CS 577 – Principles and Techniques of Data Science

Decoding Brain Signals: Building a Classifier for the P300 Signal

Group Members: Noah Bakayou, Pavithra Magendiran, Noam Joseph

Agenda

- Motivation
- What is the P300
- P300 Speller
- Dataset Overview
- Initial Approach
- Improved Approach
- Results
- Future work/Questions

Motivation

Problem Statement:

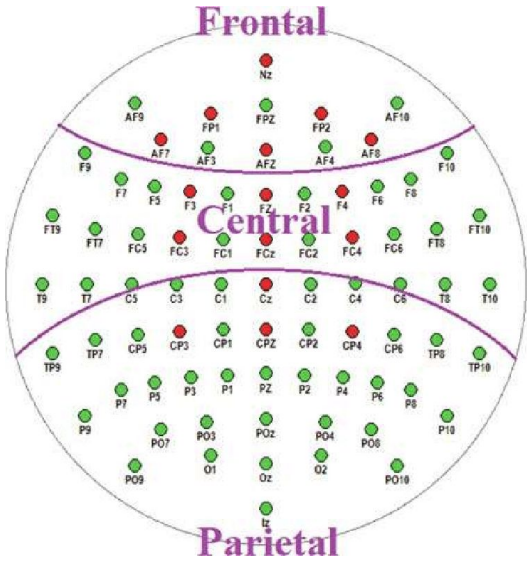
- ➢ Millions of people around the world lose the ability to speak or move due to conditions like ALS, stroke or spinal cord injuries
- ➢ Brain-Computer Interfaces (BCIs) offer a way to restore communication by decoding neural activity directly

Proposed Solution:

- ➢ Allow people with disabilities to communicate and type out words using their mind
- ➢ Create a classifier to enable decoding the brain signal that makes this system possible

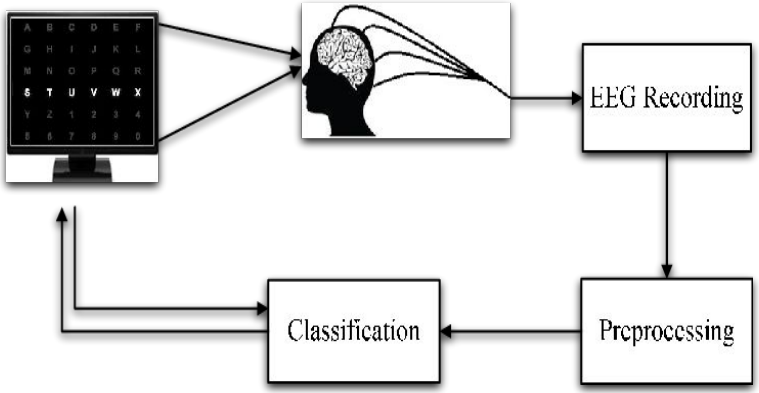
What is EEG?

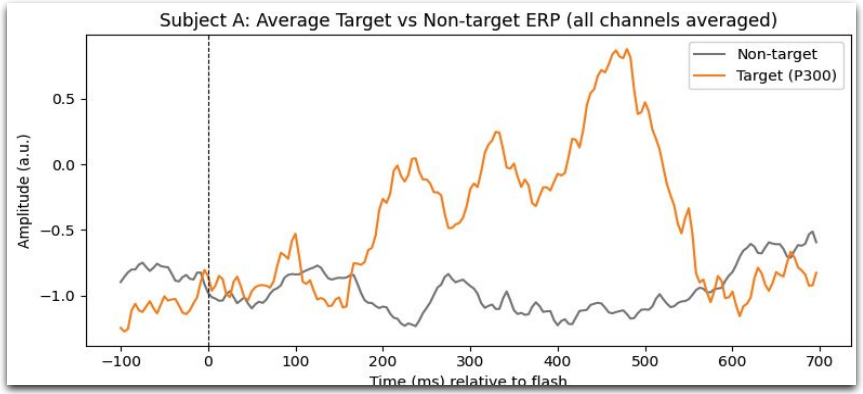
- EEG (Electroencephalography) measures tiny electrical signals produced by brain activity.
- When groups of neurons fire together, they create voltage changes that can be detected on the scalp.
- An EEG cap uses multiple electrodes placed on standardized positions based on the **10–20 international system** (e.g., Cz, Pz, F3).
- The electrodes do not read thoughts, they only detect microvolt-level electrical signals.
- These signals are amplified, digitized, and recorded as data across 64 channels, each representing one scalp location.
- EEG allows us to observe brain responses to events, such as the P300, which appears when a person notices or focuses on a meaningful stimulus.



