Machine Learning project II
predicting mouse behavior from LFPs recording

Anna Diena, Max Duparc, Erwan Seradour
based on the work of Anastasiia Oryshchuk and Sylvain Crochet from the SIP Laboratory at EPFL

***Abstract*—This report is part of the second project of the EPFL Machine Learning course (CS-433) performed in collaboration with "sensory information processing" laboratory and with subject the exploration of Determinants of Sensory Perception**

I. Introduction

The aim of our project is to analyse the recording of low field potentials(LFPs) taken from different mice and different region of the brain and predict if the animal will react to a certain stimulus. LFPs are the averaged postsynaptic activity of a few hundred to a few thousand cells. When the system activity is composed of correlated events, the analysis of this averaged activity can give much greater insight into the complete network compared with that of a single unit [1]. In a previous time with respect to the experiment we are analysing, the mouse was trained to answer to a whisker deflection stimulus by receiving a reward if he licked in certain time interval. Nonetheless even after the mouse is trained he doesn't always respond to the stimulus. There are many factors that could influence the reaction or non-reaction of the mouse to stimuli, in particular in this project we analyze the recording of the medial prefrontal cortex (`mPFC`), primary tongue-jaw motor cortex (`tjM1`), primary whisker sensory cortex (`wS1`) and we will establish if by analysing the recording before the stimulation is possible to predict the reaction or non - reaction of the mouse. In particular we analyze the three brain areas separately to observe if there is one that has a bigger influence on the future actions of the mouse, we also considered the recording across different time intervals before the stimulation to see if it's possible to select one time interval which is particularly relevant for this task.

II. Exploratory data analysis

A. dataset structure

During this this project, we worked with the row data that was collected by *Anastasiia Oryshchuk* for her PhD in the Sensory Processing Laboratory.

It consist of the recording of the local field potentials of a selected brain zone, and this for 24 sessions of 300 to 1000 tests.

A test consist in the following operation:

- A period of 2.5s where the mouse is calm.
- A stimulus. In our case, this is a whisker stimulus. It has a specific amplitude, graded increasingly form 0 to 4, where 0 denote the non-stimulation.
- A period of 1s afterwards where the mouse lick (hit) or doesn't lick (miss) a reward spout.

Each sections hold for data:

- The LFPs recording(with a sampling frequency of 2000Hz).
- The tests time.
- The amplitude tested.
- The outcome of the test (Hit/Miss).
- The zones that were studied
- Other datas that we didn't use such as the mouse name, the jaw movement tracking...

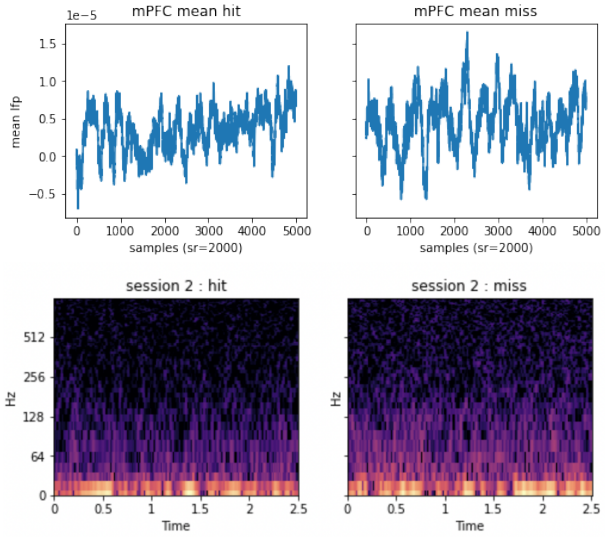

Fig. 1: Examples of the LFPs recording for the 2.5s time interval before stimulus in graph and spectrogram form

One important thing to note is that during the section, the LFPs recordings were collected in parallel from 32 electrodes located at different depth. In order to reduce the data size, and because they are very similar, we only got hold out of the median one.

The goal of our project is to explore the circumstances that would make the mouse react to or miss a stimuli. We are therefore only interested at trials where the amplitude tested was bigger than 0. In this case, an Hit is define as a success and a miss is a failure.

