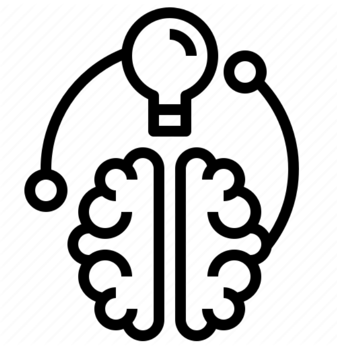
Faculdade de Engenharia da Universidade do Porto

Neuroengineering

“Prediction of seizure-like activity in *in vitro* cortical populations”



Teachers:

João Cunha jpcunha@fe.up.pt Paulo Aguiar pauloaguiar@ineb.up.pt

Brainstorm Team:

Afonso Pinto up201503316@fe.up.pt Athina Papastergiou up201810610@fe.up.pt

Beatriz P. da S. e Souza up201811344@fe.up.pt Karina Damberg up201811712@fe.up.ptt

Summary

Epilepsy is a neurological condition that affects approximately 70 million worldwide. New approaches to investigate the mechanism of this disease such as machine learning have been used to deeply understand the seizure-like behavior. Here, our goal is to predict seizure-like activity in in vitro cortical cells with 10ms advantage.

Key-words

epilepsy; machine-learning; seizure

Acknowledgment

This project was the result of several contributions and collaborations, given directly and indirectly, but all of them essential to its realization. We would like to express our sincere thanks to all those who have made this work possible.

To Professor Paulo Aguiar for the guidance given and valuable constant monitoring during the development of the project.

Index

1. Introduction

Epilepsy is a chronic disorder, the hallmark of which is recurrent, unprovoked seizures which may be related to a brain injury or a family tendency, but often the cause is completely unknown. Many people with epilepsy have more than one type of seizure and may have other symptoms of neurological problems as well [1].

Seizures are characterized by abnormal electrical activity generated by large neuronal populations. One of the main treatments is electrical brain stimulation, which focuses on suppressing the abnormal activity [2].

Patterns of human epileptic seizures and neuronal spike trains share similarities such as specific durations of events (seizures or spikes) and inter-event intervals. The challenge is to find a way of locating the points (onsets) and the extents (durations) of these abrupt changes [3].

Computational modeling is a powerful tool for understanding the underlying mechanisms involved in seizure genesis and its dynamics, and has proven to be very useful in aiding with the design of more efficient seizure onset detectors and brain stimulation protocols [4].

2. Technologies

At the base of this project are the following technologies

2.1 Python / Jupyter

Python is an interpreted, high-level, general-purpose programming language, with lots of packages at our disposal, being particularly prominent in the context of the project the ones related with machine learning algorithms.

On top of python we used Jupyter Notebook which is a tool to give interactivity to the data science and scientific computing necessary in the development.

2.2 Bokeh

Bokeh is an interactive visualization library that targets modern web browsers for presentation.

Our goal when using it was to provide elegant, concise construction of versatile graphics, and to extend this capability with high-performance interactivity over very large datasets.

2.3 HDF5

HDF5 is a unique technology suite that makes possible the management of extremely large and complex data collections. The results of the seizure-like activity were saved in this format.

2.4 Tensorflow

TensorFlow is an end-to-end open source platform for machine learning made by google and it was available and used as a python package.

3. Data Manipulation

3.1 Parse HDF5

Since the results of the seizure-like activity were saved in this format, we had to read that into python data structures, to do so we used *McsPy* which is a set of Python tools, classes and functions, to handle data recorded and processed with Hard- and Software produced by *Multi Channel Systems MCS GmbH*.

With that we were able to plot raw voltage as follows:

