ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Τμήμα Πληροφορικής



ΕΠΕΞΕΡΓΑΣΙΑ ΣΗΜΑΤΩΝ ΦΩΝΗΣ ΚΑΙ ΗΧΟΥ

***Εργασία Εαρινού Εξαμήνου 2024-2025***

*Μανωλά Αναστασία*

*Π21091*

Πειραιάς, *Μάιος 2025*

Περιεχόμενα

[1 Τεκμηρίωση κώδικα 3](#_Toc198491319)

[1.1 Αρχείο extract\_features.py 3](#_Toc198491320)

[1.2 Αρχείο LeastSquaresClassifier.py 4](#_Toc198491321)

[1.3 Αρχείο mlp\_classifier.py 5](#_Toc198491322)

[1.4 Αρχείο TrainingMLP.py 6](#_Toc198491323)

[1.5 Αρχείο post\_process.py 7](#_Toc198491324)

[1.6 Αρχείο main.py 8](#_Toc198491325)

[2 Σχολιασμός επιδόσεων ταξινομητών : Least Squares και MLP τριών επιπέδων 9](#_Toc198491326)

1. Τεκμηρίωση κώδικα
   1. Αρχείο extract\_features.py

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, λογισμικό

Το περιεχόμενο που δημιουργείται από τεχνολογία AI ενδέχεται να είναι εσφαλμένο.

extract\_features(file\_path, sr=16000, frame\_len=0.025, hop\_len=0.01, n\_mfcc=13)

Η συνάρτηση αυτή εκτελεί προεπεξεργασία σε ένα ηχητικό αρχείο και επιστρέφει έναν πίνακα χαρακτηριστικών. Δέχεται ως είσοδο :

* file\_path (str): Το μονοπάτι προς το ηχητικό αρχείο (.wav) προς επεξεργασία.
* sr (int, προεπιλογή 16000): Συχνότητα δειγματοληψίας στην οποία γίνεται επαναδειγματοληψία του ήχου.
* frame\_len (float, προεπιλογή 0.025): Διάρκεια κάθε καρέ (frame) σε δευτερόλεπτα.
* hop\_len (float, προεπιλογή 0.01): Χρονική απόσταση μεταξύ διαδοχικών καρέ.
* n\_mfcc (int, προεπιλογή 13): Αριθμός των MFCC συντελεστών που θα εξαχθούν.

Και επιστρέφει πίνακα όπου κάθε γραμμή αντιστοιχεί στα MFCCs ενός frame. Η συνάρτηση μετατρέπει το ηχητικό σήμα σε μια αναπαράσταση χαμηλής διάστασης.

create\_dataset(speech\_dir, noise\_dir)

Η συνάρτηση create\_dataset δημιουργεί ένα πλήρες σύνολο δεδομένων από δύο καταλόγους: έναν με δείγματα ομιλίας και έναν με δείγματα θορύβου. Δέχεται ως είσοδο ένα φάκελο που περιέχει .wav αρχεία ομιλίας (speech\_dir (str)) και έναν φάκελο που περιέχει .wav αρχεία θορύβου. Για κάθε αρχείο στους δύο φακέλους, εξάγονται MFCC χαρακτηριστικά με τη συνάρτηση extract\_features. Τα δείγματα ομιλίας επισημαίνονται με ετικέτα 1 (foreground) και τα δείγματα θορύβου με ετικέτα 0 (background). Τα χαρακτηριστικά στοιβάζονται κάθετα και οι ετικέτες οριζόντια, δημιουργώντας ένα ενιαίο dataset. Η συνάρτηση επιστρέφει έναν πίνακα χαρακτηριστικών με σχήμα (N, D), όπου N είναι το πλήθος των frames και D οι MFCC συντελεστές και έναν πίνακα ετικετών σχήματος (N,), με δυαδικές τιμές (0 ή 1).

