Machine Learning Engineer Nanodegree

Capstone Project

Viacheslav Nesterov
October 1, 2018

Recognizing patterns of brain activity using Brain Computer Interface

Brainwave analysis (reading vs writing state)

I. Definition

Project Overview

The purpose of this project is to experiment with electroencephalogram (EEG) device using Brain Computer Interface (BCI) with Machine Learning tools and techniques for proving the concept of human-machine interactions, putting start for more profound and deep research in the area and building technology that will allow using BCI for communication and machine interaction. Results of this project will inspire author to move forward. Therefore, this project is a starting point for future commitments.

Brain Computer Interface sometimes called a neural-control interface (NCI), mind-machine interface (MMI), direct neural interface (DNI), or brain–machine interface (BMI), is a direct communication pathway between an enhanced or wired brain and an external device. BCI differs from neuromodulation in that it allows for bidirectional information flow. BCIs are often directed at researching, mapping, assisting, augmenting, or repairing human cognitive or sensory-motor functions[1].

Brain computer interface technology represents a highly growing field of research with application systems. Its contributions in medical fields range from prevention

[1] https://en.wikipedia.org/wiki/Brain–computer_interface

to neuronal rehabilitation for serious injuries. Mind reading, and remote communication have their unique fingerprint in numerous fields such as educational, self-regulation, production, marketing, security as well as games and entertainment. It creates a mutual understanding between users and the surrounding systems[2].

Student motivation is research and learning the BCI technology with applying Machine Learning algorithms to create systems of communications between human and machine or in between humans with machine as intermediaries, using brain activity and BCI device.

Project notebook represents analysis and algorithmic model to recognize brain waves patterns for user's mental state/activity. The analysis, model training and prediction were performed on pre-collected datasets from EEG device. Electroencephalography is an electrophysiological monitoring method to record electrical activity of the brain[3]. It is typically noninvasive, with the electrodes placed along the scalp, although invasive electrodes are sometimes used such as in electrocorticography. EEG measures voltage fluctuations resulting from ionic current within the neurons of the brain. The EEG device used for this project is Ultracortex "Mark IV" EEG Headset (8 channels) from OpenBCI[4]. EEG has 8 electrodes that located on designated areas of scalp and provide 8 channels data respectively[5].

Problem Statement

For the purpose of this project it was decided to learn model to recognize two user's states while user performs two different but mentally similar activities: *reading* and *writing*. Reading and writing activities were performed in almost identic environments but within various 24 hours ranges. User was reading the same book and writing at the same desk in same light conditions and pose.

Data collection session consisted of two phases: 1) EEG setup (mounting, connecting to interface, checking); 2) Recording while performing activity (when user reads or writes, EEG is activated and recording signals to file).

[2] https://www.sciencedirect.com/science/article/pii/S1110866515000237
[3] https://en.wikipedia.org/wiki/Electroencephalography
[4] www.openbci.com
[5] http://docs.openbci.com/Headware/01-Ultracortex-Mark-IV#ultracortex-mark-iv-assembly-instructions-electrode-location-overview

For network training the significant amount of data is required. To make collected data consistent and eliminate excessive preprocessing, data samples should be coherent, and noise excluded (occurred due to distractions, unwanted muscular activity, etc.). With that purpose multiple recorded datasets need to fit 'identical' environment and psychological conditions of user. To produce datasets more coherent to each other it was decided to make recordings for around one-minute length.

Necessary information was collected before the project and researched during its execution to recognize fundamentals on how brainwaves are generated, in which diapasons, what impacts brain waves signal (electrical activity) and so on. Author understands that the field of neuroscience is enormously broad to embrace for this experiment and now being in process of discovering its cornerstones. This project is written in layman's terms, as is. Key initiative was to experiment with machine learning algorithms in area according to abovementioned purpose.

Metrics

The benchmarking goal is to test model in real life conditions, i.e. on dataset that is out of lab experiment – absolutely new dataset. For this purposed new dataset was collected at different time. A it was mentioned before, the challenge here is variability of data given users mental and physical conditions. Also, given that data collected is relatively small amount and does not refer to persons conditions, the results expected might be not very much impressive. If total accuracy of developed model is more than 65% it means that approach works, and project goal may deem accomplished. There were few new datasets used for testing. Some of them turned test to prove model's accuracy slightly more than 50%, some of them, up to 93%.

