Inner Speech Recognition

Presentation by Moulik & Shubham

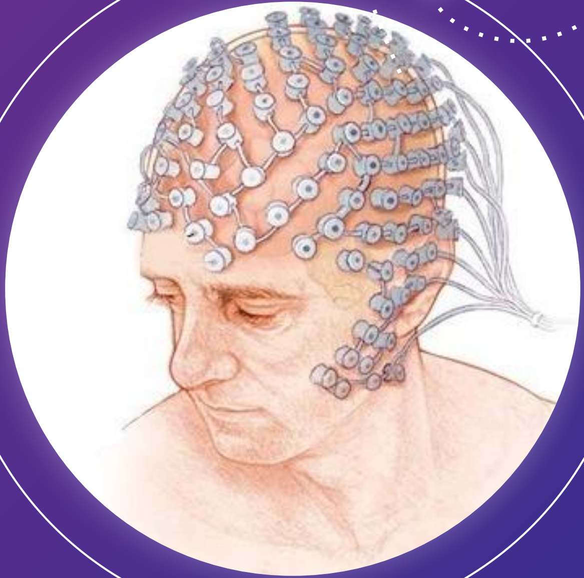
Index

"Your brain does not manufacture thoughts. Your thoughts shape neural networks."

— Deepak Chopra

Surface Electroencephalography

Surface electroencephalography (EEG) is a standard and noninvasive method for measuring electrical brain activity. Thanks to recent advances in artificial intelligence, automatic detection of brain patterns has vastly improved, resulting in faster, more reliable, and accessible Brain-Computer Interfaces. One fascinating phenomenon gaining attention is "inner speech," where individuals can execute commands simply by thinking about them, offering a natural means of controlling external devices.



Thinking Out Loud (Article)

"Thinking out loud, an open-access EEG-based BCI dataset for inner speech recognition" was published in the journal Scientific Data in 2022. The authors of the article are Nicolás Nieto, Victoria Peterson, Hugo Rufiner, Juan Kamienkowski, and Ruben Spies.

The article presents a new dataset of electroencephalography (EEG) data that was collected from 10 participants while they were thinking about four different commands: "up", "down", "left", and "right". The data was collected using a 136-channel EEG system. The authors of the article argue that this dataset is a valuable resource for researchers who are interested in developing new techniques for inner speech recognition. They also suggest that the dataset could be used to better understand the brain mechanisms underlying inner speech.

Background & Summary

Brain-Computer Interfaces (BCIs) utilize non-invasive electroencephalography (EEG) to decode neural activity, enabling control commands for devices. Different paradigms have been explored for communication, including silent, imagined, and inner speech. Inner speech involves thinking with an auditory imagery of one's own "voice." Despite datasets for imagined speech and motor imagery, publicly accessible EEG datasets for inner speech are limited. A multi speech-related BCI dataset was compiled, providing valuable data for future research and applications, with over 9 hours of continuous EEG recordings from ten participants performing mental tasks in three conditions: inner speech, pronounced speech, and visualized actions.

Methods (Participants)