* 1. Αρχείο LeastSquaresClassifier.py

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, λογισμικό, λογισμικό πολυμέσων

Το περιεχόμενο που δημιουργείται από τεχνολογία AI ενδέχεται να είναι εσφαλμένο.

Το αρχείο LeastSquaresClassifier.py υλοποιεί έναν απλό γραμμικό ταξινομητή βασισμένο στη μέθοδο των Ελαχίστων Τετραγώνων (Least Squares).

fit(X, y)

Η μέθοδος fit υπολογίζει τα βάρη (weights) του μοντέλου, χρησιμοποιώντας ψευδοαντίστροφη του κανονικοποιημένου πίνακα εισόδου. Δέχεται ως είσοδο έναν πίνακα χαρακτηριστικών με σχήμα (N, D), όπου N είναι το πλήθος των δειγμάτων και D ο αριθμός χαρακτηριστικών, και έναν δυαδικό διανυσματικό πίνακα ετικετών με σχήμα (N,). Στη μέθοδο προστίθεται ένα bias term στη μορφή μιας επιπλέον στήλης με 1s στο X και υπολογίζεται η λύση ελαχίστων τετραγώνων. Τέλος αποθηκεύει τα βάρη του μοντέλου στη μεταβλητή self.weights.

predict(X)

Η predict εφαρμόζει το εκπαιδευμένο μοντέλο σε νέα δεδομένα για να προβλέψει τις αντίστοιχες κλάσεις. Δέχεται ως είσοδο έναν πίνακα χαρακτηριστικών, χωρίς bias. Προστίθεται bias column στο X, ώστε να αντιστοιχεί στις διαστάσεις των weights και υπολογίζεται η έξοδος μέσω γραμμικού πολλαπλασιασμού. Τέλος, εφαρμόζεται κατώφλι (threshold) 0.5, τιμές μεγαλύτερες από 0.5 ταξινομούνται ως 1 (θετική κλάση), αλλιώς ως 0, Η μέθοδος επιστρέφει πίνακας προβλεπόμενων ετικετών, δυαδικού τύπου (0 ή 1).

* 1. Αρχείο mlp\_classifier.py

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, λογισμικό, γραμματοσειρά

Το περιεχόμενο που δημιουργείται από τεχνολογία AI ενδέχεται να είναι εσφαλμένο.

Το αρχείο mlp\_classifier.py περιέχει την υλοποίηση ενός Ταξινομητή MLP, με χρήση PyTorch.

init(self, input\_size, hidden1=128, hidden2=64, hidden3=32)

Ο constructor της κλάσης ορίζει την αρχιτεκτονική του MLP. Δέχεται ως παραμέτρους :

* input\_size (int): Το πλήθος των χαρακτηριστικών εισόδου
* hidden1 (int, προεπιλογή 128): Αριθμός νευρώνων στο πρώτο επίπεδο
* hidden2 (int, προεπιλογή 64): Αριθμός νευρώνων στο δεύτερο επίπεδο
* hidden3 (int, προεπιλογή 32): Αριθμός νευρώνων στο τρίτο επίπεδο

Το δίκτυο υλοποιείται ως nn.Sequential αλυσίδα, με τη ακόλουθη σειρά:

1. Πλήρως συνδεδεμένο επίπεδο: input\_size → hidden1
2. ReLU ενεργοποίηση
3. hidden1 → hidden2
4. ReLU ενεργοποίηση
5. hidden2 → hidden3
6. ReLU ενεργοποίηση
7. hidden3 → 1 (έξοδος)
8. Sigmoid ενεργοποίηση (για δυαδική ταξινόμηση)

forward(self, x)

Η μέθοδος αυτή καθορίζει τη ροή των δεδομένων μέσα από το δίκτυο (δηλαδή, το forward pass). Δέχεται ως είσοδο δεδομένα διαστάσεων (N, input\_size), όπου N το πλήθος των δειγμάτων και επιστρέφει πιθανότητες για την κλάση 1 (μία έξοδος ανά δείγμα) με τιμές στο διάστημα [0, 1].