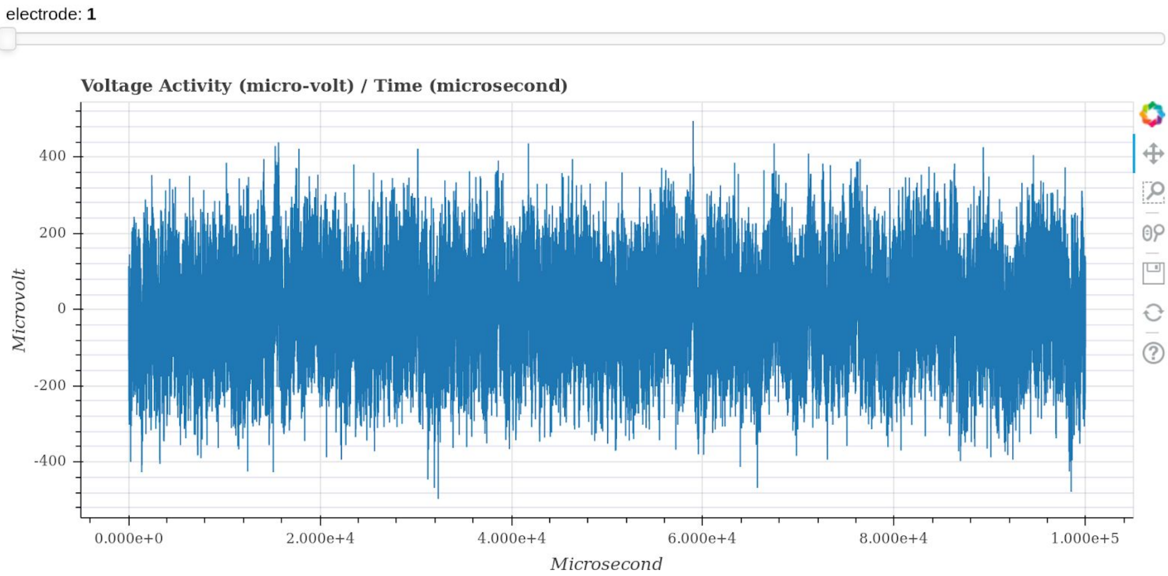


Figure 1. Raw Voltage Plot

3.2 Detecting Spikes

One of our first tasks were to detect spikes over that raw data.

To do so we calculated the standard deviation of the electrode voltage values, multiplied it by an editable factor (by default 5) and compared each absolute value with it. If it was higher we marked that activity as as spike. Here is one example of output:

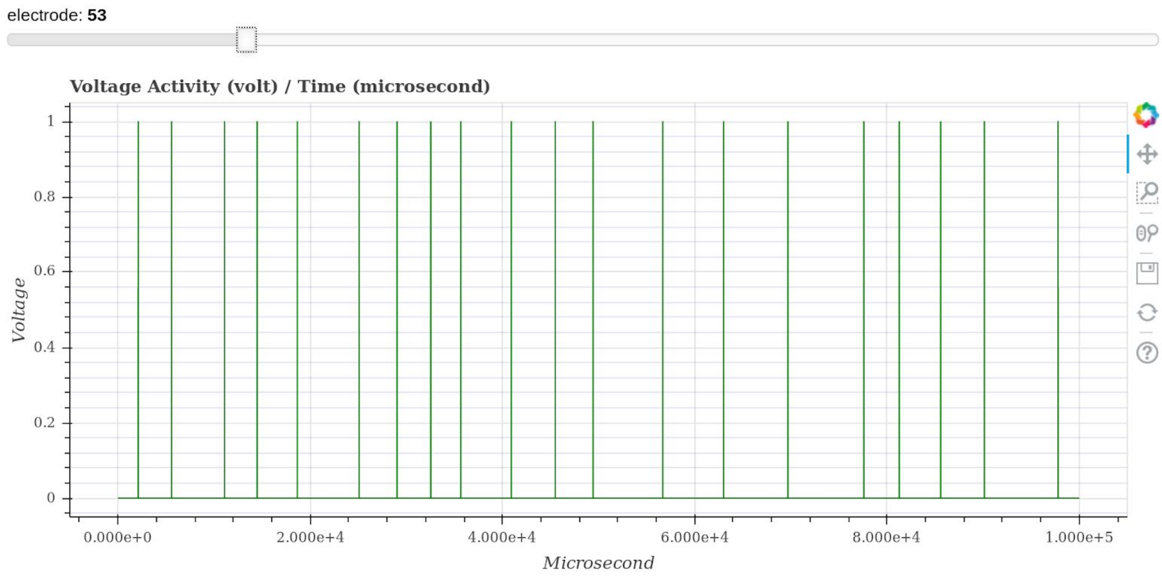


Figure 2. Spike Activity Plot

3.3 Detecting Seizures

The next step was to detect seizures. There is no concrete definition of a seizure, which makes this task somewhat partial, by default we mark some activity as seizure if at least a given percentage (25% for example) of the electrodes were spiking. Changing this to follow any other rule is easily done.

