

Artificial Intelligence - Project Report: "Disprove the existence of the free will"

Tilen Matkovic

Faculty of Computer and Information Science

Ljubljana, Slovenia

matkovic.tilen@gmail.com

ABSTRACT

The neuroscience of free will is trying to show us if we have free will or not. Some experiments have shown that there is some kind of brain signal that activates just before the moment, that the person is being aware to make some decision. This measure of activity is called Bereitschaftspotential or Readiness potential. In this project I tried to reconstruct an experiment in which we try to predict when a person is going to press a button. Using Emotiv EPOC EEG headset and OpenViBE software platform, I build blocks for capturing the data and used machine learning methods to evaluate my signals.

KEYWORDS

EEG, emotiv, EPOC, machine learning, neuroscience

1 INTRODUCTION

Do people have free will? Some studies have found that a person's brain commits to certain decisions before the person becomes aware of having made them. These delays are ranging to about half a second, but other scientists were able to predict with 60% accuracy whether subjects would press a button up to 10 seconds before the subject became aware of having made that choice [3].

Benjamin Libet made an experiment in the 1980s, in which he asked subjects to choose a random moment to flick their wrist while he measured activity in their brain [2]. Although it was known, that electrical signal, called Bereitschaftspotential (also "readiness potential"), preceded the physical action, Libet wanted to know how this corresponded to the felt intention to move (see picture 1). His findings suggest that decisions made by a subject are made on a subconscious level and only afterwards translated into a conscious decision. He found out that the readiness potential started about 0.35 seconds earlier than the subject reported conscious awareness that he/she feels the desire to make a movement [1]. Libet concluded that we have no free will in the initiation to our movements to some extent.

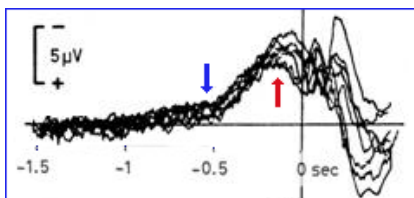


Figure 1: Bereitschaftspotential ("readiness potential") - blue arrow presents start of readiness potential, red arrow presents conscious awareness of the decision to move hand.

In this project, I tried to reconstruct this experiment with my method using *Emotiv EPOC* EEG headset and *OpenViBE* software platform.

2 METHODOLOGIES

2.1 Emotiv EPOC

Emotiv EPOC is a wireless, 14-channel mobile EEG system or headset, used for monitoring brain activity. It uses saline based wet sensors, can also monitor 9-axis motion data and transmits data wirelessly through Bluetooth. There is an official open source *Emotiv SDK - Community Edition* software for collecting data, but there also exists reverse engineered *Emokit* software, made by EPOC users, which is used for collecting raw stream data from the Emotiv EPOC headset. One of the forked repositories from *Emokit* is called *Hemokit*, which can print EEG data, format it as JSON, serve it via TCP or Websockets and read from real devices and dump files. *Hemokit* can be used to stream EEG data to OpenViBE.

2.2 OpenViBE

OpenViBE is a free and open source software platform dedicated to designing, testing and using brain-computer interfaces. It is used for real-time processing of brain signals. It can be used by programmers or people with no programming skills. With OpenViBE designer tool you can put predefined boxes and connect them according to their inputs/outputs. If some of the boxes do not exist, you can create new boxes, program them in C++, Python, Lua or Matlab, and insert them in your scenario.

3 THE EXPERIMENT

The idea of the experiment is the following - subject is pressing the buttons on the keyboard at random times, but with a few seconds in between. EEG signals are being saved into a file while doing this. Next, the computer learns when the user is going to press the button on the basis of the recorded signals, tries to predict when he/she is going to press the button next time and triggers some sound or visual effect. In my case, I concentrated on offline evaluation (using machine learning methods, e.g. cross validation) to evaluate my approach to the problem.

The first thing to do, was to establish the connection between EPOC headset and computer. Although there exists official Emotiv SDK, I used Emokit or its forked version Hemokit for getting the raw data from the headset. Hemokit has a nice feature, which can send raw data directly from headset to OpenViBE, which I discovered by chance. So now we are getting data to OpenViBE, next thing to do is to create scenarios for each task:

- (1) acquisition - save acquired signal from headset to a .csv file, separately acquire and log keyboard stimulation times when certain button is pressed (and save to a .csv file),
- (2) classifier/trainer - use .csv files from previous step and learn when user presses the button, print average classification accuracy from k-fold cross validation and save trained hypothesis to a file (this step is done offline),
- (3) online testing - use hypothesis from previous step to predict button pressing (online).

Acquisition step is pretty straightforward. Trainer step is described in detail in next subsection.

