ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών υπολογιστών

Ροή Σ: Σήματα, Έλεγχος και Ρομποτική

ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ

1η Εργαστηριακή Άσκηση

«Αναγνώριση φωνής με Κρυφά Μαρκοβιανά Μοντέλα και Αναδρομικά Νευρωνικά Δίκτυα»

Στοιχεία Ομάδας

- Μέλος 1ο: Πέππας Μιχαήλ-Αθανάσιος | Α.Μ: 03121026
- Μέλος 2ο: Αυγερινός Παναγιώτης | Α.Μ: 03121023
- Ημερομηνία Παράδοσης Αναφοράς: 13.11.2025

▪ Εισαγωγή

Έχοντας λάβει υπόψιν το πλαίσιο και τις απαιτήσεις της 1ης Εργαστηριακής Άσκησης, όπως αυτές καθορίζονται στο συνοδευτικό έγγραφο (PDF) της εκφώνησης, υποβάλλουμε τις λύσεις μας σε δύο διακριτά αρχεία:

1. **patrec1_03121026_03121023_report.pdf**: Η παρούσα αναφορά, η οποία περιέχει την αναλυτική παρουσίαση των λύσεων, τις απαντήσεις στα θεωρητικά ερωτήματα και τον σχολιασμό των αποτελεσμάτων για όλα τα ζητούμενα (Βήματα 1-8).

2. **patrec1_03121026_03121023_code.py:** Το αρχείο κώδικα (σε μορφή .py, προερχόμενο από το .ipynb notebook), το οποίο περιέχει την πλήρη υλοποίηση των λύσεων των ερωτημάτων. Ο κώδικας είναι πλήρως σχολιασμένος (με comments και Markdown cells) στα κρίσιμα σημεία.

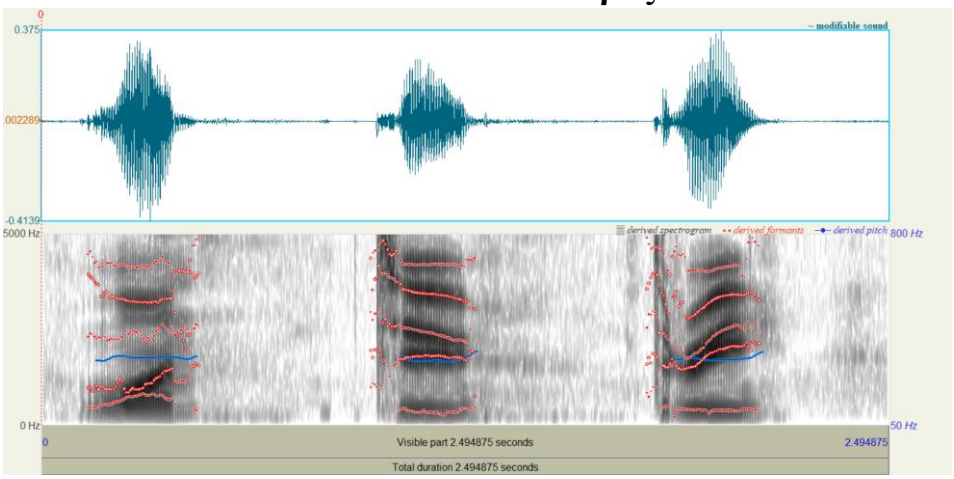
Στη συνέχεια, παρατίθενται οι απαντήσεις μας, οργανωμένες ανά βήμα και ερώτημα, σύμφωνα με τη σειρά που τέθηκαν.

- **Βήμα 1**

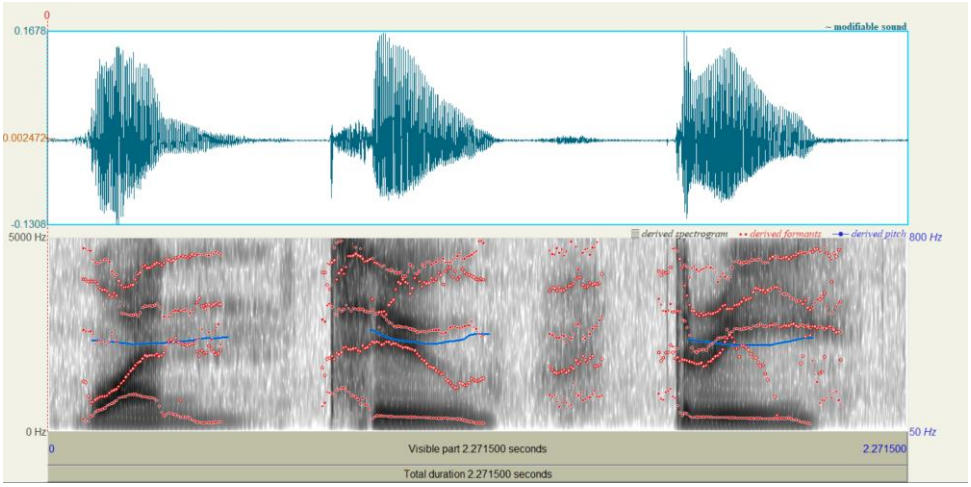
  Στο λογισμικό Praat φορτώθηκαν τα δύο σήματα ήχου. Οι κυματομορφές καθώς και τα spectrograms με σημειωμένα τα pitch και τα formants φαίνονται παρακάτω:

1. One-two-three

One-two-three Άντρας



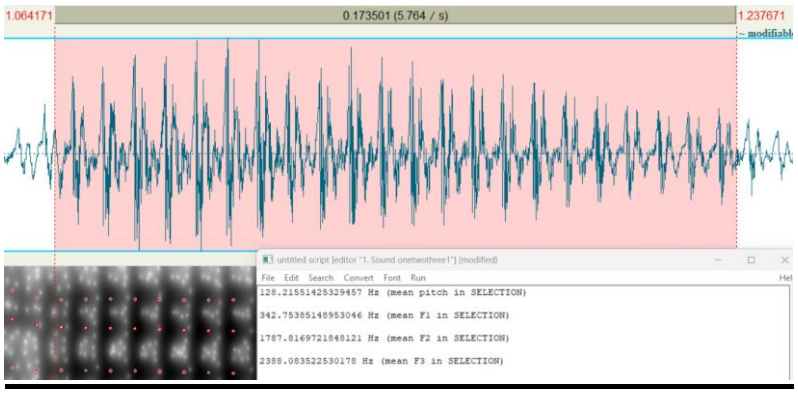
One-two-three Γυναίκα



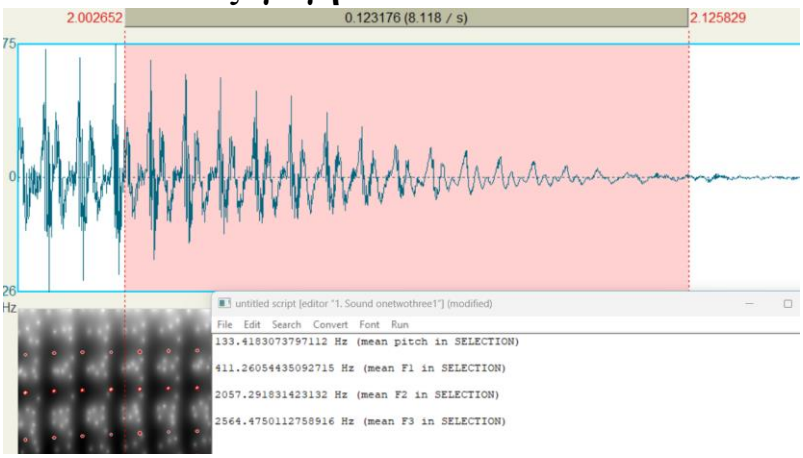
2. Pitches και Formants φωνηέντων του Άντρα

Εξαγωγή «α» από το one

Εξαγωγή «ου» από το two

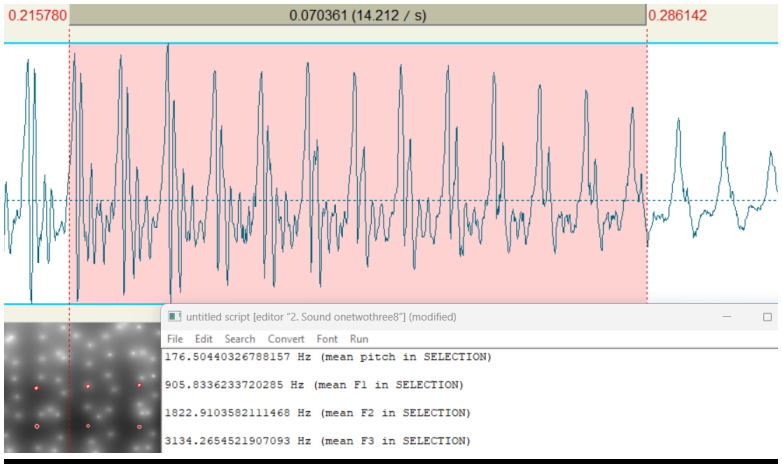


Εξαγωγή «ι» από το three

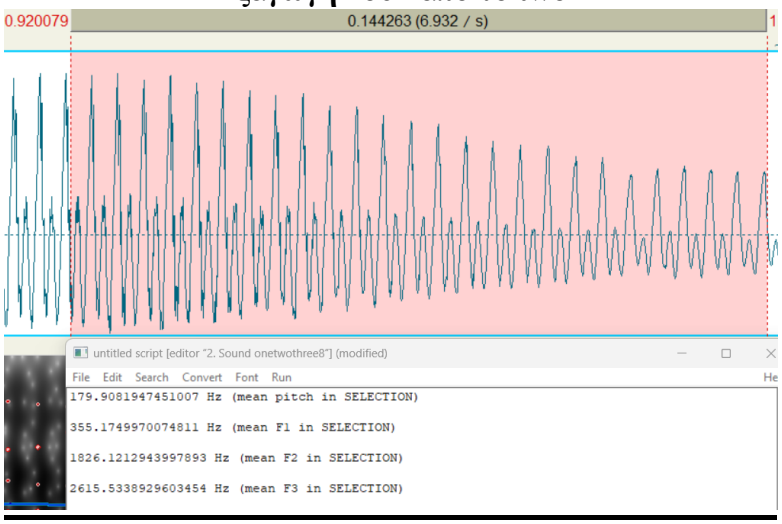


3. Pitches και Formants φωνηέντων της Γυναίκας

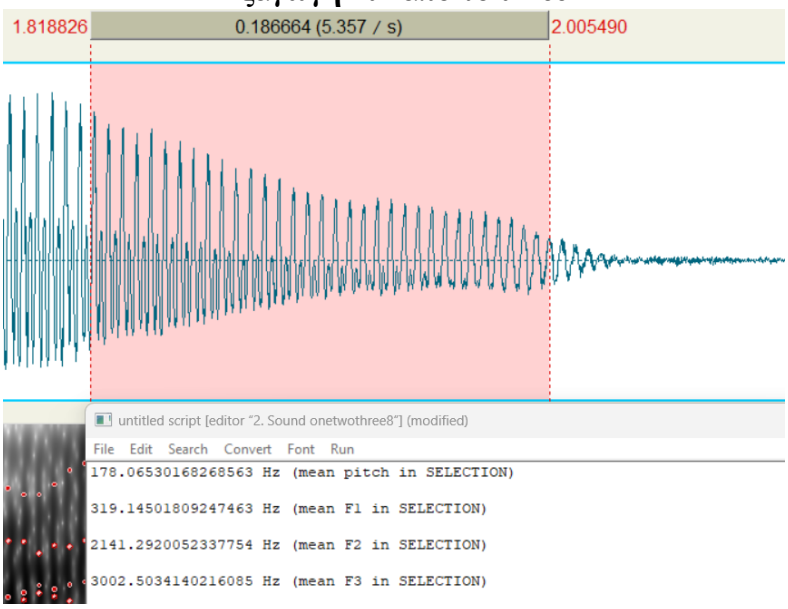
Εξαγωγή «α» από το one



Εξαγωγή «ου» από το two



Εξαγωγή «ι» από το three



Τα αποτελέσματα συνοψίζονται και στον παρακάτω πίνακα:

Ομιλητής	Φωνήεν	Pitch (Hz)	F_1(Hz)	F_2(Hz)	F_3(Hz)
Άνδρας	α	134.28	744.99	1322.64	2510.70
	ου	128.21	342.75	1787.71	2388.08
	ι	133.41	411.26	2057.29	2564.47
Γυναίκα	α	176.50	905.83	1822.91	3134.26
	ου	179.90	355.17	1826.12	2615.53
	ι	178.06	319.14	2141.29	3002.50

Παρατηρήσεις και Σχόλια Βήματος 1

1. **Pitch**

   Η γυναικεία φωνή έχει υψηλότερο pitch σε όλα τα φωνήεντα, κάτι που είναι φυσιολογικό και οφείλεται στη μικρότερη μάζα και μήκος των φωνητικών της χορδών. Η διαφορά αυτή είναι σταθερή και σαφής μεταξύ των δύο ομιλητών.