B. Data exploration

Before training any model, we explore the dataset. From this first analysis, we observe two main things. First, there is

not any obvious visual pattern either in frequency or in time that allows us to determine whether a test is a hit or a miss. We can see it in figures 1 and 3. Secondly, the last 4 samples of each test happen sometimes after the stimulation (ref figure 2).

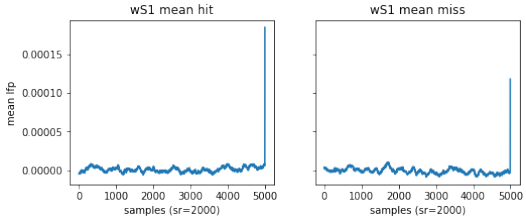

Fig. 2: Why we should cut last 4 samples.

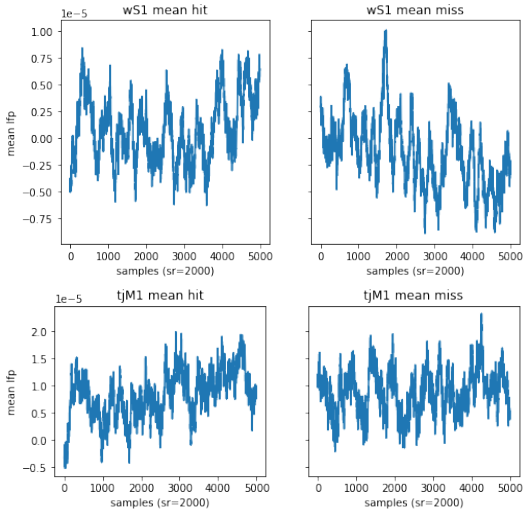

Fig. 3: Example of all hit and miss average for wS1 and tjM1

III. Data pre-processing

First, we remove the last four samples of each test for the reason mentioned above. Then, we normalize our data as some machine learning algorithm were struggling with the very small values of the time series. We apply Standard normalisation to each session separately.

A. balancing data

When we started working on the project, we were not balancing our dataset and we identify several problems resulting from that. First, we were not able to compare the results we obtained by training one model per session and one model for all the sessions, as each session was balanced differently. For instance, if session 1 has a 70% ratio of hits and session 2 has a 30% ratio of hits then a naive classifier predicting the most likely class would have 70% accuracy when trained session wise and 50% accuracy when trained with all sessions pooled together (assuming session 1 and 2 have same number of samples). Secondly, a classifier trained with all sessions pooled together could learn to predict the session and then predict the most likely class for this session, thus achieving a good accuracy but not learning the given task. Hence, we decide to balance our dataset session wise.

B. Amplitude

In the mainframe of our project, we only work on amplitude 2 and 3. We make this choice because in amplitude 1 and 4 the mouse either almost always doesn't respond to the stimulus or almost always responds to it.

IV. Models exploration

A. Baseline

We use as a baseline an SVM classifier from scikit learn with a Gaussian kernel, as it is a fast and simple model while still being pretty good. We train it on the raw preprocessed data. To measure the accuracy we use 20 fold cross-validation. We can see the results in figure 4. We obtain similar results to a random classifier. From this first experiments, we identify that we will need to use specific model to time-series and neural activity recognition and that we want to quantify the statistical significance of our results as it could be that they are not significant considering the dataset size and its noisiness. After exploring a bit the literature we identify some possible solutions :

- First, when doing exploratory analysis, we identify that the response of the mouse could slightly depends on the mean and the standard deviation of the LFPs just before the stimulation. To verify this hypothesis, we apply binning.
- Secondly, we use Ts-fresh a library specialised in extracting features from times-series, thus transforming our problem into classical classification.
- Thirdly, we use pyts a library specialized in handling times series.
- Fourthly, we transform our times-series into images and then apply standard image classification algorithm.

