ECE C147/C247, Winter 2024 **Project guidelines**
Neural Networks & Deep Learning Prof. J.C. Kao
UCLA ECE TAs: T. Monsoor, Y. Liu, S. Rajesh, L. Julakanti, K. Pang
Due: Friday, March 15, 2024 (Friday of Week 10)

The project is intended to give you experience to working on neural networks in a research application. It is also intended to give you experience working with topics covered in the last three weeks of class, particularly RNNs, for which this project is the evaluation. This means, in this project, you should evaluate CNNs, RNNs, and potentially even CNN+RNN combinations. You may also evaluate neural network topics we don't explicitly get to cover in class, including but not limited to generative models (GANs, VAEs), transformer architectures, or deep reinforcement learning.

If you do not evaluate an RNN or other post-CNN algorithm, you will lose points for creativity in the rubric described at the end of this document.

The goal will be to optimize the performance of decoding on a particular task; or else to apply one of these techniques to a research area. You may use any tools to implement these networks (e.g., PyTorch, Keras, TensorFlow etc.). Our philosophy here is that you already understand the principles of training these networks, doing backpropagation, optimization, etc., and are now well-equipped to use these packages that implement several of these operations for you.

You may work on the datasets provided below, or do a project related to your own research / interests. If you are doing a project related to your own research / interests, please e-mail Prof. Kao directly to get approval. In general, as long as the project will explore a post-CNN topic, it should be approved. We will provide approval for custom project requests as long as they are sent prior to Friday, 1 March 2024. We will not approve any custom projects after this date.

Students may work in groups of up to (and including) 4 people. To find teammates, you may use the "Search for Teammates" Piazza functionality. For more information, see `https://support.piazza.com/support/solutions/articles/48001158117-search-for-teammates`. You may also do the project individually.

Project dataset

In class, we have worked with CIFAR-10 extensively. But what about data with temporal components and structure? This project will explore this, using datasets collected from electroencephalography (EEG). A more complete description of the data is attached at the end of the project description.

EEG reflects the coordinated activity of millions of neurons near a non-invasive scalp electrode. Because these are scalp potentials, necessarily, they have relatively poor spatiotemporal resolution compared to other neural recording techniques. EEG is believed to be recording dipoles that are transmitted through the scalp.

This dataset is made publicly available through: `http://www.bbci.de/competition/iv/`. However, people typically process these datasets with `MATLAB`. We have formatted the data in a way that it will be easily loadable for you, as opposed to you working with the raw `.gdf` files. These datasets are internal for this class and should not be distributed. If you would like to publish based off of this data, you should visit the BCI competition site and download the raw gdf files after filling out the following agreement:

Each participant has to agree to give reference to the group(s) which recorded the data and to cite (one of) the paper listed in the respective description in each of her/his publications where one of those data sets is analyzed. Furthermore, we request each author to report any publication involving BCI Competiton data sets to us for including it in our list.

After filling out the form and pushing the "I Agree" button an automatic e-mail will be generated containing location and access information for the data set download area.

For each subject, they record from 22 EEG electrodes while the user imagines performing one of four actions. Therefore, this is a classification task (with four outcome classes), where the EEG is used to determine what action the subject was imagining. Follow the instructions below to load the data.

We have processed the data so that you can load the data with numpy. Further, we have removed trials that have NaN's. You can load the data as follows:

In [1]:
```
import numpy as np
```

In [2]:
```
import numpy as np
X_test = np.load("X_test.npy")
y_test = np.load("y_test.npy")
person_train_valid = np.load("person_train_valid.npy")
X_train_valid = np.load("X_train_valid.npy")
y_train_valid = np.load("y_train_valid.npy")
person_test = np.load("person_test.npy")
```

Shape of data

In [4]:
```
print ('Training/Valid data shape: {}'.format(X_train_valid.shape))
print ('Test data shape: {}'.format(X_test.shape))
print ('Training/Valid target shape: {}'.format(y_train_valid.shape))
print ('Test target shape: {}'.format(y_test.shape))
print ('Person train/valid shape: {}'.format(person_train_valid.shape))
print ('Person test shape: {}'.format(person_test.shape))
```

```
Training/Valid data shape: (2115, 22, 1000)
Test data shape: (443, 22, 1000)
Training/Valid target shape: (2115,)
Test target shape: (443,)
Person train/valid shape: (2115, 1)
Person test shape: (443, 1)
```

