

Neural Network Analysis of Electroencephalograms Graphical Representation *

Aleksandr Bragin ^[0000-0003-4148-4980] and Vladimir Spitsyn ^[0000-0001-5978-1321]

National Research Tomsk Polytechnic University, Tomsk, Russia

Abstract. The article is devoted to the problem of recognition of motor imagery based on electroencephalogram (EEG) signals, which is associated with many difficulties, such as the physical and mental state of a person, measurement accuracy, etc. Artificial neural networks are a good tool in solving this class of problems. Electroencephalograms are time signals, Gramian Angular Fields (GAF), Markov Transition Field (MTF) and Hilbert space-filling curves transformations are used to represent time series as images. The paper shows the possibility of using GAF, MTF and Hilbert space-filling curves EEG signal transforms for recognizing motor patterns, which is further applicable, for example, in building a brain-computer interface.

Keywords: Motor Imagery recognition, Electroencephalogram, Gramian Angular Field, Hilbert Space-Filling Curves, Markov Transition Field, Convolutional Neural Network.

1 Introduction

Electroencephalography is one of the most popular non-invasive methods for studying brain activity today. Electroencephalogram (EEG) signals show the total electrical activity of neurons in the cerebral cortex. You can get a lot of useful information about the human condition by studying these data.

The study of EEG is associated with many difficulties, such as the dependence of signals on age, time of day, the presence of noise, interference, and weak structuring.

Classical mathematical methods based on time-frequency, wave or component analysis can be used for EEG studies. However, often the use of these methods does not give stable results of human states recognition, and their application becomes extremely complicated due to the complexity of the algorithms [1]. Brain signals are very complex. Classical mathematical techniques (Fourier transform, wavelet analysis, etc.) are based on extracting the useful signal from the whole data array and algorithmic work with it. For signals recorded in difficult conditions of psychophysiological experiments,

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

* The reported study was funded by RFBR according to the research project № 18-08-00977 A.

it is often difficult to isolate the useful signal, and the technique may stop working with the slightest change in state.

The use of artificial neural network (ANN) in applied areas, such as the brain-computer interface, is a perspective direction today [2–6]. The ability of the ANN to adaptive learning, resistance to signal distortion and a good generalizing effect makes them an excellent tool for classification [7–9].

There are several approaches to the classification of time series using ANN [10]. A key factor in the success of human activities recognition using EEG is the effective use of data obtained from measurement sensors. In this paper, the method proposed in [11] is used. In this method, the time series is converted into images, after which the convolutional neural network is used to analyze them.

This work is devoted to the creation of systems for the recognition of motor imagery (movement of the right and left hands) based on EEG signals. The processes associated with motor imagery are extremely complex, changes in the time-frequency structure of electroencephalograms are not systematic and vary for each person. The change in the EEG signal can be used to recognize motor patterns.

2 Time Series Recognition

Convolutional Neural Network (CNN) is a special architecture of artificial neural networks aimed at effective pattern recognition. This type of network uses some features of the visual cortex and responds to straight lines at different angles and their combinations. Thus, the idea of convolutional neural networks is the alternation of convolutional layers and pooling layers. The network structure is unidirectional (without feedback), essentially multi-layered. Convolutional neural networks are an effective tool in pattern recognition tasks. It is widely used in video and image processing tasks, natural language processing systems [12]. The use of such networks helps to reduce the number of parameters of the model being taught, to reduce hardware costs in the process of learning and working. However, such a network does not work directly with time series. In [11], a method was proposed for using convolutional neural networks to classify time series. In this method, the time series is converted into images in three ways, after which the usual convolutional neural network is used to image.

2.1 GAF transform

To classify EEG signals using convolutional neural networks, the GAF (Gramian Angular Field) method was used [13, 14]. In this method, the time series is converted into a polar coordinate system. Based on the data obtained, a matrix G is constructed. Each element of the matrix is equal to the cosine of the sum of the angles. The resulting matrix is converted into an image, which is fed to the input of the convolutional neural network.

Preservation of time dependence is provided with this conversion. The main diagonal is a special case with $k = 0$, containing the original values and angular information.

