ένα δεξιόστροφο RNN, το οποίο διαβάζει μία ακολουθία από το x1 προς το xT και ένα αριστερόστροφο RNN, το οποίο διαβάζει μία ακολουθία από το xT προς x1. Κάθε χρονική στιγμή t, λοιπόν, το διάνυσμα της κρυφής κατάστασης προκύπτει ως η συνένωση των δύο διανυσμάτων των επιμέρους κατευθύνσεων. Από το Σχήμα 12d προκύπτει ότι η προσθήκη αυτή μας δίνει το βέλτιστο μοντέλο LSTM, με το validation error να λαμβάνει τις μικρότερες τιμές του, δίχως έντονα peaks.

Τέλος, σημειώνεται ότι χρησιμοποιώντας τη συνατηση *pack_padded_sequence()*, ο χρόνος εκπαίδευσης του Αναδρομικού Δικτύου **μειώνεται δραστικά**, καθώς εξοικονομούμε περιττούς υπολογισμούς που απαιτούνται έπειτα από padding των batches διαφορετικού μήκους.

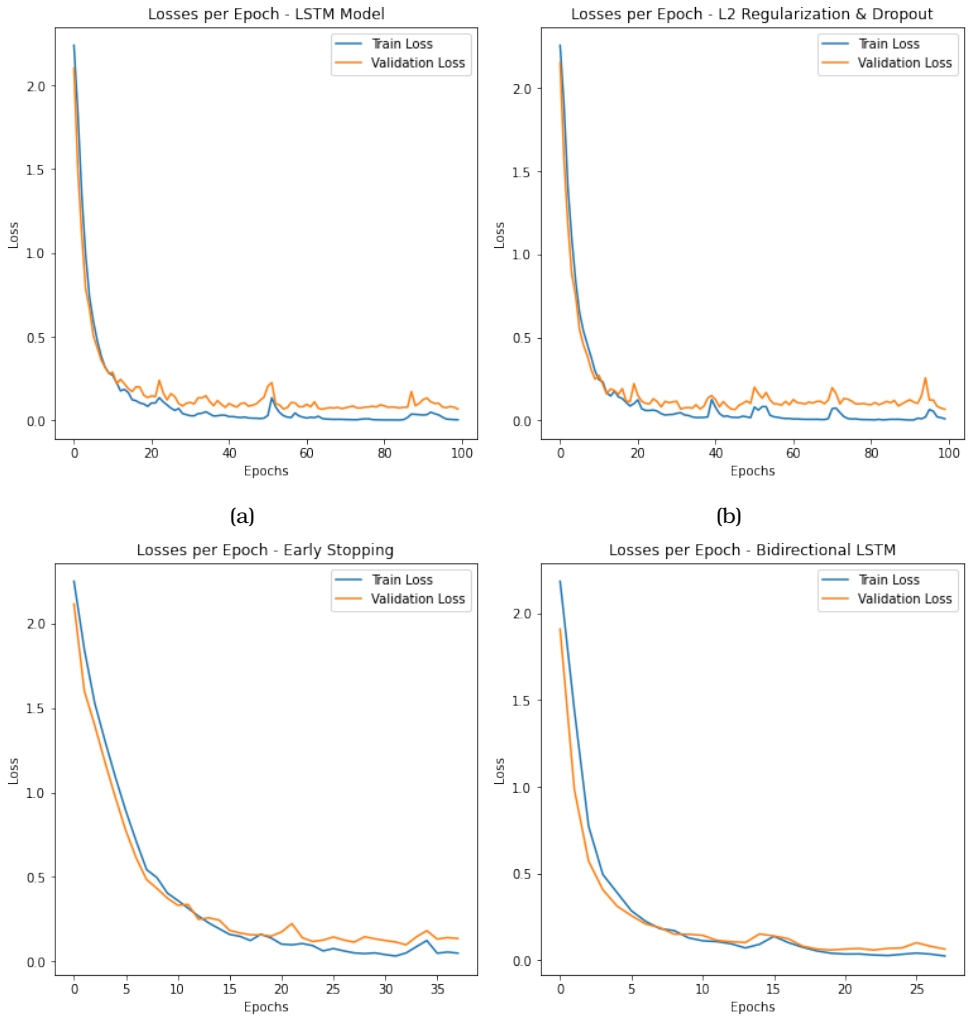
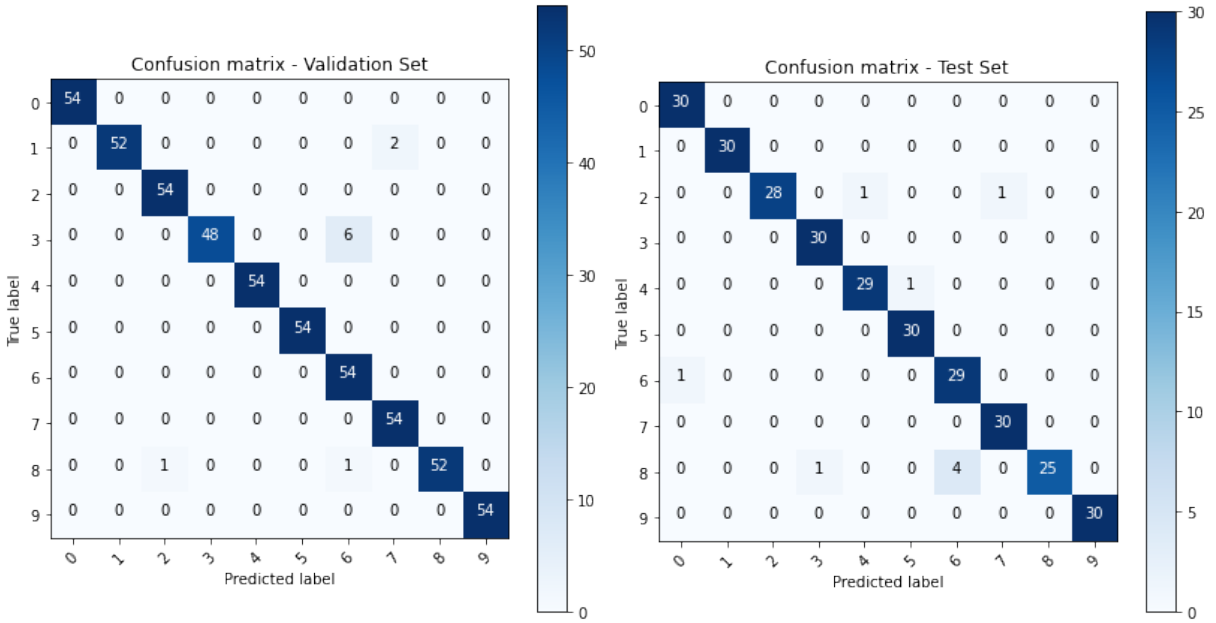