This indicates that there are 2115 trials; each trial has corresponding EEG data from 22 electrodes over 1000 time bins. Please look at the dataset documentation to know more about the data. E.g., Table 2 lists what class labels (769, 770, 771, 772) correspond to. The person files correspond to the subject performing the task, ranging from 0-8 (inclusive) and may be useful should you want to see how well you can classify on individual subjects. In the original data / documentation, you may see that there are 25 channels. We have removed 3 of the channels (so that there are 22) because those 3 channels were for recording eye movements, not brain activity.

EEG questions that we suggest you answer

There are many projects you can imagine doing from this data, but if you'd like a "default" project, we suggest you do the following:

1. Optimize the classification accuracy for subject 1. Does it help to train across all subjects?
2. Optimize the classification accuracy across all subjects. How does the classifier do? Do you notice any interesting trends?
3. Evaluate the classification accuracy as a function of time (e.g., does it increase as you have data over longer periods of time? how much time is required to get a reasonable classification accuracy?)

Feel free to innovate beyond this to earn creativity and insight points. For example, you may gain additional creativity points for creatively designing, evaluating, and comparing CNN, RNN, and CNN+RNN architectures. Additional insight points may also be rewarded for explaining how these different approaches result in better / worse performance.

Project due date

The project will be due on Friday of Week 10, 15 March, 2024.

Project submittables

Each group should submit a writeup of their project work, exceeding no more than 3 pages including figures. References are excluded from the 3 pgs (e.g., they may overflow onto a 4th page and be numerous). It is fine to be below the page limit; this is the *maximum*. We appreciate conciseness. We will also ask you to submit your code, so that we can validate your results. If you have a project where you cannot submit your code, please notify us so we can proceed accordingly.

The writeup must adhere to the following template: `https://media.icml.cc/Conferences/CVPR2023/cvpr2023-author_kit-v1_1-1.zip` – so that we can judge all writeups in the same manner without having to worry about different font sizes, etc.

In addition to this writeup, the students should submit two extra pages. One page should contain a table summarizing the performance of all algorithms they tested for the datasets they evaluated them on (e.g., if you tested on all 9 EEG datasets, in addition to testing one classifier across all subjects, report all of these accuracies in your table). The other page should summarize the architectures they used (e.g., if they used a CNN, describe its layers, the size of the activation maps at the end of each layer; if they used an RNN, how many units does it have, etc.) and details about the training (e.g., what optimizer used, if dataset augmentation was used, what activation functions, etc.). You can think of this page as the "Methods" part of the paper. You may reference these pages in the writeup (as opposed to reproducing these tables / figures in the formal writeup.)

Project writeup

In the writeup, there should be the following sections:

Abstract

A brief description of what you did in the project and the results observed.

Introduction

If you are doing the EEG project, do not use the introduction to formulate the general problem of EEG decoding, as we are all familiar with the EEG problem. Instead, use the introduction to set up and motivate the architectures you pursued and why. If you are doing a project from your own research, please give us brief background.

Results

State the results of your experiments.

Discussion

Discuss insights gained from your project, e.g., what resulted in good performance, and any hypotheses for why this might be the case.

References

List references used in your writeup.

Project grading

Here, we outline the criterion by which we will grade the project. Note, some projects will be more creative than others; some projects will achieve higher performance than others. We will provide room for extraordinary work in one category to compensate for deficiencies in another category. These are the general areas we will look into. Concretely, the final project will be graded on a scale of 20 points, but each section is assigned points so that the sum total can exceed 20 points. Your final project score will be capped at 20 points. You should aim to do a good job in all areas.

1. **Creativity (7 points)**.

   How creative and/or diverse is the approach taken by the students? Does the student implement / try various algorithms? Are multiple architectures compared? An example of a project that we would count as creative is comparing e.g., a CNN and an RNN in decoding performance, as well as combining CNN + RNN architectures. Creativity may also result from how one tackles the design of these algorithms.

2. **Insight (7 points)**.

   Does the project reveal some insight about the choice of hyperparameters, architectures, etc. on the performance of the algorithm? Is there reasonable insight / explanation / intuition into the results? (i.e., you should not just blindly apply different algorithms to a problem and compare them.)

