EE–559: Mini-project I

François Fleuret

https://fleuret.org/dlc/

[version of: March 28, 2018]

1 Introduction

The objective of this project is to train a predictor of finger movements from Electroencephalography (EEG) recordings. You can use all PyTorch tools, and you can also install and use scikit-learn if you want.

A zip file named `Proj1_StudentFamilyName1_StudentFamilyName2_StudentFamilyName3.zip` must be uploaded to the Moodle of the course before

Friday May 18th, 23:59.

It should contain, and only contain:

- the python source files, including a main executable `test.py` to call without arguments, and
- a 3–5 pages report in pdf.

The source code should be properly commented to facilitate its understanding.

Exchange of code or report snippets between groups, and materials taken "from the web" are forbidden. Also, every student should have a clear understanding of her/his group's entire source code and report. This will be checked during the oral presentation.

2 Objective

The goal of this project is to implement a neural network to predict the laterality of finger movement (left or right) from the EEG recording. This is a standard two-class classification problem.

Your report should describe the experimental process you followed, the network architectures, optimizers, and other settings you tried, and the reasoning behind your choices, if any.

It is encouraged that you implement first some very basic baselines (*e.g.* a linear predictor), and also that your main executable provides performance estimate in train and test for several configurations you tried.

3 Data files

3.1 Description

This is the Data Set 4 of the "BCI competition II" organized in May 2003 (Benjamin Blankertz and Müller, 2002).

http://www.bbci.de/competition/ii/

It is composed of 316 training recordings, and 100 test recordings, each composed of 28 EEG channels sampled at 1khz for 0.5s.

http://www.bbci.de/competition/ii/berlin_desc.html

The goal is to predict the laterality of upcoming finger movements (left vs. right hand) 130 ms before key-press.

3.2 Header

You can download a header to load the data at

https://fleuret.org/dlc/dlc_bci.py

By default, the data-sets have been downscaled to a 100Hz sampling rate. You can pass the argument `one_khz = True` to `load` to use the 1000Hz version.

To test the header, you can run

```
import dlc_bci as bci

train_input, train_target = bci.load(root = './data_bci')

print(str(type(train_input)), train_input.size())
print(str(type(train_target)), train_target.size())

test_input, test_target = bci.load(root = './data_bci', train = False)

print(str(type(test_input)), test_input.size())
print(str(type(test_target)), test_target.size())
```

References

G. C. Benjamin Blankertz and K.-R. R. Müller. Classifying single trial eeg: Towards brain computer interfacing. In *Neural Information Processing Systems (NIPS)*, 2002. Cited on page 2.