Single-trial P300 classification using deep belief networks for a BCI system

Sergio A. Cortez
Dept. of Electrical Engineering
Universidad de Ingeniería y Tecnología
Lima, Peru
sergio.cortez@utec.edu.pe

Christian Flores
Dept. of Electrical Engineering
Universidad de Ingeniería y Tecnología
Lima, Peru
cflores@utec.edu.pe

Javier Andreu-Perez
Dept. of Electronic Engineering
University of Essex
Essex, United Kingdom
javier.andreu@essex.ac.uk

***Abstract*—A brain-computer interface (BCI) aims to provide its users with the capability to interact with machines only through its brain activity. There is a special interest in developing BCIs targeted at people with mild or severe motor disabilities since this kind of technology would improve their lifestyles. The Speller is a BCI application that uses the P300 waveform to essentially allow its user to communicate without using its peripheral nerves. This paper focuses on the classification of the P300 waveform from single-trials obtained through EEG using deep belief networks (DBNs). This deep learning algorithm can identify relevant features automatically from the subject's data, making its training requiring less pre-processing stages. The network was tested using signals recorded from healthy subjects and post-stroke victims. The highest accuracy achieved was of 91.6% for a healthy subject and 88.1% for a post-stroke victim.**

***Index Terms*—brain-computer interface, stroke victims, EEG, deep belief networks**

I. Introduction

A brain-computer interface (BCI) is a technology that grants people control over computers by using only their brain signals [1]. Different brain imaging techniques, like electroencephalography (EEG) and magnetic resonance imaging (MRI), are used to register and analyze people's brain activity. Certain mental processes are of special interest in BCI design because they can be performed without problems by motor-impaired subjects, which allows engineers and medics to work on solutions towards them to improve their quality of life [2].

Speller is a well-known BCI application based on the P300 waveform, which is an event-related potential (ERP) triggered by visual stimuli [3] and recorded using EEG. This BCI acts as a typing machine using the induced P300 response related to images of letters to which the user pays attention. The P300 ERP is elicited using an oddball paradigm which basically consists on presenting the subject target stimuli blended among irrelevant stimuli while recording its physiological response. 300ms after the target stimulus is presented, a positive deflection (or potential) in the subject's EEG signals can be observed, thus taking the name P300. The simplicity of this paradigm makes this BCI require less data samples for training in comparison with other paradigms, making it a useful for developing quick solutions targeted at patients [4].

Statistical and Machine learning algorithms have been successfully used for developing classifiers that discriminate the user's responses automatically [5]. The performance of the classifier is of special importance since it will determine the correct functioning of the BCI ultimately. The work of Hoffmann [4] used Bayesian Linear Discriminant Analysis (BLDA) and Fisher's Linear Discriminant Analysis (FLDA) to classify the P300 responses of five disable subjects and four healthy subjects in a Speller-like BCI with 6 images. Using multiple blocks for classification, they were able to achieve on average a classification accuracy of 100% for the disable subjects. In [6], the authors presented a new way to detect the P300 waveform from raw EEG data by employing convolutional neural networks (CNN). They obtained a block precision of 95.5% with their best model.

Regarding deep belief networks for EEG classification, in [7] the authors proposed a method to improve the DBN's training algorithm. Testing their models with EEG P300 trials, they were able to achieve a target by block classification accuracy of 93.47% for their best subject. The work of [8] used a DBN to classify raw EEG data for a P300 based BCI. They reported their best model precision was able to reach 86.4%. There has been other works such as [9], in which they tried to classify single P300 trials using DBNs, reporting for its best subject up to 87% in precision.

We propose a method to classify P300 single trials using deep belief networks (DBNs). We tested this network with EEG data of healthy subjects and stroke victims. The model can be used in the design of any P300 based BCI. This work is presented as follows: in section II are presented the materials, including participants, experimental setup and EEG acquisition and the methods, which describes preprocessing, feature vectors and classifiers. The results and discussion are presented in section III. Finally, we present the conclusions and future work in section IV.