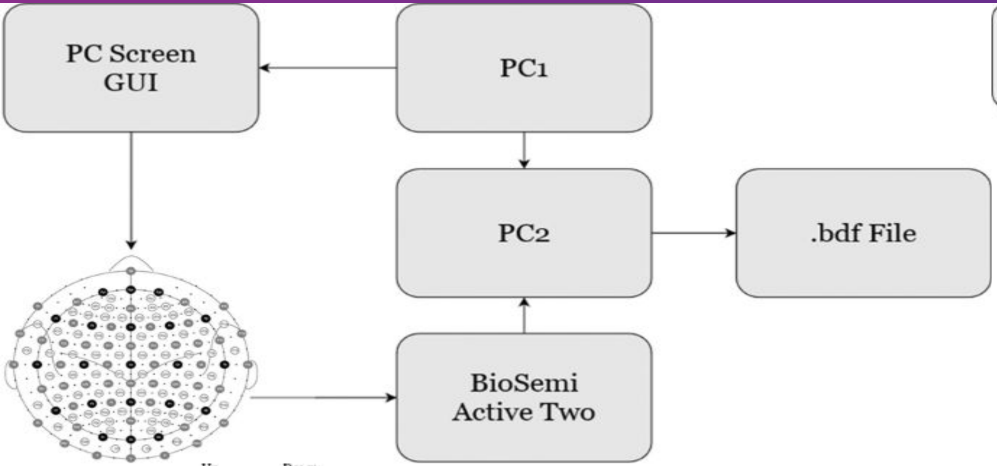
Participant	Self-declared gender	Age	Dominance	Native language
sub-01	Female	56	Right	Spanish
sub-02	Male	50	Right	Spanish
sub-03	Male	34	Right	Spanish
sub-04	Female	24	Right	Spanish
sub-05	Female	30	Right	Spanish
sub-06	Male	29	Right	Spanish
sub-07	Male	26	Right	Spanish
sub-08	Female	28	Right	Spanish
sub-09	Male	35	Right	Spanish
sub-10	Male	31	Right	Spanish

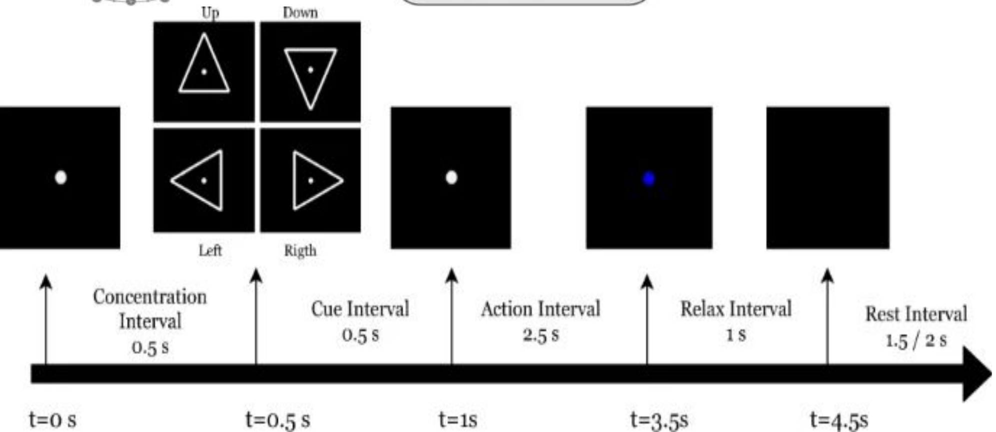
Methods (Procedures)