2. **F1 — Ύψος γλώσσας / «άνοιγμα» στόματος**

   Το **«α»** έχει το μεγαλύτερο F1**,** τόσο στον άνδρα όσο και στη γυναίκα, πράγμα που δείχνει ανοιχτή άρθρωση**.**
   Το **«ου»** και **«ι»** έχουν χαμηλότερα F1, δείχνοντας κλειστά φωνήεντα.

3. **F2 — Θέση γλώσσας εμπρός–πίσω**

   Το φωνήεν **«ι»** έχει το υψηλότερο F2 καθώς είναι εμπρόσθιο φωνήεν.
   Το **«ου»** έχει χαμηλότερο F2 καθώς είναι οπίσθιο φωνήεν.
   Για τη γυναίκα, η ίδια τάση διατηρείται, αλλά οι τιμές είναι γενικά υψηλότερες.

4. **F3 — Πρόσθετα αρθρωτικά χαρακτηριστικά**

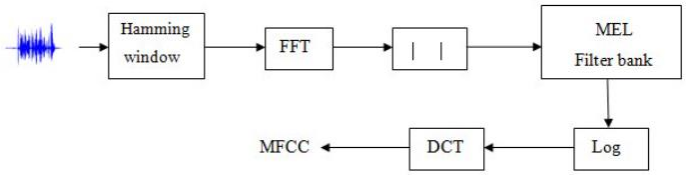
   Και εδώ, οι γυναίκες έχουν συστηματικά υψηλότερες τιμές**,** που συνδέονται με τον μικρότερο φωνητικό σωλήνα.

▪ <u>Βήμα 2</u>

Σε python υλοποιήθηκε η ζητούμενη συνάρτηση η οποία επιστρέφει τις ζητούμενες λίστες σε τούπλα. Ο κώδικας και τα σχόλια βρίσκονται αναλυτικά στο επισυναπτόμενο αρχείο .py.

- **Βήμα 3**

Σε αυτό το βήμα υπολογίζονται τα MFCC, ένα σύνολο χαρακτηριστικών που περιγράφουν το φάσμα (spectrum) του σήματος φωνής. Η διαδικασία που επιτελείται για τον υπολογισμό mfcc φαίνεται και στο παρακάτω διάγραμμα.



Εικόνα 1: Δομικό σχήμα διαδικασίας υπολογισμού MFCC

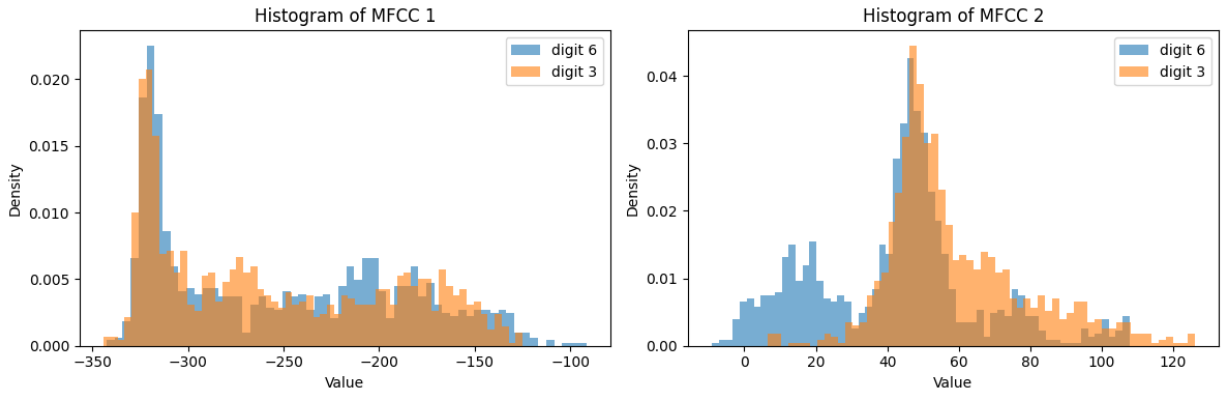
Αρχικά, το σήμα χωρίζεται σε παράθυρα Hamming. Έπειτα, με τον μετασχηματισμό Fourier (FFT) υπολογίζεται το μέτρο του φάσματος συχνοτήτων και έπειτα μέσω των φίλτρων Mel αποτυπώνεται η ενέργεια σε ζώνες που αντιστοιχούν στον τρόπο που αντιλαμβάνεται τις συχνότητες το αυτί. Ο λογάριθμος της ενέργειας αυτών των ζωνών υπολογίζεται και τέλος, μέσω του διακριτού συνημιτονικού μετασχηματισμού (DCT), προκύπτουν οι συντελεστές MFCC.

Για την εξαγωγή των παραπάνω μέσω Python, χρησιμοποιείται η βιβλιοθήκη της Librosa. Τέλος υπολογίζονται η πρώτη και δεύτερη τοπικές παράγωγοι των χαρακτηριστικών, Δ και ΔΔ μέσω της βιβλιοθήκης της librosa.

Η εργασία με σχόλια φαίνεται αναλυτικά και στον κώδικα.

- **Βήμα 4**

Έγινε αναπαράσταση του 1ου και 2ου MFCC των ψηφίων 3 και 6 , τα οποία είναι τα τελευταία ψηφία των αριθμών μητρώων μας. Η αναπαράσταση των Histograms φαίνεται παρακάτω:

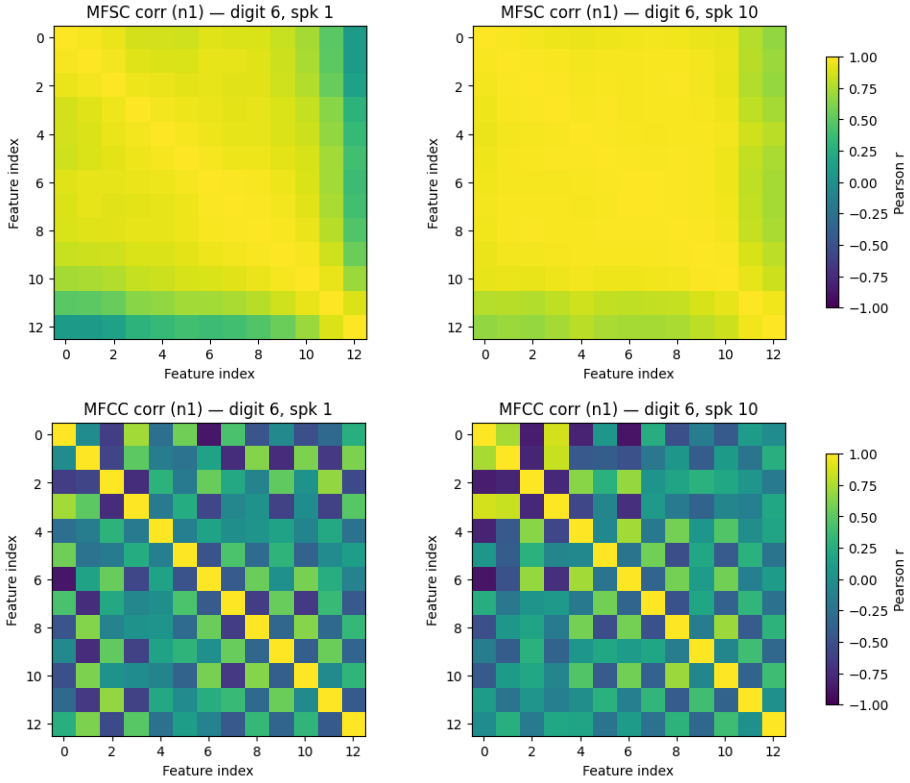


Εικόνα 2: Histograms για MFCC των ψηφίων

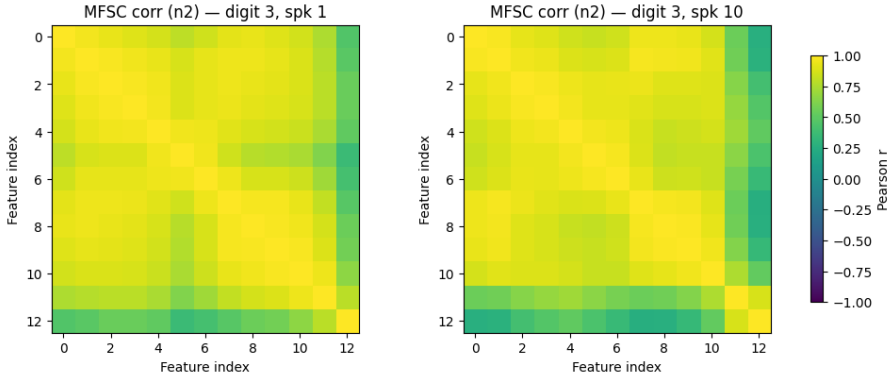
Παρατηρούμε πως ο $MFCC_1$ έχει πολύ μικρή απόκλιση στις μέσες τιμές για τα δύο ψηφία γεγονός που δείχνει ασθενή διαχωρισιμότητα.
Ο $MFCC_2$ παρουσιάζει σαφή μετατόπιση μέσης τιμής προσφέροντας, έτσι, καλύτερη ικανότητα διάκρισης.

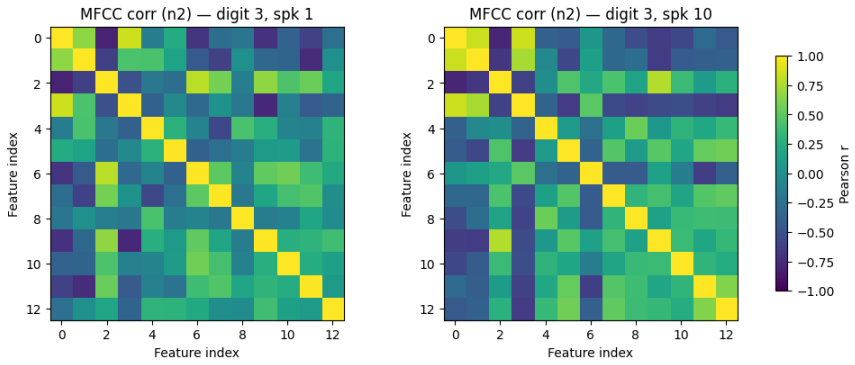
Στη συνέχεια χρησιμοποιήθηκαν δύο εκφωνήσεις ανά ψηφίο από διαφορετικούς ομιλητές και υπολογίστηκαν οι συντελεστές MFSC και οι MFCC. Οι MFSC είναι οι MFCC πριν την εφαρμογή του DCT. Η συσχέτιση των MFSC και MFCC φαίνεται παρακάτω:

<u>Διαγράμματα Συσχέτισης για το Ψηφίο 6</u>:



<u>Διαγράμματα Συσχέτισης για το ψηφίο 3</u>:





Παρατηρήσεις – Σχόλια:

1. Τα MFSCs εμφανίζουν μεγάλες συσχετίσεις εκτός της κυρίας διαγωνίου,το οποίο δείχνει πως υπάρχει πλεονάζουσα πληροφορία.
2. Αντίθετα, τα MFCCs (μετά τον DCT) παρουσίασαν σχεδόν διαγώνιους πίνακες συσχέτισης, με τις περισσότερες αλληλεξαρτήσεις να έχουν εξαλειφθεί.

