Original Article

A New Method for Detecting P300 Signals by Using Deep Learning: Hyperparameter Tuning in High-Dimensional Space by Minimizing Nonconvex Error Function

Seyed Vahab Shojaedini[1], Sajedeh Morabbi[1], MohammadReza Keyvanpour[2]

[1]Department of Biomedical Engineering, Iranian Research Organization for Science and Technology, [2]Department of Computer Engineering, Alzahra University, Tehran, Iran

Abstract

Background: P300 signal detection is an essential problem in many fields of Brain-Computer Interface (BCI) systems. Although deep neural networks have almost ubiquitously used in P300 detection, in such networks, increasing the number of dimensions leads to growth ratio of saddle points to local minimums. This phenomenon results in slow convergence in deep neural network. Hyperparameter tuning is one of the approaches in deep learning, which leads to fast convergence because of its ability to find better local minimums. In this paper, a new adaptive hyperparameter tuning method is proposed to improve training of Convolutional Neural Networks (CNNs). **Methods:** The aim of this paper is to introduce a novel method to improve the performance of deep neural networks in P300 signal detection. To reach this purpose, the proposed method transferred the non-convex error function of CNN) into Lagranging paradigm, then, Newton and dual active set techniques are utilized for hyperparameter tuning in order to minimize error of objective function in high dimensional space of CNN. **Results:** The proposed method was implemented on MATLAB 2017 package and its performance was evaluated on dataset of Ecole Polytechnique Fédérale de Lausanne (EPFL) BCI group. The obtained results depicted that the proposed method detected the P300 signals with 95.34% classification accuracy in parallel with high True Positive Rate (i.e., 92.9%) and low False Positive Rate (i.e., 0.77%). **Conclusions:** To estimate the performance of the proposed method, the achieved results were compared with the results of Naive Hyperparameter (NHP) tuning method. The comparisons depicted the superiority of the proposed method against its alternative, in such way that the best accuracy by using the proposed method was 6.44%, better than the accuracy of the alternative method.

Keywords: *Brain–computer interface, deep neural network, hyperparameter, nonconvex error function, P300 signal*

Introduction

Recently, brain–computer interface (BCI) technology has had a vast and rapid growth in control outer equipment using event-related potential (ERP) signals. These signals are produced in the human brain as response to external stimulus having great potential to make a nonmuscles communication path between disabled people and outside world.[1,2]

One of the important ERPs is P300 signal which is applied in BCIs, diagnosis of neurological disorders and lie detection.[2,3] The P300 waveform evoked between 250 and 300 ms after a brief auditory or visual stimulus.[4] Due to its wide applications, detecting of P300 is still a serious problem in BCI paradigm. Furthermore, low signal-to-noise ratio (SNR) makes this problem open and so challenging.[3]

In general, detecting of P300 signal consists of three steps. The first step is preprocessing in which some useless features of the signal are removed. In the second step, discriminative features are extracted from signal, and finally in the third step, classification makes up a model to distinguish P300 and non-P300 components. Based on the above procedure, effective feature extraction and classification methods make great impact on increase performance of the P300 detection.[1]

Several techniques have been applied to improve P300 detection. Averaging is the oldest method, which tries to obtain higher detection rate by increasing the SNR.

How to cite this article: Shojaedini SV, Morabbi S, Keyvanpour M. A new method for detecting P300 signals by using deep learning: Hyperparameter tuning in high-dimensional space by minimizing nonconvex error function. J Med Signals Sens 2018;8:205-14.

Received: February, 2018. **Accepted:** June, 2018.

Address for correspondence:
Dr. Seyed Vahab Shojaedini, Department of Biomedical Engineering, Iranian Research Organization for Science and Technology, Tehran, Iran.
E-mail: shojadini@irost.ir

Website: www.jmss.mui.ac.ir
DOI: 10.4103/jmss.JMSS_7_18

However, this idea reduces the bit rate in P300 and may deform the ERP signal.[4,5] Artificial intelligence methods have been proposed to increase SNR without loss of any considerable information in P300 detection.[6,7]

A group of investigations has depicted the application of linear and nonlinear models for P300 detection and classification.[8-11] For instance, linear discriminant analysis and support vector machines have extensively been applied in BCI applications.[12-14]

