

HW3: Deep Learning for BCI

Instructor: Chun-Shu Wei
TA:

Student Id

Introduction

In this homework, you are asked to utilize Convolutional Neural Network (CNN) for motor imagery EEG classification tasks, with experiments on 4 training schemes and 3 model architectures.

Please follow the provided code template for CNN framework implementation, model training, and dataset loading to fulfill the requirements. Try to make observations through the progress and discuss the results.

[Kaggle link](#)
[Code template](#)

****Make sure that you download the sample code before starting your implementation.**

Submission Policy

Read all the instructions below carefully before you start working on the assignment, and before you make a submission.

- **PLAGIARISM IS STRICTLY PROHIBITED. (0 point for Plagiarism)**
- The code will be graded on code completeness, model structure and algorithmic correctness.
- Submission deadline: **2023.05.05 10:00:00 AM.**
- Late submission penalty formula:

$$\text{original score} \times (0.7)^{\#(\text{days late})}$$

Submission Format

- Each student submits **1 zip file including 1 report (.pdf file) and your code (.ipynb file)**. Paper submission or programming languages other than Python/deep learning framework other than PyTorch is not allowed.
 - Remember to **upload the results for unlabeled test data to Kaggle**.
 - The source code must contain **comments**.
 - The report must contain **observations, results, and explanations**. Please name your zip file as hw3_studentID_Name.zip.
 - Illegal format penalty: **-5 points** for violating each rule of submission format.
-

Materials

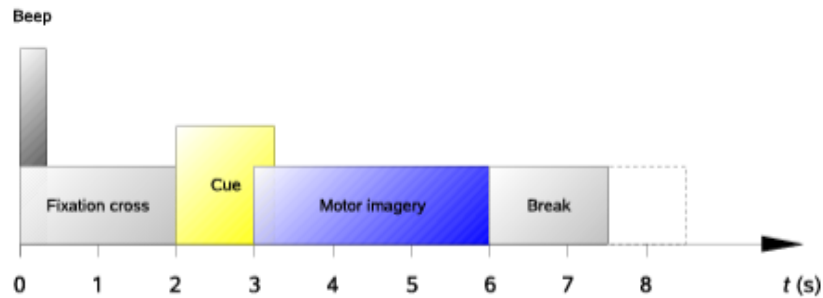
- PyTorch tutorials:
<https://pytorch.org/tutorials/>
<https://github.com/utkuozbulak/pytorch-custom-dataset-examples>
 - MNE: EEG processing & analysis Python package
[documents](#)
 - (for your own reference: [EEGNet and ShallowConvNet implementation](#))
-

1. Problem

1.1. EEG Dataset

Dataset description: ([Official document](#))

BCI competition IV 2a is a well-studied motor imagery dataset composed of **22 channel EEG data from 9 subjects**. Four motor imagery tasks are involved, namely the imagination of movement of the **left hand** (class 1), **right hand** (class 2), **both feet** (class 3), and **tongue** (class 4). The experiment paradigm is shown as follows.



In a trial, the subject is presented with a visual fixation cross. The instruction cue to imagine specific body part movement is given at 2s and lasts for 1.25s, the subject is prompted to perform the desired motor imagery task until 6s. Each subject has one training and one evaluation session, each session includes 72 trials for the 4 classes, yielding a total of 72*4 trials per session. The given dataset is filtered by FIR with band [4, 38] Hz and resampled to 125.0 Hz. The event trials are epoched using the same procedure described in referenced literature^[2].

Dataset is provided through Kaggle. You should follow the instructions in the code template to download this dataset. The dataset will be organized according to the following structure.

Kaggle data structure	
- BCI_hw3_dataset/	
- train/	
- BCIC_S0x_T.mat (x from 1~9, keys=x_train, y_train)	
- labeled_test/	
- BCIC_S0x_E.mat (x from 1~4, keys=x_test, y_test)	
- unlabeled_test/	
- BCIC_S0x_E.mat (x: 5, 6, A, B, C, keys=x_test)	

The dataset filename is under the format “BCIC_S{subject_ID}_{T(train)/E(test)}.mat”, e.g., “BCIC_S01_T.mat” is the training set for subject one and “BCIC_S03_E.mat” is the test set for subject three. For the unlabeled test data, three of the subject IDs are hidden and the trials were shuffled.

1.2. CNNs

Three baseline EEG models: **EEGNet**^[1], Spatial Component-wise Convolutional Network (**SCCNet**)^[2] and **ShallowConvNet**^[3] are included in this homework. The EEGNet and ShallowConvNet implementations are provided in the sample code, for SCCNet you will have to finish the implementation on your own. The model design ideas and more details could be found in the [BCI_ML2](#) course material and referenced literature^{[1] [2] [3]}.

1.3. Training Schemes

Four training schemes: **Individual(Ind)**, **Subject Independent(SI)**, **Subject Dependent(SD)** and **Subject Independent + Fine Tuning (SI+FT)** are included in this homework. The visualized explanation could be found in the [BCI_ML2](#) course material and the text explanation could be found in the referenced literature^[2]. The training schemes refer to different training data / test data combinations to simulate different kinds of real life BCI application scenarios.