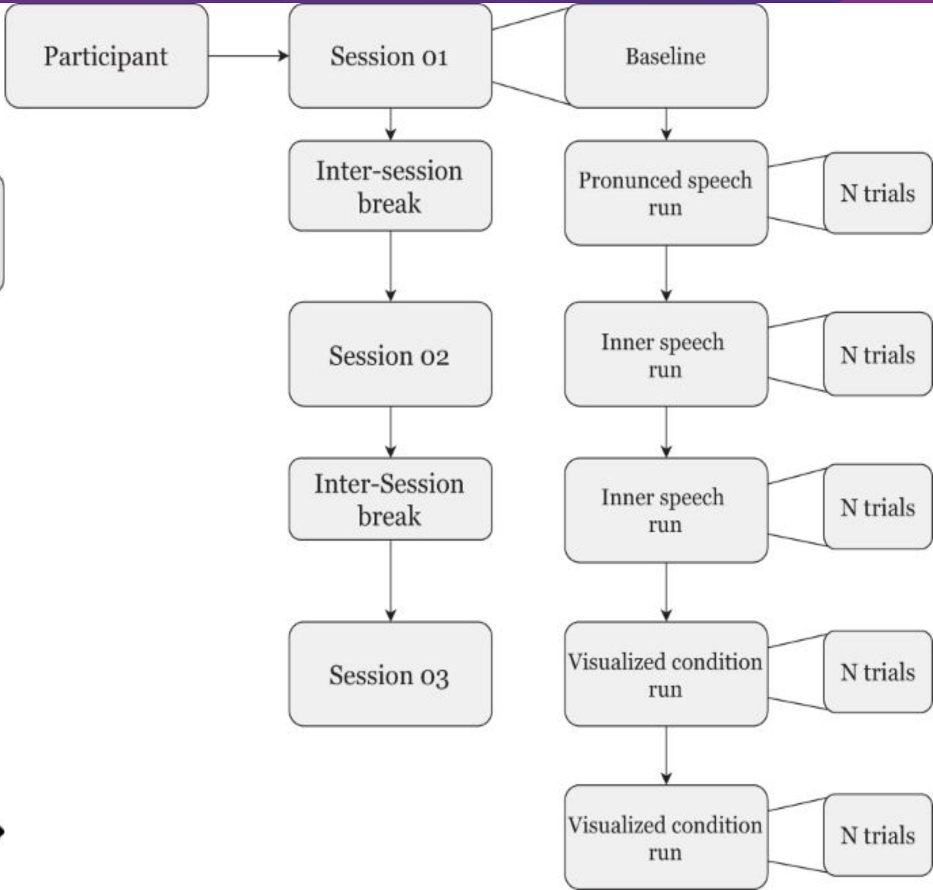
The study involved EEG recordings and Brain-Computer Interface (BCI) control using visual cues and speech conditions. Participants were seated in a shielded room in front of a computer screen. The experiment used Matlab and Psychtoolbox for stimulation. Each participant had three sessions with five runs each, including pronounced speech, inner speech, and visualized conditions. Trials involved concentration, cue, action, and relax intervals. Concentration controls were added randomly to assess attention.

The BCI control application used Spanish words for directions ("arriba," "abajo," "derecha," "izquierda" or "up," "down," "right," "left," respectively). Each trial began with a concentration interval (0.5 s) followed by a cue interval (0.5 s) where a white triangle pointing to a direction was shown. Then, an action interval (2.5 s) allowed the participant to perform the task indicated by the cue. After that, a relax interval (1 s) with a blue circle appeared, and participants were instructed to stop the activity but not blink until the blue circle disappeared. A rest interval of variable duration (1.5 s to 2 s) was given between trials.

Methods (Procedures)







Methods (Data Acquisition)

EEG, EOG, and EMG data were acquired using a BioSemi ActiveTwo system with 128 EEG and 8 external EOG/EMG channels. Electrode positions were predetermined, and impedance was checked before recording. ActiView software was used for recording with a 208 Hz low-pass filter. Data was saved in .bdf files, including EEG, EOG, EMG, and tagged events.

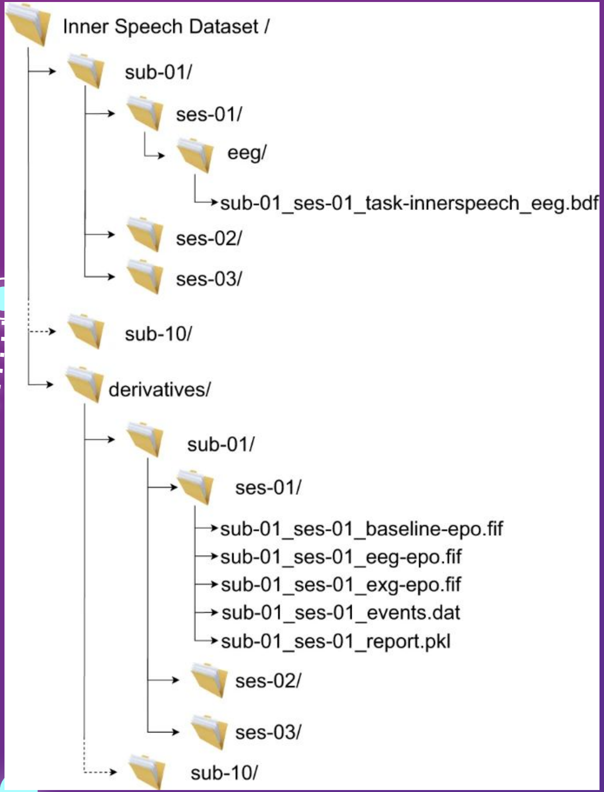
Methods (Interaction Conditions)

