**Dataset description**

1. The dataset captures EEG measurements from the same participant before and after 24 hours of sleep deprivation while performing a working memory task under different conditions. Each trial involves the presentation of pictures, and the participant's task is to recognize and respond correctly. More specifically:
2. **Experiment Type:** Working memory paradigm using the n-back test.
3. **Task:** Participants are required to recall a picture shown N times before and compare it with the current picture. In this study, N is set to 2, meaning participants need to remember the last two pictures.
4. **Experimental Conditions:** There are four conditions, each associated with a specific task or stimulus. Participants indicate their responses by pressing corresponding buttons.
5. **Number of Trials:** The experiment consists of 72 trials, with the conditions balanced across these trials.
6. **Subjects**: 20 subjects for each session (rested, fatigued)
7. **Duration:** The experiment lasts approximately 5 minutes.
8. **Inclusion Criteria:** Only correct answers are included in the dataset.
9. **Datasets:** There are two datasets corresponding to different states of the participants:
10. **Fatigue State:** After 24 hours of sleep deprivation.
11. **Rested State:** Prior to sleep deprivation.
12. **EEG Data Details:**
13. **Sampling Rate:** 256Hz.
14. **Channels:** EEG measurements are recorded from 63 electrode positions (10-20) on the scalp, with each channel providing a time series of data.

**Preprocessing**

Firstly, the application of a Finite Impulse Response (FIR) filter designed to pass frequencies within the range of 1 to 35 Hz;

secondly, the removal of the baseline through the calculation of its average value, and

finally, the detrending of the data to eliminate any remaining trends.

* Loading EEG Data:
  + EEG data is loaded from a CSV file into a Pandas DataFrame.
  + A list of EEG channel names (channels) is specified.
  + MNE is used to create an info structure (info) that describes the EEG data (channel names, sampling frequency).
* Creating MNE object:
  + MNE's RawArray is used to create a raw EEG data structure (raw) from the Pandas DataFrame.
* Filtering EEG Data:
* The raw EEG data is band-pass filtered between 1 Hz and 35 Hz using a FIR filter (raw.filter(1, 35, fir\_design='firwin')).
* EEG signals consist of several frequency bands that are associated with different types of brain activity. These include delta (1–4 Hz), theta (4–8 Hz), alpha (8–12 Hz), beta (12–30 Hz), and gamma (30 Hz and above) waves.
* A bandpass filter allows signals within a specific frequency range to pass through while filtering out signals that fall outside that range.
* By applying a 1-35 Hz bandpass filter, you're preserving the delta, theta, alpha, and beta bands, which are commonly analyzed in EEG studies for various cognitive and neurological insights. This range excludes slower fluctuations that may represent non-neuronal activity (like slow artifact) and higher frequency noise, including muscle artifacts and environmental noise.
* This step is crucial for focusing on the frequencies of interest that are most likely to contain relevant brain activity information.
* Baseline Removal:
* The baseline mean is calculated for each channel, and the baseline is subtracted from the entire EEG data.
* The baseline in EEG data refers to a period where the brain is in a neutral state, and there is no experimental stimulus or task being performed. The average signal in this period is often considered to be a reference or "baseline" level of activity.
* Removing the average baseline involves calculating the mean signal level during this neutral period and then subtracting this value from the entire EEG recording.
* This process is intended to correct for any constant background activity or slow drifts in the EEG signal that are not related to the specific experimental conditions or stimuli.
* Detrending EEG Data:
* Over time, EEG signals can exhibit trends that are not related to the brain's electrical activity. These trends can be due to a variety of factors like slow shifts in electrode potentials or physiological artifacts (e.g., changes in impedance).
* Detrending refers to the removal of these linear or non-linear trends from the data.
* The process typically involves fitting a trend line to the time series data and then subtracting this trend from the original signal.
* This is important for both visual interpretation of the EEG waveforms and for computational analyses, as it can reduce the risk of spurious results that are due to these unrelated trends rather than true brain activity.

