Deep Learning Final Project (KEN4257)

Course coordinator: Dr. Siamak Mehrkanoon

May 2020

1 Magnetoencephalography (MEG) data

MEG data comes from a neuroimaging technique that allows to scan the brain's magnetic field. Multiple sensors (eg magnetometers) are placed on the human scalp and their recordings can be of major importance in neuroscience research. One can for instance infer from brain data the state of a patient that has mental disorders [1].

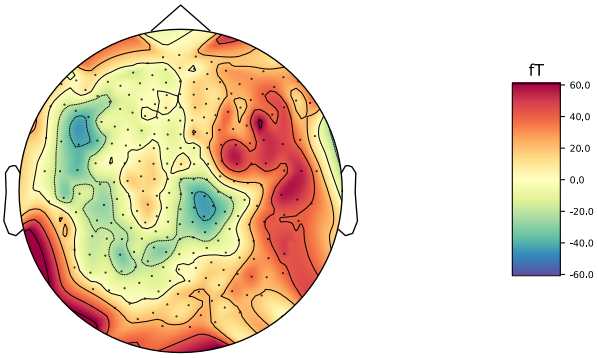

Figure 1: MEG data from a subject. The unit is in Tesla and the order of magnitude is in fT (femtoTesla = 10e-15 Tesla).

2 Data access and reading

The original data was provided by the *Human Connectome Project* [2] and a small part of it has been prepared for this assignment. You can download the MEG data in the following link : MEG data download

Once downloaded and uncompressed, you should end up with 2 folders : "Intra" and "Cross". The folder "Intra" contains 2 folders : train and test. The folder "Cross" contains 4 folders: train, test1, test2, and test3.

2.1 Reading of the data

The files contained in each of those folders have the "h5" extension. In order to read them, you need to use the h5py library (that you can install using "pip install h5py" if you don't have it already). This type of files can contain datasets identified by a name. For simplicity, each file contains only 1 dataset. The following code snippet can read the file "$Intra/train/rest\_105923\_1.h5$":

```
import h5py

def get_dataset_name(file_name_with_dir):
    filename_without_dir = file_name_with_dir.split('/')[-1]
    temp = filename_without_dir.split('_')[:-1]
    dataset_name = "_".join(temp)
    return dataset_name

filename_path="Intra/train/rest_105923_1.h5"
with h5py.File(filename_path,'r') as f:
    dataset_name = get_dataset_name(filename_path)
    matrix = f.get(dataset_name)[()]
    print(type(matrix))
    print(matrix.shape)
```

After executing the code above, "matrix" variable will be a numpy array with shape 248 x 35624

2.2 Explanation of the files

The files have the following format: "taskType_subjectIdentifier_number.h5" where taskType can be rest, task_motor, task_story_math, and task_working_memory. In practice, these tasks correspond to the activities performed by the subjects:

- **Resting Task**: Recording the subjects' brain while in a relaxed resting state.
- **Math & Story Task** : Subject performs mental calculation and language processing task.
- **Working Memory task**: Subject performs a memorization task.
- **Motor Task** : Subject performs a motor task, typically moving fingers or feets.

The subject identifier is made of 6 numbers, and the number at the end corresponds to a chunk part. This number has no particular meaning (splitted files are easier to handle in terms of memory management). The folder "Intra" contains the files of 1 subject only. In the folder "Cross", 2 subjects are contained in the train folder while the 3 test folders contain different subjects from the ones contained in the train folder. As seen in the section above, each

file is represented by a matrix of shape 248 x 35624. The number of rows, 248, corresponds to the number of magnetometer sensors placed on the human scalp. The number of columns, 35624, corresponds to the time steps of a recording.

3 Investigation and questions

In brain decoding, 2 types of classifications are performed. The first one is intra-subject classification, where deep learning is used to train and test models using the same subject(s). The second type, called cross-subject classification, happens when we train a model with a set of subjects, but test the model on new, unseen subjects. In this assignment, you are asked to perform **both** intra-subject and cross-subject classification. **The goal will be to accurately classify whether the subject is in one of the following states: rest, math, memory, motor.**

Tasks:

- (a) Choose a suitable deep learning model for the involved classification tasks. Justify your choice.
- (b) Compare the accuracy of the 2 types of classification, i.e. intra-subject and cross subject data using your model. Explain your results.
- (c) Explain the choices of hyper-parameters of your model architecture and analyze their influence on the results (for both 2 types of classification). How they are selected?
- (d) Explore separately the impact of batch normalization as well as dropout layer on the accuracy of your model for both 2 types of classification.
- (e) Explore the impact of learning rate as well as the choice of weight initialization on the accuracy of your model for both 2 types of classification.
- (f) If there is a significant difference in training and testing accuracies, what could be a possible reason? What are the alternative models or approaches you would select? Select one and implement to further improve your results. Justify your choice.

3.1 Hints

3.1.1 Data preprocessing

As you have seen in figure 1, the order of magnitude of this data is 10e-15, which might not be adapted for deep learning tasks. A common approach to tackle this problem is to do min-max scaling, making all the data scale to values between 0 and 1. Another common approach is Z-score normalization. More specifically, a time wise scaling/normalization is more suitable.

3.1.2 Data downsampling

The machine that made the recording of this data used a sample rate of 2034 Hz, meaning that every second corresponds to 2034 samples, or data points. Therefore every file corresponds to a duration of approximately 17.5 seconds. A common approach in neuroscience research is to consider that not every samples are significant, and to perform downsampling. A major advantage of this technique is that it makes deep learning training faster, while not necessarily having a negative impact on the accuracy.

3.1.3 Memory management during training

Since the train folder of the "Cross" directory contains 64 files, it might be difficult to load everything in the memory for the training. A simple workaround is to use a loop. For instance, the first iteration would load a small subpart of all the files (eg: 8 files), to fit the model to this data. The second iteration would load the next subpart (the next 8 files), to fit it etc ...

References

[1] Stefan Klöppel, Cynthia M Stonnington, Carlton Chu, Bogdan Draganski, Rachael I Scahill, Jonathan D Rohrer, Nick C Fox, Clifford R Jack Jr, John Ashburner, and Richard SJ Frackowiak. Automatic classification of mr scans in alzheimer's disease. *Brain*, 131(3):681–689, 2008.

[2] David C Van Essen, Kamil Ugurbil, E Auerbach, D Barch, TEJ Behrens, R Bucholz, Acer Chang, Liyong Chen, Maurizio Corbetta, Sandra W Curtiss, et al. The human connectome project: a data acquisition perspective. *Neuroimage*, 62(4):2222–2231, 2012.