Data Augmentation Techniques for Deep Learning in Motor Imagery EEG Classification

Human Computer Interaction Course Project

Alberto Baldrati, Serban Cristian Tudosie

Supervisors: Prof. Andrew David Bagdanov

Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Firenze

Index

Introduction

Introduction

- Brain computer interfaces (BCIs) provide a new communication bridge between human minds and devices
- The ability to control devices with our minds largely depends on the accurate identification of non-invasive EEG signals
- Recent advantages in deep learning have helped the progress in BCI field with CNNs that are becoming the new cutting-edge tools to tackle the problem of EEG recognition
- To successfully train a CNN a large amount of data is needed, but due to strict requirements it is difficult to collect large-scale and high-quality EEG data.
- The aim of this work to investigate the performance of different data augmentation methods for the classification of Motor Imagery data

Dataset

Dataset
Data acquisition

- The dataset we collected is made of 240 samples equally split in three classes which corresponds to three different motor imagery actions.

- As acquisition board we used the *OpenBCI Cyton Bio-sensing Board* which is a 8-channel neural interface with a 32-bit processor

- Data is sampled at 250 Hz on each of the eight channels

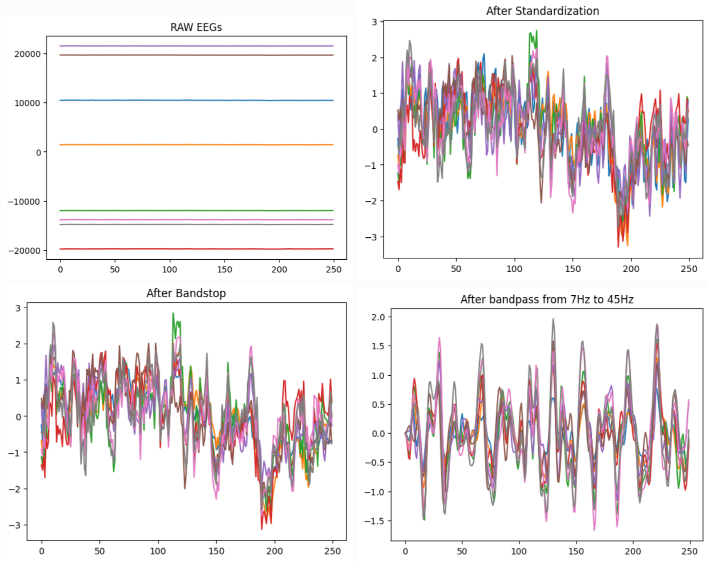
- The acquisition procedure was very strict in order to mitigate various issues such as electronic interference at 50 Hz and limited capacity of human to rapidly switch among thoughts

Dataset
Data preprocessing

- Data acquisition procedure saves each sample as a raw EEG signal embedded in a matrix where each row correspond to a different electrode and each column to a different timestamp. Such matrix has a shape of 8×250
- Data is not preprocess during acquisition to keep the possibility of tuning preprocess parameters at a later stage
- In our experiments we have performed three preprocessing steps:
  - A channel wise standardization
  - A bandstop filtering at 50Hz (power line frequency in Italy)
  - A bandpass filtering from 7Hz to 45Hz

Dataset

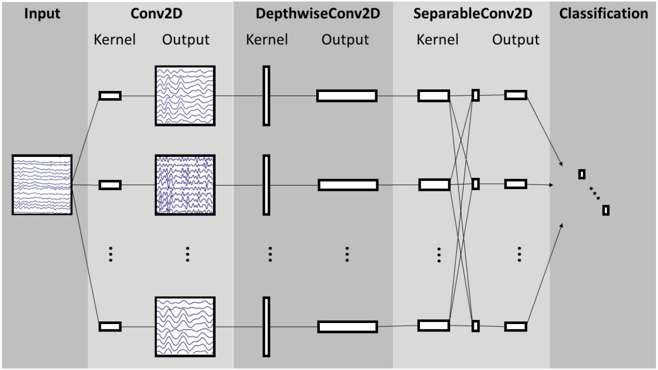
Data preprocessing



EEGNet

EEGNet

- In all our experiments we have used EEGNet (Lawhern et al., 2016) as end-to-end convolutional neural network. It is a compact CNN architecture for EEG-based BCIs that can be applied across several different BCI paradigms



