Lab2: EEG Motor Imagery Classification

Introduction

The meaning of Motor Imagery is that when a person imagines moving their limbs (or muscles) without actual physical output, specific brain regions remain activated. Common areas of motor imagery include left hand, right hand, feet, and tongue. In this assignment, we are going to predict motor imagery task by training models using deep learning techniques.

Dataset

We use BCI Competition IV 2a [1] as the dataset, there are total four labels, including left hand, right hand, feet, and tongue. This dataset consists of EEG signals from several subjects, each subject contains 2 sessions. There are 288 trails in each session, 72 for each label.

Important Date

- Code and report submission deadline: 7/23 (Tues.) 18:30 (24-hour clock)
  - Zip the code and report files and name it: ”DLP_Lab2_StudentID_name.zip”
  - Example: “DLP_Lab2_312551077_薛祖恩.zip”
  - You will get 10 points penalty in lab 2 if the file name is wrong
- Demo date: 7/23 (Tues.)

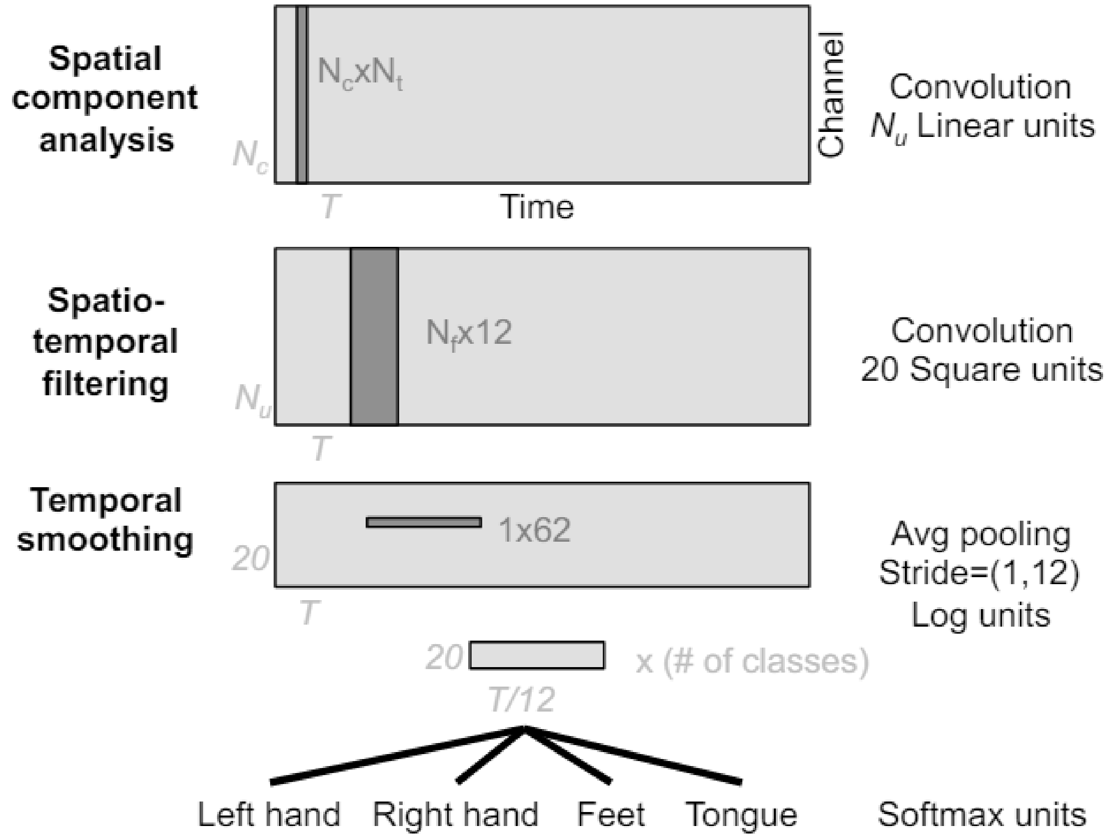
Requirements

- Train the model from scratch, do not load parameters.
- Implement the python scripts by yourself.
- **Plagiarism is not allowed.**

BCI Competition IV 2a

The data provided in this assignment has been already preprocessed, it is not necessary to do the preprocessing in the dataloader.py. The preprocessing steps include remove EOG signals (we focus on EEG signals), bandpass filter, normalization, and down sampling. For the labels, 0, 1, 2, 3 represents left hand, right hand, feet, and tongue respectively.

Model Architecture (SCCNet)



A parameter was misused in the SCCNet paper, please refer to the following description to find the correspondence of the defined parameters and notations in the paper.