The GAF matrix is a matrix built on the series as follows:

1. First, the series is normalized to the segment $[-1, 1]$, (1):

$$\hat{x}_i = \frac{(x_i - \max(X)) + (x_i - \min(X))}{\max(X) - \min(X)} \quad (1)$$

2. Next, the resulting values are translated to the polar coordinate system:

$$\begin{cases} \varphi_i = \arccos(x_i) \\ r_i = \frac{t_i}{N} \end{cases} \quad (2)$$

3. The GAF matrix is calculated by the following expression:

$$G = \begin{bmatrix} \cos(\varphi_1 + \varphi_1) & \cdots & \cos(\varphi_1 + \varphi_n) \\ \cos(\varphi_2 + \varphi_1) & \cdots & \cos(\varphi_2 + \varphi_n) \\ \vdots & \ddots & \vdots \\ \cos(\varphi_n + \varphi_1) & \cdots & \cos(\varphi_n + \varphi_n) \end{bmatrix} \quad (3)$$

The final matrix saves all the information about the series, except for the initial bounds of the values that we lose in step (1) after the normalization procedure - that is, we can restore the original series using the obtained matrix but only scaled to the segment $[-1, 1]$.

Images for further use are formed on the basis of the obtained matrices. Since the color channel in this case does not carry useful information, binary images are used in the work, which made it possible to reduce the number of image channels by 3 times in comparison with the RGB version. Binary GAF image is shown in Fig. 1.

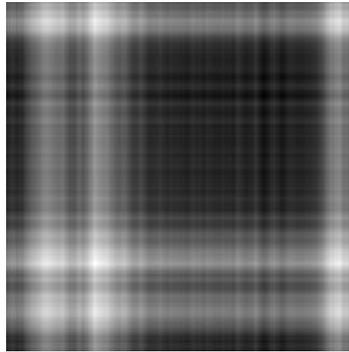


Fig. 1. GAF image example

2.2 MTF transform

Another method of converting the original time series into images is the MTF (Markov Transition Field) method. Unlike GAF, the original bounds of the series and the distribution of the values of the series that are lost when applying convolutions and pooling to GAF are stored in the MTF matrix. The idea of this method is to consistently represent the probabilities of Markov transitions to store information in the time dimension. The first step is the quantization of time series to build a first-order Markov matrix. The Markov matrix, which includes Markov dynamics, rejects the conditional connection between the distribution of X and the time dependence on the time steps t_i . MTF extends the Markov matrix to equalize each probability in a temporal order.

The data conversion order:

1. The whole set of observation values is divided into m quantile bins — segments with the same probability that the observation value falls into each of them. This can be done simply with the help of a training sample – to combine all the values of a variable into one set, sort it, and then arrange $m - 1$ boundaries in it so that there is approximately the same number of values between two adjacent boundaries. The space between two adjacent borders will be a quantile bin.
2. Let ω_{ij} be the number of pairs of neighboring observations in the time series, which we are considering - such that the left observation lies in the bin i and the right ob-

servation in the bin j . $\hat{\omega}_{ij}$ is a normalized ω_{ij} such that $\sum_i \hat{\omega}_{ij} = 1, \forall j$. Then the MTF matrix for a time series is a matrix of transition probabilities between bins:

$$G = \begin{bmatrix} \hat{\omega}_{11} & \cdots & \hat{\omega}_{1m} \\ \hat{\omega}_{21} & \cdots & \hat{\omega}_{2m} \\ \vdots & \ddots & \vdots \\ \hat{\omega}_{m1} & \cdots & \hat{\omega}_{mm} \end{bmatrix}$$

Example of MTF image generated from EEG data is shown in Fig. 2.

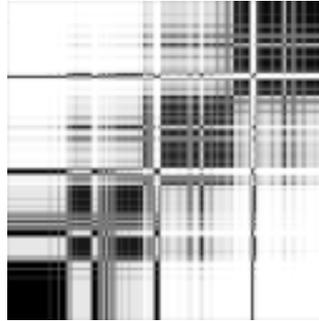


Fig. 2. MTF image example

2.3 Hilbert space-filling curves

A Hilbert curve is a continuous fractal space-filling curve. If such a curve is spread along a straight line, points located close to each other in a two-dimensional representation will also tend to be close to each other in a linear sequence. Good clustering properties are one of the main advantages of Hilbert curves for use in computer science. [15].

Hilbert curves have found application in many fields of science, technology and medicine. They are used to index and improve database performance [16, 17], video and image processing [18], visualization of genetic data [19, 20].