The detrending operation is applied to the baseline-corrected EEG data using scipy.signal.detrend().

**Temporal features**

The python script **raw\_preprocess.py** processes EEG data related to "fatigued" and "rested" conditions. It employs MNE-Python and SciPy for preprocessing, including bandpass filtering,

and detrending. Then it extracts the raw preprocessed EEG data as temporal features with a 3s time segmentation. The resulting EEG raw data have the dimensions (2401 samples x 63 channels x769 time points). Processed EEG data and labels are stored as NumPy arrays in the file **raw\_mental.npz** that will be used in a classification model.

**Classification**

Our model is a classification approach for time series data with multiple channels, focusing on distinguishing between fatigued and rested states from a private preprocessed dataset. Traditional convolutional neural networks (CNNs) may not capture the varying importance of different channels, so our attention-enhanced CNN addresses this by using both temporal and spatial information of the EEG dataset.

**Model Architecture**:

The model is fed with EEG inputs of shape (63 channels x 769 time points) for each of the 2401 samples. The architecture is the following:

**Convolution and MaxPooling layers**: Two 2D Convolution layers with relu activation functions to extract temporal and spatial features from the raw EEG signals. 2D convolutions apply a 2-dimensional filter to the dataset and the filter moves in 2-direction (x, y) to calculate the low level feature representations The feature maps connected to multiple adjacent time points of the previous layer which can obtain deeper time information. Each convolution layer is followed by MaxPooling layers to reduce temporal features and overfitting.

The kernel sizes are (3x3) for convolution and (1x2) for max pooling, for an aggressive filtering for the temporal dimension.

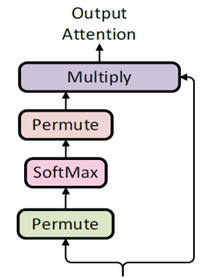
**Attention Modules**: Two attention modules follow the convolution layers for spatial feature extraction. Various regions of the brain, for example, the prefrontal and temporal brain regions, occupy different weights in the process of working memory, so the enhancement or suppression of a certain brain region will affect the classification. Attention mechanism can learn the weight map of different EEG channels to enhance the EEG channels with high correlation and suppress the EEG channels with low correlation. Each attention module is consisted of 4 layers:

**Permute**: Changes the order of the weight matrixes for each channel.

**Dense**: Uses softmax to learn the attention weight matrix of the channel.

**Permute**: Reverts the dimension order.

**Multiply**: Fuses the AM block input and weight matrix.



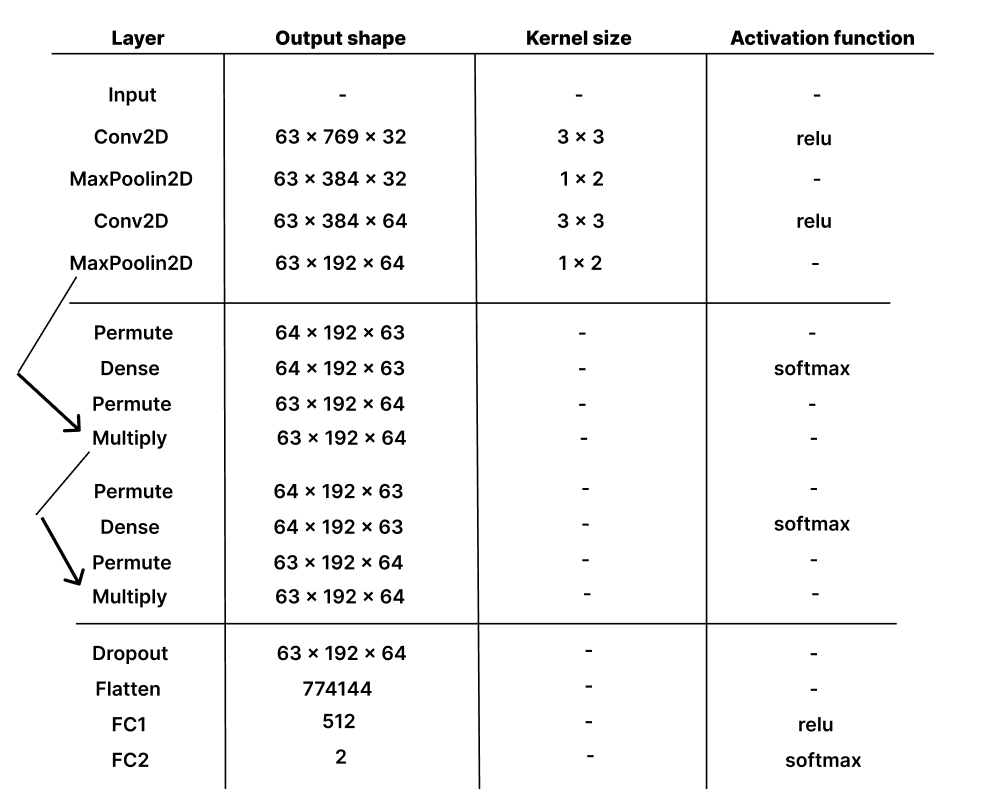
**Dropout Layer:** Included to reduce overfitting by randomly omitting a subset of features during training.