Certainly, linear techniques are not enough strong to deal with complex real-world problems and their nonlinear counterparts have faced with overfitting.[7] One of the most important techniques to detect P300 is artificial neural network.[6,15,16] Although usefulness of this idea, its basic drawback is getting stuck in local minimum which degrades its performance in P300 detection. In recent years, studies have demonstrated the potential of deep neural networks in the field of P300 detection. One of the most popular types of deep neural networks is convolutional neural network (CNN) and recurrent CNN (RCNN) which have been widely used for P300 detection, thanks to their ability in extracting high-level features.[17,18] Based on the literature, RCNN is used to learn electroencephalography (EEG) signals for mental activity classification. Furthermore, the RCNNs successfully preserved the spectral, spatial, and temporal structure of the data during classification.[19] In another research, spatial and temporal features of the EEG signal have been combined to train a two-dimensional CNN. Furthermore, three-dimensional CNNs were used to preserve spatiotemporal features and also employ transfer learning to further increase classification performance.[18]

Another type of method employs EEG data in time and space domains in order to offer better results regardless of the number of layers of CNN.[20] In another group of researches, new CNN architectures were proposed which used a depthwise and separable convolution to more efficiently extract relevant features for EEG-based BCIs.[21] It has been shown that such techniques resulted in more accurate classification in parallel with being as compact as possible. Deep neural networks have a deep structure with multiple levels of data representations.[22] However, they have some drawbacks mainly containing the increase in dimensions of the deep neural network, which leads to higher process volume and the ratio of the number of saddle points to local minima increases exponentially.[23,24]

Such saddle points are enclosed by some high error plateaus that may extremely slow down learning process and give the illusory impression of the existing local minimum.[24] This phenomenon seriously hampers the detection of P300 signals. Therefore, escaping from saddle points has been acquainted as a vital challenge in P300 detection by deep neural networks. It is NP-hard problem, therefore, as a solution minimizing nonconvex error function in high-dimensional spaces may be demonstrated.

A vast variety of methods have been proposed to escape saddle points containing the first order, second order, and evolutionary algorithms. The first-order algorithms try to improve P300 detection using gradient information. They are simple to use and converge fast.[25,26] The second-order algorithms compute Hessian matrix, which highly depends on dimensions of objective function. As the dimensions grow, the required memory also increases.[23,26] In some researches, evolutionary approaches have been used as learning schemes for deep neural networks. Certainly, these methods have high volume of process and computational complexity in each generation.[27]

The stochastic gradient descent (SGD) family is a *de facto* optimization paradigm for tuning a deep architecture.[28] A critical problem in SGD is to set hyperparameters seriously influences the convergence and the performance of the deep neural network.[29] Adaptive methods may improve the hyperparameters efficiently which lead to obtain higher performance in parallel with speed up training process.

In this paper, a novel method is introduced to improve training of deep neural networks, which is based on adaptive hyperparameter tuning. The proposed method minimizes nonconvex error function in high-dimensional space. For this purpose, the objective function is transferred to Lagrangian paradigm as a two-constrained optimization problem including either quality or inequality constraints. Then, Karush–Kuhn–Tucker (KKT) system is used to translate the problem into a standard nonlinear framework. Finally, the hyperparameter tuning is achieved using iterative Newton and dual active set techniques. The proposed method is applied on Adadelta module of a CNN to obtain a well-fit model for distinguishing P300 and non-P300 signals.

The structure of the paper is as follows: Section 2 includes description of dataset and proposed protocol. In Section 3, the results of evaluating the proposed method against its alternatives are demonstrated. In Section 4, the obtained results are compared to the state-of-the-art method using some effective indexes. The conclusion is presented in the last section of the paper.

Materials and Methods

In this section, firstly the details about the utilized dataset and preprocessing techniques were described. Then, the proposed scheme is introduced to address saddle point problem using a new optimal hyperparameter tuner. Finally, the proposed technique is applied on CNN to improve P300 detection.

Dataset overview

In this paper, EPFL BCI dataset was applied. It has been captured by the Biosemi system with 32 electrodes located according to standard 10-20 international system position at 2048 HZ. The EPFL BCI dataset is composed of eight available subjects as its specifications have been represented in Table 1.