Συμπέρασμα:

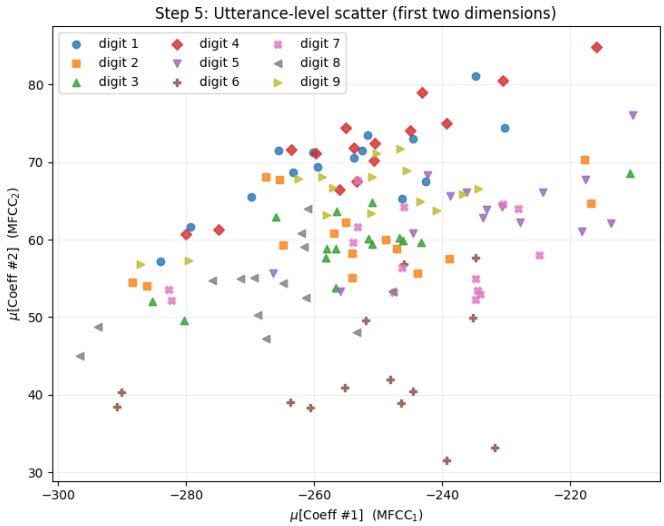
Η εφαρμογή του DCT στα MFSC μειώνει την εσωτερική συσχέτιση μεταξύ χαρακτηριστικών, οδηγώντας σε πιο ανεξάρτητες αναπαραστάσεις. Επομένως είναι καλύτερο να χρησιμοποιούμε MFCC, καθώς προσφέρουν πιο καθαρή και αποδοτική αναπαράσταση του φάσματος της ομιλίας αφαιρώντας την πλεονάζουσα πληροφορία μεταξύ γειτονικών συχνοτήτων.

Έτσι, τα χαρακτηριστικά γίνονται πιο ανεξάρτητα, συμπαγή και εύκολα αξιοποιήσιμα από στατιστικά μοντέλα.

▪ Βήμα 5

Τα χαρακτηριστικά ενώθηκαν σε ένα σταθερού μήκους διάνυσμα για κάθε εκφώνηση, ώστε να μπορέσουμε να δούμε πώς διαχωρίζονται τα ψηφία σε χαμηλές διαστάσεις. Για κάθε εκφώνηση, από τα 39-διάστατα χαρακτηριστικά του Βήματος 3 (13 MFCC + 13 Δ + 13 ΔΔ), υπολογίζουμε τον μέσο όρο και την τυπική απόκλιση στο χρόνο. Έτσι προκύπτει ένα 78-διάστατο διάνυσμα [μ(39) | σ(39)].

Το scatter plot με τις 2 πρώτες διαστάσεις των διανυσμάτων φαίνεται παρακάτω:



Κάθε σημείο αντιστοιχεί σε μία εκφώνηση, και κάθε ψηφίο (1–9) έχει διαφορετικό σύμβολο.
Παρατηρείται μεγάλη επικάλυψη μεταξύ κλάσεων όταν χρησιμοποιούνται μόνο οι δύο πρώτοι μέσοι όροι, κάτι αναμενόμενο για τόσο συνοπτική αναπαράσταση. Ωστόσο, ο άξονας του μ(MFCC$_2$) εμφανίζει καλύτερη διαχωριστική ικανότητα .
Για παράδειγμα : ψηφία όπως το 1 και το 7 τείνουν να έχουν υψηλότερες τιμές, ενώ το 6 χαμηλότερες.
Η διασπορά εντός κλάσης οφείλεται σε διαφορετικούς ομιλητές και παραλλαγές εκφώνησης και συμπεραίνουμε πως δύο μόνο στατιστικά δεν επαρκούν για καθαρό διαχωρισμό μεταξύ ψηφίων.

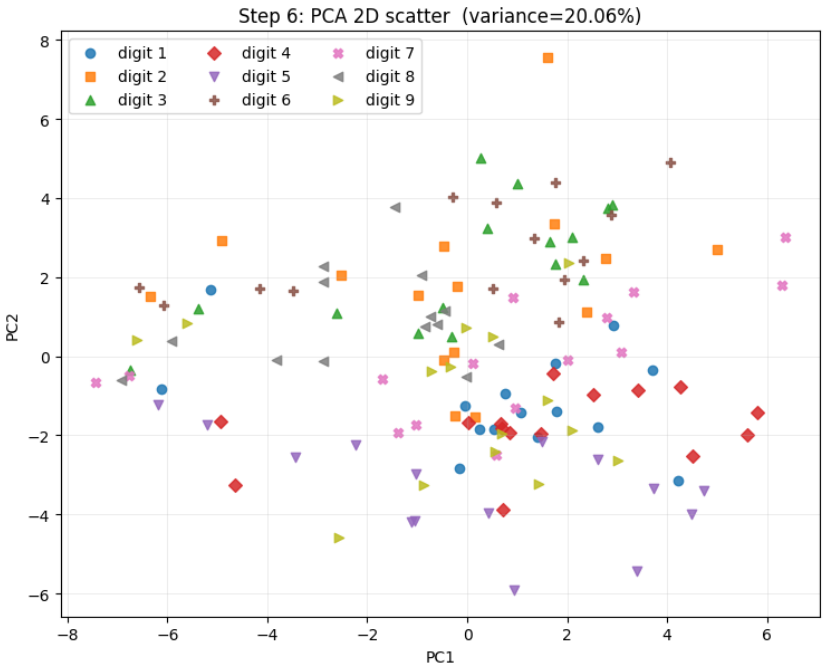
▪ Βήμα 6

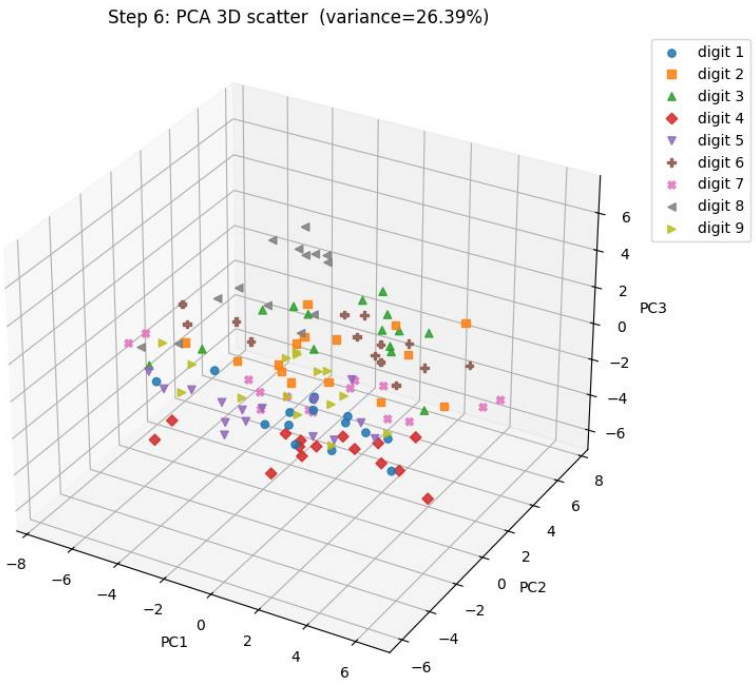
Εφαρμόζουμε Ανάλυση Κύριων Συνιστωσών (PCA) στα χαρακτηριστικά επιπέδου εκφώνησης, ώστε να προβάλλουμε τα δεδομένα σε 2D και 3D χώρους και να εκτιμήσουμε πόση από τη συνολική διασπορά (variance) εξηγείται από τις κύριες συνιστώσες.

Για το τελευταίο θα χρησιμοποιηθεί και ο δείκτης EVR που ορίζεται όπως παρακάτω:

$$EVR_k = \frac{\lambda_k}{\sum_j \lambda_j}$$

όπου λ_kείναι η ιδιοτιμή της k-ης συνιστώσας. Το άθροισμα όλων των EVR ισούται με 1. Επιπλέον, το άρθροισμα των EVR δείχνει πόση συνολική πληροφορία διατηρείται αν κρατήσουμε τις πρώτες m συνιστώσες.
Τα διαγράμματα φαίνονται παρακάτω:





EVR: PC1 = 12.54%, PC2 = 7.51%, PC3 = 6.33%
Cumulative EVR: 2D = 20.06%, 3D = 26.39%

Παρατηρούμε πως στο 2D γράφημα (PC1–PC2) διακρίνονται κάποιες ομάδες, αλλά υπάρχει μεγάλη επικάλυψη μεταξύ ψηφίων.

Το 3D γράφημα προσφέρει μικρή βελτίωση, ωστόσο η διαχωρισιμότητα παραμένει περιορισμένη.

Συμπέρασμα**:**
Οι πρώτες 2–3 κύριες συνιστώσες εξηγούν μόνο περίπου **20–26%** της συνολικής διασποράς, γεγονός που δείχνει ότι η πληροφορία είναι κατανεμημένη σε πολλές διαστάσεις.
Η μείωση διαστάσεων δεν είναι επιτυχημένη.

- **Βήμα 7**

Τα δεδομένα χωρίζονται σε 70% εκπαίδευση και 30% έλεγχο. Εφαρμόζεται StandardScaler, που προσαρμόζεται μόνο στα δεδομένα εκπαίδευσης και στη συνέχεια εφαρμόζεται στο test set, για να αποφευχθεί διαρροή πληροφορίας.

Εκπαιδεύτηκαν 6 ταξινομητές:

- Bayesian classifier (full covariance)
- Gaussian Naive Bayes
- SVM (RBF kernel)
- k-NN (k=5)
- Logistic Regression (multinomial)
- Decision Tree

Για την αξιολόγηση των ταξινομητών χρησιμοποιήθηκαν οι παρακάτω έννοιες :

TP (True Positives): σωστές προβλέψεις για μια συγκεκριμένη κλάση

FP (False Positives): περιπτώσεις που προβλέφθηκαν λανθασμένα ως αυτή η κλάση

FN (False Negatives): περιπτώσεις αυτής της κλάσης που το μοντέλο δεν αναγνώρισε σωστά

Δείκτης Precision: Δείχνει όταν το μοντέλο προβλέπει αυτή την κλάση, πόσο συχνά έχει δίκιο.

$$\text{Precision} = \frac{TP}{TP + FP}$$

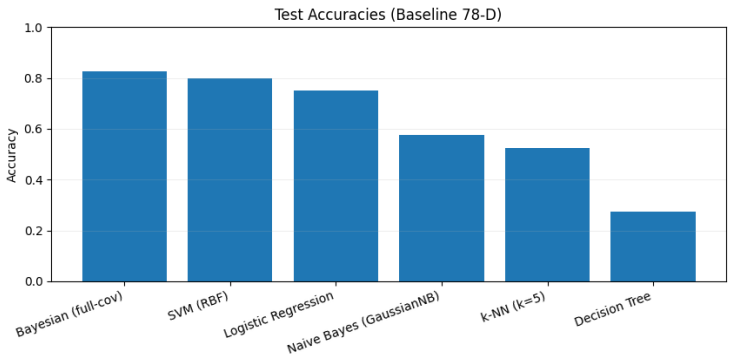
Δείκτης Recall: Δείχνει από τα πραγματικά δείγματα αυτής της κλάσης, πόσα αναγνωρίζει σωστά το μοντέλο

$$\text{Recall} = \frac{TP}{TP + FN}$$

Δείκτης F1: Ο αρμονικός μέσος του precision και recall.

$$F1 = \frac{2PR}{P + R}$$

Παρακάτω φαίνονται τα διαγράμματα για τους ταξινομητές και την ακρίβεια του εκάστοτε κατα τη δοκιμή με το test set.



Εικόνα 3: Ραβδογραφήματα ακρίβειας για όλους τους ταξινομητές

Παρατηρούμε πως Bayesian (full-cov) είχε την καλύτερη επίδοση (accuracy = 0.825) επειδή μπορεί να μοντελοποιεί τις συσχετίσεις μεταξύ των 78 χαρακτηριστικών, κάτι που τα δεδομένα μας εμφανώς έχουν. Η ridge regularization σταθεροποιεί τις εκτιμήσεις των πινάκων συνδιακύμανσης (Σ), αποτρέποντας υπερπροσαρμογή.

SVM και Logistic Regression είχαν επίσης καλή απόδοση, αφού μετά την κανονικοποίηση λειτουργούν ως ισχυροί γραμμικοί/μη γραμμικοί ταξινομητές.