Total model accuracy is calculated by measuring average accuracy for both reading test and writing test. Each test accuracy is measured by number of correct predictions of state. For example, if model predicts 50 seconds of reading from 100 seconds of reading test set, the accuracy is 50%. The formula for this approach is following:

Accuracy = n(correct predictions) / n(seconds)

II. Analysis

Data Exploration

During this project it was noted that the environment, user's mood and psychological condition play very important role. Such conditions like daydreaming, twilight state (when person is drifting to sleep or from awake)[6] become significant factors from data standpoint. Within those factors brain waves of specific diapason - alpha, beta, theta, gamma may skew sample data significantly[7].

Therefore, there were three options for collecting and analyzing data identified:

> **Option 1:** To collect necessary amount of data with multiple datasets obtained within multiple sessions for short period of time. For instance, in between 5pm and 8pm within single 24h period.
>
> **Option 2:** To collect large number of datasets during multiple sessions completed within long range of time - one week, month or couple of months.
>
> **Option 3:** Mixed option - collect data in similar conditions within unidentified period of time. For instance, each day between 6pm and 8pm within couple of weeks.

Option 1 is good to exclude mental and psychological conditions that may impact research results, while option 2 might be good to embrace most occurred conditions for long period of time to consider them and train the model appropriately.

Benefit from 1-st option, where the focus is just on current user's state and psychological condition is to get relatively fast and proved results for research, train model on more coherent data and obviously achieving goal of the project, distinguishing 2 activities' states. However, within this option there is no opportunity to use pre-trained model for prediction of user's states at any other time in future due to lack of background data generated from users mental and psychological conditions. This option is also not perfect to proceed due to persons physical and emotional variance - working on project in lab conditions reading and

[6] https://brainworksneurotherapy.com/what-are-brainwaves
[7] https://en.wikipedia.org/wiki/Neural_oscillation

writing states blur and merge into single 'dreaming/abandoned' passive condition, where person starts loosing focus. This drawback was noticed after numerous attempts and analysis of brainwaves' patterns.

There is much more benefit from 2-nd option, where users mental and psychological conditions may be considered due to long term period of data collection.

Using this option, unique dataset will be created that is useful not only for purposes of this research. Also, dataset collected within such option will help to train the model, which may be used in future any time to predict user's state of given actions (reading/writing). Such model will be more reliable. However, this option is very time and resource consuming.

Given the purpose of the project, option 3 was selected and performed.

8 channels EEG data is represented in µV and collected with 1000Hz sample rate. Filters are applied within equipment firmware to generate data with minimum noise. For this particular dataset the 7–13Hz bandpass and a 60Hz notch filters were applied.

Exploratory Visualization

<u>Obtaining data and first look:</u>

It is necessary to get 8 channels of data from EEG recorded dataset[8]. Raw datafile contains more than that and requires preprocessing and cleanup.

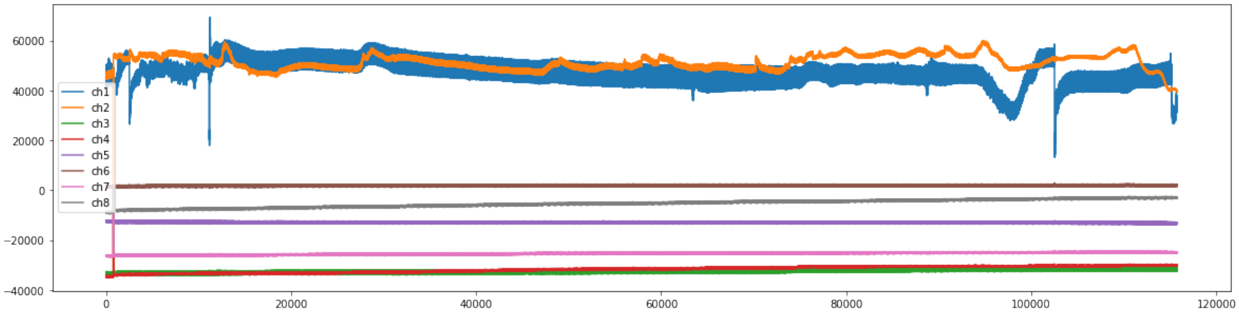
`On below image first 5 rows of data are depicted:`