The dataset includes three BCI interaction conditions: inner speech, pronounced speech, and visualized condition. Inner speech involves imagining one's own voice giving a direct order to the computer without articulation gestures. Pronounced speech requires audibly repeating the word indicated by the cue. The visualized condition involves mentally moving a circle in the direction of the visual cue, exploring spatial thinking and its neural activation. These conditions aim to understand inner speech generation and its potential for BCI applications.

Methods (Data Processing)

The dataset was processed using Python and the MNE library. Raw data, including EEG, EOG, and EMG, were loaded from .bdf files. Event tags were checked and corrected. Re-referencing eliminated noise and drift. Data were filtered, epoching was performed, and ICA was used to separate EEG sources from artifacts. EMG control detected mouth movements. Ad-hoc tags correction was applied to balance trials by condition for one participant. The processed dataset and code are available for further analysis.

Data Records



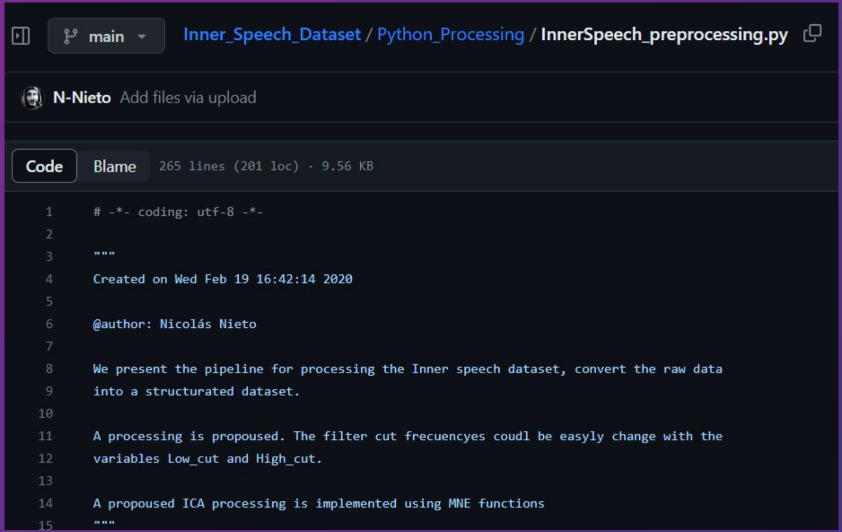
The dataset, accessible at the OpenNeuro repository, consists of EEG recordings organized in ten subfolders, one for each participant's session. The raw data file contains continuous recordings for all 136 channels, with a mean duration of 1554 seconds. Additional processed files include EEG data, external electrodes data, events data, and a report file. The EEG data is stored in.fif format, representing trials in each session with dimensions [Trials × 128 × 1154]. The external electrodes data has dimensions [Trials × 8 × 1154]. The events data is in.dat format, and the baseline data in.fif format, with dimensions [1 × 136 × 3841]. The report file contains general participant information and session processing results.