Gaussian Naive Bayes απέδωσε χειρότερα, καθώς υποθέτει ανεξαρτησία μεταξύ χαρακτηριστικών — υπόθεση μη ρεαλιστική για τα MFCCs και τα στατιστικά τους.

k-NN είναι ευαίσθητος στην κλίμακα και στην πυκνότητα των δεδομένων, ειδικά με μικρό πλήθος δειγμάτων ανά κλάση.

Decision Tree είχε τη χειρότερη επίδοση (accuracy = 0.275), καθώς υπερπροσαρμόστηκε (overfitting) στο μικρό train set και δεν γενικεύει καλά, ειδικά χωρίς ρύθμιση (tuning/pruning).

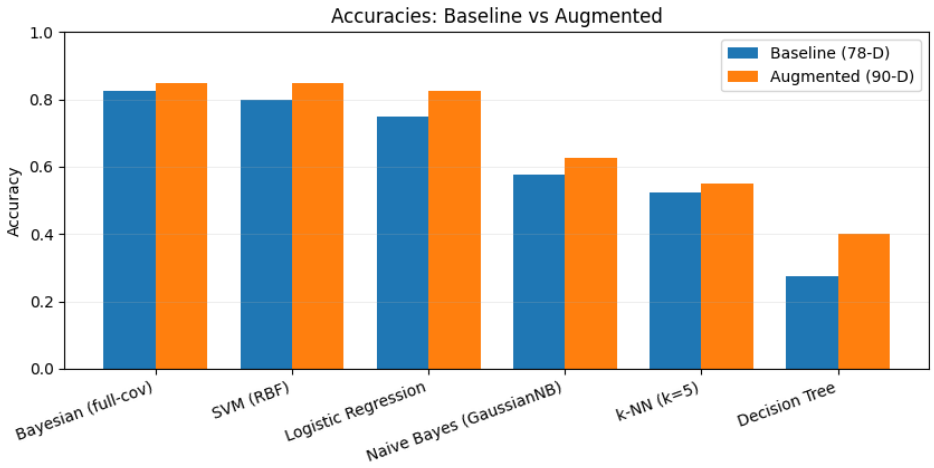
Μπόνους ερώτημα:

Για κάθε ηχητικό δείγμα (waveform), υπολογίζονται πρόσθετοι ακουστικοί δείκτες. Αυτά τα στατιστικά συνενώνονται με το βασικό 78-διάστατο διάνυσμα (MFCC + Δ + ΔΔ) , δημιουργώντας νέα 90-διάστατα διανύσματα χαρακτηριστικών.

Πρόσθετα Χαρακτηριστικά**:**

- Zero-Crossing Rate (ZCR): Ρυθμός αλλαγής πρόσημου στο σήμα, υποδεικνύει παρουσία φωνής ή θορύβου.
- Spectral Centroid: Το "κέντρο μάζας" του φάσματος, δείκτης brightness του ήχου.
- Spectral Bandwidth: Διασπορά γύρω από το centroid μετρά το φασματικό εύρος.
- Spectral Rolloff (0.85): Η συχνότητα κάτω από την οποία βρίσκεται το 85% της συνολικής ενέργειας του σήματος.
- RMS Energy: Ριζική μέση τετραγωνική ενέργεια, αντιπροσωπεύει τη βραχυχρόνια ένταση του ήχου.
- Spectral Flatness: Δείκτης τονικότητας έναντι θορύβου.

Με αυτά τα πρόσθετα χαρακτηριστικά παίρνουμε το παρακάτω διάγραμμα σε σύγκριση με πρίν:



Εικόνα 4: Ραβδογραφήματα Ακρίβειας για όλους τους ταξινομητές και με επιπλέον χαρακτηριστικά

Παρατηρούμε πως σε όλους τους ταξινομητές έχουμε βελτιωμένη ακρίβεια. Οι έξι νέοι δείκτες (ZCR, Centroid, Bandwidth, Rolloff, RMS, Flatness) προσθέτουν πληροφορία για την ενέργεια και τη φασματική υφή του ήχου, η οποία δεν αποτυπώνεται πλήρως στα MFCCs.

Τα γραμμικά και μη γραμμικά διαχωριστικά μοντέλα (π.χ. Logistic Regression, SVM) εκμεταλλεύονται αποτελεσματικά τον πλουσιότερο κανονικοποιημένο χώρο χαρακτηριστικών, οδηγώντας σε σταθερές βελτιώσεις.

Το GaussianNB παραμένει περιορισμένο λόγω της υπόθεσης διαγώνιας συνδιακύμανσης, ενώ το full-cov Bayesian επωφελείται περισσότερο επειδή μοντελοποιεί τις νέες συσχετίσεις μεταξύ χαρακτηριστικών.

Το k-NN βελτιώνεται ελαφρώς, καθώς είναι ευαίσθητο στην απόσταση και τη διάσταση, ειδικά με μικρό αριθμό δειγμάτων.

Το Decision Tree παρουσιάζει τη μεγαλύτερη απόλυτη βελτίωση, αλλά παραμένει το πιο αδύναμο συνολικά λόγω υψηλής διακύμανσης και έλλειψης ρύθμισης (tuning/pruning).

Γενικότερα: Η σύγκριση είναι απόλυτα δίκαιη (ίδιο split, ίδια κανονικοποίηση, ίδια πρωτόκολλα εκπαίδευσης). Παρόλα αυτά το test set είναι μικρό (~4–5 δείγματα ανά κλάση), επομένως τα αποτελέσματα πρέπει να παρουσιαστούν με επιφύλαξη ή να επιβεβαιωθούν με πολλαπλά splits.

▪ <u>Βήμα 8</u>

Δημιουργούμε μικρές, ομοιόμορφα δειγματοληπτημένες ακολουθίες από σήμα ημιτόνου και εκπαιδεύουμε τρία αναδρομικά νευρωνικά δίκτυα (RNN).
Κάθε ακολουθία n έχει τυχαία αρχική φάση $\phi_n \sim U[0,2\pi)$ και ορίζεται ως:

$$x_{n,k} = \sin(\phi_n+\omega_k\Delta t), y_{n,k} = \cos(\phi_n+\omega_k\Delta t)$$

$L = 10$ (μήκος ακολουθίας)
$\Delta t = 1$ ms (δειγματοληψία στα 1 kHz)
$F = 40$ Hz $\Rightarrow \omega = 2\pi f$

Η περίοδος είναι $T = 25$ ms.

- Το παράθυρο των 10 ms ≈ 0.4T καλύπτει αρκετό μέρος του κύκλου ώστε το μοντέλο να εξάγει τη φάση.
- Η συχνότητα δειγματοληψίας (1 kHz) είναι πολύ μεγαλύτερη από το όριο Nyquist (80 Hz), άρα χωρίς aliasing.
- $N = 4096$ ακολουθίες συνολικά → 64% train / 16% validation / 20% test (με σταθερό, ντετερμινιστικό διαχωρισμό).

Στην υλοποίηση έχουμε : 1 αναδρομικό block με batch_first=True και hidden size = 32. Υλοποιούνται τρεις παραλλαγές:

- RNN (tanh) — απλό, χωρίς πύλες.
- GRU — με πύλες ενημέρωσης και επαναφοράς.
- LSTM — με πύλες εισόδου, εξόδου και μνήμης.

Head: γραμμικό στρώμα (Linear) ανά χρονικό βήμα που μετατρέπει το κρυφό διάνυσμα σε έξοδο $y \in \mathbb{R}^{L\times 1}$.

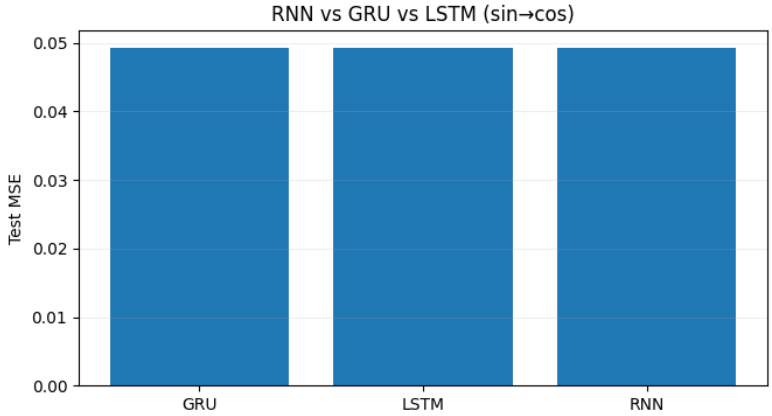
Θεωρούμε πως η τιμή H = 32 επαρκεί για την εκμάθηση μιας ομαλής 1D συνάρτησης σε μικρές ακολουθίες (L = 10), αποφεύγοντας ταυτόχρονα overfitting.

Οι μονάδες LSTM (Long Short-Term Memory) και GRU (Gated Recurrent Unit) χρησιμοποιούνται επιπλέον, γιατί ξεπερνούν το πρόβλημα της εξαφάνισης βαθμίδων (vanishing gradients), που εμποδίζει τα απλά RNN να αντιληφθούν έντονες εξαρτήσεις. Χάρη στα gates, οι LSTM/GRU μπορούν να "θυμούνται" ή να "ξεχνούν" πληροφορίες επιλεκτικά, διατηρώντας τη ροή της πληροφορίας για περισσότερο χρόνο. Έτσι, καθιστώνται πιο σταθερά, συγκλίνουν γρηγορότερα και έχουν καλύτερη απόδοση ειδικά σε χρονοσειρές ή ακολουθιακά δεδομένα όπως σήματα ομιλίας, ήχου ή φυσικής γλώσσας.

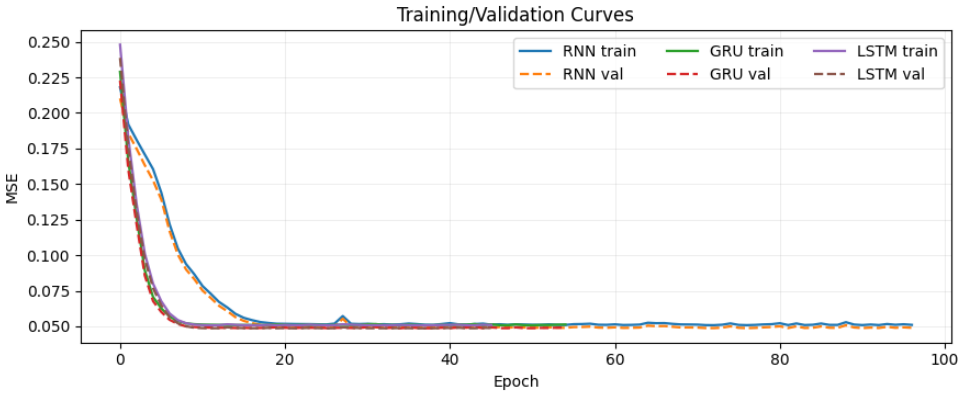
Ως δείκτης αξιολόγησης χρησιμοποιείται το Mean Squared Error (MSE).

$$\mathrm{MSE} = \frac{1}{NL}\sum_{n,k}(\hat{y}_{n,k} - y_{n,k})^2$$

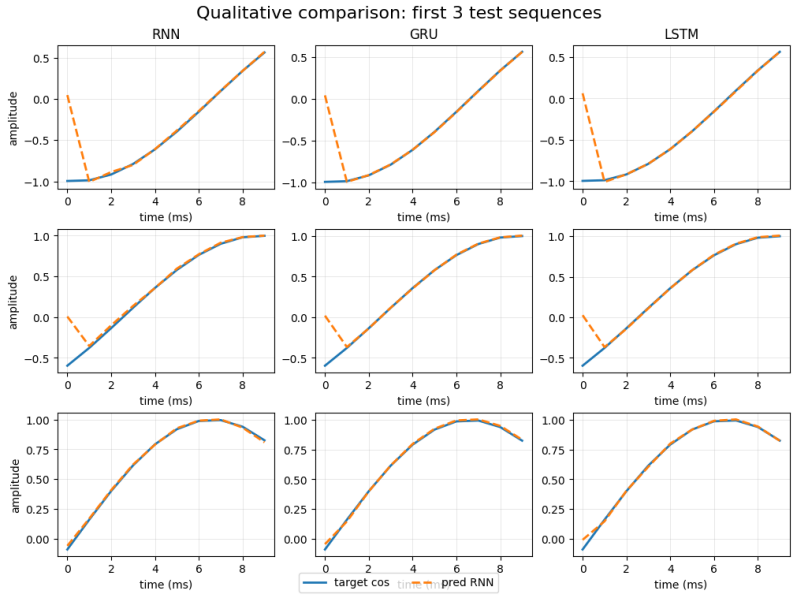
Παρακάτω φαίνονται οι οπτικοποιήσεις των αποτελεσμάτων μας:



Εικόνα 5: Ραβδόγραμμα test MSE για τα 3 μοντέλα



Εικόνα 6: Καμπύλες train/validation MSE



Εικόνα 7: Γραφήματα πρόβλεψης vs πραγματικού σήματος για τις 3 πρώτες test ακολουθίες

- Η απώλεια μειώνεται γρήγορα στα πρώτα ~15 epochs και μετά σταθεροποιείται ομαλά, οι καμπύλες train και validation σχεδόν ταυτίζονται, δείχνοντας καλή γενίκευση χωρίς υπερεκπαίδευση (λόγω του καθαρού, χωρίς θόρυβο προβλήματος).
- Οι προβλέψεις (διακεκομμένες γραμμές) ευθυγραμμίζονται πλήρως με τα πραγματικά συνημίτονα για όλες τις ακολουθίες δεν παρατηρείται μετατόπιση φάσης.