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
0	185	58618.86	53523.57	-30090.90	-35064.43	-23330.84	-8979.72	-20677.42	-5400.47	0.0	0.0	0.0	21:34:27.470
1	186	56983.38	53428.82	-31389.02	-35162.67	-23361.28	-8974.02	-20932.99	-5383.91	0.0	0.0	0.0	21:34:27.471
2	187	54007.64	53505.58	-34059.57	-36148.29	-23147.31	-8638.86	-22330.82	-5117.63	0.0	0.0	0.0	21:34:27.471
3	188	52158.42	53686.63	-35635.79	-36097.86	-22749.87	-8340.93	-22689.63	-4718.21	0.0	0.0	0.0	21:34:27.471
4	189	51213.07	53778.96	-36408.46	-35831.50	-22692.63	-8360.49	-22636.42	-4754.48	0.0	0.0	0.0	21:34:27.471

[8] http://docs.openbci.com/Hardware/03-Cyton_Data_Format#cyton-data-format-interpreting-the-eeg-data

There are columns 1 - 8 representing corresponding channels (obtained data from respective electrodes 1 - 8). Column 12 is required to break out data by seconds.

```
Below is sample of data visualized:
```



Algorithms and Techniques

Convolutional 1D Neural Network model from Keras library has been selected due to specifics and volume of data to train on. As far as the problem deals with classification and the model will be used for multiclassification in future, the decision was to build corresponding model architecture.

Parameters of model were adjusted in course of training. Current parameters are perfect to work with project problem, the only tweakable parameter that need to be altered is training batch. Usually 16 batches are good for the task. However, if there is less data to train the number of batches could be reduced to 8. It was critical to achieve as much higher accuracy as possible to perform further benchmark testing. So, the architecture of the model and final tweaks were adjusted after finding optimal base variables for data preprocessing.

Benchmark

As it was mentioned earlier for benchmark test new generated dataset is required. New dataset was collected for test in different time frame from those when most of data was collected for model training.

The length of benchmark dataset should be not less than 30 seconds to ensure clean results of the test.

For benchmark test there were several data samples collected. Two datasets, each for reading and writing were picked, 50 seconds and 100 seconds respectively. Such difference in length was selected to make benchmark test even more balanced due to model specific to predict reading state better than writing state.

III. Methodology

Data Preprocessing

Data logged and recorded in high resolution with 1000 Hz sample rate. Ideally each row of data represents 1/1000 fraction of data point sample for 1 second. Other words, there are 1000 samples/ impulses recorded per 1 second. However, due to equipment specifics some data cycles/impulses may deviate from desirable rate causing unequal number of impulses per second rather than 1000. Therefore, the goal is to compile each sample of data input consisting of equal number of impulses for 1 second.

Therefore, balance of the data is needed to get each sample of training data to be equal. It was decided to equalize each sample of data to have 990 rows (recorded impulses). Seconds which will have different number of impulses will be lost.

Data rescaling and spikes removing:

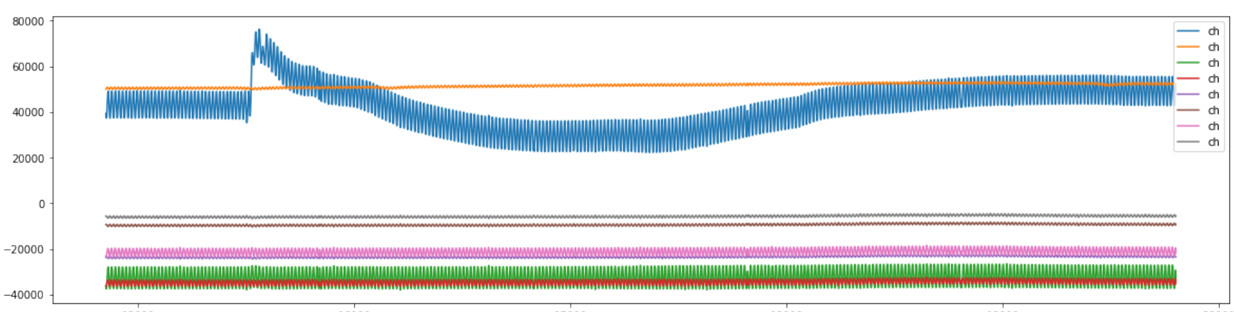
Data received needs to be rescaled. But it can't be rescaled against whole dataset. If to rescale wholistic dataset, which will be hundreds of seconds long, it will not be possible to make prediction on shorter datasets that represent couple of seconds.