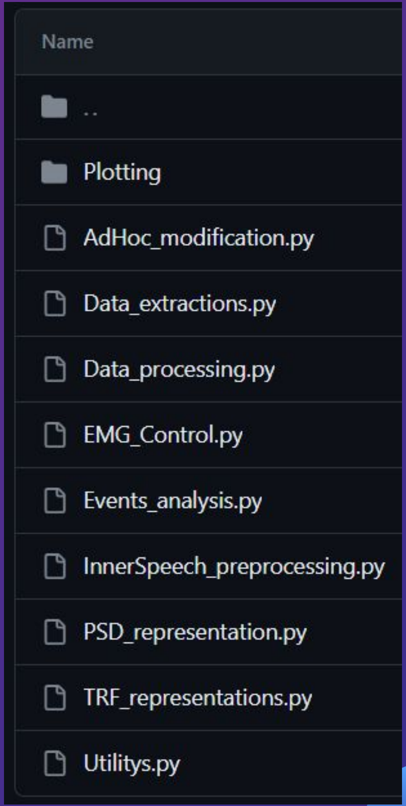
Limitations & Final Remarks

The authors present a novel EEG dataset featuring three speech-related conditions, encompassing 5640 trials and over 9 hours of continuous recording. It stands as the first dataset with native Spanish speakers and a high-density acquisition system (128+8 channels). However, limitations include potential variations in participants' mental activity interpretation and the unclear distinction between imagined and inner speech processes. Nonetheless, the dataset, accompanied by raw and processed data, and implementation codes, holds significant value for the scientific community.

Usage Notes

The processing script was developed in Python 3.7[66], using the MNE-python package v0.21.0[51], NumPy v1.19.2[67], Scipy v.1.5.2[68], Pandas v1.1.2[69] and Pickle v4.0[70]. The main script, "InnerSpeech_processing.py", contains all the described processing steps and it can be modified to obtain different processing results, as desired. In order to facilitate data loading and processing, six more scripts defining functions are also provided.

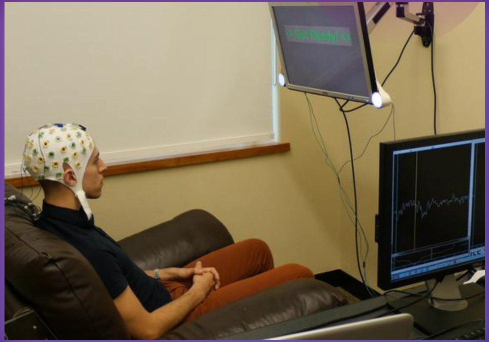


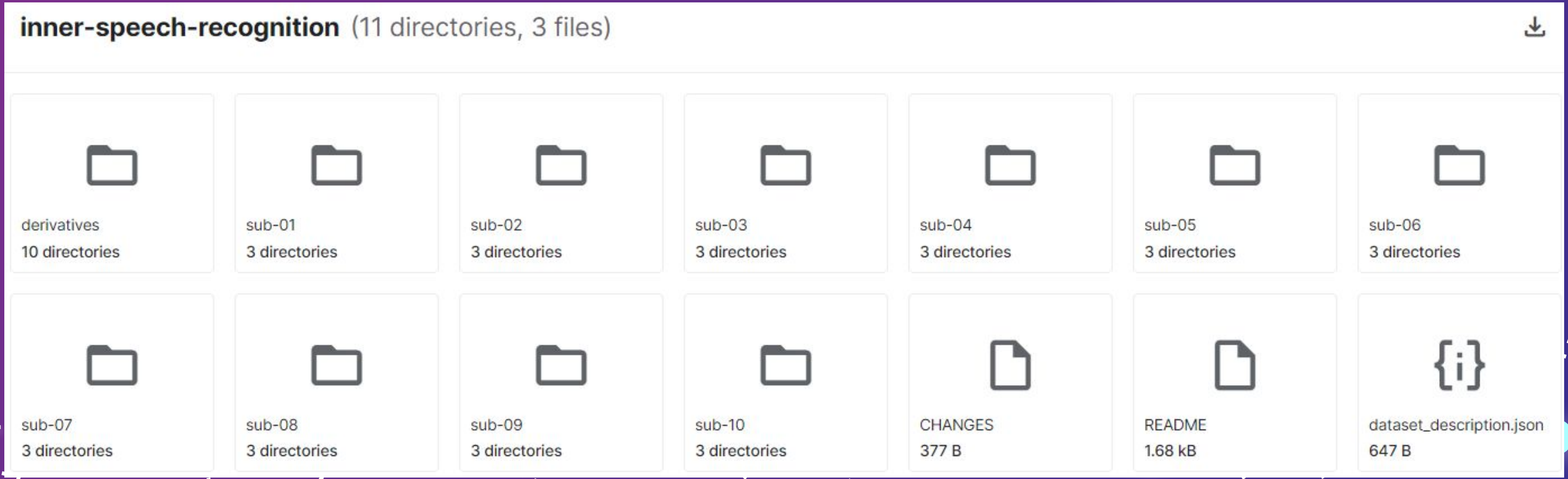


Inner Speech Recognition

Some images of the dataset

(Sourced from Inner Speech Recognition | Kaggle)