Παρατηρούμε πως όλα τα μοντέλα πετυχαίνουν σχεδόν ίδια απόδοση (διαφορές τάξης 10^{-4} MSE), κάτι αναμενόμενο σε ένα απλό, χωρίς θόρυβο και χαμηλής πολυπλοκότητας έργο με πολύ μικρές ακολουθίες.

Τα μοντέλα με πύλες (GRU, LSTM) δείχνουν λίγο πιο σταθερή σύγκλιση, αλλά το τελικό αποτέλεσμα είναι πρακτικά ισοδύναμο με το απλό RNN.

Συμπέρασμα**:**

Σε αυτήν την ελεγχόμενη ρύθμιση, όλα τα RNN μαθαίνουν τέλεια τη φασική σχέση

ημιτόνου–συνημιτόνου. Οι GRU/LSTM υπερέχουν ελαφρά σε σταθερότητα και σύγκλιση, αλλά η εργασία είναι τόσο απλή που το vanilla RNN παραμένει πλήρως ανταγωνιστικό.

Κύριο Μέρος Εργαστηρίου

- **Βήμα 9**

Έγινε προεπεξεργασία του συνόλου δεδομένων Free Spoken Digit Dataset (FSDD) για εκπαίδευση και δοκιμή 80%-20% αντίστοιχα.
Συγκεκριμένα, τα αρχεία με αναγνωριστικό ["0","1","2","3","4"] ορίζονται ως test (~10% των δεδομένων), ενώ τα υπόλοιπα 2700 δείγματα χωρίζονται σε train (2160, 72%) και validation (540, 18%) με stratified split ώστε να διατηρηθεί η αναλογία των ψηφίων.
Για κανονικοποίηση, υπολογίζονται ο μέσος όρος και η τυπική απόκλιση των MFCC μόνο από το train set, και στη συνέχεια εφαρμόζεται σε όλα τα σύνολα:

$$x_{norm} = \frac{x - \mu}{\sigma}$$

Αυτό εξασφαλίζει ομοιόμορφη κλίμακα χαρακτηριστικών, σταθερή εκπαίδευση και αποφυγή data leakage.

- **Βήμα 10**

Σε αυτό το σημείο γίνεται αρχικοποίηση GMM–HMM, χρησιμοποιώντας τα κανονικοποιημένα χαρακτηριστικά MFCC από το Βήμα 9.
Κάθε εκφώνηση (utterance) αντιστοιχεί σε μία ηχογράφηση ενός ψηφίου, μετατρεπόμενη σε ακολουθία χαρακτηριστικών (T×D), όπου T είναι τα χρονικά παράθυρα και D ο αριθμός των MFCC.
Η ομιλία εξελίσσεται χρονικά, οπότε χρησιμοποιείται μοντέλο “left–right” (Bakis HMM), που επιτρέπει μόνο μετάβαση προς τα εμπρός ή επανάληψη της ίδιας κατάστασης. Για αρχικοποίηση, κάθε εκφώνηση χωρίζεται σε n ισομεγέθη τμήματα.
Όλα τα παράθυρα από το ίδιο τμήμα συγκεντρώνονται για κάθε κατάσταση και πάνω τους εφαρμόζεται GMM με n_mixtures Γκαουσιανές (με διαγώνια συνδιακύμανση) ώστε να εκτιμηθούν τα αρχικά στατιστικά.
Η αρχική κατανομή έχει τη μορφή π = [1, 0, …, 0], ενώ οι μεταβάσεις δεν έχουν οπισθοδρόμηση ή “άλματα”. Μόνο η τελευταία κατάσταση μπορεί να οδηγήσει σε τερματισμό.
Παράδειγμα: για το ψηφίο “3” έχει 216 δείγματα και n =3, κάθε εκφώνηση χωρίζεται σε 3 τμήματα (T~11 παράθυρα το καθένα). Από τη συγκέντρωση όλων των παραπάνω δημιουργούνται τρεις ομάδες (states), καθεμία μοντελοποιημένη από ένα GMM. Αυτές συναρμολογούνται σε ένα πλήρες HMM.

- **Βήμα 11**

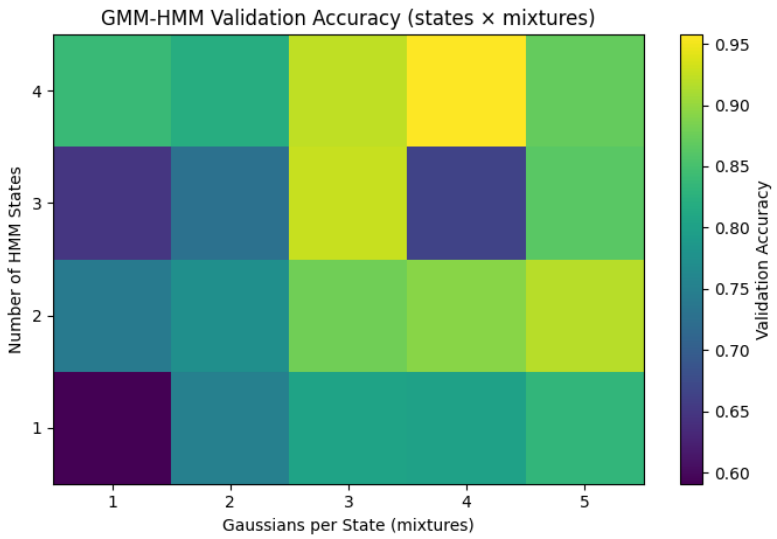
Στο βήμα αυτό εκτελείται εκπαίδευση EM & Αξιολόγηση. Επιπλέον, πραγματοποιείται η συστηματική διερεύνηση των παραμέτρων των GMM–HMM για βελτιστοποίηση της απόδοσης στο σύνολο επικύρωσης.

Δοκιμάζονται 20 συνδυασμοί: αριθμός καταστάσεων (states) ∈ {1,2,3,4} × αριθμός μειγμάτων Gaussians (mixtures) ∈ {1,2,3,4,5}.

Για κάθε συνδυασμό:

- Δημιουργείται αριστερό–δεξιό HMM με αρχικοποίηση από το Βήμα 10 (equal-chunk bootstrap).
- Όταν mixtures=1, κάθε κατάσταση έχει μία Gaussian, όταν mixtures ≥ 2, χρησιμοποιείται GMM.
- Η εκπαίδευση γίνεται με Baum–Welch (EM) έως 50 εποχές ή μέχρι να σταθεροποιηθεί το likelihood (βελτίωση < 10^{-3} για 3 συνεχόμενες εποχές).
- Η αξιολόγηση γίνεται στο validation set, ταξινομώντας κάθε εκφώνηση στο ψηφίο με το μέγιστο log-likelihood. Υπολογίζεται η συνολική ακρίβεια (val_acc).
- Εφαρμόζονται όρια σε διακυμάνσεις (≥ 8e−4) και βάρη μειγμάτων (≥ 8e−4) για αποφυγή αριθμητικής αστάθειας.
- Χρησιμοποιούνται σταθεροί seeds (NumPy, torch, random) για δίκαιη σύγκριση μεταξύ συνδυασμών.

Τα αποτελέσματα φαίνονται παρακάτω:



Κορυφαίες επιδόσεις:

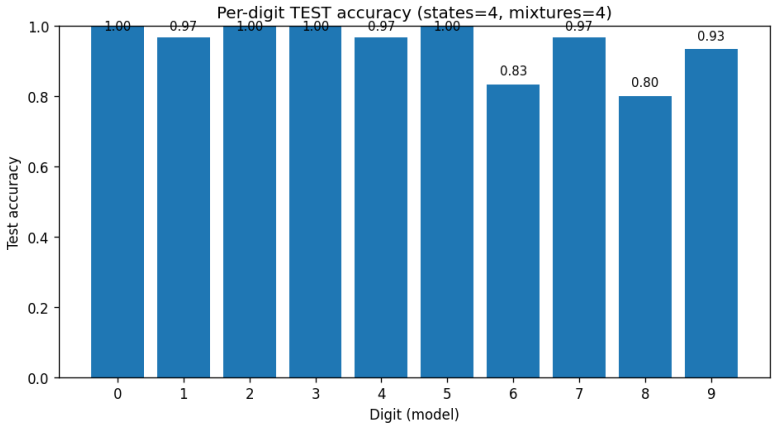
- (4 states, 4 mixtures) → 0.9574 ακρίβεια (254.27s)
- (3,3) → 0.9259
- (4,3) → 0.9222
- (2,5) → 0.9185
- (4,5) → 0.8704

Παρατηρούμε πως η αύξηση από 1 σε 4 βελτιώνει σταθερά την ακρίβεια (0.83 → 0.96), δείχνοντας ότι περισσότερες χρονικές φάσεις μοντελοποιούν καλύτερα την εξέλιξη του ψηφίου.
Ακόμη, η ακρίβεια βελτιώνεται ως τις 3 έως 4 Gaussians ανά state, μετά όμως μειώνεται λόγω υπερπροσαρμογής και κακής σύγκλισης του EM.
Το (4,4) προσφέρει τα καλύτερα χρονικά αλλά και ακριβέστερα αποτελέσματα.

- **Βήμα 12**

Έχοντας τα μοντέλα από το προηγούμενο βήμα, πραγματοποιήθηκε χρήση των βέλτιστων υπερπαραμέτρων (4 καταστάσεις, 4 μίγματα) για την εκπαίδευση των τελικών μοντέλων ανά ψηφίο και να αξιολόγηση.
Συνοπτικά ταθεροποιούμε τις υπερπαραμέτρους από το Βήμα 11: states=4, mixtures=4, ενώνουμε train + validation και εκπαιδεύουμε ένα HMM ανά ψηφίο (0–9) με Baum–Welch (EM) και early stopping.
Κάθε δείγμα του TEST ταξινομείται στο ψηφίο με το μέγιστο log-likelihood.

Αποτελέσματα (4 states × 4 mixtures) :



Η απόδοση είναι εξαιρετική για τα περισσότερα ψηφία (0,2,3,5 φτάνουν το 100%), ενώ τα 6 και 8 εμφανίζουν χαμηλότερη ακρίβεια (0.83 και 0.80), πιθανόν λόγω ακουστικής ομοιότητας με άλλα ψηφία και διαφορετικότητας ομιλητών.

Το μοντέλο των 4 καταστάσεων × 4 μειγμάτων προσφέρει ισορροπία μεταξύ πολυπλοκότητας και γενίκευσης, αποφεύγοντας την υπερπροσαρμογή.

Η σύγκριση με τα αποτελέσματα επικύρωσης δείχνει καλή συνέπεια και ικανότητα γενίκευσης.