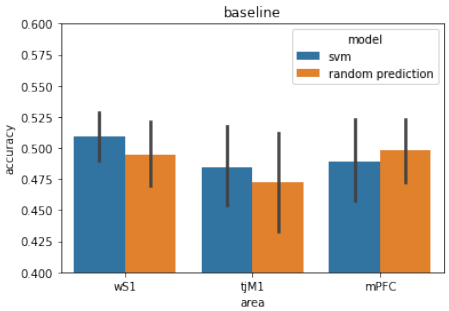

Fig. 4: Baseline accuracy, the error bar represents the 95% confidence interval

B. binning

After prepossessing our data, we splits the times series into bins of equal length and then for each bin compute the mean, the standard deviation, the minimum value and the maximum

model	area	parameters	accuracy
svm	wS1	bin size=100, kernel=rbf, C=0.8	0.52489
svm	tjM1	bin size=500, kernel=rbf, C=0.8	0.5465
svm	mPFC	bin size=50, kernel=rbf, C=0.8	0..4984

TABLE I: Binning best accuracy per area

value of the bin. We come up with the first 2 features during exploratory analysis and the last 2 are easy to compute and seem interesting. We train an SVM classifier with a Gaussian kernel using different bin sizes. We measure the accuracy using 20 fold cross-validation and the results are in figure 5. We obtain a small accuracy gain going from 0 to 5 percent depending on the area, but we only obtain significant results for tjM1 with an accuracy of 54.65 percent.

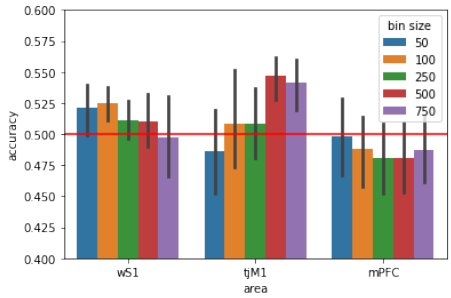

Fig. 5: Accuracy using binning (bin size in sample size), the error bar represents the 95% confidence interval.

C. *tsfresh*

Tsfresh is a python package specialized for handling Time series. It indeed automatically calculates a large number of time series characteristics, that it defines as features. In literature we saw that Tsfresh can be used in the context of neural activity classification [2].

Therefore, we used the `extract features` command from tsfresh on our dataset. We then tried this new dataset with some simple model and we got a small improvement, although not significant.

This is why we decided to use Time series forest.

D. *pyts*

Pyts is a python library designed for time series classification which has already been used in literature for time series classification of neural activity [2]. Among the available algorithms that we tried, we selected "Time Series Forest" which is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control overfitting. Firstly, we tuned the hyper parameters of the model separately on the different brain area, but without making any distinction between the stimulus amplitude. Afterwards, since we want to analyze thoroughly stimulus 2 and 3, we try to tune the hyper parameters of the model on the two stimulus separately. From the different results obtained we see a maximum accuracy gain of 5 percent for the model with stimulus 2 on mPFC, however due to the high standard deviations seen in the predictions we decided that improvement with this second method was not stable and consequently we decided to continue with the first set of hyper-parameters calculated. To estimate the reliability of our prediction we used 10 - fold cross validation and calculated the mean accuracy obtained and its standard deviation across different brain areas and pre-stimulus time intervals. We trained our model both on stimulus 2 and 3 separately, on the two stimulus together and on the two stimulus together adding a feature to describe the stimulus amplitude but we didn't notice any relevant improvement with it (ref. figure 9). Using this method, analyzing stimulus 2 and 3 together without added feature we got the following maximum accuracies:

area	parameters	max accuracy
wS1	d=2, n=900, w=1	0.5207
mPFC	d=10, n=700, w=9	0.5237
tjM1	d=6, n=100, w=7	0.5462

n=estimators numbers, w=windows number, d=max depth

1) Session based prediction vs all session: This question is very interesting. Indeed, from a biological point of view, we should consider every session as independent, because those were not done using the same mouse, and even if the brain area recorded is always the same, different neurons may have slightly different characteristics.

But, for every session, we only get between 100 and 500 usable tests, something that, in the world of machine learning, isn't a lot. If we consider every session together, then we get an average of 2500 tests, which is more in line with what is used in the field.

Figures 6 and 7 are representing the accuracy we got when considering both methods.