What is the P300?

- **P300 Event Related Potential (ERP)** - Positive EEG brain response **deflection at ~300 ms** after a person detects a meaningful or rare event
- **It is part of the Oddball paradigm**, where the brain reacts more strongly to infrequent, important stimuli
- The P300 reflects a decision-making process. It produces a brief positive voltage bump strongest over centro-parietal electrodes.
    - The exact P300 shape varies across people because its timing and amplitude depend on individual reaction speed and cognitive processing differences.

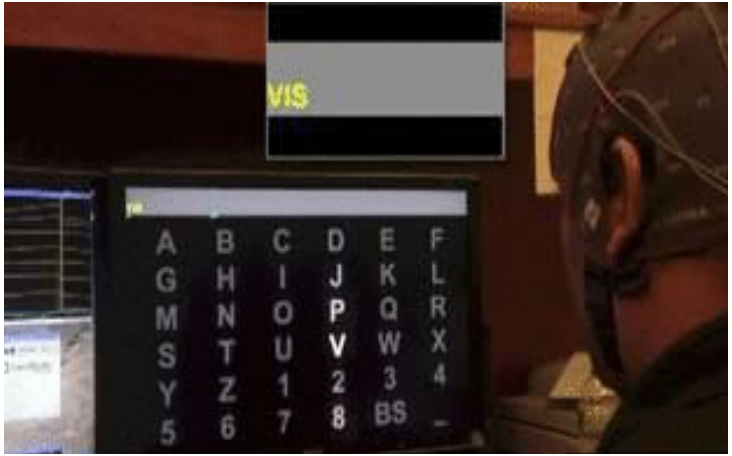




Developed by Farwell & Donchin (1988)

P300 Speller

- A 6×6 matrix of characters that are shown on the screen.
- Rows and columns flash in a random sequence while the user focuses on the intended character.
- When the target row or column flashes, the brain generates a **P300 response**
    - **Target flashes** produce a strong P300 peak because the user is attending to that row/column.
    - **Non-target flashes** do not produce a clear peak
- Each of the 12 row/column codes flashes 15 times per character selection.
- For every flash, the system extracts a flash-locked EEG segment from all 64 channels.
- A classifier evaluates the EEG segments for all 12 codes and identifies:
    - the row with the strongest P300
    - the column with the strongest P300
- The intersection of the detected row and column reveals the character the user intended to type.



Dataset Overview: BCI Competition III P300 Speller

- The EEG is originally stored as a 3-D array: (epochs × full recording length × 64 channels)
  - For Subject A: 85 epochs (one trial per character)
- In each epoch, the speller produces 180 flashes
  - 12 row/column codes × 15 repetitions
  - Each flash has a
    - StimulusCode (which row/column flashed; 1-6, 7-12)
    - StimulusType label indicating whether that flash contained the target character (1 = target, 0 = non-target)
- For every flash, we cut out a short EEG segment from all 64 channels:
  - Sampling Window: 0 ms to +600 ms around the flash
  - Sampling rate: 240 Hz → 1 sample ≈ 4.17 ms
- This produces a flash-locked EEG slice with shape:
  - (window_length × 64 channels) (≈145 samples × 64 channels)
    - Later, we reduce the 64-channel window to 3 channels (‘Cz’, ‘Fc1’, ‘Cpz’).
    - We found that mean positive amplitude difference between target and non target was strongest on "Cz", "FC1", "Cpz" (in line with current research)
- These flash-locked segments form the basis for our machine-learning inputs after flattening.