Οι υπερπαράμετροι επιλέγονται μέσω validation set (όχι test) ώστε να αποφεύγεται διαρροή πληροφορίας. Στη συνέχεια, το τελικό μοντέλο εκπαιδεύεται σε TRAIN+VAL και ελέγχεται στο TEST, παρέχοντας αντικειμενική εκτίμηση της γενίκευσης.

Αν δεν γίνει η διαδικασία αυτή υπάρχουν οι παρακάτω κίνδυνοι:

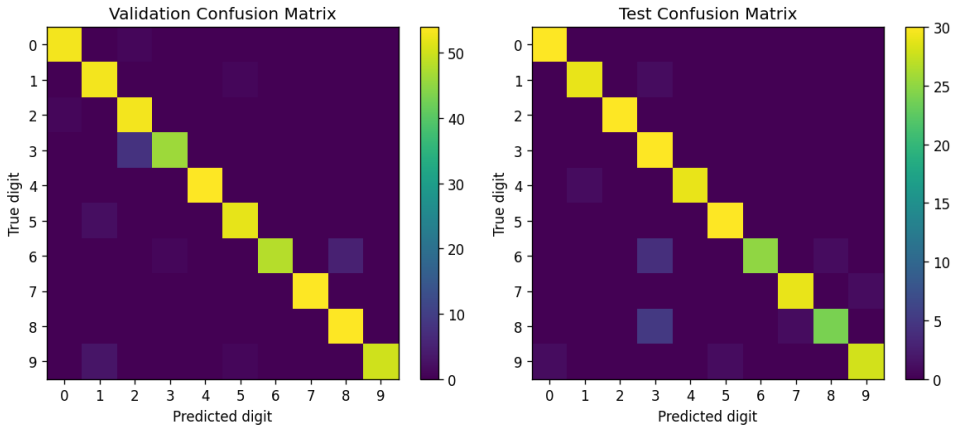
1. Data leakage (διαρροή πληροφορίας): Αν "δούμε" τα δεδομένα του TEST στη φάση εκπαίδευσης ή ρύθμισης, τότε η μέτρηση απόδοσης δεν είναι πλέον αξιόπιστη. Το μοντέλο έχει ουσιαστικά «δει τις απαντήσεις».

2. Ψευδώς υψηλή απόδοση: Το μοντέλο φαίνεται να έχει πολύ καλή ακρίβεια, αλλά στην πραγματικότητα έχει προσαρμοστεί στα συγκεκριμένα δείγματα και αποτυγχάνει σε νέα δεδομένα. Υπερπροσαρμογή.

3. Αδυναμία γενίκευσης: Χωρίς ανεξάρτητο TEST set, δεν μπορούμε να εκτιμήσουμε αν το σύστημα λειτουργεί σωστά σε πραγματικές συνθήκες.

- **<u>Βήμα 13</u>**

Δημιουργήθηκαν δύο 10×10 πίνακες για το σετ επικύρωσης (VALIDATION) και ένας για το σετ ελέγχου (TEST).
Κάθε γραμμή αντιστοιχεί στο πραγματικό ψηφίο και κάθε στήλη στο προβλεπόμενο ψηφίο.
Οι πίνακες αυτοί μας βοηθούν να δούμε ποιες κατηγορίες αναγνωρίζονται σωστά και ποιες συγχέονται μεταξύ τους.



Παρατηρολυμε πως οι πίνακες είναι κυρίως διαγώνιοι, άρα η πλειονότητα των ψηφίων αναγνωρίζεται σωστά.
Παρόμοια λάθη εμφανίζονται και στα δύο διαγράμματα, αρα έχουμε σταθερό σφάλμα σε κάθε σύνολο.
Τα ψηφία 6 και 8 μπερδεύονται περισσότερο, πιθανώς λόγω ομοιότητας στα ακουστικά ή χρονικά χαρακτηριστικά.

Η συνολική απόδοση είναι πολύ υψηλή:

- Validation: ≈95.7%
- Test: ≈94.7%

Μικρές στοχευμένες βελτιώσεις όπως προσθήκη περισσότερων μειγμάτων Gaussians μόνο για τα ψηφία 6,8 μπορούν να αυξήσουν ακόμα παραπάνω την απόδοση.

▪ Βήμα 14

1. Αρχικά προετοιμάζουμε το Free Spoken Digit Dataset (FSDD) για χρήση με RNN/LSTM μοντέλα.
2. Τα δεδομένα χωρίζονται σε TRAIN, VALIDATION και TEST και από κάθε ηχογράφηση εξάγονται 6 MFCC χαρακτηριστικά ανά πλαίσιο (~30 ms παράθυρο), σχηματίζοντας μεταβλητού μήκους ακολουθίες.
3. Η κανονικοποίηση γίνεται με βάση τα στατιστικά του TRAIN: υπολογίζονται ο μέσος όρος (μ_train) και η τυπική απόκλιση (σ_train) όλων των πλαισίων, και εφαρμόζεται σε όλα τα σύνολα (TRAIN, VAL, TEST).
4. Οι ακολουθίες προσαρμόζονται σε κοινό μήκος μέσω zero-padding και οργανώνονται σε DataLoaders για εκπαίδευση, επικύρωση και έλεγχο.

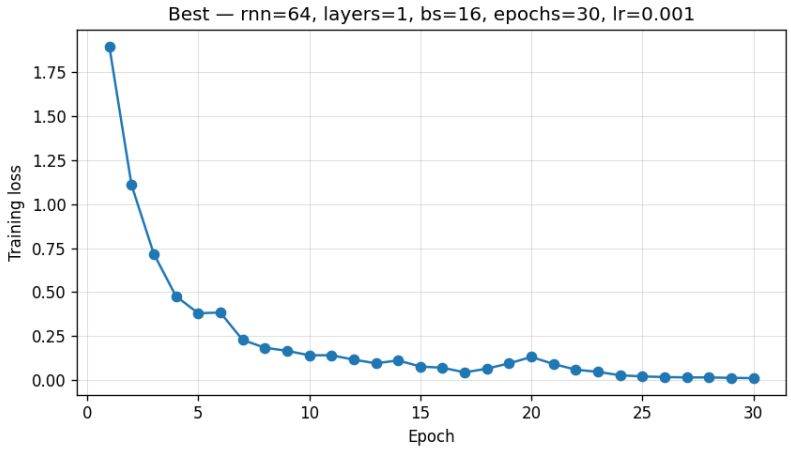
Συνολικά, το βήμα αυτό εξασφαλίζει ότι τα δεδομένα είναι καθαρά, σωστά κανονικοποιημένα και έτοιμα για χρήση στα επόμενα στάδια εκπαίδευσης των μοντέλων.

14.1-14.3:

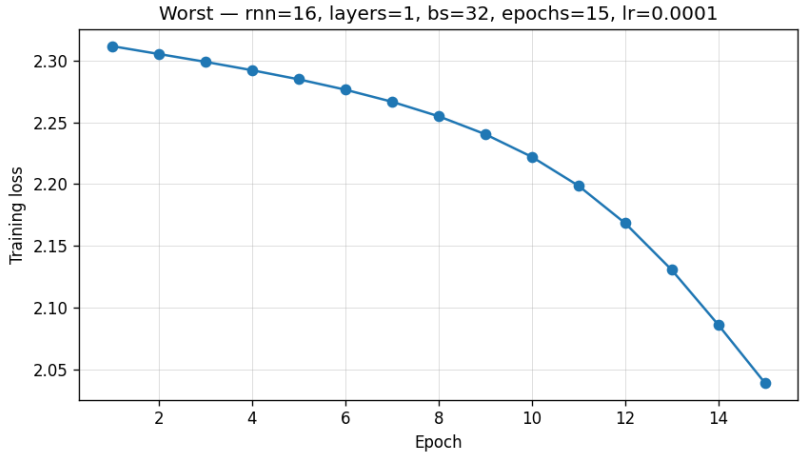
Με τη χρήση του βοηθητικού κώδικα υλοποιήθηκε και εκπαιδεύτηκε ένα απλό LSTM. Πραγματοποιήθηκε ένα grid search για τις διάφορες τιμές των παρακάτω παραμέτρων: Δοκιμάζονται διάφορες τιμές για:

- Μέγεθος κρυφών νευρώνων (rnn_size): 16, 32, 64
- Αριθμό επιπέδων (num_layers): 1, 2, 3
- Μέγεθος παρτίδας (batch_size): 16, 32
- Εποχές (epochs): 15, 23, 30
- Ρυθμός μάθησης (lr): 1e-4, 5e-4, 1e-3

Η τιμές των απωλειών για όλα τα μοντέλα φαίνονται και στον κώδικα. Από όλα τα νευρωνικά δίκτυα που προκύπτουν από αυτή την αναζήτηση κρατάμε το καλύτερο και το χειρότερο μοντέλο για να παρουσιάσουμε τα αποτελέσματα της εργασίας μας.
Παρακάτω φαίνονται τα αντίστοιχα διαγράμματα:



Εικόνα 8: Καλύτερο Μοντέλο και οι τιμές των παραμέτρων του

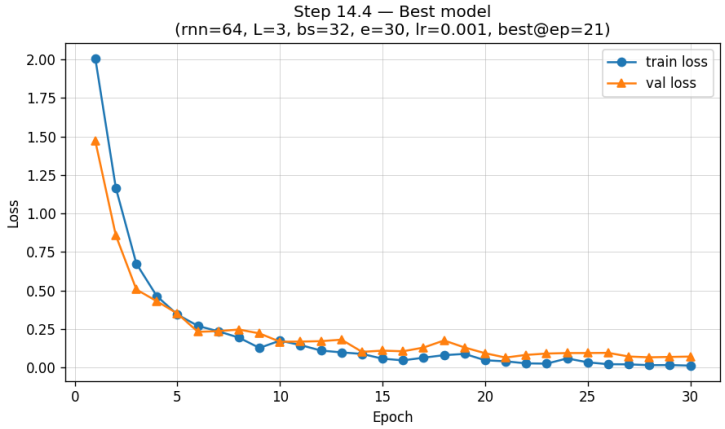


Εικόνα 9: Χειρότερο Μοντέλο και οι τιμές των παραμέτρων του

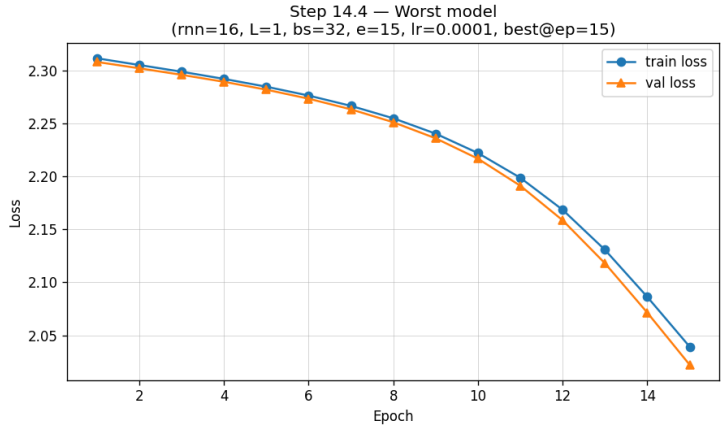
14.4:

Έγινε αξιολόγηση όλων των μοντέλων LSTM που εκπαιδεύτηκαν στα προηγούμενα βήματα (14.1–14.3) πάνω στο validation set, χωρίς νέα εκπαίδευση, ώστε να εξαχθούν καμπύλες απώλειας και ακρίβειας ανά εποχή (val_loss, val_acc).
Για κάθε μοντέλο, φορτώνονται τα αποθηκευμένα στιγμιότυπα (snapshots) ανά εποχή και αξιολογούνται στο σύνολο validation. Αν δεν υπάρχουν snapshots, χρησιμοποιούνται τα τελικά βάρη.
Ομοίως με πρίν, η αξιολόγηση ταξινομεί τα μοντέλα με βάση τη χαμηλότερη απώλεια επικύρωσης (validation loss), με ισοπαλία υπέρ του μοντέλου με υψηλότερη ακρίβεια. Τα μοντέλα αυτά δοκιμάζονται και στο training set.