Table 1: The details of EPFL dataset

Subject	Diagnosis	Description
Subject 1	Cerebral palsy	The stimulation type is visual, the data of subject 5 is not available in dataset
Subject 2	Multiple sclerosis	
Subject 3	Late-stage amyotrophic lateral sclerosis	
Subject 4	Traumatic brain and spinal cord injury, C4 level	
Subject 5	Postanoxic encephalopathy	
Subject 6	Able-body	
Subject 7	Able-body	
Subject 8	Able-body	
Subject 9	Able-body	

The data of each subject is composed of four sessions. Each of the sessions consisted of six runs; they are corresponding to six images which had displayed in a six-cell paradigm.

The images were flashed at random order. Each flash of an image lasted for 100 ms and then during 300 ms none of them were flashed. In a six-cell paradigm, the interstimulus interval (ISI) was 400 ms. More details about the dataset may be found in.[30]

Proposed protocol

An overview of the proposed protocol is shown in Figure 1 which consists of five stages which are illustrated as below.

Data acquisition

The first stage is data acquisition, during this step, the raw EEG signals were captured using 32 electrodes which are located on the scalp.

Preprocessing

Before learning deep model, preprocessing steps were applied. This stage was composed of six successive steps which are described as bellow.

Referencing

The EEG voltages have been recorded using 32 electrodes which have depended on each other. Hence, activities in the reference electrode may be reflected in the recorded signal of others. Referencing method used to remove such anomaly effects in the activity of electrodes.

Filtering

A sixth-order forward–backward Butterworth bandpass filter was used at 1.0 and 12.0 Hz cutoff signal frequency,[30] to remove additional frequencies and noise.

Downsampling

The EEG was downsampled from 2048 to 32 Hz by selecting each 64th sample from the bandpass-filtered data.[30]

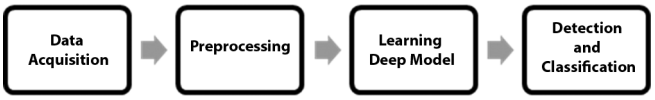

Figure 1: An overview of the proposed scheme

Signal trial extraction

Single trials of duration 1000 ms were extracted from the data. These trials were started at stimulus onset, that is, at the beginning of the intensification of an image and were ended 1000 ms after stimulus onset. Due to the ISI of 400 ms, the last 600 ms of each trial overlapped with the last 600 ms of the following trial.[30]

Windsorzing

To reduce the effects of outliers in EEG signal such as muscles activity and eye blinks, the 10th percentile and the 90th percentile of data of each electrode were computed. Amplitude values lying out of this range were replaced by start and end limits of this range, respectively.[30]

Normalizing

Normalizing is a way to control behavior of data in the model. The last stage of preprocessing is mapping the EEG signal to the range of (0, 1) to set data onto the same range, so the computational complexity has a sharp decrease.

Learning deep model

Deep CNNs have been based on deep learning theory which has large scale and different types of layers. A portion of this structure is responsible for extracting discriminative features of input data, while others are responsible for classification the data based on extracted features. Therefore, deep architecture leads to better data generalization and representation thanks to its complex structure which provides the capability of either feature extraction or classification. The basic part of the deep CNN is convolutional layer. This layer contains a collection of filter banks which are applied to input data to extracts various features. Furthermore, a nonlinear activation function is applied on neurons. Then, the pooling, which is also known as a downsampling layer is employed. Finally, the resultant features were assigned to the last layer (i.e., fully connected) in order to classify the data. Several candidates (for example, Adadelta, Adam, RMSprop) have been proposed to perform the above classification.

