Brain-Computer Interface, Spring 2022, NYCU-CS

**Lab 3: Deep Learning for BCI**

**Rules**

1. Each group submits 1 report (.pdf) and 1 code (.ipynb).
2. Report must contain observations, results and explanations. Please name your files as Lab3\_GroupNum.pdf.
3. Paper submission is not allowed. Please use our Docs. template to complete

your report.

1. Code must contain comments to explain your code. Only Python and Pytorch

are allowed. Please name your code as Lab3\_GroupNum.ipynb

1. Implementation will be graded by completeness, algorithm correctness, model

description, and discussion.

1. Illegal format penalty: -5 points for violating each rule of formats.
2. PLAGIARISM IS STRICTLY PROHIBITED. (0 point for Plagiarism)
3. Late submission penalty: original score\*0.8 only within a week
4. Due on May 6, 2022 at 09:00 AM

**Introduction**

In this lab, we will exercise on applying convolutional neural networks for EEG classification in motor-imagery-based BCI.

**Part 1 (50%)**

Use the PyTorch code of the EEGNet [1] to perform 4-class motor imagery EEG classification.

*Data description*

‘BCI competition IV 2a’ is a well-known motor imagery dataset which is composed of 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). Each subject has training and testing sessions, and each session involves 72 trials for each class, yielding a total of 72\*4 trials per

session.

The dataset name will be like BCIC\_S{subject\_ID}\_{train/test}.mat for each file, like “BCIC\_S01\_T.mat” is a training set for subject one and “BCIC\_S03\_E.mat”is a testset for subject three. We will provide the dataset called “BCICIV\_2a\_mat” to you in the attached file to implement and we will give you a tutorial to teach you how to build neural networks using PyTorch.

Problems

1. (10%) Please build an EEGNet-based subject-independent model with training set for all subjects except subject S01(“from BCIC\_S02\_T.mat to BCIC\_S09\_T.mat”) and testing accuracy with test set for subject S01(“ BCIC\_S01\_E.mat”). (with learning rate = 0.001, batch size = 32)
   1. Show overall accuracy which is trained with an subject-independent model.
   2. Show accuracy of each of the four classes.
   3. Construct a 4\*4 confusion matrix using the classification results.

| a. Overall | | 66.66666666666666% |
| --- | --- | --- |
| b. | Left hand (class1) | 67% |
| Right hand (class 2) | 86% |
| Both feet (class 3) | 56% |
| Tongue (class 4) | 58% |
| c. Confusion matrix | |  |

1. (10%) Still build EEGNet to train a classifier with training set for all subjects except subject S01(“from BCIC\_S02\_T.mat to BCIC\_S09\_T.mat”) and testing accuracy with test set for subject S01(“ BCIC\_S01\_E.mat”)with different learning rate and different batch size. Please fill in the table below.

Test accuracy with different learning rate and different batch size.

| Learning rate\Batch size | 64 | 32 | 8 | 4 |
| --- | --- | --- | --- | --- |
| 0.003 | 68.75% | 66.32% | 65.97% | 69.79% |
| 0.001 | 71.88% | 66.67% | 65.97% | 67.01% |
| 0.0003 | 68.06% | 68.75% | 67.36% | 69.10% |
| 0.0001 | 62.15% | 62.15% | 65.28% | 65.63% |

From the table above, we can see that when the learning rate is 0.0001, test accuracy is relatively lower than others. I think that the possible reason is that the learning rate is too small, which may lead to convergence at an undesirable local minimum. Additionally, when observing the validation accuracy while training, it is quite easy to find that the validation accuracy oscillates when the learning rate is 0.003 and 0.001. The phenomenon may result from setting the learning rate too large. If the learning rate is too large, it might be possible to skip over the minimum value accidently when doing gradient descent. However, there is not much difference in test accuracy between different batch sizes, and neither does the validation accuracy.