The first-order Hilbert curve (H1) has the shape of an inverted letter “U”, located on three sides of the square (fig.3). Higher-order curves are obtained by replacing the upper vertices with the curves of the previous order, the lower left vertex — a curve rotated 90 degrees clockwise, and the lower right vertex — a curve rotated 90 degrees counterclockwise. The second-order Hilbert curve (H2) is shown in Fig. 3 and consists of first-order curves. Curves H1 are connected by three-line segments called bundles. The third-order curve H3 (Fig. 3) consists of four H2 curves and three bundles. A curve of each order begins in the lower left corner and ends in the lower right corner. The zero-order Hilbert curve is a point. Thus, the n -order Hilbert curve H_n covers 4^n points and is located on a surface of size $2^n \times 2^n$.

There are several effective algorithms for obtaining the coordinates of a point on a curve, such as the Butz algorithm [21], algorithm 781 [22].

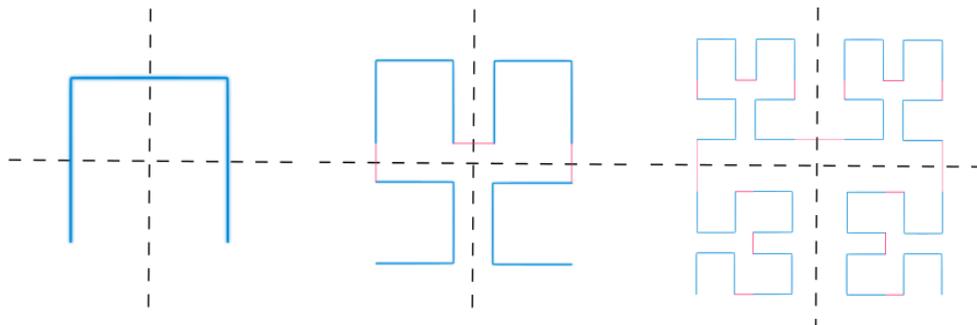


Fig. 3. Hilbert curves of the first, second and third orders.

The time series $f(t)$, consisting of p points, can be represented on a two-dimensional plane in the form of a Hilbert curve by linking the index i of each point in the time series with the corresponding coordinate (x, y) on the curve. In this case, the point value in the coordinate (x, y) will have the corresponding value from the time series.

An example of EEG data obtained from one measuring channel and their presentation in the form of a Hilbert curve are shown in Fig. 4. The maximum value of the amplitude of the electroencephalogram in the time dependence can be easily compared with the corresponding region on the Hilbert curve.

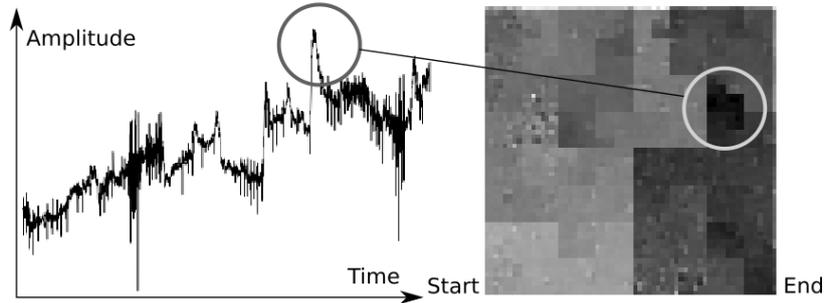


Fig. 4. Representation of the time series in the form of a Hilbert curve

3 Deep convolutional neural network architecture

The architecture of the deep convolutional neural network (CNN) adapted for the classification of the EEG signal according to the number of measuring electrodes was developed in this paper. Input images are fed to the network input in the form of a 64 channel image, where each channel is a transformed electroencephalogram signal. The main layers of the deep CNN are three convolutional and three fully connected layers. Network parameters were determined experimentally.

4 Dataset

The data presented in [23] were selected for the study. Each subject was in a chair with armrests and watched the image on the monitor. At the beginning of each test, a black screen with a fixing cross was displayed on the monitor for two seconds, and then the subject had to imagine a hand movement depending on the instructions on the monitor for next three seconds. Then followed a short break for a few seconds, after which the action was repeated.

The data set is EEG signals recorded using the BCI 2000 system [24] and using 64 electrodes at a sampling frequency of 512 Hz. Frequency filters for data conversion were not used.

The order of the experiment and the transformation of the original data into GAF images are shown in Fig.5. Data from the first subject were used for the training of neural networks.