II. Materials and Methods

A. Participants and EEG Acquisition

Nine volunteers agreed to participate in this study. Table I shows the age, gender and medical diagnosis of each volunteer. The healthy subjects acted as the control group for the three post-stroke patients. The Ethics Committee from the Universidad Peruana Cayetano Heredia issued the ethical approval for the experiment and informed written consent. The participants

978-1-7281-9377-9/20/$31.00 ©2020 IEEE

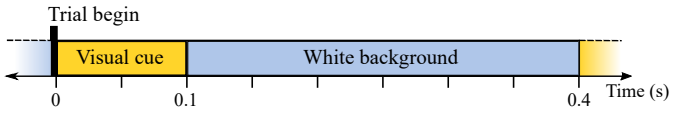

Fig. 1. Protocol's time diagram. One of the six images (visual cues) is randomly selected and displayed on the screen for 100ms followed by a white background for the next 300ms. After displaying all six images, this process repeats itself between 20 and 25 times.

were informed about the objectives of this study and ensured the preservation of their anonymity. Subjects S08 and S09 presented mild aphasia, but only subject S09 showed signs of upper limbs paresis. Subject S07 exhibited severe apraxia and aphasia.

The EEG signals were acquired using sixteen bipolar electrodes and the g.USBamp amplifier (g.tec medical engineering GmbH, Austria). The electrodes were placed following the 10-20 system on the positions: Fz, FC1, FC2, C3, Cz, C4, CP1, CP2, P7, P3, Pz, P4, P8, O1, O2, and O3. The ground electrode was placed at the subject's right mastoid and, the reference electrode, on its left earlobe.

TABLE I
PARTICIPANTS INFORMATION

Subject	Age	Gender	Diagnosis
S01	33	Male	Healthy
S02	21	Male	Healthy
S03	20	Male	Healthy
S04	21	Male	Healthy
S05	24	Male	Healthy
S06	29	Male	Healthy
S07	20	Male	Hemorrhagic post-stroke
S08	52	Female	Ischemic post-stroke
S09	55	Male	Ischemic post-stroke

B. Experimental Setup

The protocol used here was based on Hoffmann's work [4]. To summarize, six different images were randomly flashed on a screen with white background. Each one of these tries to represent an action the subject would like to carry out. The Fig. 1 shows the timing scheme of the experiment. It is called a block to the time interval in which the six images are flashed only once. Between 20 and 25 blocks make a run, and each session had 6 runs. Four sessions, recorded in two days, were obtained from the nine participants. Through all the experiment, the participants were asked to count how many times the image they were told to pay attention to appeared on the screen.

C. Signal Preprocessing

The data was downsampled from 2400Hz to 120Hz and then filtered using a sixth order Butterworth bandpass filter with cut-off frequencies in 1 and 15 Hz. The data points of each electrode recorded in one second after an image was flashed were extracted and stored in a 16×120 matrix, defining a trial. Any artifacts or outliers were removed by winsorization [4]. Finally, the signals on each trial were standardized. The feature vectors were constructed rearranging the data from all trials. Specifically, the data points of each channel were concatenated in the following way:

$$\boldsymbol{I} = \begin{bmatrix} S_1^1 & S_1^2 & S_1^3 & \dots & S_1^{16} & S_2^1 & S_2^2 & \dots & S_{120}^{16} \end{bmatrix},$$

wherein for a single point S_k^i, i indicates the channel it belongs and k its position with respect of time. Then, the vectors were re-normalize from 0 to 1 separately. Each trial and thus, each feature vector, has a label which indicates whether or not if its related visual stimulus triggered a P300 waveform. If that is the case, the trial is called *target*, and when not, *non-target*. However, since consecutive trials overlap, the adjacent *non-target* trials to the *target* ones will also present the P300 waveform at some degree. These trials were removed before applying the balancing process.

Due to the paradigm employed, the subject's data will have uneven amounts of *target* and *non-target* trials. Any unevenness may bias the classifier, producing non-reliable performance metrics. To avoid this, a balancing process was applied. It consisted on randomly selecting the same number of *target* trials from a *non-target* trial pool. The resulting vectors were stacked along with the *non-target* ones in random order, shaping the training matrix.