* 1. Αρχείο TrainingMLP.py

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης

Το περιεχόμενο που δημιουργείται από τεχνολογία AI ενδέχεται να είναι εσφαλμένο.

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμματοσειρά

Το περιεχόμενο που δημιουργείται από τεχνολογία AI ενδέχεται να είναι εσφαλμένο.

train\_mlp(X, y, epochs, batch\_size, lr)

Η συνάρτηση train\_mlp εκπαιδεύει έναν MLP ταξινομητή, με χρήση της βιβλιοθήκης PyTorch. Η συνάρτηση περιλαμβάνει στάδια προεπεξεργασίας, εκπαίδευσης και αξιολόγησης και επιστρέφει το εκπαιδευμένο μοντέλο και τον κανονικοποιητή εισόδων. Αρχικά, δέχεται ως είσοδο:

* X (np.ndarray): Πίνακας χαρακτηριστικών διαστάσεων (N, D), όπου N είναι το πλήθος των δειγμάτων και D το πλήθος των χαρακτηριστικών.
* y (np.ndarray): Πίνακας ετικετών διαστάσεων (N,), με δυαδικές τιμές (0 ή 1) για κάθε δείγμα.
* epochs (int): Αριθμός epoch εκπαίδευσης.
* batch\_size (int): Μέγεθος batch για την εκπαίδευση.
* lr (float): Ρυθμός εκμάθησης (learning rate) για τον βελτιστοποιητή Adam.

Η διαδικασία που ακολουθείται κατά την εκπαίδευση του μοντέλου περιλαμβάνει μία σειρά από στάδια. Αρχικά, τα δεδομένα χωρίζονται σε δύο υποσύνολα, ένα σύνολο εκπαίδευσης και ένα σύνολο testing, με αναλογία 80% - 20%. Στη συνέχεια, εφαρμόζεται κανονικοποίηση στα χαρακτηριστικά εισόδου με χρήση του μετασχηματισμού *StandardScaler*, έτσι ώστε τα δεδομένα να έχουν μέση τιμή μηδέν και μοναδιαία διασπορά. Μετά την κανονικοποίηση, τα δεδομένα μετατρέπονται σε αντικείμενα τύπου *tensor* με τύπο torch.float32. Το σύνολο εκπαίδευσης μετατρέπεται σε δομή TensorDataset και στη συνέχεια χρησιμοποιείται DataLoader για τη φόρτωση των δεδομένων σε μικρά *batche*, κατάλληλες για την εκπαίδευση του μοντέλου. Το μοντέλο ορίζεται μέσω της κλάσης MLPClassifier, η οποία υλοποιείται στο αρχείο mlp\_classifier.py και παρέχει δυαδική έξοδο μέσω της συνάρτησης ενεργοποίησης sigmoid. Ως συνάρτηση κόστους χρησιμοποιείται η *Binary Cross-Entropy Loss.* Η εκπαίδευση του μοντέλου πραγματοποιείται για προκαθορισμένο αριθμό epoch. Σε κάθε epoch, το μοντέλο επεξεργάζεται όλα τα δεδομένα εκπαίδευσης σε παρτίδες και υπολογίζει τη μέση τιμή του κόστους. Τέλος, πραγματοποιείται αξιολόγηση της απόδοσης του μοντέλου στο σύνολο επικύρωσης, όπου οι τιμές εξόδου της συνάρτησης sigmoid στρογγυλοποιούνται σε 0 ή 1, και υπολογίζεται η ακρίβεια (accuracy) της ταξινόμησης.

* 1. Αρχείο post\_process.py

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, λογισμικό, λογισμικό πολυμέσων

Το περιεχόμενο που δημιουργείται από τεχνολογία AI ενδέχεται να είναι εσφαλμένο.