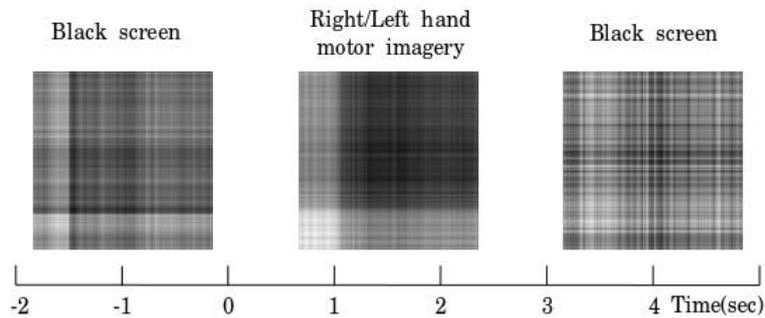


Fig. 5. The order of the experiment

5 Model structure

The signals received from 64 electrodes in the electroencephalograms under study are presented, so we have the corresponding number of images as input parameters of the model. The images are converted by analogy with the RGB image into the input vector, however, if in the case of the RGB image 3 channels are used, one for each color, in this case a 64 channel image containing the converted EEG signal is input. The block diagram of the final model is presented in Fig. 6.

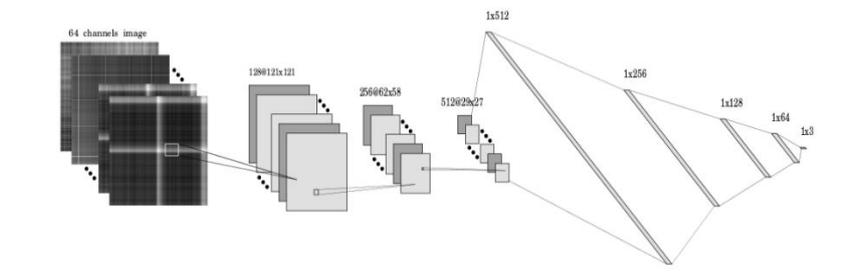


Fig. 6. Model structure

6 Recognition of motor imagery

For the recognition of motor imagery were used data of the first test subject [23]. At the input of the neural network were fed the image size of 128 by 128 pixels. The recognition accuracy for the GAF method was about 97%, for the MTF method – 93%, for the Hilbert space-filling curves method - 98%, recognition accuracy from the data source [23] is equal to 80%. The full EEG signal (in the case of the GAF transformation, normalized in the [-1;1] range) was used to generate images, without removing any artifacts, as well as highlighting characteristic rhythms and patterns. This approach

avoids the need to filter the EEG signal for further investigation, which may result in the loss of useful information.

7 Conclusion

The article showed the possibility of using the GAF, MTF and Hilbert space-filling curve transformations methods for the detection of motor imagery in EEG signals. It is shown that without the use of additional filtering of the initial data (selection of brain rhythms) it is possible to achieve high results. The accuracy of recognition of motor imagery of the movement of the right and left hand, as well as the state of rest for the studied EEG signals was 97% (GAF), 93% (MTF), 98% (Hilbert space-filling curves). General results indicate that these methods provide higher accuracy than the methods described in the data source. The Hilbert space-filling curves method showed better results in the recognition of motor imagery and allows you to restore the original signal, after processing, for further study. The discussed methods of electroencephalogram classification can later be used to build a brain-computer interface.

8 Acknowledgment

The reported study was funded by RFBR according to the research project № 18-08-00977 A and in the framework of Tomsk Polytechnic University Competitiveness Enhancement Program.

References

1. V.V. Grubov, A.E. Runnova, M.K. Kurovskaya, A.N. Pavlov, A.A. Koronovskii, A.E. Hramov, "Demonstration of brain noise on human EEG signals in perception of bistable images", Proc. SPIE. 2016. V. 9707. DOI: 10.1117/12.2207390
2. P. Sotnikov, K. Finagin, S. Vidunova, "Selection of optimal frequency bands of the electroencephalogram signal in eye-brain-computer interface", Procedia Computer Science. 2017, vol. 103, pp. 168-175.
3. A.N. Vasilyev, S.P. Liburkina, A.Y. Kaplan, "Lateralization of EEG patterns in humans during motor imagery of arm movements in the brain-computer interface", Zhurnal Vysshei Nervnoi Deyatelnosti Imeni I.P. Pavlova, 2016, vol. 66, № 3, pp. 302-312.
4. V.A. Maksimenko, S. Heukelum, V.V. Makarov, J. Kelderhuis, A. Lüttjohann, A.A. Koronovskii, A.E. Hramov, G. Luijtelaar, "Absence seizure control by a brain computer interface", Scientific Reports. 2017, vol. 7, p. 2487.
5. W. Hsu, I. Chiang, "Application of neural network to brain-computer interface", 2012 IEEE International Conference on Granular Computing, Hangzhou, 2012, pp. 163-168. doi: 10.1109/GrC.2012.6468559
6. K. Nakayama, K. Inagaki, "A brain computer interface based on neural network with efficient pre-processing", 2006 International Symposium on Intelligent Signal Processing and Communications, Tottori, 2006, pp. 673-676. doi: 10.1109/ISPACS.2006.364745
7. R. Östberg, "Robustness of a neural network used for image classification: The effect of applying distortions on adversarial examples", Dissertation, 2018.

