Brain Computer Interface-HW3: Deep learning for BCI

109350008 張詠哲

1. (45%) 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 constructed with fixed constants instead of the provided arguments, the penalty is HW3 score -5. For training scheme implementation, make sure your dataset contents meet the scheme definitions. Plot the confusion matrix of your best performed model-scheme-curriculum combination.

The implement part will be present in the .ipynb file. And following tables are the performance comparison of SCCNet & EEGNet on 4 subjects with difference training schemes (both models' hyperparameter use default setting).

SCCNet:

Curriculums	Subjects id	SI (test accuracy)	SI + FT (test accuracy)
	01	0.591	0.613
Easy to all	02	0.541	0.563
	03	0.613	0.826
	04	0.598	0.680
	01	0.711	0.784
all	02	0.589	0.591
	03	0.654	0.718
	04	0.648	0.666
	01	0.609	0.725
All to easy	02	0.503	0.542
	03	0.586	0.868
	04	0.523	0.649

EEGNet:

Curriculums	Subjects id	SI (test accuracy)	SI + FT (test accuracy)
	01	0.609	0.755
Easy to all	02	0.507	0.557
	03	0.688	0.864

	04	0.533	0.564
all	01	0.645	0.739
	02	0.513	0.586
	03	0.618	0.843
	04	0.522	0.575
All to easy	01	0.587	0.788
	02	0.518	0.574
	03	0.583	0.831
	04	0.521	0.631

By the tables we can find that. In the Subject Independent (SI) training scheme, both models generally performs better after fine-tuning (SI + FT), indicating the effectiveness of fine-tuning on the targeted test subject. And the training scheme on "all" usually get the better performance than other training scheme (easy to all & all to easy). When SI on subject 01 & 03 usually get better accuracy than other subject.

And the experimental results maybe can be attributed to following factors:

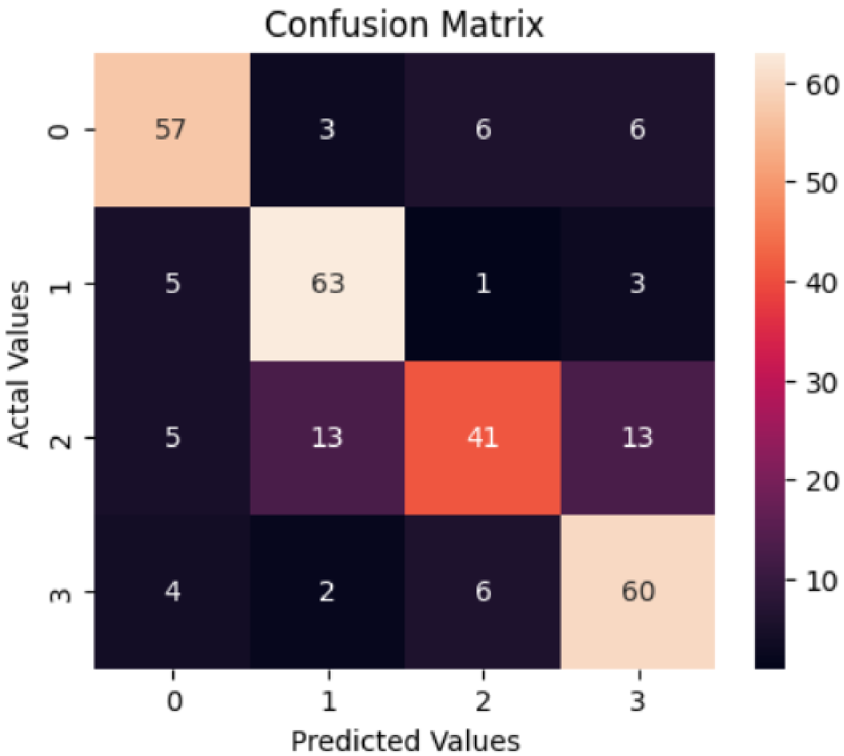
- **Effectiveness of Fine-Tuning (SI + FT):**
  The consistent improvement in test accuracy after fine-tuning (SI + FT) across both models (SCCNet and EEGNet) suggests that adapting the models to subject-specific characteristics enhances their performance on the targeted test subject. This is quite intuitive, fine-tuning helps overcome subject-specific variability and improves the generalization of the models on specific subject.

- **Subject-Specific Variability:**
  The superior performance of Subject Independent (SI) training on subjects 02 and 04 compared to other subjects. I don't have clear evidence to prove that, but i think maybe these subjects have unique EEG patterns and characteristics. Make the models better capture and adapt to the features of other subjects, but harder to predict the specific subject.

- **Training on "All" Subjects:**
  The variations in performance across different training curriculums (easy to all & all to easy) suggest that the order and difficulty progression of the training tasks influence the model's adaptability. The "all" training scheme usually get better performance. In my opinion, maybe because it involves training on the entire dataset, provide a more comprehensive learning experience, contributing to its

superior performance.