1. (30%) Go through the paper “Spatial Component-wise Convolutional Network (SCCNet) for Motor-Imagery EEG Classification” [2], use each of the training schemes: the subject-independent, subject-dependent, and subject-independent plus fine-tuning, to train the EEGNet and test on S01. For example, in the subject-independent training, a total of 16 sessions for 8 subjects selected from S02 to S09 are served as the training set, while the test set is the same as above (“BCIC\_S01\_E.mat”). (with learning rate = 0.001, batch size = 32)

| **scheme** | **subject-independent** | **subject-dependent** | **subject-independent + fine-tuning** |
| --- | --- | --- | --- |
| **accuracy** | 66.67% | 69.44% | 76.39% |
| **confusion**  **matrix** |  |  |  |

Table 3. Test accuracy and corresponding confusion matrices with different training schemes

1. Show the S01 accuracy using different training schemes.

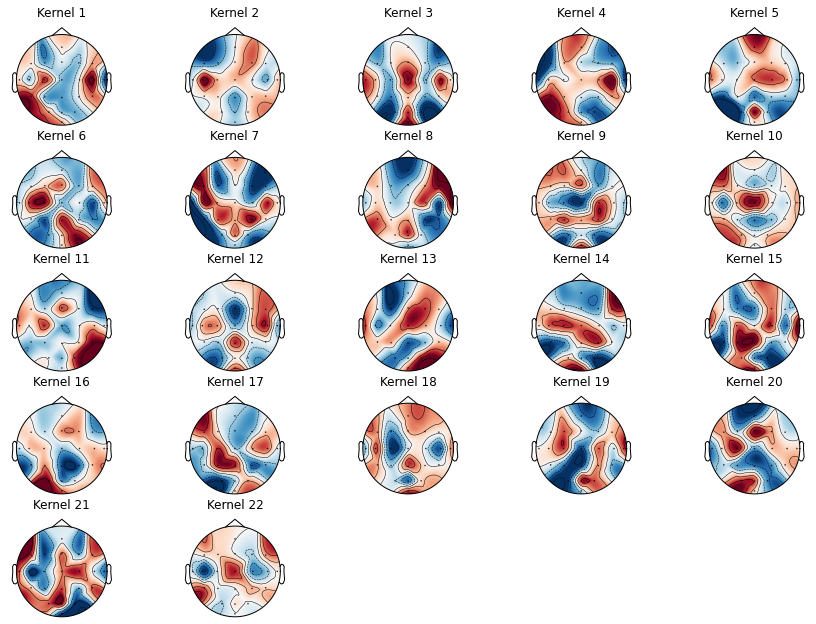
In this part, we trained EEGNet with three schemes, then used the same dataset to evaluate their accuracy. We can see that the accuracy of SI+FT is apparently higher than SI and SD, which has the same trend as the paper(SCCNet). It claims that learning from the new data in a separate

fine-tuning phase helps SCCNet to adapt itself for a single user.

1. Construct a 4\*4 confusion matrix using the classification results. Illustrate which 4 values in the confusion matrix are accuracies of the four classes.

The four values on the diagonal line from upper left to lower right represent the accuracies of the four classes, we can conclude that the accuracies of class 1 are higher than others in three schemes.

1. Use topographic maps to show values of spatial kernel weights corresponding to the first layer of SCCNet, which is obtained from the pretrained model of SCCNet. Notice that a total of 22 topographic maps should be included in a big figure.



The topoplot indicates the weights distribution of different channels in each kernel. It means which brain region the model puts more emphasis on.

**Part 2 (50%)**

In a field like deep learning applications in BCI is moving faster and faster each day. Various CNN architectures are developed to improve the accuracy of EEG pattern recognition. In this part, you implement three neural networks: 1.EEGNet, 2.Shallow ConvNet, 3.SCCNet

1. (25%) Perform four training schemes (individual, subject-independent, subject-dependent, subject-independent plus fine-tuning) on all three neural network models with “BCIC\_S01\_E.mat” as the test set.