That's why it was decided to rescale dataset by seconds' batches. Dedicated function will iterate through all dataset and take the amount of data equal to batch variable to scale each batch separately and consecutively. The size of batch was defined given two goals:

1) Shortest possible interval to predict (mostly when using the model with live dataflow to be able to predict 'x' number of seconds);

2) Get highest model accuracy. Given the amount of training data and specifics of this project it was challenging to pick the right size of batch.

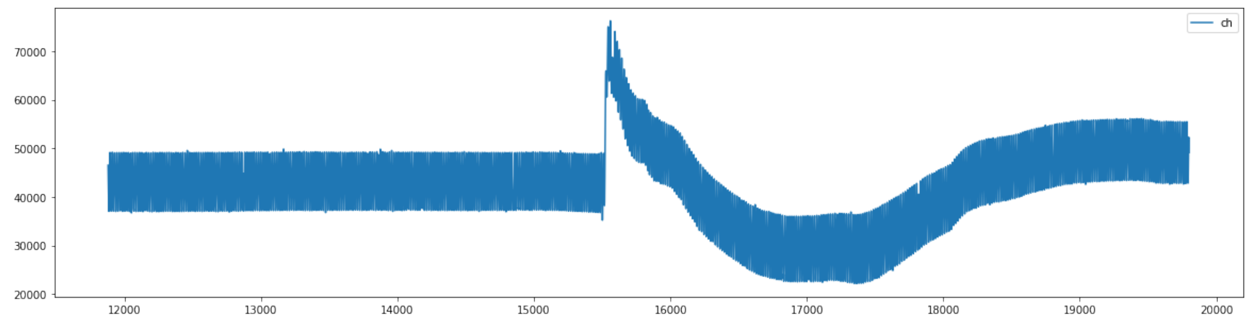
`On below image there is example on how sample data looks before rescaling:`



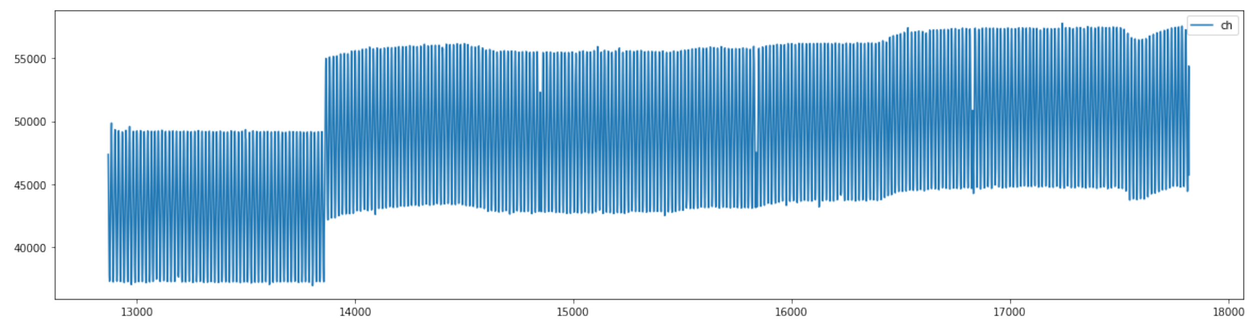
Before even rescaling the dataset, there is another problem to solve. Although data received from EEG was filtered, there are still spikes could be identified from muscle activity or other noise that need to be addressed.

To handle that it was decided to remove spikes on non-rescaled dataset first. The main parameter for spikes removing is '*margin'* variable, which sets up the 'corridor' for wave oscillation. Such 'corridor' is set up by function '*variance_clean()'*. Wave taken by that 'corridor' function will be trimmed to fit. It is important to note that spikes trimming was performed for each second of wave – such approach helped to preserve wave's larger dynamic range observed through whole recorded time long.