With the detection of seizures done, we were able to do a Raster plot:

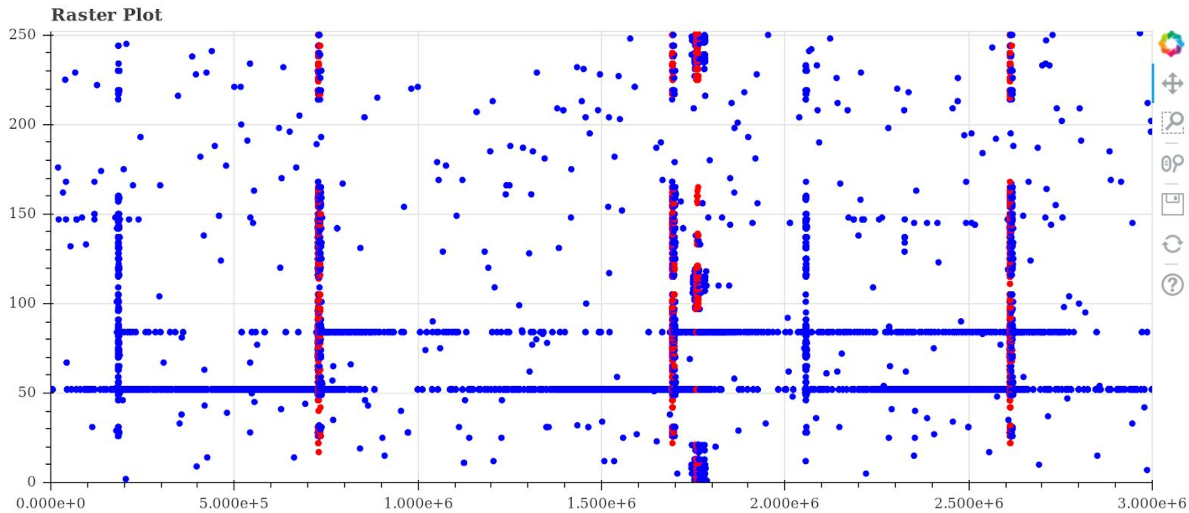


Figure 3. Raster Plot

4. Machine Learning

We modelled the machine learning part of the project as a binary classification problem.

Given all the electrodes voltage values at a given time the algorithm should classify it between a *pre-seizure activity* [True] or *normal activity* [False].

Multiple algorithms were considered, as they were mentioned in suggested literature, we opted in for a Neural Network as it typically works better than the other strong candidate (k-nearest neighbors)[5].

4.1 Neural Network

We build a 2-hidden layers fully connected neural network (also known as multilayer perceptron) with TensorFlow. We are using some of TensorFlow higher-level wrappers (tf.estimators, tf.layers, tf.metrics, ...)

There are 256 nodes in each hidden layer.

The input layer has 256 nodes as well, one for each electrode voltage.

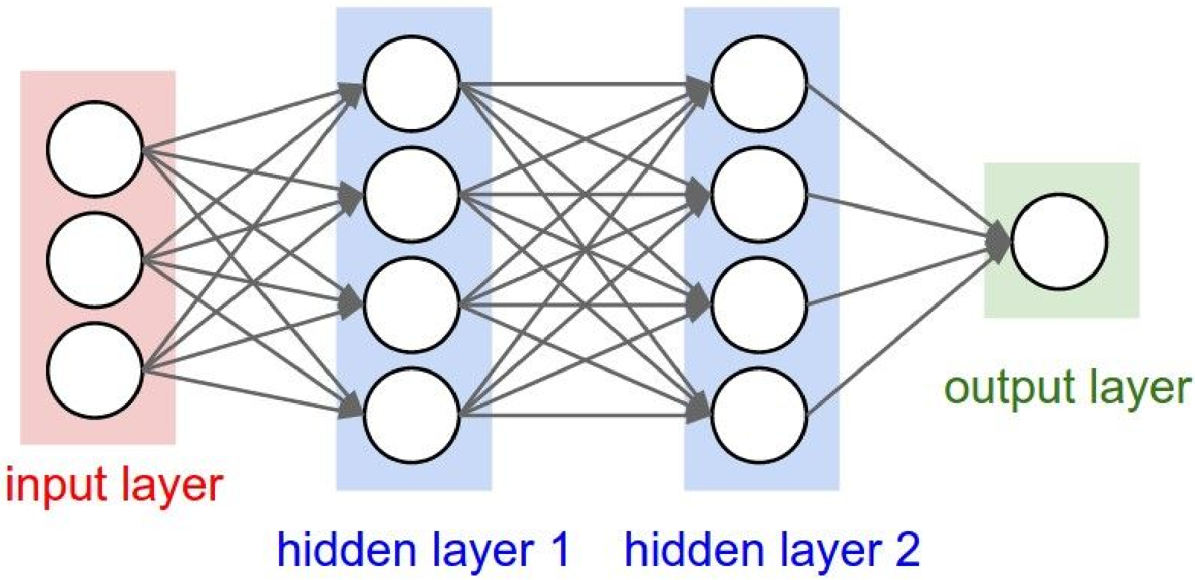


Figure 4. Neural Network Representation

Given the unbalance of the dataset some measures were applied:

- Instead of approaching this problem as prediction pre-seizure **at** a given time (~100ms) before, we are now trying to predict a pre-seizure **until** a given time. This increased the % of *True* values (classified as pre-seizures) from less than 1% to 12% of the dataset.
- The neural network was configured to give more credit to classifying something as *True* (seizure) order than *False* (not seizure). [6]

4.2 Results

We random splitted the data into (80%) training, (20%) testing.