8. Q. Wang, W. Guo, K. Zhang, A. G. Ororbia, II, X. Xing, X. Liu, C. Lee Giles, "Adversary resistant deep neural networks with an application to malware detection", Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 13-17, 2017, Halifax, NS, Canada
9. J. Yim, K. Sohn, "Enhancing the performance of convolutional neural networks on quality degraded datasets", 2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA), Sydney, NSW, 2017, pp. 1-8. doi: 10.1109/DICTA.2017.8227427
10. N. Hatami, Y. Gavet, J. Debayle, "Classification of time-series images using deep convolutional neural networks" 2017 The 10th International Conference on Machine Vision (ICMV 2017), ICMV Committees, Nov 2017, Vienne, Austria. 10.1117/12.2309486. hal-01743695
11. Z. Wang, T. Oates, "Spatially Encoding Temporal Correlations to Classify Temporal Data Using Convolutional Neural Networks", B: CoRR abs/1509.07481 (2015). url: <http://arxiv.org/abs/1509.07481>.
12. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Adv. Neural Inf. Process. Syst. (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), vol. 25, pp. 1097–1105, Curran Associates, Inc., 2012.
13. Z. Wang, T. Oates, "Imaging time-series to improve classification and imputation", In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015, pp. 3939–3945.
14. Z. Wang, T. Oates, "Encoding time series as images for visual inspection and classification using tiled convolutional neural networks", Association for the Advancement of Artificial Intelligence (AAAI) conference, 2015.
15. Bongki Moon, H.V. Jagadish, Christos Faloutsos, and Joel Salz, Analysis of the Clustering Properties of Hilbert Space-filling Curve, IEEE Trans. on Knowledge and Data Engineering (IEEE-TKDE), vol. 13, no. 1, pp. 124-141, Jan./Feb. 2001
16. Daniel A. Keim. Pixel-oriented Visualization Techniques for Exploring Very Large Databases
17. Knut Stolze. "DB2 Spatial Extender performance tuning", <https://www.ibm.com/developerworks/data/library/techarticle/dm-0510stolze/index.html>
18. Giap Nguyen, Patrick Franco, Remy Mullot, Jean-Marc Ogier, "Mapping high dimensional features onto Hilbert curve: applying to fast image retrieval". 21st International Conference on Pattern Recognition (ICPR 2012) November 11-15, 2012. Tsukuba, Japan
19. Md. Monowar Anjum, Ibrahim Asadullah Tahmid and Dr. M. Sohel Rahman. "CNN Model With Hilbert Curve Representation of DNA Sequence For Enhancer Prediction"
20. Pak Chung Wong, Kwong Kwok Wong, Harlan Foote, and Jim Thomas. "Global Visualization and Alignments of Whole Bacterial Genomes"
21. A.R. Butz. "Alternative algorithm for Hilbert's space filling curve". // IEEE Trans. On Computers. — 1971. — T. 20. — DOI:10.1109/T-C.1971.223258
22. G. Breinholt and C. Schierz. "Algorithm 781: Generating hilbert's space-filling curve by recursion". ACM Trans. on Math. Soft., 24:184–189, 1998.
23. Hohyun Cho, Minkyu Ahn, Sangtae Ahn, Moonyoung Kwon, Sung Chan Jun, "EEG datasets for motor imagery brain-computer interface", GigaScience, Volume 6, Issue 7, July 2017, gix034, <https://doi.org/10.1093/gigascience/gix034>
24. B. Blankertz, Kr. Müller, G. Curio, Tm. Vaughan, G. Schalk, Jr. Wolpaw, A. Schlögl, C. Neuper, G. Pfurtscheller, T. Hinterberger, M. Schröder, N. Birbaumer, "The BCI competition 2003: Progress and perspectives in detection and discrimination of EEG single trials", IEEE Transactions on Biomedical Engineering, 2004, №6 (51), pp. 1044–1051.