EEG Data Augmentation

EEG Data Augmentation

Overview

- Data augmentation is a technique that generates new effective data training samples from origin datasets when we do not have sufficient data to train a deep learning model
- This method has been applied for years in computer vision tasks where building a new sample from an image is easy
- For EEG data, which are dynamic times series, standard computer vision augmentation pipelines make no sense
- We then explore several techniques tailored for EEG data reporting which of them can bring an actual improvement

EEG Data Augmentation

Channel Swapping

- **Channel Swapping** technique is very simple and is not taken from literature
- It consists of randomly swapping a channel of a sample with the same channel of another sample having the same label
- This channel swapping is made possible by the channel-wise standardization which makes each channel independent
- The whole process is controlled by an hyperparameter which is the probability of swapping each single channel

EEG Data Augmentation
Channel Mirroring

- **Channel Mirroring** is also very simple and can be seen as an adaptation of computer vision x-flip augmentation
- As the name suggests it consists of randomly flipping (along the temporal axis) a channel
- The whole process is controlled by an hyperparameter which is the probability of mirroring each single channel in a sample

EEG Data augmentation

Gaussian Noise

- In Wang et al., 2018 multiple ways of augmentation based on noise addiction in time domain were explored
- They show that due to the strong randomness and non-stationarity some local noises such as Poisson noise or salt(pepper) noise change the features of EEG data locally
- They focus on the addiction of gaussian noise with zero mean, in order to ensure that overall amplitude samples will not be changed
- They applied such technique to EEG based emotion recognition improving the performance in dataset SEED dataset (Zheng et al., 2018) and MAHNOB-HCI dataset (Soleymani et al., 2012)

EEG Data Augmentation

Gaussian Noise on STFT

- In Zhang et al., 2018 is pointed out the simple noise addiction on EEG data is not effective

- They propose a noise addiction novel technique that add gaussian noise in frequency domain after performing a Short-time Fourier transform:

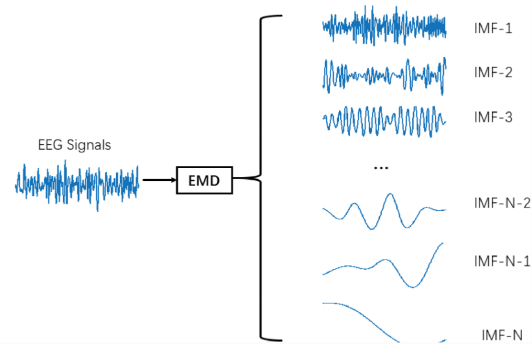
1. $$Z(\tau, f) = \int_{-\infty}^{+\infty} x(t) g(t-\tau) e^{-j2\pi ft} dt$$

2. $$Z'(\tau, f) = (A+p)\cos\phi + j(A+p)\sin\phi$$
$$\text{where } p \sim \mathcal{N}(\mu, \sigma^2)$$

3. $$x'(t) = STFT^{-1}(Z'(\tau, f))$$

EEG Data Augmentation
Empirical Mode Decomposition

- The empirical mode decomposition algorithm allows to conduct a data-driven analysis for nonlinear and non-stationary signals.
- The algorithm decomposes the original signals into a finite number of functions called intrinsic mode functions (IMFs), each of which represents a non-linear oscillation of the signals