And Following picture is the confusion matrix of my best model-scheme-curriculum-subject combination.
(model = SCCNet, training scheme = ‘all’, subject id = ‘01’)



2. (5%) Fill in your hyper-parameter settings (Batch size, learning rate, epochs, optimizer, etc) of your best performed model-scheme-curriculum combination.
※ if you didn’ t modify the sample code, fill in the hyper-parameters specified in the sample code.

Following is the hyperparameter and training scheme of my best performance model
Hyperparameter:

```
# config training scheme, mode, hyperparam
eegmodel = SCCNet # function alias, should be EEGNet, SCCNet, SCCNet_v2
kwargs = dict(Nt=22, Nc=20, fs=125.0, dropoutRate=0.5) # custom args for different EEG model
#kwargs = dict(kernLength=32, F1=16, F2=32, D=2, dropoutRate=0.2) # for EEGNet
scheme = "all"  # "all", "easy"
easy_list = ["01", "02", "03", "04"]  #TODO: Modify this list!!! Put the easy subject id here ex. ["01", "02" ......]
epochs = 350
batch_size = 64
lr = 1e-4
savepath = "/content/checkpoints/"
os.makedirs(savepath, exist_ok=True)

subject_id = "01"
```

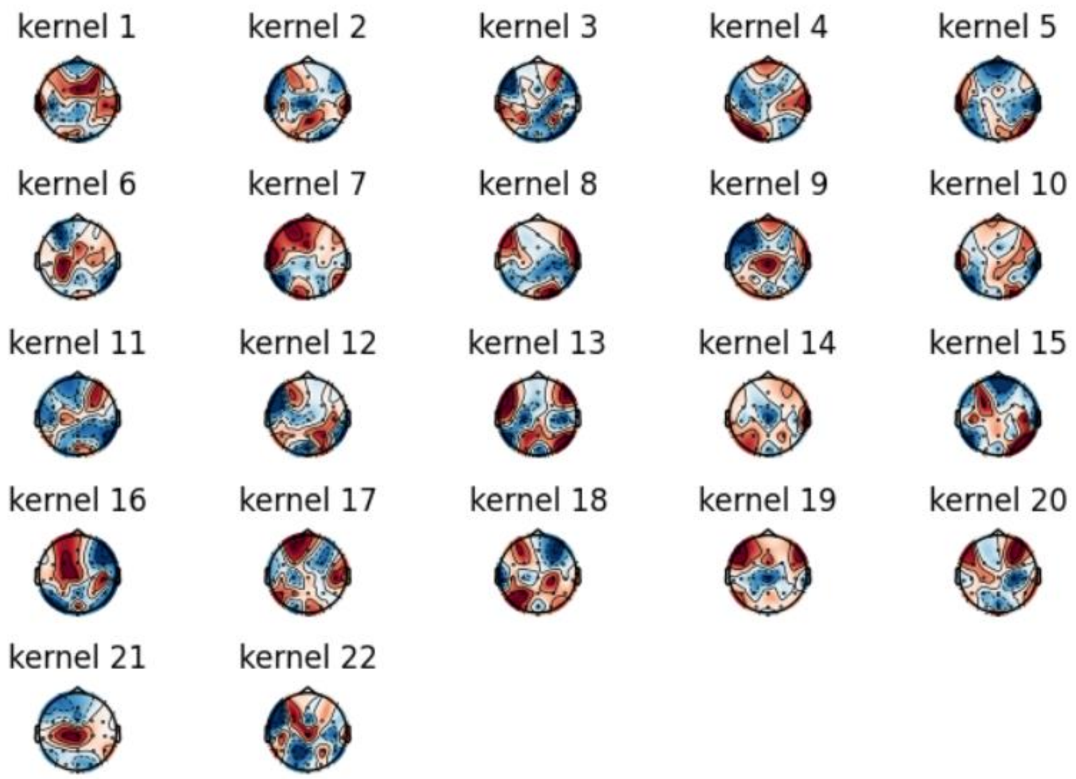
Training scheme: all

the Kaggle score of my best performance models

9	109350008	0.60000	14	1d

Your Best Entry!

3. (10%) Obtain spatial kernel weights from the first convolutional layer of your best performed SCCNet model, visualize the weights as topographic maps using the MNE package.



Discussion

1. Strengths and weaknesses of the 2 CNN models and your observations regarding the models (structures, parameter size .etc)

The following statements are my observations and some thought about the strengths and weaknesses of SCCNet & EEGNet:

SCCNet:

- **Strengths:**

SCCNet leverages spatial filtering techniques through two-step 2-dimensional convolution procedures. Although Both models (SCCNet & EEGNet) incorporate spatial filtering techniques, but SCCNet seems to emphasize spatial analysis more explicitly. This maybe beneficial for feature extraction, noise suppression, and artifact removal in EEG data. And the second convolutional block of SCCNet performs spatial-temporal convolution, which is expected to perform spectral filtering, inter-component correlation, and other spatial-temporal analyses on EEG spatial components. SCCNet dynamic structure allows for more adaptability in terms of architecture, but its effectiveness depends on the specific application and provided arguments.