We evaluate our model with the following measures (more can easily be added):

'accuracy': 0.872285,

'f1_score': 0.22751863,

'false_negatives': 76607.0,

'false_positives': 22.0,

'true_negatives': 522961.0,

'true_positives': 410.0,

'specificity': 0.1999903630344357,

'sensitivity': 0.0010647442406425862

4.3 Analysis

The overall results of the project are not encouraging.

The major indicative of this is the sensitivity result, given the context of the problem, our goal would be to have a much higher true positive rate. We are uncertain why this is happening but if we had more time to spend on the issue, these are some of the things we would like to try (following the given order):

- Try other anti-unbalanced dataset strategies:
    - Oversampling - Using *smote* (Synthetic Minority Over-sampling Technique) - which roughly looks at the minority class and generates new values considering the k nearest neighbours.
- Allow the retraining of the model with other simulation files - currently the training was only made with the *cortical_same_chip/2019-02-11T11-42-58Chip_3_DIV17.h5*. Since we used tensorflow this can be easily achieved but we don't support it in our notebook yet.

- Try other machine learning algorithms, a tool like RapidMiner allows the application of multiple algorithms to a given problem. We could use some tool like that to point us in the right direction (algorithm with better results for this problem) and then tweak it further.

5. Deployment

5.1 Requirements

We strongly recommend the use of a virtual environment with:

- Python 3
- All the pip requirements: `pip install -r requirements.txt`

5.2 How to use

- In the command line type: *jupyter notebook* (a tab should open in your default browser)
- Go to the location of the *Brainstorm.ipynb* and run it.
- In the Import Data module you may want to change the path to file you want to read
- Run all the cells in order.

6. Scrum

During the development phase we tried to follow a scrum approach, with 2 weeks sprints and weekly meetings. The sprint reviews and retrospective were made outside the redmine scope due to difficulties using the tool. In the other hand, the creation of a backlog and the assignment of issues to sprints and persons was done in the mentioned tool.

7. Conclusions

Before most we are thankful to be given the opportunity of working in a project with the relevance of the one we had in our hands, in terms of evolving the neuroscience field.

The final results of our work were not great; however, given the low to none data science background of the team, the use of very large datasets with very unbalanced classes and the novelty flavour of the project, our team believes that the general requirements of the project were met.

- Data manipulation
- Plotting
- Machine Learning algorithm

Here are some of the strong points of our work:

- Very flexible/editable.
- Good code structure.
- Good base for plot capabilities.
- Use of anti-unbalanced machine learning technics.
- Good model evaluation metrics.

And here are some weak points and possible features that would improve the project even further:

- The data manipulation takes about 1 hour when we use all the dataset.
  - We don't know what we can do better here, we tried to parallelize the work with multiprocessing pool maps but the results were even worse.
- Plotting all the raw voltage as we are doing makes the bokeh interactions extremely slow. Bokeh allows the plotting of streaming data, our we could try to parametrize the viewing range to load only a part of the data.
- Seizure detection formula could be improved.
- The slight change in the project scope is not something we have done willingly.
- The machine learning improvements mentioned in 4.3.

Bibliography

[1]. https://www.epilepsy.com/learn/about-epilepsy-basics/what-epilepsy (accessed in 11/06/2019).

[2]. Sohal, V. S., and Sun, F. T. (2011). Responsive neurostimulation suppresses synchronized cortical rhythms in patients with epilepsy. Neurosurg. Clin. N. Am. 22, 481–488, vi. doi: 10.1016/j.nec.2011.07.007

[3]. Seneviratne, U., Karoly, P., Freestone, D. R. and Boston, R. C. (2019). Methods for the detection of Seizure bursts in epilepsy. Front. Neurol. 10:156. doi: 10.3389/fneur.2019.00156

[4]. Ehrens, D., Sritharan, D. and Sarma, S. V. (2015). Closed-loop control of a fragile network: application to seizure-like dynamics of an epilepsy model. Front. Neurosc. doi: 10.3389/fnins.2015.00058

[5]. https://pdfs.semanticscholar.org/bdf7/8610dce0cd6808221e8df179597b028d272e.pdf (accesed in 17/06/2019)

[6]. https://datascience.stackexchange.com/questions/12886/tensorflow-adjusting-cost-function-for-imbalanced-data (accessed in 17/06/2019)