**Hyper parameter:**

* Epoch: 100
* Batch size: 128
* Learning rate: 0.001
* Optimizer: Adam

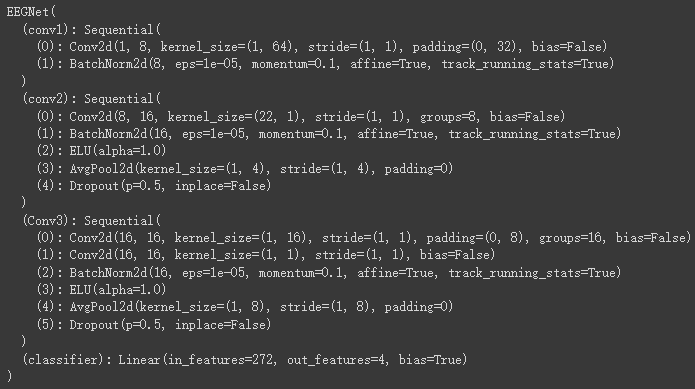
|  | Individual | Independent | Dependent | SI+Fine-tune |
| --- | --- | --- | --- | --- |
| EEGNet | **76.4%** | 66.3% | 76.04% | 75.3% |
| Shallow ConvNet | 71.18% | 72.25% | 77.78% | **84.37%** |
| SCCNet | 78.81% | 79.51% | 82.29% | **90.27%** |

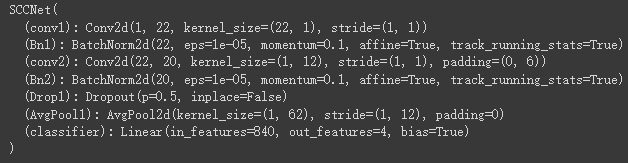
1. (25%) Compare the results among all three CNNs under different training schemes. Summarize and explain the pros and cons of each model.

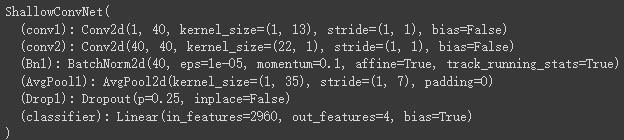
For these 4 training schemes, before our experiment, we can simply guess independent training schemes may lead to the worst result. Because EEG varies from subject to subject, even session to session. In these 4 schemes, only the independent scheme doesn’t contain any information about the new (testing) subject. so, models trained by this scheme may involve more bias. As for other 3 schemes, the dependent and fine-tune schemes are possible to outperform other schemes because they can obtain certain information about new subjects and also have more data to train a more generalized model.

And back to our experiment result. As expected, two over three best results of each model come from independent + fine-tune schemes.

From the viewpoint of the model architecture, EEGNet and Shallow ConvNet apply temporal filters first, and then extract spatial features by spatial filters at the 2nd level. This means that these 2 models put more emphasis on temporal information. On the other hand, SCCNet applies spatial filters first in order to retain more spatial information. From the result, maybe, we can conclude that spatial information is a more important factor in motor imagery classification.







As for the total number of parameters, EEGNet and SCCNet are more compact. And Shallow ConvNet has nearly 50k parameters. In my opinion, generally, the EEG dataset is smaller than the image dataset in CV. Additionally, In EEG, it isn't suitable to learn the patterns or features that are too specific to each subject because EEG has larger variation between each subject or session. Therefore, the compact model is more likely to give better results.

| **Model** | **#parameters** |
| --- | --- |
| **EEGNet** | 2,548 |
| **Shallow ConvNet** | 9,254 |
| **SCCNet** | 47,644 |

In short, from **the properties of the task**, if the task is more correlated to the spatial information, **SCCNet** may take some advantages. On the other hand,when temporal information impacts the task more, EEGNet or Shallow ConvNet may lead to a better performance. Furthermore, from **the size of dataset**, when the data is insufficient (such as individual scheme), the relatively compact model(i.e **EEGNet and SCCNet**) may fit better than Shallow ConvNet

*PyTorch tutorial link*

<https://pytorch.org/tutorials/>

<https://github.com/utkuozbulak/pytorch-custom-dataset-examples>

**Reference**

[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] Schirrmeister, Robin Tibor, et al. "Deep learning with convolutional neural networks for EEG decoding and visualization." Human brain mapping 38.11 (2017): 5391-5420.