Inner Speech Recognition (Dataset)

The Inner Speech Recognition dataset on Kaggle is a collection of EEG recordings of people thinking words. The dataset was created by researchers at the University of Trento in Italy, and it is designed to help train and evaluate algorithms for inner speech recognition.

The dataset contains 100 recordings of people thinking 100 different words. Each recording is 2 seconds long, and the words are spoken in Italian. The dataset also includes a ground-truth transcript of the words that were spoken.

The Inner Speech Recognition dataset is a valuable resource for researchers who are working on inner speech recognition. The dataset is well-curated and easy to use, and it provides a large corpus of data for training and evaluating algorithms.

Key Points

The dataset contains 5640 trials of EEG data.

The trials were collected from 10 participants while they were thinking about four different commands.

The data was collected using a 136-channel EEG system.

The dataset is available for free online.

The dataset could be used to better understand the brain mechanisms underlying inner speech.

The authors argue that the dataset is a valuable resource for researchers who are interested in developing new techniques for inner speech recognition.

Tutorial
(For loading the Inner Speech Database)

Set up - Download and import required libraries

In [1]:

```
#@title Install dependencies
!git clone https://github.com/N-Nieto/Inner_Speech_Dataset -q
!pip3 install mne -q
```

```
|████████████████████████████████| 7.4 MB 21.3 MB/s
```

In [2]:

```
#@title Imports
import os
import mne
import pickle
import random
import warnings
import numpy as np
import matplotlib.pyplot as plt

from google.colab import drive, files

from Inner_Speech_Dataset.Python_Processing.Data_extractions import  Extract_data_from_subject
from Inner_Speech_Dataset.Python_Processing.Data_processing import  Select_time_window, Transform_for_classificator, Split_

np.random.seed(23)

mne.set_log_level(verbose='warning') #to avoid info at terminal
warnings.filterwarnings(action = "ignore", category = DeprecationWarning )
warnings.filterwarnings(action = "ignore", category = FutureWarning )
```

Tutorial
(For loading the Inner Speech Database)

Data Loading.

In [5]:
```
# Mount drive with data. You have to download and store the dataset in your own Drive
drive.mount('/gdrive', force_remount=True)
```

```
Mounted at /gdrive
```

In [6]:
```
### Hyperparameters

# The root dir have to point to the folder that cointains the database
root_dir = "/gdrive/My Drive/..."

# Data Type
datatype = "EEG"

# Sampling rate
fs = 256

# Select the useful par of each trial. Time in seconds
t_start = 1.5
t_end = 3.5

# Subject number
N_S = 1   #[1 to 10]
```

In [7]:
```
#@title Data extraction and processing

# Load all trials for a sigle subject
X, Y = Extract_data_from_subject(root_dir, N_S, datatype)

# Cut usefull time. i.e action interval
X = Select_time_window(X = X, t_start = t_start, t_end = t_end, fs = fs)
```

In [ ]:
```
print("Data shape: [trials x channels x samples]")
print(X.shape) # Trials, channels, samples

print("Labels shape")
print(Y.shape) # Time stamp, class , condition, session
```

Tutorial
(For loading the Inner Speech Database)

Create the different groups for a classifier. A group is created with one condition and one clase.