3. **Performance (6 points)**.

   Does the project achieve relatively good performance on the problem, given that the students are training with a CPU (and may not have GPU access)? How do different algorithms compare? If the project is related to one's research, how do results compare to the literature? (i.e., you should not just train a few different algorithms without optimizing them reasonably.) We do recognize that students may not have access to GPUs; if this is a problem for your optimization, state it clearly in your results that you believe performance could be increased with more time; this should be apparent from e.g., a loss function plot with physical time on the x-axis (e.g., showing that after some number of hours, the loss had decreased, but still had a long way to go). We will account for this in grading your performance. Last year, students also had success using Google Colaboratory.

4. **Write-up (4 points)**.

   Are the approach, insight, and results clearly presented and explained? Dissemination of results is an important component to any project.

Last note

A potentially helpful resource as you start to tackle this problem may be this paper, which used CNNs with the same dataset: `https://arxiv.org/pdf/1703.05051.pdf`

There have also been other CNN based architectures developed, such as EEGNet: `https://iopscience.iop.org/article/10.1088/1741-2552/aace8c`

BCI Competition 2008 – Graz data set A

C. Brunner[1], R. Leeb[1], G. R. Müller-Putz[1], A. Schlögl[2], and G. Pfurtscheller[1]

[1]Institute for Knowledge Discovery, Graz University of Technology, Austria
[2]Institute for Human-Computer Interfaces, Graz University of Technology, Austria

Experimental paradigm

This data set consists of EEG data from 9 subjects. The cue-based BCI paradigm consisted of four different motor imagery tasks, namely the imagination of movement of the left hand (class 1), right hand (class 2), both feet (class 3), and tongue (class 4). Two sessions on different days were recorded for each subject. Each session is comprised of 6 runs separated by short breaks. One run consists of 48 trials (12 for each of the four possible classes), yielding a total of 288 trials per session.

At the beginning of each session, a recording of approximately 5 minutes was performed to estimate the EOG influence. The recording was divided into 3 blocks: (1) two minutes with eyes open (looking at a fixation cross on the screen), (2) one minute with eyes closed, and (3) one minute with eye movements. The timing scheme of one session is illustrated in Figure 1. Note that due to technical problems the EOG block is shorter for subject `A04T` and contains only the eye movement condition (see Table 1 for a list of all subjects).

The subjects were sitting in a comfortable armchair in front of a computer screen. At the beginning of a trial ($t = 0\,\mathrm{s}$), a fixation cross appeared on the black screen. In addition, a short acoustic warning tone was presented. After two seconds ($t = 2\,\mathrm{s}$), a cue in the form of an arrow pointing either to the left, right, down or up (corresponding to one of the four classes

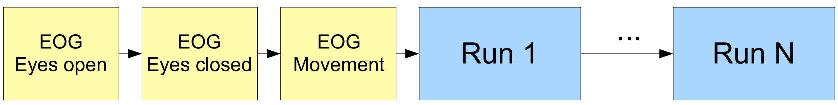

Figure 1: Timing scheme of one session.

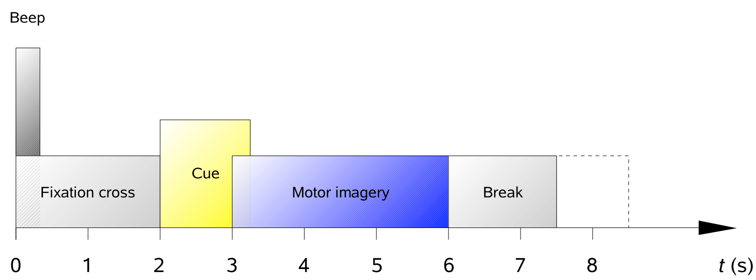

Figure 2: Timing scheme of the paradigm.

left hand, right hand, foot or tongue) appeared and stayed on the screen for 1.25 s. This prompted the subjects to perform the desired motor imagery task. No feedback was provided. The subjects were ask to carry out the motor imagery task until the fixation cross disappeared from the screen at $t = 6$ s. A short break followed where the screen was black again. The paradigm is illustrated in Figure 2.