Η συνάρτηση post\_process\_predictions εκτελεί μετα-επεξεργασία σε ακολουθία δυαδικών προβλέψεων, με σκοπό την εξαγωγή χρονικών τμημάτων (segments) σταθερής ετικέτας και την εξομάλυνση του αποτελέσματος. Δέχεται ως είσοδο:

* preds (array-like): Ακολουθία δυαδικών τιμών (0 ή 1), που αντιστοιχούν στις προβλέψεις του μοντέλου για κάθε frame.
* frame\_hop (float, προεπιλογή 0.01): Χρονική απόσταση (σε δευτερόλεπτα) μεταξύ διαδοχικών frames.
* min\_duration (float, προεπιλογή 0.2): Ελάχιστη διάρκεια (σε δευτερόλεπτα) που πρέπει να έχει ένα segment ώστε να θεωρείται έγκυρο και να συμπεριληφθεί στο αποτέλεσμα.
* audiofile\_name (str, προεπιλογή "File1"): Όνομα του ηχητικού αρχείου στο οποίο αναφέρονται οι προβλέψεις. Χρησιμοποιείται προαιρετικά για καταγραφή ή παρουσίαση.

Η συνάρτηση εφαρμόζει median filter με μέγεθος kernel ίσο με 5 για να μειώσει τον θόρυβο στις προβλέψεις και να ενισχύσει τη συνοχή τους. Έπειτα, εντοπίζει τα χρονικά διαστήματα κατά τα οποία η πρόβλεψη παραμένει σταθερή (0 ή 1) και υπολογίζει τη διάρκεια καθενός. Όσα segments έχουν διάρκεια μικρότερη από το όριο min\_duration απορρίπτονται. Τέλος, πραγματοποιείται διόρθωση των ενδιάμεσων κενών μεταξύ διαδοχικών segments, ώστε να εξασφαλιστεί η χρονική συνέχεια στην έξοδο. Η συνάρτηση επιστρέφει μια λίστα με χρονικά τμήματα, όπου κάθε στοιχείο είναι ένα tuple της μορφής (start\_time, end\_time, class\_label), που αντιστοιχεί στην έναρξη, λήξη και κατηγορία (0 ή 1) κάθε segment.

* 1. Αρχείο main.py

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, λογισμικό

Το περιεχόμενο που δημιουργείται από τεχνολογία AI ενδέχεται να είναι εσφαλμένο.

Το αρχείο main.py αποτελεί το κύριο πρόγραμμα του συστήματος και ενοποιεί όλα τα επιμέρους στάδια της διαδικασίας ταξινόμησης ήχου: εξαγωγή χαρακτηριστικών, εκπαίδευση και αξιολόγηση μοντέλων, πρόβλεψη σε νέο ηχητικό αρχείο και μετα-επεξεργασία των αποτελεσμάτων. Περιλαμβάνει τα εξής βασικά βήματα:

1. **Εξαγωγή χαρακτηριστικών**:  
   Χρησιμοποιείται η συνάρτηση create\_dataset από το αρχείο extract\_features.py για τον διαχωρισμό των δεδομένων σε σύνολο εκπαίδευσης και test με τη χρήση της train\_test\_split.
2. **Εκπαίδευση με Least Squares Classifier**:  
   Τα δεδομένα κανονικοποιούνται με StandardScaler και χρησιμοποιούνται για την εκπαίδευση ενός *Least Squares Classifier*, που υλοποιείται στο αρχείο LeastSquaresClassifier.py. Το μοντέλο εκπαιδεύεται και αξιολογείται στο σύνολο επικύρωσης, με υπολογισμό ακρίβειας μέσω της accuracy\_score.
3. **Εκπαίδευση με MLP**:  
   Το σύνολο δεδομένων X, y χρησιμοποιείται για την εκπαίδευση ενός MLP μέσω της συνάρτησης train\_mlp από το αρχείο TrainingMLP.py. Η εκπαίδευση πραγματοποιείται με : αριθμός epoch 15, μέγεθος batch 128, και ρυθμός εκμάθησης 0.001.
4. **Πρόβλεψη σε νέο αρχείο ήχου**:  
   Το αρχείο επεξεργάζεται με τη συνάρτηση extract\_features, ώστε να εξαχθούν τα MFCC χαρακτηριστικά του. Στη συνέχεια, τα χαρακτηριστικά κανονικοποιούνται με βάση τον κανονικοποιητή scaler\_mlp που προέκυψε από την εκπαίδευση. Τα δεδομένα μετατρέπονται σε PyTorch tensors και διέρχονται από το εκπαιδευμένο MLP μοντέλο για την παραγωγή προβλέψεων.
5. **Μετα-επεξεργασία και αποθήκευση αποτελεσμάτων**:  
   Οι δυαδικές προβλέψεις υποβάλλονται σε μετα-επεξεργασία με τη συνάρτηση post\_process\_predictions από το αρχείο post\_process.py, ώστε να εξαχθούν συνεκτικά χρονικά segments. Τα αποτελέσματα καταγράφονται σε πίνακα δεδομένων και αποθηκεύονται σε αρχείο CSV με όνομα "output.csv".
6. Σχολιασμός επιδόσεων ταξινομητών : Least Squares και MLP τριών επιπέδων

Στο πλαίσιο της παρούσας εργασίας υλοποιήθηκαν οι ταξινομητές Least Squares και MLP. Παρακάτω παρατίθενται οι παρατηρήσεις που προέκυψαν από την σύγκριση των επιδόσεων τους.

Ο ταξινομητής Least Squares βασίζεται σε γραμμική προσέγγιση, όπου το πρόβλημα της ταξινόμησης μετατρέπεται σε πρόβλημα παλινδρόμησης μέσω ελαχίστων τετραγώνων. Βασικό πλεονέκτημα του αποτελεί η ταχύτητα εκπαίδευσης και η ευκολία υλοποίησης. Κατά την εκτέλεση του προγράμματος παρατηρούμε ότι ο ταξινομητής Least Squares πετυχαίνει ακρίβεια 95,27% ( Least Squares Accuracy: 0.9527), γεγονός που δηλώνει ικανοποιητική απόδοση. Η ικανότητα του να μοντελοποιήσει σύνθετες και μη γραμμικές σχέσεις, που συχνά εμφανίζονται σε πραγματικά δεδομένα, περιορίζεται από την γραμμική φύση του ταξινομητή.

Ο ταξινομητής MLP (Multilayer Perceptron) τριών επιπέδων ανήκει στην κατηγορία των νευρωνικών δικτύων και βασίζεται σε μια πολυεπίπεδη αρχιτεκτονική με πλήρως συνδεδεμένα επίπεδα και μη γραμμικές συναρτήσεις ενεργοποίησης. Η συγκεκριμένη υλοποίηση περιλαμβάνει τρία κρυφά επίπεδα με 128, 64 και 32 νευρώνες αντίστοιχα, ενώ στην έξοδο χρησιμοποιείται η συνάρτηση Sigmoid, κατάλληλη για προβλήματα δυαδικής ταξινόμησης.

Σημαντικό προτέρημα του MLP αποτελεί η ικανότητά του να μοντελοποιεί πολύπλοκες, μη γραμμικές σχέσεις μεταξύ των χαρακτηριστικών εισόδου. Αυτό του επιτρέπει να αποδίδει εξαιρετικά σε δεδομένα που δεν είναι γραμμικά διαχωρίσιμα, όπως συμβαίνει συχνά στα ηχητικά σήματα. Στην πράξη, το μοντέλο παρουσίασε συνεχή βελτίωση κατά την εκπαίδευση, με το loss να μειώνεται σταθερά και την τελική ακρίβεια επικύρωσης να φτάνει το εντυπωσιακό 98.93%.