Παρακάτω φαίνονται τα νέα διαγράμματα για το καλύτερο και το χειρότερο μοντέλο:



Εικόνα 10: Καλύτερο μοντέλο στο validation set



Εικόνα 11: Χειρότερο Μοντέλο στο validation set

14.5-14.6:

Εξετάζεται η επίδραση της κανονικοποίησης (regularization) μέσω Dropout και L2 (weight decay) πάνω στα 5 καλύτερα μοντέλα LSTM που προέκυψαν από το Βήμα 14.4. Οι επιπλέον παράμετροι που χρησιμοποιούνται είναι :

- Dropout ∈ {0.0, 0.2, 0.4}
  Κατά την εκπαίδευση απενεργοποιεί τυχαία ένα ποσοστό νευρώνων σε κάθε βήμα, ώστε το δίκτυο να μη βασίζεται υπερβολικά σε συγκεκριμένες συνδέσεις. Έτσι αναγκάζεται να γενικεύσει, βελτιώνοντας την απόδοσή του σε άγνωστα δεδομένα.

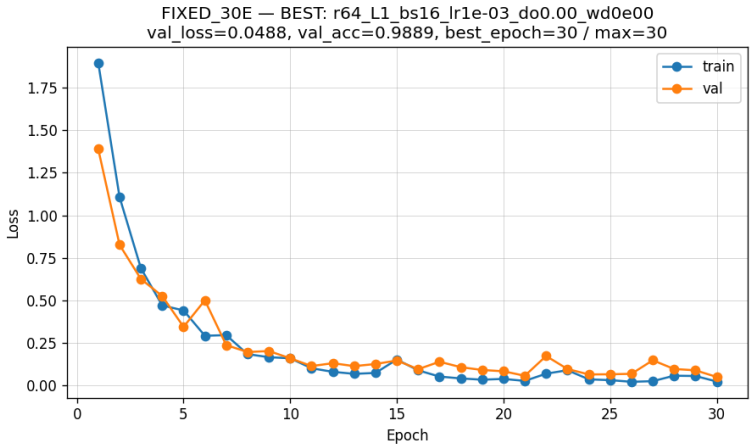
- L2 Regularization μέσω weight_decay ∈ {0.0, 1e−4, 1e−3}
  Προσθέτει έναν όρο ποινής στο κόστος, που είναι ανάλογος με το τετράγωνο των βαρών.
  Με αυτόν τον τρόπο, αποτρέπει τα βάρη από το να γίνουν πολύ μεγάλα και βοηθά στη διατήρηση της σταθερότητας, μειώνοντας επίσης την υπερεκπαίδευση.

Με αυτές τις παραπάνω γίνεται επανεκπαίδευση σε δύο φάσεις:

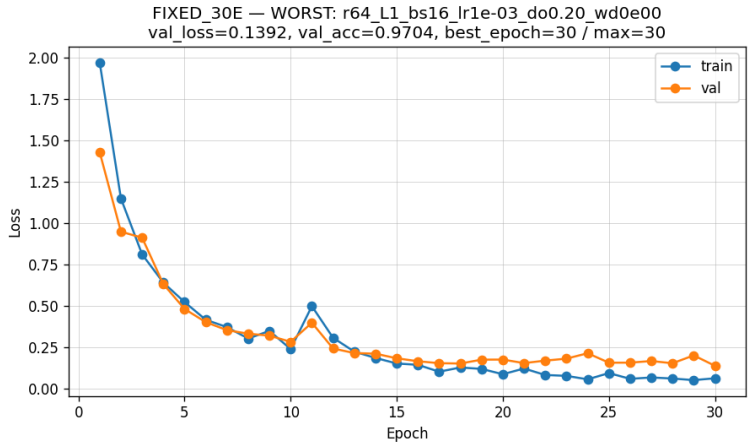
Φάση 1 : Με σταθερές 30 εποχές (χωρίς early stopping)
Φάση 2 : Με early stopping έως 30 εποχές με πρόωρη διακοπή αν η απώλεια επικύρωσης (VAL loss) δεν βελτιώνεται. (patience=3, min_delta=1e−4)

Τα διαγράμματα φαίνονται παρακάτω:

Σταθερές 30 εποχές (No early stopping):

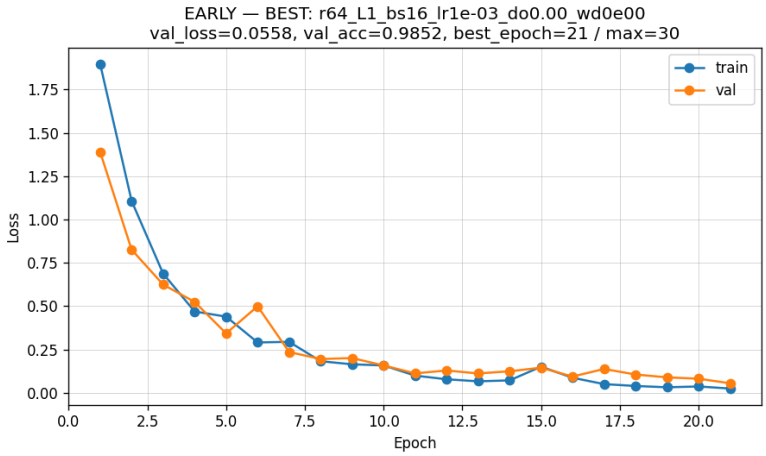


Εικόνα 12: Καλύτερο Μοντέλο στα train-val sets no early stopping/fixed 30 epochs

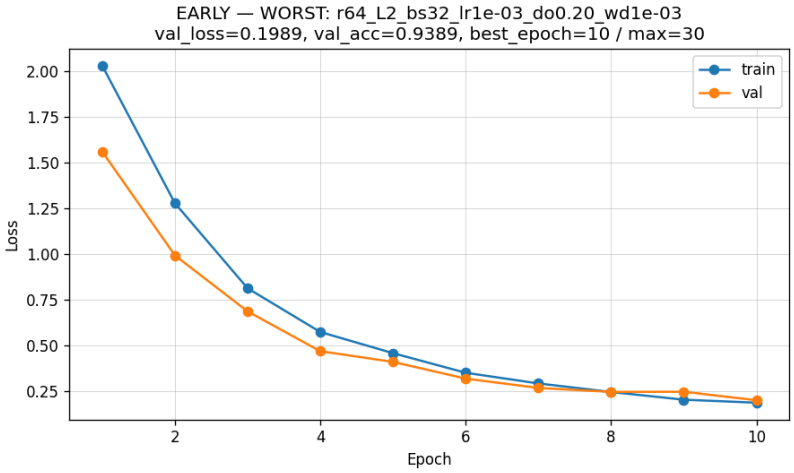


Εικόνα 13: Χειρότερο Μοντέλο στα train-val sets no early stopping/fixed 30 epochs

Early stopping:



Εικόνα 14: Καλύετρο Μοντέλο στα train-val sets με early stopping



Εικόνα 15: Χειρότερο Μοντέλο στα train-val sets με early stopping

Η χρήση του Early Stopping είναι σωστή γιατί βοηθά στην πρόληψη της υπερπροσαρμογής (overfitting) κατά την εκπαίδευση ενός μοντέλου. Καθώς το μοντέλο εκπαιδεύεται, το training loss) μειώνεται, αλλά μετά από κάποιο σημείο το validation loss αρχίζει να αυξάνεται —το μοντέλο "μαθαίνει" υπερβολικά καλά τα δεδομένα εκπαίδευσης και αποτυγχάνει να γενικεύσει. Το Early Stopping σταματά αυτόματα την εκπαίδευση όταν η απόδοση στο σύνολο επικύρωσης πάψει να βελτιώνεται (π.χ. για μερικές συνεχόμενες εποχές).

Εξασφαλίζει δηλαδή ότι «σταματάμε» όταν το μοντέλο είναι στο καλύτερό του σημείο, πριν αρχίσει να χειροτερεύει. Παρόλα αυτά δεν είναι σίγουρο πως το μοντέλο με αυτές τις επιπλέον παραμέτρους θα έχει την καλύτερη απόδοση σε κάποιο test set.

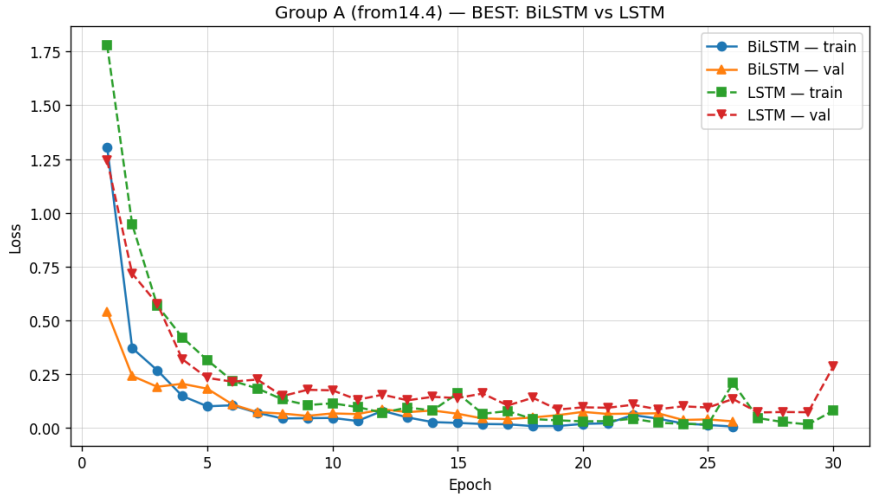
14.7:

Έγινε αλλαγή των LSTM μοντέλων από τα προηγούμενα ερωτήματα σε BiLSTM.
Συγκρίνονται τα μοντέλα διατηρώντας ίδιες τις υπόλοιπες υπερπαραμέτρους ώστε η διαφορά να οφείλεται μόνο στην αμφίδρομη λειτουργία των BiLSTM.
Συγκεκριμένα:
Δημιουργήθηκαν δύο ομάδες μοντέλων:

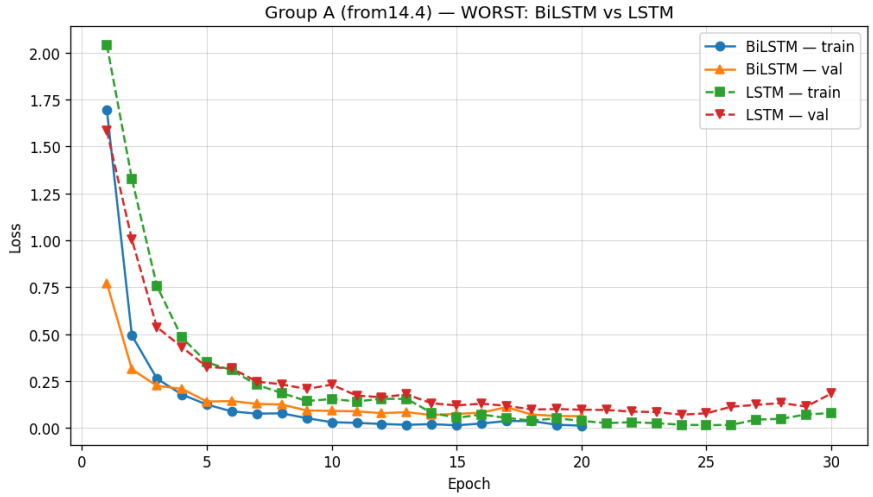
- **Ομάδα Α:** Τα 5 καλύτερα μοντέλα του Βήματος 14.4 (χωρίς regularization ή early stopping).
- **Ομάδα Β:** Τα 5 καλύτερα μοντέλα των Βημάτων 14.5–14.6 (με dropout, L2 και πιθανό early stopping).

Κάθε μοντέλο επανεκπαιδεύεται ως BiLSTM (bidirectional=True) με τις ίδιες ρυθμίσεις και καταγράφεται η καλύτερη εποχή ως προς τη validation loss.