D. Classification

A deep belief network (DBN) was used to classify single trials automatically. Neural networks can be seen as function approximators [10], in which their inside parameters are adjusted trying to match actual outputs with desired outputs by comparing the error between both. A DBN is a deep neural network made of two or more restricted Boltzmann machines (RBMs) stacked on top of each other [11]. A RBM is a simple neural net with an input (or visible) layer fully connected to a single hidden layer. The DBN training is divided in two stages: the first one is the unsupervised training (or pretraining) of each RBM and the second stage is the supervised training of the whole network. The pretraining stage is motivated by the problems related to regular training by backpropagation. The minimization of the performance function of a deep network by the gradient descent method becomes troublesome due to its highly non-convex character. The unsupervised greedy layer-wise training initialize the network's weights to reduce the possibilities of the gradient getting stuck in local maxima (or minima) when the supervised training takes place. In the unsupervised training stage, each RBM learns to detect the most relevant characteristics of their respective inputs using the contrastive divergence (CD) algorithm [12]. A RBM's hidden layer will act as the input layer for the next RBM once its pretraining is complete. Once all the RBMs have been trained, the whole network can be fine-tuned using error backpropagation in a supervised manner. The simplified pretraining stage of each RBM is briefly presented below.

Let us assume a binary RBM for simplicity, which then can be used to generalize the model for real value inputs [13]. Let $\boldsymbol{v}$ and $\boldsymbol{h}$ be the state vectors of the visible and hidden

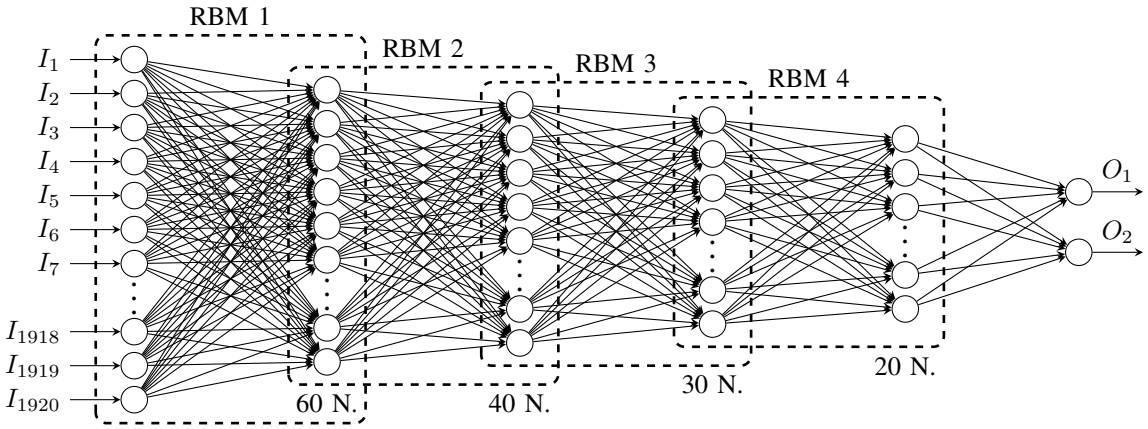

Fig. 2. Architecture of the deep belief network proposed. The first RBM is type Gaussian - Bernoulli, and the rest, Bernoulli- Bernoulli ones.

neurons of the RBM. Also, let $\boldsymbol{w}$ be the weight matrix which describes the interaction between the ith visible neuron and the jth hidden neuron. The energy of a joint configuration $(\boldsymbol{v}, \boldsymbol{h})$, is defined as:

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\sum_{i=1}^{V}\sum_{j=1}^{H} v_i h_j w_{ij} - \sum_{i=1}^{V} b_i v_i - \sum_{j=1}^{H} c_j h_j,$$

where V and H are the total number of visible and hidden neurons whilst b_i and c_j are their bias terms respectively.