**Flattening and Fully Connected Layers:**

Flatten: Converts the multi-dimensional tensor into a 1D array.

FC1: A dense layer with relu activation processes the flattened data.

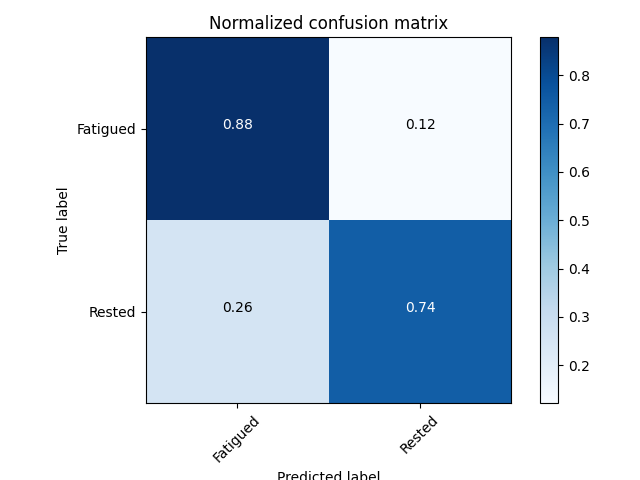
FC2: The final output layer with softmax activation yields the classification probabilities for the two states.



**Validation**

For the validation of the performance of the model a 5-fold cross validation is employed, and we use the metrics of average accuracy from the validation accuracies of all the folds and the confusion matrices are plotted for more details on sensitivity and specificity.

The ***average accuracy*** of this model is 81.34% with a varying performance of ***std***= 2.95% across different validation sets.

Confusion matrix
**Confusion matrices**:

From the first confusion matrix we have the following results:

**True Positives (TP)** for 'Fatigued': 1084 (correctly predicted as Fatigued)

**False Negatives (FN)** for 'Fatigued': 150 (incorrectly predicted as Rested)

**False Positives (FP)** for 'Rested': 298 (incorrectly predicted as Fatigued)

**True Negatives (TN)** for 'Rested': 869 (correctly predicted as Rested)

From the normalized confusion matrix, we have the following results:

**Sensitivity/Recall for Fatigued**: 0.88 (88% of the actual Fatigued instances were correctly predicted)

**Specificity for Rested**: 0.74 (74% of the actual Rested instances were correctly predicted)

After these results we conclude that the CNN model with attention shows a strong ability to classify the Fatigued state, with higher sensitivity and there is room for improvement on recognizing the Rested state. Overall, the model shows a strong ability to distinguish between the two states of Fatigued and Rested with an 81.34% average accuracy utilizing well the raw EEG data from 40 subjects. That is mainly because of the temporal and spatial feature extraction with the use of CNN layers and attention modules.

Classification python file: **2d\_classify.py**