1.4. Requirements (60%)

- 1.4.1. (40%) Please go through the referenced literature^[2] carefully and complete the “#TODO” parts. For SCCNet implementation, you have to use the predefined arguments to construct each layer. If your SCCNet structure is not changed along with the provided arguments, the penalty is HW3 score -5. For training scheme implementation, make sure your dataset contents meet the scheme definitions. After finishing the code, train 3 models with 4 training schemes and test on **subject 1 (BCIC_S01_E.mat)**. Show **testing accuracies and plot confusion matrices** for your 12 models. [confusion matrix wikipedia](#)
[matplotlib.cm \(optional\)](#)

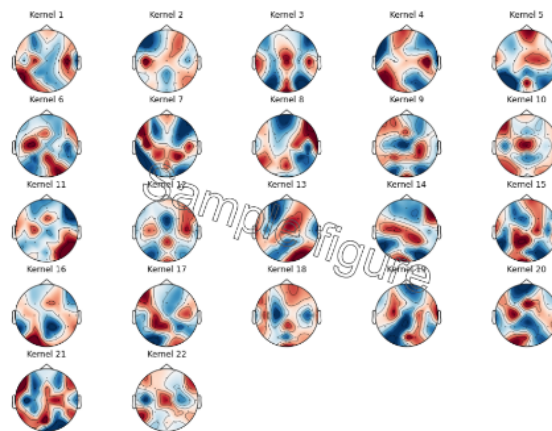
	EEGNet	SCCNet (or SCCNet_v2)	ShallowConvNet
Ind			
SI			
SD			
SI+FT			

- 1.4.2. (10%) Fill in your **hyper-parameter settings (Batch size, learning rate, epochs, optimizer, etc)**.

- ※ if you didn't modify the sample code, fill in the hyper-parameters specified in the sample code
- ※ if you have different settings for each subjects, **fill in the hyper-parameters for your subject 1 model.**

	EEGNet	SCCNet (or SCCNet_v2)	ShallowConvNet
Ind			
SI			
SD			
SI+FT			

- 1.4.3. (10%) Obtain **spatial kernel weights from the first convolutional layer of your SCCNet model trained with Ind scheme on subject 1, visualize the weights as topographic maps using the MNE package.**
[PyTorch Conv2D attributes](#)
[mne.channels.make_standard_montage](#) (Hint: electrode position)
or [mne.info](#) (Hint: EEG recording metadata structure)
[mne.viz.plot_topomap](#)
(Hint: There should be 22 topo maps for one model in your figure, each corresponding to a channel. Model weights and their meanings [article1](#), [article2](#))



- 1.4.4. (0%) Make predictions to the 5 test subjects without true labels (unlabeled_test) with the one with best performance from your trained model, export the result as a .csv file using the sample code (section 'Generate Submission csv File') and **submit it to kaggle**. This part is to make sure your code is executable and your models are trainable, which should beat the 0.6 accuracy baseline (the baseline will be adjusted

dynamically according to your submission). Otherwise you will only get half of the score in your implementation (1.4.1).

(Hint1: You may need to tune your hyper-parameter setting to get a higher accuracy)

(Hint2: Generally, subjects 5 and 6 result in lower test accuracy)

1.5. Discussion (40%)

- 1.5.1. Pros and cons of the 3 CNN models and 4 training schemes
- 1.5.2. Your observations regarding the models (structures, size .etc)
- 1.5.3. Difficulties you encountered in this homework.
- 1.5.4. (Optional) For models trained with different subjects, what are the possible reasons for the difference in model performance.
- 1.5.5. Other topics you find worthy to discuss.

1.6. Bonus Challenge

1.6.1. SCCNet_v2

In the SCCNet architecture there is a “permutation layer”, whose functionality can be achieved simply through some modification of the kernels in the convolutional layer(s). To get bonus points, try to implement SCCNet without the permutation layer. (Hint: go through the referenced literature^[2] and try to understand the kernel direction, their corresponding data dimensions, and the kernels’ operation objectives.)

Correct implementation: HW3 +10 points (+10(0.45/3) = +1.5 for semester score)

1.6.2. Kaggle leaderboard

The top 3 submissions in the Kaggle leaderboard will get bonus points.

Top 1: HW3 +30 points (+ 4.5 for the semester score)

Top 2: HW3 +20 points (+ 3 for semester score)

Top 3: HW3 +10 points (+ 1.5 for semester score)

1.7. References

- [1] Lawhern, Vernon J., et al. "EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces." *Journal of neural engineering* 15.5 (2018): 056013.
- [2] Wei, Chun-Shu, Toshiaki Koike-Akino, and Ye Wang. "Spatial component-wise convolutional network(SCCNet) for motor-imagery EEG classification." 2019 9th International IEEE/EMBS Conference on Neural Engineering (NER). IEEE, 2019.
- [3] Schirrneister, Robin Tibor, et al. "Deep learning with convolutional neural networks for EEG decoding and visualization." *Human brain mapping* 38.11 (2017): 5391-5420.