Figure 12: Train & Validation Loss ανά Εποχή για (a) απλό LSTM, (b) με L2-Regularization & Dropout, (c) με Early Stopping, (d) Bidirectional LSTM.

Συνοψίζοντας, λοιπόν, το βέλτιστο μοντέλο δίνεται από ένα LSTM Bidirectional Αναδρομικό Νευρωνικό Δίκτυο, εφοδιασμένο με Dropout, L2-Regularization και Early Stopping τεχνικές. Το ποσοστό ευστοχίας για το validation set μετά από περίπου 30 εποχές εκπαίδευσης είναι **98.15%**, ενώ το αντίστοιχο ποσοστό στο test set είναι **97%**. Από τα αντίστοιχα confusion matrices (Σχήμα 13) φαίνεται ότι το εν λόγω μοντέλο ταξινομεί εσφαλμένα 10 ψηφία (με συνηθέστερο λάθος το 3) στο validation set και 9 ψηφία (με συνηθέστερο λάθος το 8) στο test set.



(a) (b)

Figure 13: Confusion Matrices του βέλτιστου LSTM στο (a) Validation Set, (b) Train Set.

Σύνοψη

Συμπερασματικά, για τη δημιουργία ενός ASR συστήματος αναγνώρισης εκφερόμενων ψηφίων από εκφωνητές, προτιμώνται μοντέλα που μπορούν να κάνουν capture τη χρονική αλληλουχία που διέπει τα ηχητικά σήματα. Τέτοιου είδους μοντέλα είναι αφενός τα GMM-Hidden Markov Models και αφετέρου τα LSTM Αναδρομικά Νευρωνικά Δίκτυα, καθώς και τα δύο αποδίδουν ποσοστά ευστοχία άνω του 90%. Για την αποφυγή φαινομένων overfitting στα δεύτερα, επιστρατεύονται διάφορες τενικές βαθειάς μάθησης - όπως είναι το Dropout, Ridge Regression και Early Stopping - μέσω των οποίων βελτιώνεται η συνολική επίδοση του δικτύου.