Suppose ω as Adadelta model parameters and T (ω) as its objective function which may be minimized to obtain a well-fit model. Gradient descent procedure is a way to obtain this optimization which makes use of the gradient information of the objective function to update model parameters as bellow:

$$E(\nabla\omega^2)_t = \alpha.E(\nabla\omega^2)_{t-1} + (1-\alpha)\nabla\omega_t^2 \quad (1)$$

In $E(\nabla\omega^2)_t$ which shows running average over the gradient of the squared model's parameter which is only depends on previous average and the current gradient. The parameter α refers to momentum and the optimization criteria for Adadelta error may be written as a set of qualities and inequalities as below:

$$J(\alpha) = \min_{\alpha} \alpha X^2 + \mu^2 A \tag{2}$$

$$P_1(\alpha) = \frac{(1-\sqrt{\alpha})^2}{k_{\min}} - \mu$$

$$P_2(\alpha) = \left(\frac{\sqrt{k_{\max}/k_{\min}} - 1}{\sqrt{k_{\max}/k_{\min}} + 1} \right)^2 - \alpha$$

Where X denotes the distance between the current model and local quadratic approximation's minimum, A denotes the estimate for gradient variance, furthermore, $k_{\max}$ and $k_{\min}$ refer to the maximum and minimum generalized curvature.[31-33] To obtain the well-fit model, it is necessary to minimize the nonlinear j (α) (i.e., Eq. 2) subject to equality constraint $P_1$and inequality constraint P_2. To perform such optimization first supposes quality constraint of Eq. 2 as:

$$\min_{\alpha \epsilon \mathbb{R}^n} J(\alpha) \tag{3}$$

$$\text{subject to:} \quad P_1(\alpha) = 0$$

Putting the mentioned equation in Lagrangian paradigm leads to:

$$L(\alpha, y) = J(\alpha) - y^{\mathrm{T}} P_1(\alpha) \tag{4}$$

Where y denotes the Lagrangian multiplier. To achieve optimum parameter, the KKT system is written in the form of nonlinear equations as follows:

$$\nabla_{\alpha} L(\alpha, y) = \nabla J(\alpha) - \nabla P_1(\alpha) y = 0 \tag{5}$$
$$\nabla_{y} L(\alpha, y) = -P_1(\alpha) = 0$$

Let us write this system of equations more compactly as $H(\alpha, y) = 0$:[34]

$$H(\alpha, y) = \begin{pmatrix} H_1(\alpha, y) \\ H_2(\alpha, y) \end{pmatrix} \tag{6}$$
$$= \begin{pmatrix} \nabla_{\alpha} L(\alpha, y) \\ \nabla_{y} L(\alpha, y) \end{pmatrix}$$
$$= \begin{pmatrix} \nabla J(\alpha) - \nabla P_1(\alpha) y \\ -P_1(\alpha) \end{pmatrix} = 0$$

Our target is to solve Eq. 6 using Newton paradigm, therefore, the gradient of $H(\alpha, y)$ is computed as:

$$\nabla H(\alpha, y) = \nabla \begin{pmatrix} H_1(\alpha, y) \\ H_2(\alpha, y) \end{pmatrix} \tag{7}$$
$$= \begin{pmatrix} \frac{\partial H_1}{\partial \alpha_1} & \frac{\partial H_2}{\partial \alpha_1} \\ \frac{\partial H_1}{\partial y_2} & \frac{\partial H_2}{\partial y_2} \end{pmatrix}$$
$$= \begin{pmatrix} \nabla^2_{\alpha\alpha} L(\alpha, y) - \nabla P_1(\alpha) \\ -\nabla P_1(\alpha)^{\mathrm{T}} \quad 0 \end{pmatrix} = 0$$

In Eq. 7, $\nabla^2_{\alpha\alpha} L(\alpha, y)$ is the Hessian of $L(\alpha, y)$ and is defined as:

$$\nabla^2_{\alpha\alpha} L(\alpha, y) = \nabla^2 J(\alpha) - \sum_{i=1}^{m} y_i \nabla^2 P_{1,i}(\alpha) \tag{8}$$

Based on the symmetry of $\nabla H(\alpha, y)$ which leads to $J(\alpha, y) = \nabla H(\alpha, y)^{\mathrm{T}} = \nabla H(\alpha, y)$, Newton's method (i.e., Eq. 7) gives

$$\begin{pmatrix} \nabla^2_{\alpha\alpha} L(\alpha, y) - \nabla P_1(\alpha) \\ -\nabla P_1(\alpha)^{\mathrm{T}} \quad 0 \end{pmatrix} \cdot \begin{pmatrix} \nabla\alpha \\ \nabla y \end{pmatrix} = -\begin{pmatrix} \nabla J(\alpha) - \nabla P_1(\alpha) y \\ -P_1(\alpha) \end{pmatrix} \tag{9}$$