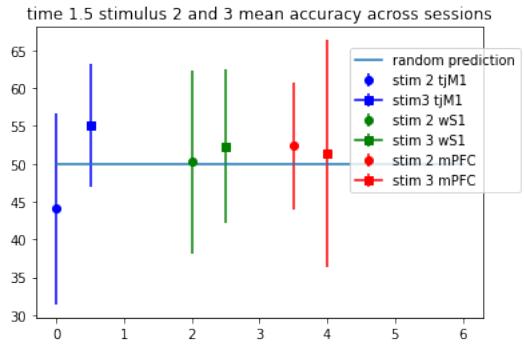

Fig. 6: Average accuracy when working section based with Time series forest

If our average accuracy is only slightly improved when working on all sessions combined, it's standard deviation is far lower. This is what we expected, as a consequence of big number law.

2) Time Sensitivity: We can't say anything really conclusive about how the pre-stimulus time interval considered is linked accuracy. The results we found are in figures 7.

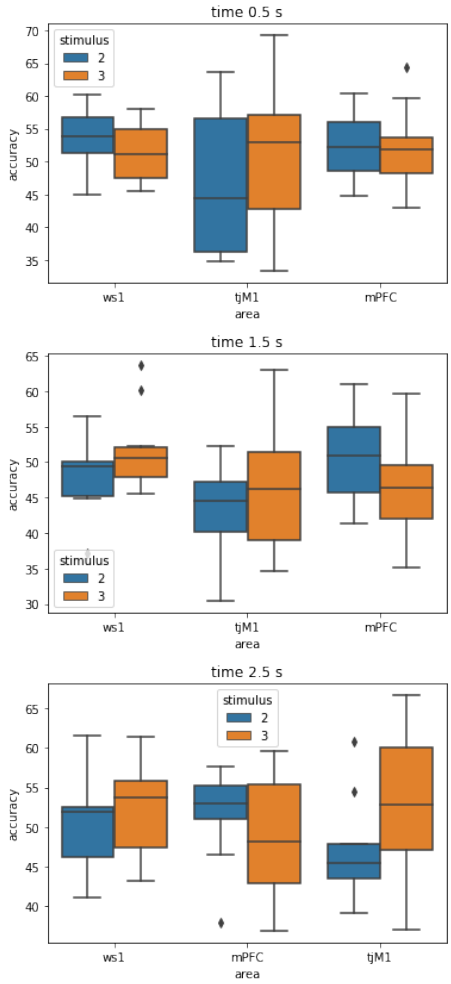

Fig. 7: Accuracy box using Time Series Forest on our data set with different pre-stimulus time, stimulation amplitude and recorded zone.

3) Area importance: Now, figures 5, 7 and 9 (appendix) gives us some information on the zone that hold some importance in predicting the reaction.

Indeed, it seems very difficult also in this case to predict which brain zone holds more information, we reached the maximum accuracies with tjM1. However taking in consideration,the large standard deviation that greatly affects our predictions it seems that no zone can be reliably used for the prediction.

By combining this discovery with the time sensitivity, it seems that using the methods analyzed until now is not possible to have a reliable prediction, furthermore even if the three brain areas analyzed have different functions, it does not seem possible to establish which one has more importance for the reaction of the mouse.

E. Image recognition

Transforming short times-series into images is sometimes used in neural activity recognition especially when the images are spectrograms. It is an interesting idea if you can identify patterns when looking at your plots. Even if during exploratory analysis, we do not find any such patterns, we decide to do it as we have explored only a portion of the 2500 tests and it will allows to determine if any such patterns exist.

model	area	image kind	parameters	accuracy
resnet34	wS1	time	epochs=15, freeze epochs=2	0.5096
resnet34	wS1	frequency	epochs=15, freeze epochs=2	0.4958
resnet34	tjM1	time	epochs=15, freeze epochs=4	0.6034
resnet34	tjM1	frequency	epochs=15, freeze epochs=3	0.5258
resnet34	mPFC	time	epochs=14, freeze epochs=4	0.5271
resnet34	mPFC	frequency	epochs=10, freeze epochs=2	0.4961

TABLE II: Image recognition results

To do this experiment, we consider stimulus 2 and 3 together with all the sessions pooled together. For each tests, we create a graph in time domain and a spectrogram. The results is something similar to what you can see in figure 1. For each area, we train two models one using the time images, the other the frequency images. To train our model, we fine tuned a pretrained resnet34 using fast.ai. Fast.ai is deep learning library built on pytroch specialised in transfer learning and fine tuning is a way to do transfer learning. As the model takes some time to train we don't use k-fold cross-validation, but we split our dataset in a training set with 0.8% of the samples and a test set with 0.2% of the samples. Thus, the accuracy we measure is less precise especially with tjM1 that has only 582 data points. We get the following results (Table 2).