3.1 Trainer

After acquisition step, we have to preprocess signal data first. In our case, we have 14 electrodes, so this means 14 attributes. The idea for preprocessing is that we break each electrode signal into chunks/windows of equal length (e.g. 50 ms). For each of these windows we have to assign a class - in my case, I did the following:

- if a button was pressed in some window, mark that window as 1; also get a few windows before this event (e.g. 4 windows) and mark them also as 1 - we are interested in part of the signal that happened just before the pressed button,
- if some windows are before those windows marked as 1, mark these windows as 0 (number of these windows is e.g. 10),
- if some windows are after those windows marked as 1, mark these windows as 0 (number of these windows is e.g. 2),
- rest of windows are ignored and are not contained in learning phase.

Preprocessing is abstracted in the picture 2. Last step is calculating the mean value of each window, so because we have 14 electrodes, we get 14 attributes and one class.

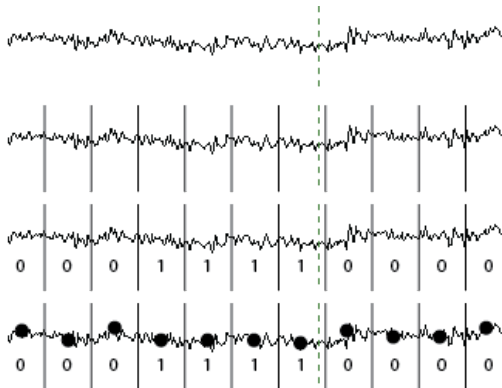


Figure 2: Abstract preview or preprocessing and marking data for learning phase. Each row presents phase of preprocessing (1. - original signal together with pressed button time (broken line), 2. windowing, 3. assigning classes, 4. calculating mean value of each window).

For learning algorithm I used *scipy's SVM*.

Table 1: Average classification accuracy for 10-fold cross validation for 50 ms long window size. X-Y-Z notation means X-zero classes (before), Y-one classes and Z-zero classes (after). Each of the two rows presents different acquisition example.

5-10-2	10-10-2	6-8-0
0.476	0.517	0.527
0.579	0.590	0.667

Table 2: Average classification accuracy for 10-fold cross validation for 100 ms long window size.

5-5-1	3-8-2	4-4-0
0.502	0.523	0.435
0.612	0.576	0.665

Table 3: Average classification accuracy for 10-fold cross validation for 200 ms long window size.

3-3-1	1-2-1	4-4-0
0.476	0.486	0.502
0.591	0.568	0.625

4 RESULTS

For calculating the results, I first acquired my signal data while pressing the keyboard buttons. Then I used these data for calculating the k-fold cross validation (offline). My results are evident from the following tables (see tables 1, 2 and 3). I have made 10-fold cross validations on 2 different acquisition examples:

- first was 126 seconds long and has 21 button presses,
- second was 471 seconds long and has 63 button presses.

We see that classification accuracies for every setting are pretty low. The highest value is at the 50 ms long window size, setting 6-8-0, which is 0.667. Being sceptical, that some values are biased, I was still not sure if this test could prove that there is some connection between our classes. Later I discovered that most of the times, just before when I pressed a button, I moved my eyes and this could be seen from EEG signal (see picture 3). Eye movement can in EEG signal be seen as letter 'U' or horizontally flipped version of it. This could lead to slightly higher classification accuracies in the second captured signal.

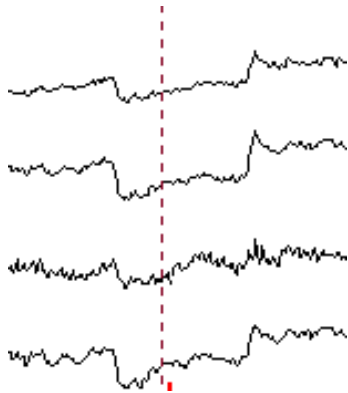


Figure 3: EEG signal with eye movement ('U' shape).

5 CONCLUSIONS

In my opinion, bad results are showing that this is probably not the best way for testing the free will experiment. Perhaps the problem is in the Emotiv EPOC headset, which is not as precise as the professional EEG devices. The problem may also be in the small number of attributes. Another idea for additional attribute is calculating the maximum frequency of each window.

Because this was my first project using EEG signals, I did not know how deal with these kind of signals. Later I found out that I could get a lot of more informations with filtering the EEG data and removing eye blinks and other noise. My current implementation is only dealing with raw EEG data, without filters. The project could have been done better considering filters, and also a more precise EEG headset.

The code that I used for the free will experiment, or at least my approach towards it, can be seen on the following *github* repository: <https://github.com/matkovic/free-will>.

REFERENCES

- [1] Benjamin Libet. 1985. Unconscious cerebral initiative and the role of conscious will in voluntary action. *Behavioral and brain sciences* 8, 4 (1985), 529–539.
- [2] Benjamin Libet, Curtis A Gleason, Elwood W Wright, and Dennis K Pearl. 1983. Time of conscious intention to act in relation to onset of cerebral activity (readiness-potential). *Brain* 106, 3 (1983), 623–642.
- [3] Chun Siong Soon, Marcel Brass, Hans-Jochen Heinze, and John-Dylan Haynes. 2008. Unconscious determinants of free decisions in the human brain. *Nature neuroscience* 11, 5 (2008), 543.