Why we chose Logistic Regression

- P300 signals are subtle and linear:
  - One characteristic of the P300 is a small, positive bump around ~300 ms after a flash
  - The difference between target and non-target is a linear amplitude shift, not a sharp threshold
- EEG features are high-dimensional and correlated:
  - 64 channels produce many correlated measurements
  - Logistic Regression works naturally with continuous, correlated features, especially after flattening.
- PCA + LR matches the structure of the data:
  - PCA reduces dimensionality across channels and time, extracting the general shape of the P300.
  - Logistic Regression learns a weighted combination of these components
- Why not decision trees?:
  - Trees split on exact thresholds (“if Cz > X…”), but P300 timing and amplitude vary across people, so these splits don’t generalize.
  - With only 85 character trials, trees easily overfit noise and become unstable.

Initial Approach: Classifying Individual Flash Segments

- First attempt: treat each flash as its own training sample.
- For every flash, we extracted a 0–600 ms EEG segment from key P300 channels (Cz, FC1, CPz).
- Each segment is a small 2-D matrix (window × channels).
  - Window length: 0–600 ms at 240 Hz → 145 samples
    - At 240 Hz, 1 sample ≈ 4.17 milliseconds
  - Flattening 145 samples × 3 channels → 435 features per flash
- Across 85 epochs:
  - 180 flashes per epoch → **≈15,300 flash samples**
  - Only 2 of the 12 codes per repetition are targets (1 row + 1 column)
  - Class balance ≈ 2,537 target vs 12,678 non-target
- These feature vectors (X) with labels (target vs non-target, y) formed our supervised dataset for logistic regression.

Why this approach performed poorly?

- Single-trial EEG is extremely noisy:
  - Individual flashes show almost no visible P300 peak
  - Noise dominates the subtle target/non-target differences
- Logistic regression struggled to find a stable decision boundary
- Results:
  - AUC: 65%
  - Accuracy: 59%
  - …barely better than flipping a coin
- This motivated our shift to ERP averaging, which boosts SNR.

Our New Preprocessing Approach

- Instead of thousands of individual flash segments, we averaged the 15 repetitions for each row/column code.
- Each epoch has 12 codes (6 rows + 6 columns), and each code flashes 15 times → we average those 15 flash-locked segments.
- This turns our dataset from:
  - ~15,000 noisy single-flash samples (flash-level) into
  - 85 epochs × 12 rows/columns = 1,020 averaged ERPs (code-level)
- For each of the 12 averaged ERPs per epoch:
  - Extract 0–600 ms from our top 3 channels (Cz, FC1, CPz)
  - Window length ≈ 145 samples
  - Flattening → 145 samples × 3 channels = 435 features per averaged ERP
- Across 85 training trials:
    - 1020 total samples
    - 170 targets, 850 non-targets
- We then use PCA to reduce dimensionality:
  - Kept the top 30 components
  - Explained ~90% of ERP variance
- Performance?...

Results

Accuracy	ROC AUC	Target Recall	Non-Target
84.3%	**0.918**	**84.7%**	**82.4%**

- AUC .918
  - The model very reliably distinguishes target P300 responses from non-targets across all decision thresholds.
- Target Recall 84.7%
  - Recall (True Positive Rate) measures how many real P300 targets the model correctly identifies.
- Non-Target Recall 82.4%
  - This indicates how often the model correctly recognizes non-target flashes. High non-target recall reduces false alarms and stabilizes classification.
- Overall Accuracy 84.3
  - Given that we used only 3 channels and a simple classical ML pipeline, achieving ~80% row/column classification accuracy is strong.
  - State of the art systems can reach ~95% using more channels and advanced filtering, but our results still show clear and robust P300 detection.

Conclusion

- ➢ We have achieved strong performance using classical ML methods by preprocessing EEG signals to isolate meaningful ERPs and built a robust classifier for target vs. non-target codes.
- ➢ We have also demonstrated the feasibility of decoding user intent from EEG and explored the neuroscience behind the P300 waveform

Future work

- Add a column to the 6x6 matrix
- Use SLM for on device inference
- Keep a history of past and current conversational info (e.g. family member names, what you previously typed) for predicting words, instead of relying on individual characters to spell