B. SCCNet

This section presents the design of the proposed SCCNet and its architecture. A major part of SCCNet consists of convolutional kernels that capture the spatial and temporal characteristics of the EEG data. The design of SCCNet focuses on leveraging the benefits from applying spatial filtering to EEG data for purposes such as feature extraction and noise suppression [5], [6]. The architecture of SCCNet is illustrated in Fig. 1. The input to SCCNet is multi-channel EEG data arranged in 2-dimensions, with $\cancel{N}$ C channels and T time points. The architecture of SCCNet consists of four blocks: the first convolution block, the second convolution block, the pooling block, and the softmax block. The SCCNet herein is implemented using the Keras platform [13].

In the first and second blocks, the SCCNet performs two-step 2-dimensional convolution procedures. The initial convolution extracts EEG features, mimicking a spatial component analysis that decomposes the original EEG data from the channel domain to a component domain, where N_u filters with a kernel size of ($\cancel{N_c}$ C, N_t). When $N_t = 1$, this convolution step essentially performs a linear combination of EEG signals

Implementation Details

Upload the files mentioned below (related to code), you will get 10 points penalty in lab 2 if the code folder does not match.

```
→ DLP_Lab2_312551077_薛祖恩.zip
    → lab2
        → model
            → SCCNet.py
        → Dataloader.py
        → trainer.py
        → tester.py
        → utils.py
    → Report.pdf
```

- SCCNet.py
    - Implement the SCCNet architecture, read the paper before doing the implementation.
- Dataloader.py
    - Implement the data loader to read the dataset.
- Trainer.py
    - Implement the code for training the SCCNet model, including functions related to training, losses, optimizer, backpropagation, etc, remember to save the model weight.
- Tester.py
    - Implement the code for testing, load the model, print out the accuracy for lab demo and report.
- Utils.py
    - Implement the drawing figure function and more if needed.

Assignment Details

In this assignment, you must implement three training methods (thus you must train three models, one for each training method), subject dependent, leave-one-subject-out (LOSO), and LOSO with fine tuning, you will get 0 points in lab 2 if you miss any of them.

- Subject dependent dataset folder
  - ./dataset/SD_train for training and ./dataset/SD_test for testing
- LOSO dataset folder
  - ./dataset/LOSO_train for training and ./dataset/LOSO_test for testing
- Finetune dataset folder
  - ./dataset/FT for training, use ./dataset/LOSO_test for testing

Please check your dataloader.py to make sure that the folder is correct before training. For the LOSO with fine-tune training method, please load your model for LOSO training method and apply finetune with the finetune dataset mentioned above.

Report Spec.

Your report should follow the structure of the topics (including the subtopics), except for topics conatins "(Optional)", otherwise you will get 10 points penalty in report score. Complete your report as detail as possible. Moreover, it's better to add graphs with your report content for clearer explanation.

- Overview (total 10%)
- Implementation Details (total 20%)
  - Details of training and testing code (10%)
  - Details of the SCCNet (10%)
  - (Optional) Anything you want to mention
- Analyze on the experiment results (total 50%)
  - Discover during the training process (20%)
  - Comparison between the three training methods (30%)
  - (Optional) Anything you want to mention
- Discussion (total 20%)
  - What is the reason to make the task hard to achieve high accuracy? (10%)
  - What can you do to improve the accuracy of this task? (10%)
  - (Optional) Anything you want to mention

Scoring

Final score: Experiment results (30%) + Report (30%) + Demo score (40%)

- Experiment Results
  - Subject dependent (10%)

Accuracy	Grade
score $\geq$ 70%	100%
60% $\leq$ score < 70%	70%
score < 60%	0%

  - LOSO (10%)

Accuracy	Grade
score $\geq$ 60%	100%
50% $\leq$ score < 60%	70%
score < 50%	0%

  - LOSO with fine-tune (10%)

Accuracy	Grade
score $\geq$ 80%	100%
70% $\leq$ score < 80%	70%
score < 70%	0%

In the demo phase, you need to load the trained model for three different training methods, and show the results of them, if you fail to do so, you will get 0 points on the experiment results.

Only train your model on the training dataset, if there are data leakage problems, you will get 0 points in lab 2.

Good luck and have fun!

Reference

[1] C. Brunner, R. Leeb, G. Muller-Putz, A. Schl ¨ ogl, and G. Pfurtscheller, ¨ "BCI competition 2008–Graz data set a," Institute for Knowledge Discovery (Laboratory of Brain-Computer Interfaces), Graz University of Technology, vol. 16, 2008.

[2] C.-S. Wei, T. Koike-Akino, and Y. Wang, "Spatial component-wise convolutional network (SCCNet) for motor-imagery EEG classification," in Proc. 9th Int. IEEE/EMBS Conf. Neural Eng., Jun. 2019, pp. 328–331.