The probability distribution for every possible joint configuration is defined using the energy function

$$p(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{Z} e^{-E(\boldsymbol{v}, \boldsymbol{h})},$$

where $Z = \sum_{\boldsymbol{v}}\sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v}, \boldsymbol{h})}$ is the partition function. The probability the neural net assigns to an input $\boldsymbol{v}$ is computed by summing all hidden vectors, resulting in

$$p(\boldsymbol{v}) = \frac{1}{Z}\sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v}, \boldsymbol{h})}.$$

A RBM can be assigned to a specific input by modifying its parameters. The derivative of the log probability of an input vector with respect to the weight matrix can be used to define a learning rule

$$\frac{\partial \log p(\boldsymbol{v})}{\partial w_{ij}} \propto \Delta w_{ij} = \eta(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{reconst})$$

where η is the learning rate and the terms in the angle brackets are the expectations under the distributions of the training set and its reconstruction respectively. The $\langle v_i h_j \rangle_{data}$ term can be calculated, for a $\boldsymbol{v}$ input, using the conditional probability

$$p(h_j = 1|\boldsymbol{v}) = f(c_j + \sum_{i=1}^{V} v_i w_{ij}) \tag{1}$$

where $f(x) = \frac{1}{1+e^{-x}}$ (logistic sigmoid). The second term $\langle v_i h_j \rangle_{reconst}$ can be calculated by applying Eq. 1 to the reconstructions after these are computed using

$$p(v_i = 1|\boldsymbol{h}) = f(b_i + \sum_{j=1}^{H} h_j w_{ij}) \tag{2}$$

with the hidden states of $\langle v_i h_j \rangle_{data}$. To summarize, this training consists on reconstructing the input by using only the hidden layer outputs. The goal is to make the real inputs and the reconstructions as similar as possible adjusting the net's parameters. Real input values are processed using a Gaussian–Bernoulli RBM, in which the conditional distribution described in Eq. 2 is modeled with a Gaussian [14].

Four RBMs stacked and an output classification layer make up the network proposed here. The Fig. 2 shows the network's architecture and the RBMs. The logistic sigmoid function was used to model each neuron's activation function, except for the ones in the output layer which used the softmax function. The unsupervised training of each RBM was fixed for 100 iterations using the CD algorithm. The learning step was set to 0.01 with an initial momentum of 0.5 and a final moment of 0.9, following the work of [15]. The algorithm used for the supervised training process was the scaled conjugate gradient (SCG) backpropagation. All subject's networks converged using only 600 epochs. The whole training stage took in average fifteen minutes per subject using a Nvidia GTX 1050 GPU and an Intel core i7 CPU on MATLAB R2019a.

III. Results and Discussion

Table II shows the classification accuracy of the DBN obtained for each subject using a 5-fold cross-validation method. It also makes a comparison with another three classifiers used in our previous works. [17] [16]. The ANFIS classifier was not tested with subject S05 because, for that moment, he was still in the process of having its EEG signals recorded. For each classifier, subjects S01, S03 and S09 obtained the best results whilst subject S07 performance was the lowest. It is very likely that subject low performance was due to its critical condition (hemorragic poststroke). Even though the P300 is an endogenous response that can be elicited in post-stroke victims, the subject's concentration is an important factor that

TABLE II
CROSS-VALIDATED MODEL ACCURACY

Subject	Classification models			
	DBN	MLP [16]	SVM [16]	ANFIS [17]
S01	91.6	91.8	91.5	85.3
S02	80.7	80.3	79.1	77.4
S03	85.8	85.3	83.9	79.3
S04	81.7	75.7	78.9	79.5
S05	83.5	84.9	83.4	-
S06	82.5	83.0	81.7	72.4
S07	66.6	68.6	69.2	70.1
S08	88.1	89.6	85.5	74.9
S09	85.8	86.9	87.4	78.4

will determine overall if the P300 potential is generated or not. Subject S07 performance suggest its concentration decreased over time mainly due to its medical condition and fatigue. A possible solution for this kind of subjects would be to increase the amount of sessions and reduce the number of runs recorded in a day, making the recording periods shorter.

For all the subjects, the results from the DBN, the MLP and SVM were similar, except for subject S04 in which the DBN clearly outperform the rest. On average, subjects S08 and S09 obtained even a better performance than most of the healthy subjects which clearly indicates a P300 based BCI for post-stroke victims using this classifier will work correctly. An important factor that may have influenced the classifier's performance was the balancing process. Although all *non-target* trials were recorded under the same circumstances and should share the same information, differences may arise between them due to the artifacts generated by subject's fatigue, external events, among others. For that reason, it is possible another *non-target* trial combination may yield better or worst results.