Data recording

Twenty-two Ag/AgCl electrodes (with inter-electrode distances of 3.5 cm) were used to record the EEG; the montage is shown in Figure 3 left. All signals were recorded monopolarly with the left mastoid serving as reference and the right mastoid as ground. The signals were sampled with 250 Hz and bandpass-filtered between 0.5 Hz and 100 Hz. The sensitivity of the amplifier was set to 100 μV. An additional 50 Hz notch filter was enabled to suppress line noise.

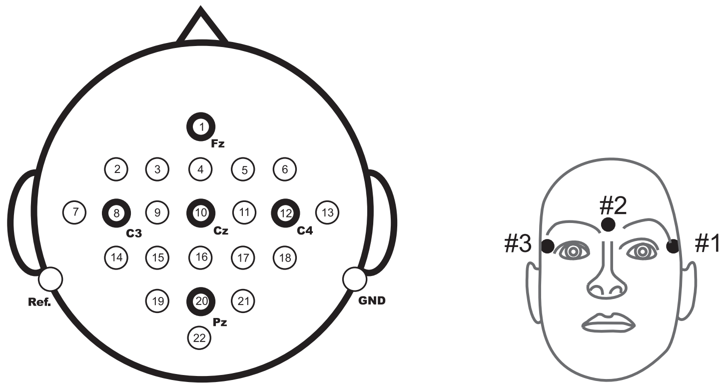

Figure 3: Left: Electrode montage corresponding to the international 10-20 system. Right: Electrode montage of the three monopolar EOG channels.

In addition to the 22 EEG channels, 3 monopolar EOG channels were recorded and also sampled with 250 Hz (see Figure 3 right). They were bandpass filtered between 0.5 Hz and 100 Hz (with the 50 Hz notch filter enabled), and the sensitivity of the amplifier was set to 1 mV. The EOG channels are provided for the subsequent application of artifact processing methods [1] and must not be used for classification.

A visual inspection of all data sets was carried out by an expert and trials containing artifacts were marked. Eight out of the total of nine data sets were analyzed in [2, 3].

Data file description

All data sets are stored in the General Data Format for biomedical signals (GDF), one file per subject and session. However, only one session contains the class labels for all trials, whereas the other session will be used to test the classifier and hence to evaluate the performance. All files are listed in Table 1. Note that the evaluation sets will be made available after the deadline of the competition (except for one file from subject `A01` which serves as an example). The GDF files can be loaded using the open-source toolbox BioSig, available for free at http://biosig.sourceforge.net/. There are versions for Octave[1]/FreeMat[2]/MATLAB[3] as well as a library for C/C++.

ID	Training	Evaluation
1	`A01T.gdf`	`A01E.gdf`
2	`A02T.gdf`	~~`A02E.gdf`~~
3	`A03T.gdf`	~~`A03E.gdf`~~
4	`A04T.gdf`	~~`A04E.gdf`~~
5	`A05T.gdf`	~~`A05E.gdf`~~
6	`A06T.gdf`	~~`A06E.gdf`~~
7	`A07T.gdf`	~~`A07E.gdf`~~
8	`A08T.gdf`	~~`A08E.gdf`~~
9	`A09T.gdf`	~~`A09E.gdf`~~

Table 1: List of all files contained in the data set, the striked out evaluation data sets will be provided after the deadline of the competition. Note that due to technical problems the EOG block is shorter for subject `A04T` and contains only the eye movement condition.

A GDF file can be loaded with the BioSig toolbox with the following command in Octave/FreeMat/MATLAB (for C/C++, the corresponding function `HDRTYPE* sopen` and `size_t sread` must be called):

```
[s, h] = sload('A01T.gdf');
```

[1] http://www.gnu.org/software/octave/
[2] http://freemat.sourceforge.net/
[3] The MathWorks, Inc., Natick, MA, USA

Event type		Description
276	0x0114	Idling EEG (eyes open)
277	0x0115	Idling EEG (eyes closed)
768	0x0300	Start of a trial
769	0x0301	Cue onset left (class 1)
770	0x0302	Cue onset right (class 2)
771	0x0303	Cue onset foot (class 3)
772	0x0304	Cue onset tongue (class 4)
783	0x030F	Cue unknown
1023	0x03FF	Rejected trial
1072	0x0430	Eye movements
32766	0x7FFE	Start of a new run

Table 2: List of event types (the first column contains decimal values and the second hexadecimal values).