The above equation may be rewritten in below quadratic form to find and the optimum $(\nabla\alpha^{\mathrm{T}}, \nabla y^{\mathrm{T}})$:

$$\min_{\Delta\alpha \epsilon \mathbb{R}^n} \frac{1}{2} \Delta\alpha^{\mathrm{T}} \left(\nabla^2_{\alpha\alpha} L[\alpha, y] \right) \Delta\alpha + (\nabla_{\alpha} L[\alpha, y])^{\mathrm{T}} \Delta\alpha \tag{10}$$

$$\text{subject to: } \nabla P_1(\alpha)^{\mathrm{T}} \Delta\alpha = -P_1(\alpha)$$

Now Eq. 9 is rewritten in a simpler form, by replacing ∇y with $\vartheta - y$ as:

$$\begin{pmatrix} \nabla^2_{\alpha\alpha} L(\alpha, y) - \nabla P_1(\alpha) \\ -\nabla P_1(\alpha)^{\mathrm{T}} \quad 0 \end{pmatrix} \cdot \begin{pmatrix} \Delta\alpha \\ \vartheta - y \end{pmatrix} = -\begin{pmatrix} \nabla J(\alpha) - \nabla P_1(\alpha) y \\ -P_1(\alpha) \end{pmatrix} \tag{11}$$

The above equation may be arranged as:

$$\begin{pmatrix} \nabla^2_{\alpha\alpha} L(\alpha, y) - \nabla P_1(\alpha) \\ -\nabla P_1(\alpha)^{\mathrm{T}} \quad 0 \end{pmatrix} \cdot \begin{pmatrix} \Delta\alpha \\ \vartheta \end{pmatrix} + \begin{pmatrix} \nabla P_1(\alpha) y \\ 0 \end{pmatrix} \tag{12}$$
$$= -\begin{pmatrix} \nabla J(\alpha) \\ -P_1(\alpha) \end{pmatrix} + \begin{pmatrix} \nabla P_1(\alpha) y \\ 0 \end{pmatrix}$$

Combining Eqs. 9 and 12 may result in Eq. 13, as follows:

$$\begin{pmatrix} \nabla^2_{\alpha\alpha} L(\alpha, y) & -\nabla P_1(\alpha) \\ -\nabla P_1(\alpha)^{\mathrm{T}} & 0 \end{pmatrix} \cdot \begin{pmatrix} \Delta\alpha \\ \vartheta \end{pmatrix} = -\begin{pmatrix} \nabla J(\alpha) \\ -P_1(\alpha) \end{pmatrix} \tag{13}$$

Therefore, the quadratic Eq. 10 may be commented as:

$$\text{mini}_{\Delta\alpha} \frac{1}{2} \Delta\alpha^{\mathrm{T}} \left(\nabla^2_{\alpha\alpha} L[\alpha, y] \right) \Delta\alpha + \nabla J(\alpha)^{\mathrm{T}} \Delta\alpha \tag{14}$$

$$\text{subject to: } \nabla P_1(\alpha)^{\mathrm{T}} \Delta\alpha = -P_1(\alpha)$$

Furthermore, the explained process is extended to include inequality constraint in Eq. 2 which results in bellow equation:

$$\min_{\alpha \epsilon \mathbb{R}^n} J(\alpha) \tag{15}$$

$$\text{subject to:} \quad P_2(\alpha) \geq 0$$

In a similar manner with Eqs 2–14, the Eq. 15 leads to 16

which is quadratic form of inequality constraint (i.e., similar to Eq. 14):

$$\min_{\Delta\alpha} \frac{1}{2}\Delta\alpha^{T}\left(\nabla^{2}_{\alpha\alpha}L[\alpha, y]\right)\Delta\alpha + \nabla J(\alpha)^{T}\Delta\alpha \quad (16)$$

$$\text{subject to: } \nabla P_2(\alpha)^{T}\Delta\alpha \geq -P_2(\alpha)$$

Now we have simplified Eqs. 14 and 16 which solving them leads to the optimization of Eq. 2 in such way that the optimal value of the hyperparameter is achieved. A sequence of Newton iterations is applied to find an acceptable solution for Eq. 14. At every iteration, the direction improving is performed; therefore, the process is called sequential quadratic programming.[34] To solve Eq. 16, the active set or dual active set method[34] is used to obtain the optimal solution.