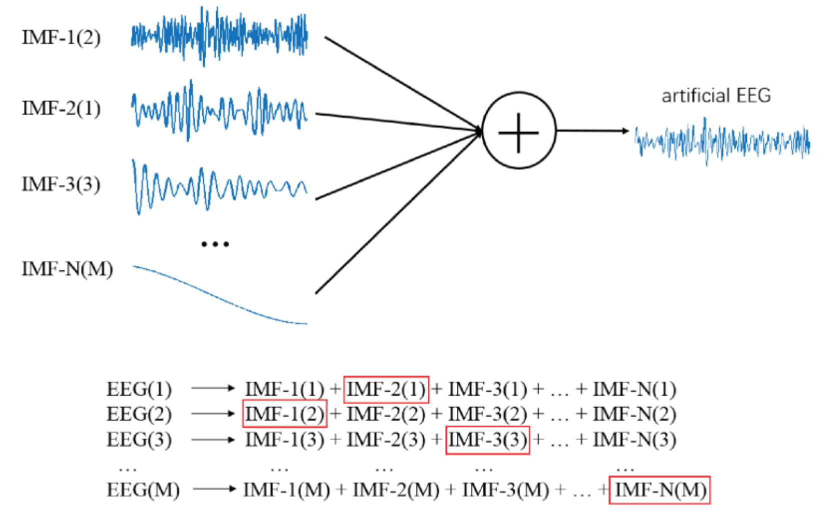


EEG Data Augmentation
Empirical Mode Decomposition

- We follow the data generation process suggested in Zhang et al., 2019:

1. Define the number of artificial frame to be create
2. Randomly select the frames that contribute with their IMFs to generate the artificial EEG frames
3. Randomly select EEG frames from the set of frames belonging to the same class. The first selected EEG frame contributes with all its first IMFs (8 IMFs, one IMF for channel), the second one with its second IMFs, and successively until n-th frame, which contributes with its n-th IMFs.
4. Generate a new artificial 8-channel EEG frame by adding up all the IMFs corresponding to the same channel

EEG Data Augmentation
Empirical Mode Decomposition



EEG Data Augmentation
Generative Adversarial Networks

- Finally we explore the possibility of using a Generative Adversarial Network to create synthetic EEG samples (Arjovsky et al., 2017) (Zhang et al., 2020)
- To overcome mode collapse issues in our experiments we used a Wasserstein GAN (with gradient penalty) which replace the Jensen-Shannon divergence with the Earth-Mover distance
- The GAN is conditional in order to generate data with known labels
- Both discriminator and generator structure are inspired to EEGNet
- The GAN has been trained for 30k epochs, with Adam optimizer with learning rate set to 2e-5, $\beta_1 = 0.5$ and $\beta_2 = 0.9$.

EEG Data Augmentation
Generative Adversarial Networks

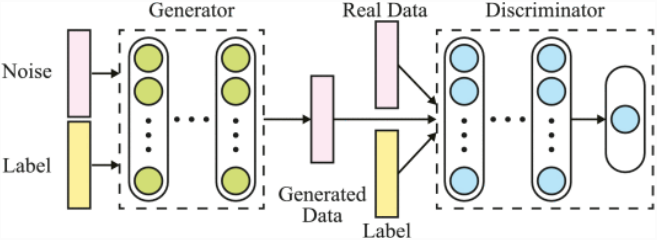


Figure: Illustration of proposed conditional GAN

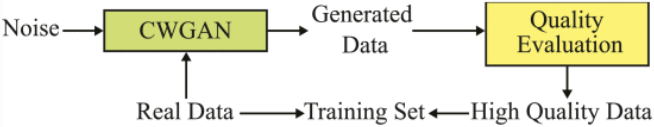


Figure: The framework for EEG data augmentation

Experiments

Experiments
Experimental Setup

- We conducted experiments on all data augmentations techniques we talked about
- We performed a 10-fold cross validation where during each step 70% of data is used as training set, 20% as validation set and 10% as test set
- The model which has been tested at each k-fold step is the one with the smaller loss on the validation set
- We repeated the splitting process five times and took as results the average
- All models have been trained for a maximum of 10k epochs with Adam optimizer (learning rate equal to 5e-5) and a batch size of 32

Experiments
Experimental results

	Channel Mirroring					Channel Swapping				
Probability	0.01	0.02	0.05	0.10	0.15	0.01	0.02	0.05	0.10	0.15
Accuracy	67.58 -0.25	66.33 -1.50	65.75 -2.08	63.75 -4.08	64.08 -3.75	66.08 -1.75	67.53 -0.30	66.33 -1.50	63.15 -4.68	62.83 -5.00

	Gaussian Noise					Gaussian Noise on STFT				
σ	1e-3	5e-3	1e-2	5e-2	1e-1	1e-3	5e-3	1e-2	5e-2	1e-1
Accuracy	67.25 -0.58	67.83 0	68.12 +0.29	66.66 -1.17	67.25 -0.58	67.50 -0.33	67.43 -0.40	68.67 +0.84	66.91 -0.92	65.99 -1.84