`Below image represents the wave with spike:`



`Here is how the region with spike looks after trimming:`

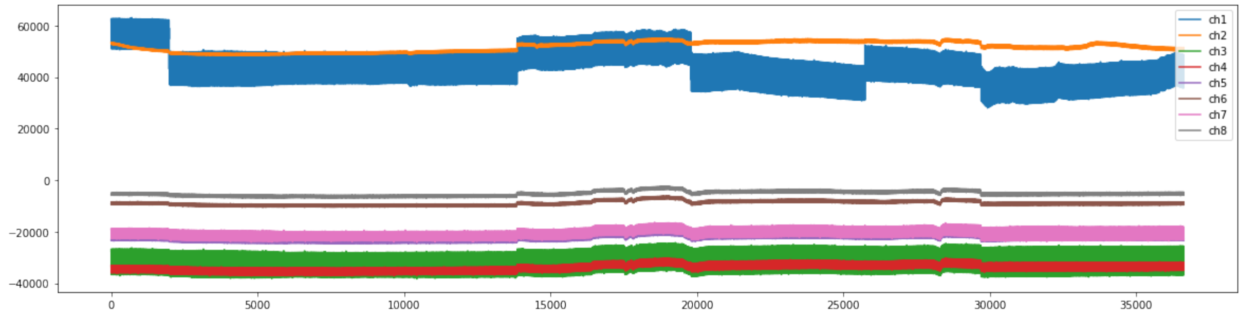


As it can be noticed, there is still shift where spike has been removed, however such slip has significantly less impact on data integrity.

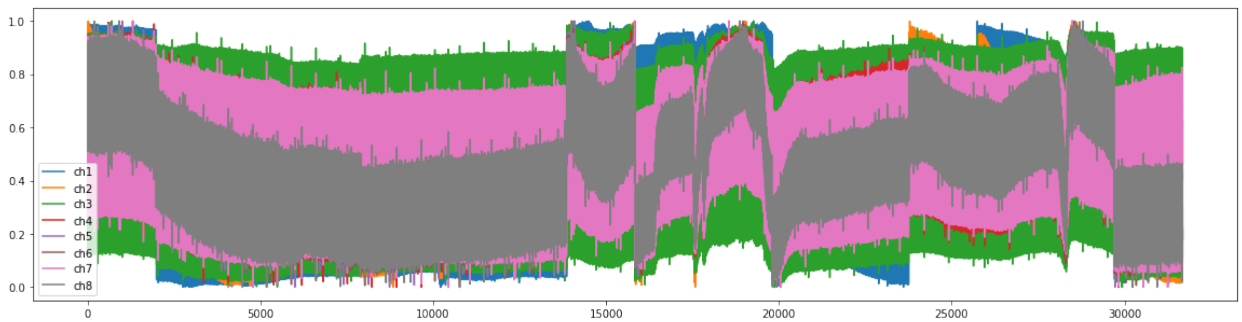
It also noticeable that the length of dataset (number of seconds we want to work with) decreases while we clean up the data. For this reason simple '*seconds()*' function was used to monitor the remaining useful data left for training.

One of the most important functions for data preprocessing is scaling function -'*scaler()*'. With it help the dataset will be rescaled with equal seconds batches. Given the specifics of data it was decided to rescale in between 0 and 1.

On below image there is sample of data as an example after seconds balancing and spikes cleaning:



Below image depicts data sample after rescaling – last phase before training the model:



Shifts may be noticeable on the chart above that looks like dataset is still skewed. However, this is acceptable as far as shifts' edges coincide with ends of seconds.

Implementation

After data preprocessing it should be properly labelled and concatenated to represent one dataset. For that purpose, two labels were created: 0 and 1 - 'reading' and 'writing' correspondingly. Dataset that contained recorded 'reading' state has been measured in length and respective series of labels of the same size has been created. The same approach was taken for 'writing' dataset. After these two datasets has been concatenated: features data representing 'x' data variable and labels data representing 'y' data variable. It is important to note that output variable or labels were one-hot encoded using '*keras.utils.to_categorical()*' function.

Before model training itself, the compile dataset has been split on training and testing samples. For this purpose '*train_test_split()*' function was used from python *sklearn* library. The volume of data allowed to use this approach.

As it was mentioned earlier, 1D Convolutional neural network model has been compiled for the purpose of this project. The model consists of 2 convolutional layers. 2 MaxPooling layers present and BatchNormalization. Such architecture allows to train model fast and with higher accuracy.

```
Model summary is represented below:

_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_1 (Conv1D)            (None, 990, 32)           800
_________________________________________________________________
batch_normalization_1 (Batch (None, 990, 32)           3960
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 198, 32)           0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 196, 64)           6208
_________________________________________________________________
batch_normalization_2 (Batch (None, 196, 64)           784
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 66, 64)            0
_________________________________________________________________
flatten_1 (Flatten)          (None, 4224)              0
_________________________________________________________________
dense_1 (Dense)              (None, 32)                135200
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 66
=================================================================
Total params: 147,018
Trainable params: 144,646
Non-trainable params: 2,372
```

For model optimization Adam optimizer has been selected as best practice. Learning rate was set on 0.001. Such configuration for optimizer worked best.