We only have a significant increase for tjM1 in the time domain. We have a 10% accuracy for gain compare to baseline.

V. Conclusion

It seems that it is very complicated to have a good predicting model using LFPs. Everything tried in this project was not spectacular, but there is hope.

Indeed, we also tried to use time forest on the tsfresh dataset that we created earlier and got results listed in figure 8.

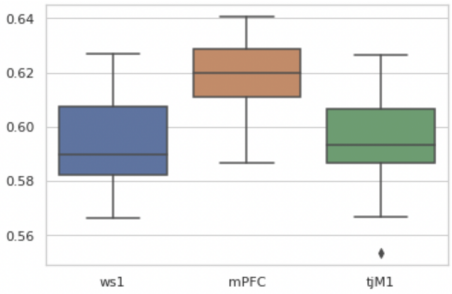

Fig. 8: boxplot of the accuracy using time forest on tsfresh all sessions datasets.

If theses results are very promising, they should be taken with precaution. Indeed, the dataset used was not balaced and for lack of time[1], we were not able to recompute the features. Thus, the results are not comparable to what we have now,

[1]As extracting tsfresh data from our requires hours of computations

but furthering this angle could be more fruitful than what we did.

REFERENCES

[1] *Shils, J. L., Tagliati, M., Alterman, R. L. (2002). "Neurophysiological monitoring during neurosurgery for movement disorders. Neurophysiology in neurosurgery, 405-IX."*

[2] *Ivan Lazarevich, Ilya Prokin, Boris Gutkin " Neural Activity Classification with Machine Learning Models Trained on Interspike Interval Series Data."* https://arxiv.org/pdf/1810.03855.pdf

[3] *"Developing Machine Learning Algorithms for Behavior Recognition from Deep Brain Signals", 2020 University of Denver* https://digitalcommons.du.edu/cgi/viewcontent.cgi?article=2762

VI. APPENDIX

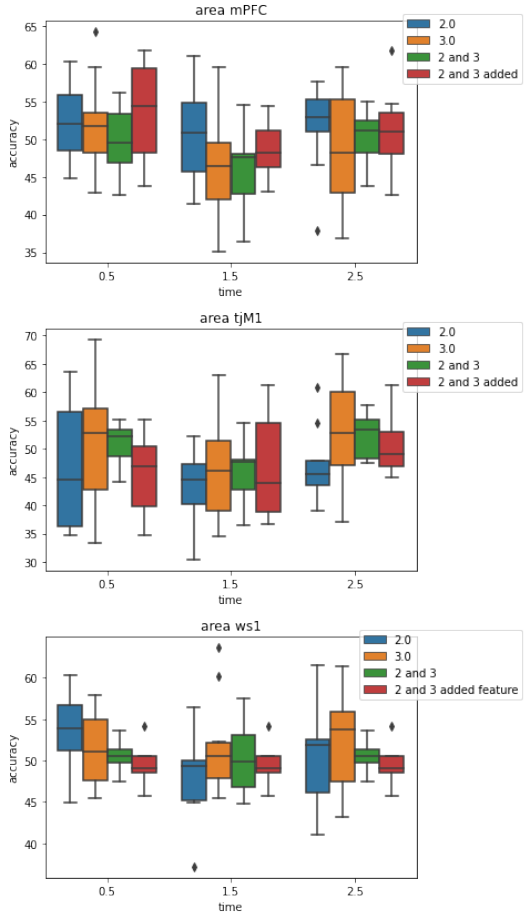

Fig. 9: Average accuracy obtained with time series forest when working on all sessions considering stimulus 2 and 3 separate, together, and the effect of an additional feature indicating the stimulus amplitude when the two amplitude are considered together