In [9]:
```
# Conditions to compared
Conditions = [["Inner"],["Inner"]]
# The class for the above condition
Classes    = [  ["Up"] ,["Down"] ]
```

In [10]:
```
# Transform data and keep only the trials of interes
X , Y =  Transform_for_classificator(X, Y, Classes, Conditions)
```

In [11]:
```
print("Final data shape")
print(X.shape)

print("Final labels shape")
print(Y.shape)
```

```
Final data shape
(100, 128, 508)
Final labels shape
(100,)
```

Tutorial
(For running the Inner Speech Database)

- **Install Dependencies:** The notebook begins by cloning the "Inner_Speech_Dataset" repository from GitHub and installing the required libraries. Specifically, it installs the "mne" library using pip.
- **Imports:** The notebook then imports various libraries that will be used throughout the tutorial. Some of the imported libraries include "os," "mne," "pickle," "random," "warnings," "numpy," "matplotlib.pyplot," and some modules from the custom package "Inner_Speech_Dataset.Python_Processing."
- **Data Loading:** The code mounts your Google Drive to access the dataset. The dataset is assumed to be stored in your Google Drive at the specified root directory ("/gdrive/My Drive/..."). The Extract_data_from_subject function is used to load all trials for a single subject from the specified root directory and datatype ("EEG"). The data is stored in the variable "X," and the corresponding labels are stored in "Y."

Tutorial
(For running the Inner Speech Database)

- **Data Processing:** The loaded data is then processed to select a specific time window of interest. The Select_time_window function is used to cut the useful time interval from each trial. The specified time window is from 1.5 seconds to 3.5 seconds. The data is stored in the variable "X."
- **Creating Classifier Groups:** The code creates different groups for a classifier. Each group is defined by one condition and one class. In this example, two groups are created, both comparing the condition "Inner," but with different classes ("Up" and "Down").
- **Transforming Data for Classifier:** The Transform_for_classificator function is used to transform the data and keep only the trials that match the specified classes and conditions. The processed data is stored in the variable "X," and the corresponding labels are stored in "Y."
- **Printing Shapes:** Finally, the code prints the shape of the processed data "X" and the corresponding labels "Y."

Tutorial
(For running the Inner Speech Database)

The final processed data "X" has a shape of (100, 128, 508), which represents 100 trials, 128 channels, and 508 samples. The final labels "Y" have a shape of (100,), indicating 100 data points corresponding to the labels for each trial.

Additions/Improvements

Changes/Improvements

IntegratIng two new functions Select_time_window and Transform_for_classificator

In [1]:

```
# Load all trials for a single subject and extract the data and labels
X, Y = Extract_data_from_subject(root_dir, N_S, datatype)

# Select a specific time window (e.g., from 1.5 to 2.5 seconds) for analysis
t_start = 1.5  # Start time in seconds
t_end = 2.5    # End time in seconds
fs = 256       # Sampling frequency

X = Select_time_window(X, t_start, t_end, fs)

print("Data shape after selecting time window:", X.shape)  # Print the updated data shape

# Create different groups for classification
Conditions = [["Inner"], ["Inner"]]
Classes = [["Up"], ["Down"]]

X, Y = Transform_for_classificator(X, Y, Classes, Conditions)

print("Data shape after creating classifier groups:", X.shape)  # Print the updated data shape
```

Additions/Improvements

In the original code provided in the Jupyter Notebook, we had the initial setup for working with the "Inner Speech" dataset. This included loading the necessary libraries, mounting Google Drive to access the data, and loading data for a specific subject. We then performed some basic data processing using functions like **Extract_data_from_subject**, **Select_time_window**, and **Filter_by_class**.

However, we realized that we could improve our data processing to focus on specific time intervals and make it more suitable for classification tasks. So, we introduced two new powerful functions: **Select_time_window** and **Transform_for_classificator**.

With the **Select_time_window** function, we can now easily narrow down our EEG data to analyze only a specific time window. For example, we can choose to examine brain activity from 1.5 to 2.5 seconds after a particular event. This allows us to zoom in on the most relevant part of the data for our analysis.

The **Transform_for_classificator** function is a game-changer for classification tasks. It helps us organize the data and labels into different groups that we can use for classification purposes. We specify the conditions and classes for each group we want to create. Then, the function cleverly filters the data and labels based on these conditions and assigns a unique number to each group. This simplifies the process of training and testing classification models significantly.

THANKS!