This classifiers can also be used to classify trials by block, as in BCI systems proposed by [4], [6]. Their single trial classification accuracy would allow the system to reduce dramatically the amount of blocks it needs to be certain about the user's command, resulting in a 100% classification accuracy in less time.

IV. CONCLUSIONS

A P300 single trial classifier based on a DBN was presented and tested on six healthy subjects and three post-stroke patients. The results surpassed our previous work in most of the healthy subjects but, performance along the patient cohort was maintained. This classifier can be employed in the designing stage of a P300 based BCI. Patients with severe medical conditions may require another type of classifiers or more training sessions to increase their performance.

The greedy layer-wise training the DBN goes through makes it require less processed inputs since it can detect by itself relevant characteristics for optimal discrimination. The main drawback is the computational power it requires for training. The balancing process to which the subject's data is put through may be preventing the classifier model to achieve its optimum classification accuracy due to the selection of non-target trials been without any specific criteria.

As future work, we intend to include amyotrophic lateral sclerosis (ALS) patients and also to improve the balancing process discriminating the non-target trials to select the most appropriate ones.

ACKNOWLEDGMENT

This work was supported by the Programa Nacional de Innovacion para la Competitividad y Productividad of Peru, under the grant PIAP-3-P-483-14 and a seed fund by UTEC.

REFERENCES

[1] H.-J. Hwang, S. Kim, S. Choi, and C.-H. Im, "Eeg-based brain-computer interfaces: a thorough literature survey," *International Journal of Human-Computer Interaction*, vol. 29, no. 12, pp. 814–826, 2013.

[2] E. W. Sellers and E. Donchin, "A p300-based brain–computer interface: initial tests by als patients," *Clinical neurophysiology*, vol. 117, no. 3, pp. 538–548, 2006.

[3] L. A. Farwell and E. Donchin, "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalography and clinical Neurophysiology*, vol. 70, no. 6, pp. 510–523, 1988.

[4] U. Hoffmann, J.-M. Vesin, T. Ebrahimi, and K. Diserens, "An efficient p300-based brain–computer interface for disabled subjects," *Journal of Neuroscience methods*, vol. 167, no. 1, pp. 115–125, 2008.

[5] F. Lotte, L. Bougrain, A. Cichocki, M. Clerc, M. Congedo, A. Rakotomamonjy, and F. Yger, "A review of classification algorithms for eeg-based brain–computer interfaces: a 10 year update," *Journal of neural engineering*, vol. 15, no. 3, p. 031005, 2018.

[6] H. Cecotti and A. Graser, "Convolutional neural networks for p300 detection with application to brain-computer interfaces," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 3, pp. 433–445, 2010.

[7] S. Morabbi, M. Keyvanpour, and S. V. Shojaedini, "A new method for p300 detection in deep belief networks: Nesterov momentum and drop based learning rate," *Health and Technology*, vol. 9, no. 4, pp. 615–630, 2019.

[8] Z. Lu, N. Gao, Y. Liu, and Q. Li, "The detection of p300 potential based on deep belief network," in *2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pp. 1–5, 2018.

[9] A. Sobhani, "P300 classification using deep belief nets," in *European Symposium on Artificial Neural Networks (ESANN)*, 2014.

[10] P. H. Winston, *Artificial Intelligence (3rd Ed.)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1992.

[11] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[12] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[13] G. E. Hinton, "A practical guide to training restricted boltzmann machines," in *Neural networks: Tricks of the trade*, pp. 599–619, Springer, 2012.

[14] A.-r. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE transactions on audio, speech, and language processing*, vol. 20, no. 1, pp. 14–22, 2011.

[15] M. Tanaka and M. Okutomi, "A novel inference of a restricted boltzmann machine," in *2014 22nd International Conference on Pattern Recognition*, pp. 1526–1531, IEEE, 2014.

[16] S. A. Cortez, C. Flores, and J. Andreu-Perez, "A smart home control prototype using a p300-based brain-computer interface for post-stroke patients," in *Smart Innovation, Systems and Technologies*, p. to appear, Springer Nature, 2020.

[17] D. Achanccaray, C. Flores, C. Fonseca, and J. Andreu-Perez, "A p300-based brain computer interface for smart home interaction through an anfis ensemble," in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1–5, IEEE, 2017.