For the purposes of saving best training results and use them top upload model any time, '*ModelCheckpoint()*' function has been used to store model weights in separate file.

Model was trained on 16 batches through 25 epochs. Such number of epochs proved to be optimal for training. If model was not successfully trained within 25 batches, it was sign to change the model parameters rather that number of batches.

Given the task for this project the higher model accuracy the better it will help in recognizing 'reading' and 'writing' states through complex noisy states. Therefore, as an outcome it was possible to adjust model parameters to train it on relevant dataset and receive 100% accuracy.

Refinement

Through the path of this project there were a lot of stumbling blocks and issues as it possible to imagine and encounter. However, most of them were related to data quality, cleansing and preprocessing. As it was mentioned earlier, the main challenge was persons mental and physical condition that significantly impacts measuring dataset.

First result was achieved relatively easy, while whole dataset has been rescaled towards itself, however as it was addressed earlier, such approach will not allow to predict short intervals of several seconds with given model. Therefore, the decision was made to rescale dataset by seconds batches. After this the most challenging part started – more quality data was required. Many functions were rewritten in order to help making dataset as 'clean' as possible. When the data was cleaned by means described in this project, it was possible to train the model with high accuracy.

IV. Results

Model Evaluation and Validation

The final model accuracy was 100% that argues for model parameters have been chosen reasonably.

The final model has been tested on different data layouts and various pieces of dataset to prove that model works effectively. However, it was noted that if dataset is skewed by mental/physical condition of user, or dataset has different length, model parameters required some tweaks to reach highest accuracy. The following parameters of the model need to be tuned in case is dataset features like length change:

- Pool size of MaxPooling layer;
- Batch size.

Rest of parameters were proven to be reliable. In case if model failed to train that was the result of unclean or unqualify dataset that need to be addressed.

Justification

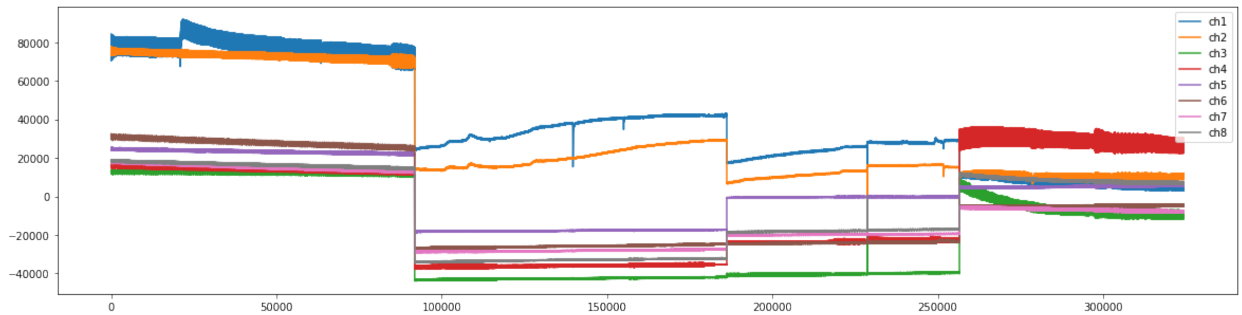
Final model benchmark test has proved that the model corresponds to the problem and resolves it. Accuracy achieved was over 70%. The final solution was thoroughly analyzed and reviewed. The problem for this project may be considered as solved.