Note that the runs are separated by 100 missing values, which are encoded as not-a-numbers (NaN) by default. Alternatively, this behavior can be turned off and the missing values will be encoded as the negative maximum values as stored in the file with:

```
[s, h] = sload('A01T.gdf', 0, 'OVERFLOWDETECTION:OFF');
```

The workspace will then contain two variables, namely the signals `s` and a header structure `h`. The signal variable contains 25 channels (the first 22 are EEG and the last 3 are EOG signals). The header structure contains event information that describes the structure of the data over time. The following fields provide important information for the evaluation of this data set:

```
h.EVENT.TYP
h.EVENT.POS
h.EVENT.DUR
```

The position of an event in samples is contained in `h.EVENT.POS`. The corresponding type can be found in `h.EVENT.TYP`, and the duration of that particular event is stored in `h.EVENT.DUR`. The types used in this data set are described in Table 2 (hexadecimal values, decimal notation in parentheses). Note that the class labels (i. e., 1, 2, 3, 4 corresponding to event types 769, 770, 771, 772) are only provided for the training data and not for the testing data.

The trials containing artifacts as scored by experts are marked as events with the type 1023. In addition, `h.ArtifactSelection` contains a list of all trials, with `0` corresponding to a clean trial and `1` corresponding to a trial containing an artifact.

In order to view the GDF files, the viewing and scoring application SigViewer v0.2 or higher (part of BioSig) can be used.

Evaluation

Participants should provide a continuous classification output for each sample in the form of class labels (1, 2, 3, 4), including labeled trials and trials marked as artifact. A confusion matrix will then be built from all artifact-free trials for each time point. From these confusion matrices, the time course of the accuracy as well as the kappa coefficient will be obtained [5]. The algorithm used for this evaluation will be provided in BioSig. The winner is the algorithm with the largest kappa value `X.KAP00`.

Due to the fact that the evaluation data sets will not be distributed until the end of the competition, the submissions must be programs that accept EEG data (the structure of this data must be the same as used in all training sets[4]) as input and produce the aforementioned class label vector.

Since three EOG channels are provided, it is required to remove EOG artifacts before the subsequent data processing using artifact removal techniques such as highpass filtering or linear regression [4]. In order to enable the application of other correction methods, we have opted for a maximum transparency approach and provided the EOG channels; at the same time we request that artifacts do not influence the classification result.

All algorithms must be causal, meaning that the classification output at time k may only depend on the current and past samples $x_k, x_{k-1}, \ldots, x_0$. In order to check whether the causality criterion and the artifact processing requirements are fulfilled, all submissions must be open source, including all additional libraries, compilers, programming languages, and so on (for example, Octave/FreeMat, C++, Python, ...). Note that submissions can also be written in the closed-source development environment MATLAB as long as the code is executable in Octave. Similarily, C++ programs can be written and compiled with a Microsoft or Intel compiler, but the code must also compile with g++.

References

[1] M. Fatourechi, A. Bashashati, R. K. Ward, G. E. Birch. EMG and EOG artifacts in brain computer interface systems: a survey. Clinical Neurophysiology 118, 480–494, 2007.

[4] One evaluation data set is distributed from the beginning of the competition to enable participants to test their program and to ensure that it produces the desired output.

[2] M. Naeem, C. Brunner, R. Leeb, B. Graimann, G. Pfurtscheller. Seperability of four-class motor imagery data using independent components analysis. Journal of Neural Engineering 3, 208–216, 2006.

[3] C. Brunner, M. Naeem, R. Leeb, B. Graimann, G. Pfurtscheller. Spatial filtering and selection of optimized components in four class motor imagery data using independent components analysis. Pattern Recognition Letters 28, 957–964, 2007.

[4] A. Schlögl, C. Keinrath, D. Zimmermann, R. Scherer, R. Leeb, G. Pfurtscheller. A fully automated correction method of EOG artifacts in EEG recordings. Clinical Neurophysiology 118, 98–104, 2007.

[5] A. Schlögl, J. Kronegg, J. E. Huggins, S. G. Mason. Evaluation criteria in BCI research. In: G. Dornhege, J. del R. Millán, T. Hinterberger, D. J. McFarland, K.-R. Müller (Eds.). Toward brain-computer interfacing, MIT Press, 327–342, 2007.