- **Weaknesses:**

The kernel sizes of average pooling are fixed, which may limit adaptability to different input data characteristics. And the model uses square activation and log activation, which may not capture complex non-linearities in the data.

EEGNet:

- **Strengths:**

EEGNet uses depthwise convolution for spatial filtering, this maybe reducing the number of trainable parameters and efficiently extracting frequency-specific spatial filters. And the use of separable convolution in the second block allows for reduced parameters and explicit decoupling of relationships within and across feature maps, which contribute to efficient parameter usage, potentially making it more suitable for scenarios with limited data.

- **Weaknesses:**

My thought is similar with SSCNet part. The kernel sizes of average pooling are fixed, potentially limiting the model's adaptability to different input data characteristics. And the model predominantly uses ELU and dropout, which might be too simplistic for capturing complex relationships. The depthwise convolution in the first block is a relatively simplistic form of spatial filtering compared to SCCNet.

2. **Strengths and weaknesses of the training schemes and curriculums**

Strengths:

- Adaptability:

The curriculum learning approach allows the model to adapt to a range of difficulty levels, maybe making it more versatile in real-life BCI scenarios where the complexity of tasks may vary. For example, by starting with easier tasks and gradually increasing difficulty (easy to all), the model is likely to develop a more generalized understanding of the data, potentially improving performance on a variety of tasks. But by the experience above, the training scheme on "all" usually get the better performance than other (easy to all & all to easy). Possibly, its superior performance could be attributed to the fact that it undergoes training on the entire dataset, offering a more learning experience.

■ Subject Independence + Fine-tuning:

The Subject Independent (SI) training scheme, combined with curriculums, promotes subject-independent learning, ensuring that the model is not overly specialized to a particular individual's characteristics. And the inclusion of fine-tuning after each curriculum under SI training allows for additional refinement of the model's performance on the specific characteristics of each test subject.

Weaknesses:

■ Increased Training Complexity:

Implementing multiple curriculums and fine-tuning stages can increase the complexity of the training process, requiring careful management of hyperparameters. And I think fine-tuning on specific subjects could potentially lead to overfitting, especially if the training dataset for each subject is limited. The success of the curriculums may be sensitive to the composition of the training set, and the chosen "easy" subjects may not be universally applicable.

■ Computational Resources:

Training on multiple subjects, applying different curriculums, and fine-tuning for each subject may demand substantial computational resources, especially when dealing with large datasets.

3. **For models trained with different subject sets, what are the possible reasons for the difference in model performance.**

Here are some possible reasons i think, but the main concept is focus on the different subject:

■ Subject Variability:

Individual subjects may exhibit variations in EEG patterns, brain activity, and cognitive responses. If the training set includes subjects with diverse characteristics, the model may struggle to generalize across different subject types. And this dataset includes motor imagery tasks for different body parts (left hand, right hand, both feet, tongue). The neural representation of these tasks may vary significantly between subjects, introducing inter-subject variability. Maybe some subjects may have unique EEG patterns or cognitive processes that are not well represented in the training set. The model might perform well on subjects with similar characteristics but struggle when faced with novel patterns.

- Subject Expertise and Skill Level:

  Subjects may have varying levels of expertise or skill in performing motor imagery tasks. This discrepancy in proficiency can impact the complexity and patterns of EEG signals, influencing model performance.

- Individual Learning and Adaptation:

  Subjects may adapt differently to the training paradigms, resulting in variations in the patterns learned by the model. Some subjects may respond more effectively to the curriculum learning approach, while others may benefit more from fine-tuning.

4. **Other topics you find worthy to discuss.**

   I have some question about BCI (not in SCCNet & EEGNet). Maybe is worth discussing.

- **Real-Time Processing Requirements:**

  Real-time processing enables immediate feedback and interaction, crucial for applications like BCI where timely responses are essential. Real-time processing minimizes delays, improving the overall user experience and facilitating quicker decision-making in applications such as neurofeedback or prosthetic control. But the intricate nature of BCI algorithms may hinder their efficient execution on devices with limited processing capabilities. It maybe demands substantial computational power, posing challenges for deployment on devices with limited resources. Therefore i think the feasibility of deploying these models on resource-constrained devices and the challenges associated with real-time processing may worth to discuss.

- **Adaptability to Clinical Settings:**

  I also want to know the applicability of the models (BCI model) in clinical settings. How well do they adapt to variations in patient conditions, and what considerations are important for translating research findings into practical clinical tools? And how good (or general) must the performance be before a model be used in clinical situations?