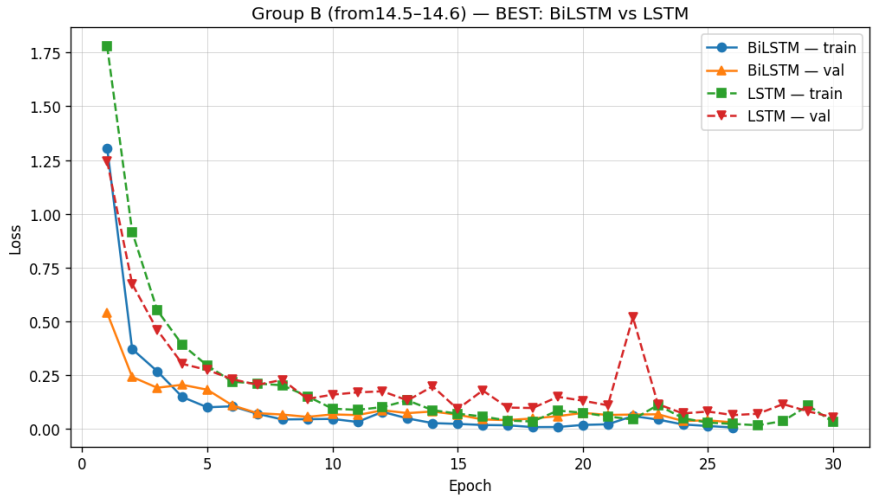
Παρακάτω φαίνονται τα διαγράμματα που συγκρίνουν καμπύλες εκπαίδευσης/επικύρωσης.



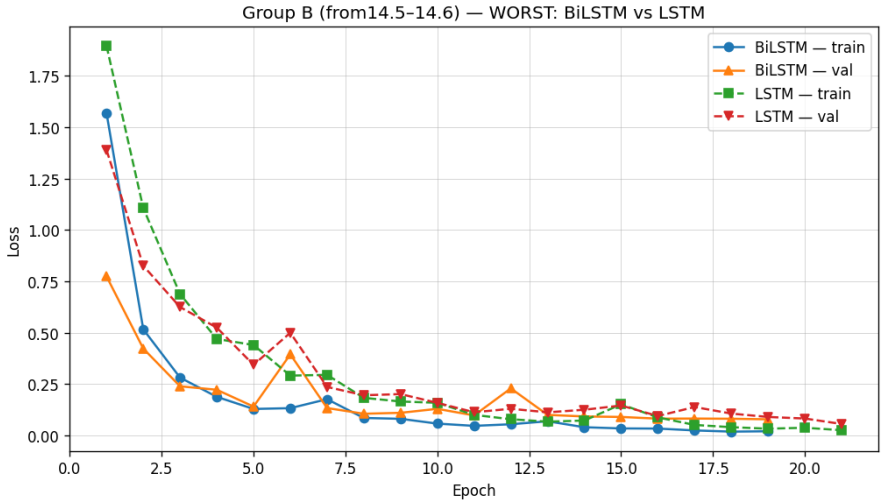
Εικόνα 16: Σύγκριση του καλύτερου εκ των 5 LSTM από το ερώτημα 14.4 (χωρίς L2 ή early stopping) με το αντίστοιχο του BiLSTM



Εικόνα 17: Σύγκριση του χειρότερου εκ των 5 LSTM από το ερώτημα 14.4 (χωρίς L2-dropout ή early stopping) με το αντίστοιχο του BiLSTM

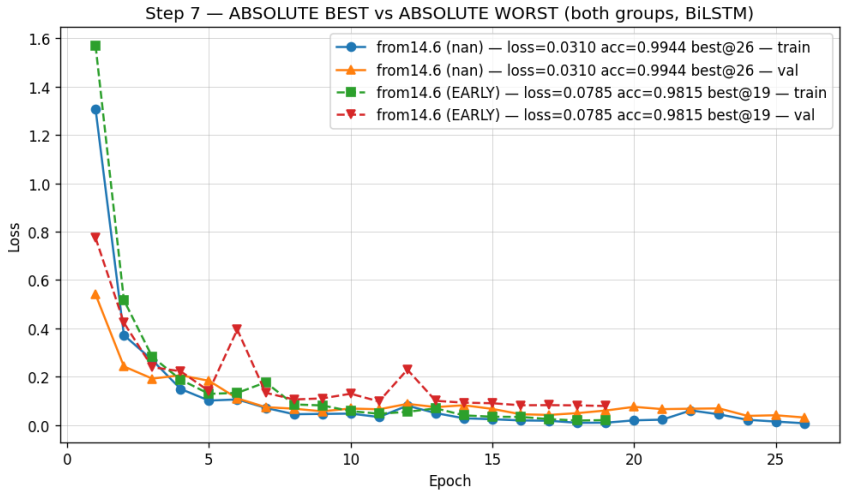


Εικόνα 18: Σύγκριση του καλύτερου εκ των 5 LSTM από το ερώτημα 14.5-14.6 (με L2-dropout- Early stopping και Checkpoints) με το αντίστοιχο του BiLSTM



Εικόνα 19: Σύγκριση του χειρότερου εκ των 5 LSTM από το ερώτημα 14.5-14.6

(με L2-dropout- Early stopping και Checkpoints) με το αντίστοιχο του BiLSTM



Εικόνα 20: Σύγκριση του καλύτερου συνολικά BiLSTM με το χειρότερο

Όπως φαίνεται στα διαγράμματα η χρήση BiLSTM προσδίδει σε κάθε μοντέλο μικρότερες απώλειες.

Σε σχέση με το απλό LSTM είναι ότι στο BiLSTM υπάρχουν δύο ξεχωριστά LSTM επίπεδα που επεξεργάζονται την ακολουθία σε δύο κατευθύνσεις:

Το ένα διαβάζει τα δεδομένα από την αρχή προς το τέλος (forward direction).

Το άλλο διαβάζει από το τέλος προς την αρχή (backward direction).

Εσωτερικά:

Το δίκτυο αποκτά διπλάσιους υπολογισμούς στο επαναλαμβανόμενο επίπεδο (δύο ροές πληροφορίας), αφού κάθε χρονική στιγμή επεξεργάζεται τόσο το παρελθόν όσο και το μέλλον της ακολουθίας.

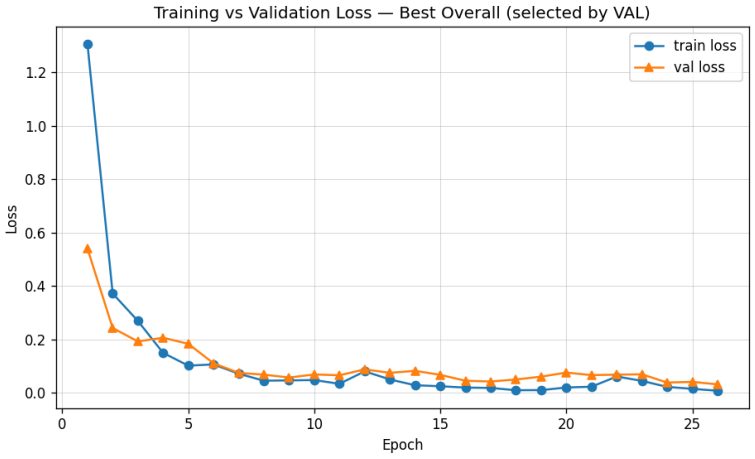
Μας προσδίδει καλύτερη εκπαίδευση όταν η πληροφορία εξαρτάται και από προηγούμενα και από επόμενα χρονικά βήματα (π.χ. αναγνώριση φωνής, ανάλυση ακολουθιών).

Παρόλα αυτά απαιτεί περισσότερη μνήμη-χρόνο και δεν είναι κατάλληλο για εφαρμογές πραγματικού χρόνου.

Τελικό Βήμα Αξιολόγηση του καλύτερου μοντέλου στο Test set

Τέλος, διαλέγουμε το καλύτερο από όλα τα μοντέλα που έχουμε εκπαιδεύσει με βάση το validation set.

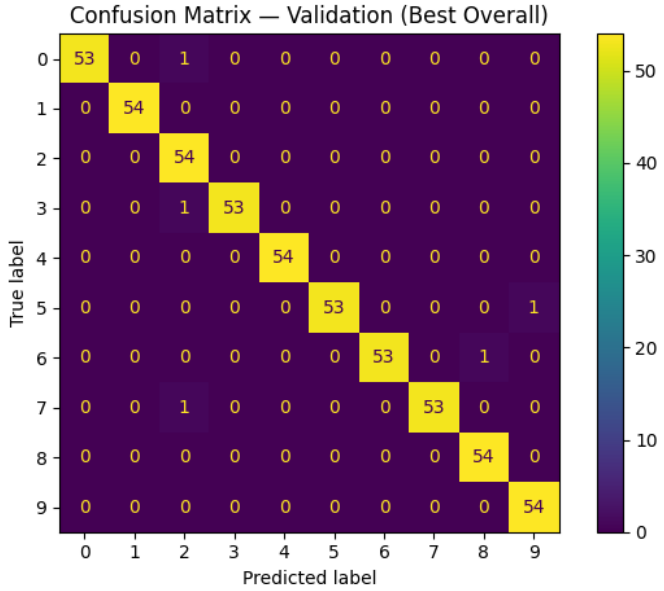
Το καλύτερο μοντέλο προέρχεται από το βήμα 14.7 με αμφίδρομο μοντέλο για LSTM του βήματος 14.4. Δηλαδή το καλύτερο μοντέλο είναι ένα BiLSTM από το ερώτημα 14.4 χωρίς τις επιπλέον παραμέτρους dropout, L2, early stopping των επομένων ερωτημάτων του.
Το παραπάνω αποδεικνύει πως η χρήση επιπλέων παραμέτρων στο μοντέλο δεν θα επιφέρει κατ' ανάγκη και καλύτερη απόδοση, παρόλο που μπορεί εν γένει να βοηθήσει.
Το διάγραμμα του τελικού μοτέλου φαίνεται παρακάτω:



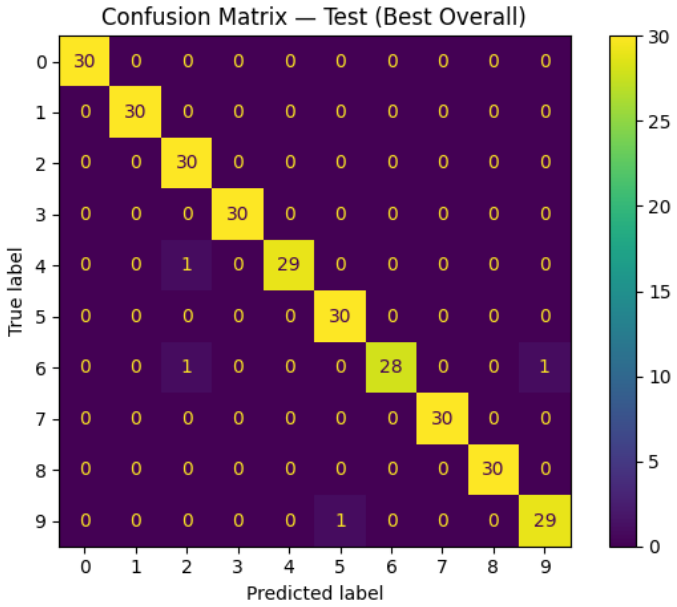
Αποτελέσματα στο Validation Set:
VAL loss: 0.032040
VAL accuracy: 99.0741%

Αποτελέσματα στο test Set:
TEST loss: 0.045889
TEST accuracy: 98.6667%

Παρακάτω φαίνονται και τα confusion matrices:



Εικόνα 21: Confusion Matrix για δοκιμή του καλύτερου μοντέλου με το Validation Set



Εικόνα 22: Confusion Matrix για δοκιμή του καλύτερου μοντέλου με το Test Set

Το μοντέλο έχει εξαιρετική απόδοση.

Όπως φαίνεται από τα παραπάνω διαγράμματα έχει ελάχιστες λάθος ταξινομήσεις.

Βιβλιογραφία

Διαμαντάρας, Κ., & Μπότσης, Δ. Α. (2019). Μηχανική μάθηση. Αθήνα: Κλειδάριθμος. ISBN: 978-960-461-995-5.
https://www.researchgate.net/publication/257365356_Speaker_Recognition_for_Mobile_User_Authentication_An_Android_Solution/figures?lo=1

Bishop, C. M. (2019). *Αναγνώριση προτύπων και μηχανική μάθηση*. Εκδόσεις Fountas. ISBN: 9789603307907

Σ.Η.Μ.Μ.Υ. Ε.Μ.Π.
Νοέμβριος 2025