V. Conclusion

Free-Form Visualization

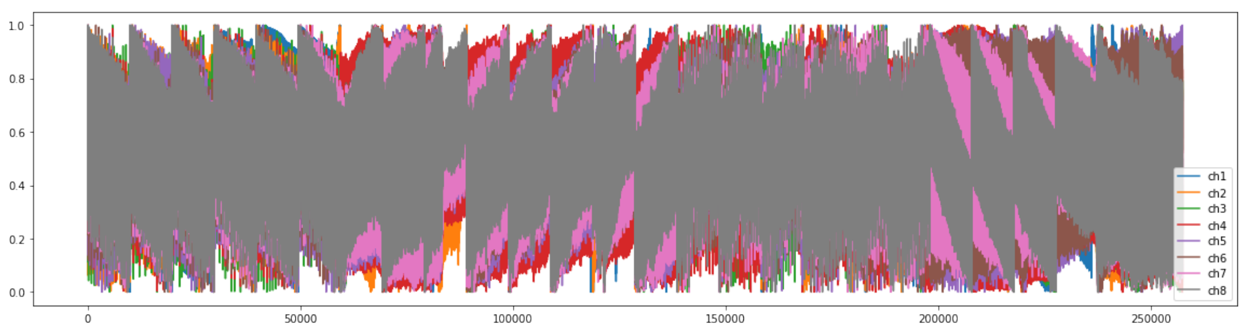
Due to challenges of the project it was important to work on integral dataset. As far as data set were consisting of multiple sets recorded within different time ranges it was difficult to compile whole integral dataset.

However, as a result below illustration shows how whole 'reading' state dataset looks like:

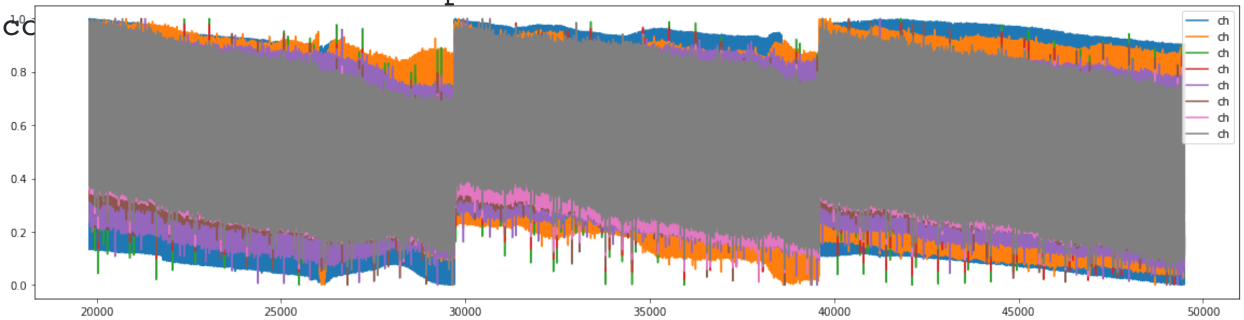


It can be noted where multiple sets are concatenated and how scattered data is. However, this problem has been addressed within preprocessing task. As a result, data (wave) spikes have been removed, data has been rescaled.

Below is depiction on how same preprocessed data looks:



And it is good to see how the data been preprocessed through batches of seconds. Slips are noticeable where seconds batches co



Reflection

The idea of recognizing ‘*reading’* and ‘*writing’* states were picked as simple task to train and test the model for the purpose of this project. However, in course of research it was identified that those two states may not drastically differ from each other. First of all, when person performs routine exercises it is very hard to capture expressive signal patterns, secondly it was noted that person while reading or writing is being mostly in idle state. What might help to contribute in distinguishing those state are eyeballs muscular patterns that may be captured from electrodes located on forehead and mid-scalp. In the rest of it, both writing or reading activities may activate similar areas of brain. For example, when person writes not just piece of text but something that requires thinking ahead, person is planning the story and vision-processing regions in the brain become active[9]. Same process may appear while reading thoughtfully and focused[10]. Author noticed that unfocused reading as well as compulsive writing does not bring any benefit for the purpose of research. To address that challenge the user’s data was collected within different time ranges. There was also control put in place to maintain user focused on specific task. At the end of the day only reliable datasets were picked for training with most confidence of being appropriate for each ‘reading’ and ‘writing’ state. Finally, the model was able to distinguish such activities given the patterns learned. The accuracy tested may not be impressive, however this is good start to research this problematic further.

[9] https://www.sciencealert.com/this-is-what-happens-in-your-brain-when-youre-writing
[10] https://degree.astate.edu/articles/k-12-education/brain-function-affects-reading.aspx

Improvement

Obtained model is not perfect to resolve current problem. However, the way data was collected, noise, psychological condition of user must be considered. Finally, this model proved to set up a basement for further development in the area. More will be achieved with live data collection and preprocessing. The work currently is in progress.