	Empirical Mode Decomposition					GAN				
Ratio	1:0.25	1:0.5	1:1	1:2	1:3	1:0.25	1:0.5	1:1	1:2	1:3
Accuracy	67.08 -0.75	66.50 -1.33	63.50 -4.33	65.75 -2.08	63.83 -4.00	68.83* +1.00	68.00* +0.17	68.41* 0.58	68.75* +0.92	67.89* +0.06

Table: Classification Accuracy of data augmentation methods in our private dataset (baseline: **67.83**)

Conclusions And Future Work

Conclusions
And Future Work

- In EEG motor imagery classification is possible to improve accuracy using data augmentation but is not so easy as in computer vision
- We show that augmentations based on channel swapping, channel mirroring and empirical mode decomposition are not effective on our dataset
- Noise based techniques can improve performances but heavily depend on the σ hyperparameter
- We obtain very promising results using GANs for generating synthetic data
- As future work we suggest to redo the experiments linked to GAN using only training data in order to dispel doubts about the goodness of such technique

Code available at

https://gitlab.com/ABaldrati/AugmentBrain

References I

- Arjovsky, M., Chintala, S., and Bottou, L. (2017).
  Wasserstein gan.

- Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. (2017).
  Improved training of wasserstein gans.

- Lawhern, V. J., Solon, A. J., Waytowich, N. R., Gordon, S. M., Hung, C. P., and Lance, B. J. (2016).
  Eegnet: A compact convolutional network for eeg-based brain-computer interfaces.
  CoRR, abs/1611.08024.

- Soleymani, M., Lichtenauer, J., Pun, T., and Pantic, M. (2012).
  A multimodal database for affect recognition and implicit tagging.
  IEEE Transactions on Affective Computing, 3(1):42–55.

- Wang, F., Zhong, S.-h., Peng, J., Jiang, J., and Liu, Y. (2018).
  Data Augmentation for EEG-Based Emotion Recognition with Deep Convolutional Neural Networks, pages 82–93.

References II

- Zhang, K., Xu, G., Han, Z., Ma, K., Zheng, X., Chen, L., Duan, N., and Zhang, S. (2020).
  Data Augmentation for Motor Imagery Signal Classification Based on a Hybrid Neural Network.
  Sensors (Basel), 20(16).

- Zhang, X.-R., Lei, M.-Y., and Li, Y. (2018).
  An amplitudes-perturbation data augmentation method in convolutional neural networks for eeg decoding.

- Zhang, Z., Duan, F., Solé-Casals, J., Dinarès-Ferran, J., Cichocki, A., Yang, Z., and Sun, Z. (2019).
  A novel deep learning approach with data augmentation to classify motor imagery signals.
  IEEE Access, PP:1–1.

- Zheng, W., Liu, W., Lu, Y., Lu, B., and Cichocki, A. (2018).
  Emotionmeter: A multimodal framework for recognizing human emotions.
  IEEE Transactions on Cybernetics, pages 1–13.

Supplementary Material

Empirical Mode Decomposition

Algorithm

Algorithm 1: The Empirical Mode Decomposition algorithm

Input: $x(t)$
Output: imf(t)

1 initial $r_0(t) = x(t)$
2 **repeat**(i=1,2...)
3   **repeat**(j=1,2...)
4         $h_{j-1}(t) = r_{i-1}(t)$
5   Detect the local maxima and the local minima of $h_{j-1}(t)$
6   Interpolate all local maxima to generate the upper envelope; and interpolate all local minima to generate the lower envelope
7   Obtain the local mean $m_{j-1}(t)$ by averaging the upper and lower envelopes
8         $h_j(t) = h_{j-1}(t) - m_{j-1}(t)$
9   **until** $h_j(t)$ satisfies the IMF's conditions
10     $imf_i(t) = h_j(t)$
11     $r_i(t) = r_{i-1}(t) - imf_i(t)$
12 **until** $r_i(t)$ is a monotonic function or does not have enough extrema to calculate the upper and lower envelopes
13 **return** imf(t)

where IMF's conditions at line 9 are:

1. The number of maxima is the same as the number of zero-crossing, or differs by at most one
2. The mean value between the envelope of the local maxima and the envelope of the local minima is zero

Generative Adversarial Networks

Mathematical details

- GANs consist of two competing components which are both parametrized as neural networks
- Given real data distribution X_r and generated data distribution X_g, the generator G produces realistic-like X_g to confuse the discriminator D, while the discriminator D tries to distinguish whether a sample comes from X_r or X_g
- The adversarial training procedure can be formulated as a min-max problem:

$$\min_{\theta_G} \max_{\theta_D} L(X_r, X_g) = \mathbb{E}_{x_r \sim X_r}[log(D(x_r))] \\ + \mathbb{E}_{x_g \sim X_g}[log(1 - D(x_g))]$$

Generative Adversarial Networks
Mathematical details

- The former optimization problem can be formalized as the minimization of Jensen-Shannon (JS) divergence
- The discontinuity of such divergence makes it hard to provide useful gradients
- To overcome this issues, JS divergence is replaced with Earth-Mover distance in Wasserstein GANs

$$W(X_r, X_g) = \inf_{\gamma \sim \Pi(X_r, X_g)} \mathbb{E}_{(x_r, x_g) \sim \gamma}[||x_r - x_g||] \quad (1)$$

where $\Pi(X_r, X_g)$ denotes all possible joint distributions of real distribution X_r and generated distribution X_g defined in traditional GANs.

Generative Adversarial Networks
Mathematical details

- Earth-mover distance is continuous and differentiable almost everywhere

- Since it's difficult to implement the infimum norm of equation 1 in reality, an alternative approach is to apply Kantorovich-Rubinstein duality of earth-mover distance:

$$W(X_r, X_g) = \frac{1}{K} \sup_{||f||_L \leq K} \mathbb{E}_{x_r \sim X_r}[f(x_r)] - \mathbb{E}_{x_g \sim X_g}[f(x_g)] \tag{2}$$

  where f denotes the set of 1-Lipschitz functions.

- In realistic implementation, f is replaced by discriminator D and $||f||_L \leq K$ is replaced by $||D||_L \leq 1$

Generative Adversarial Networks

Mathematical details

- In order to make the training procedure more stable and make convergence faster, Gulrajani et al., 2017 enforced Lipschitz constraint with gradient penalty instead of weight clipping to directly constrain the gradient norm

- In their approach, an extra penalty term is appended to the loss function:

$$\begin{aligned}\min_{\theta_G} \max_{\theta_D} L(X_r, X_g) = & \mathbb{E}_{x_r \sim X_r}[D(x_r)] \\ & - \mathbb{E}_{x_g \sim X_g}[D(x_g)] \\ & - \lambda \mathbb{E}_{\hat{x} \sim \hat{X}}[(||\nabla_{\hat{x}} D(\hat{x}||_2 - 1)^2]\end{aligned} \tag{3}$$

where λ is a hyperparameter controlling the trade-off between original objective and gradient penalty, and $\hat{x}$ denotes:

$$\hat{x} = \alpha x_r + (1 - \alpha) x_g, \alpha \sim \mathbb{U}[0, 1], x_r \sim X_r, x_g \sim X_g \tag{4}$$

Generative Adversarial Networks

Mathematical details

- In order to generate data with multiple categories, and auxiliary label Y_r is fed into both discriminator and generator

- In the generator, we concatenate X_z with Y_r. And in the discriminator, we concatenate both X_r and X_g with Y_r to construct a hidden representation, which controls the categories of generated data

- Then the proposed CWGAN can be formulated by:

$$\begin{aligned} &\min_{\theta_G} \max_{\theta_D} L(X_r, X_g, Y_r) = \\ &\mathbb{E}_{x_r \sim X_r, y_r \sim Y_r}[D(x_r|y_r)] - \mathbb{E}_{x_g \sim X_g, y_r \sim Y_r}[D(x_g|y_r)] \\ &- \lambda \mathbb{E}_{\hat{x} \sim \hat{X}, y_r \sim Y_r}[(||\nabla_{\hat{x}|y_r} D(\hat{x}|y_r)||_2 - 1)^2] \end{aligned} \tag{5}$$