Require: Normalized function *J*, nonlinear equality constraint P_1, nonlinear inequality constraint P_2, and the electroencephalography raw signals.

1. Preprocessing electroencephalography signals
2. Make train, validation, and test sets using the data of step 1
3. For epoch = 1 to 100 do
   Find the optimum value of hyperparameter using the following steps:
   a. Convert *J*, P_1, and P_2 to corresponding Lagrangian function (Eq. 4)
   b. Corresponding Karush–Kuhn–Tucker system of step a to find optimum value (Eq. 5)
   c. Convert Karush–Kuhn–Tucker system to a system of nonlinear equations (Eq. 6)
   d. Newton's method approximates the root of a given function at step c
      d.1. Compute the gradient of the given function at step c (Eqs. 7 and 8)
      d.2. Apply Newton's method (The Newton method, derived from Eq. 7, transforms Eqs. 13, and 15 into equality and inequality forms. In fact, they are the Karush–Kuhn–Tucker system for the quadratic equations in Eqs. 14 and 16)
         d.3. While (non [cpnverged to optimum vaue])
         d.3.1 finding an optimum value for Eqs. 14 and 16 using minimizer
   e. $\alpha_{optimum}$ = optimum value of the hyperparameter
   Training a deep model using Adadelta which use the optimum hyperparameter $\alpha_{optimum}$ at each epoch of training step and train and validation sets.
4. End of for
5. Give the test set as input to pretrained deep model
6. Detection P300 signals
7. Classification the detected signals as P300 and non-P300 signals

Figure 2: Pseudocode of P300 detection

Detection and classification

A pretrained CNN is applied to detect P300 signals. The CNN builds up a model to map between the features and EEG signals categories, which are known as P300 and non-P300 with binary labels. A comprehensive pseudocode of the proposed P300 detection protocol is shown in Figure 2.

Results

The proposed method was implemented on MATLAB R2017a, with an Intel Core i7 and 2 TB RAM. It was applied on EPFL dataset which was recorded using 32 channels from 8 subjects. The electrodes were located in predetermined positions based on 10–20 system, as shown in Figure 3.

The data of each subject composed of four sessions. Hence, in each subject, the data from two sessions were used to train and one another used to validation and the data from leave-off session was used to test. This method was repeated four times, hence, each session was presented once for test. At least, for each subject, the average of four steps at four different examined methods was evaluated.

The EEG data of each subject were first preprocessed to make it ready to feed up to CNN. Then, P300 was detected by applying CNN which had been trained by Adadelta based Naive Hyperparameter Tunning which is called as NHP for brevity in the rest of article,[18] Adam,[35] RMSprop,[36] and proposed schemes, respectively. The structure of CNN is presented in Figure 4. As it illustrated, the CNN composed of four convolutional layers including two max pooling and two activation function layers. In the above and bottom of each layer, some information about the current layer is presented. For example, in the above of the first convolutional layer, the phrase Conv (5, 1, 0, and 20) means that in this layer, a 5 by 5 filter, stride size 1, no

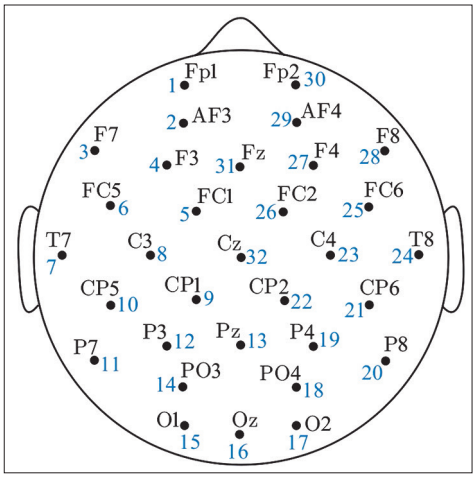

Figure 3: The position of 32 electrodes based on international 10–20 system

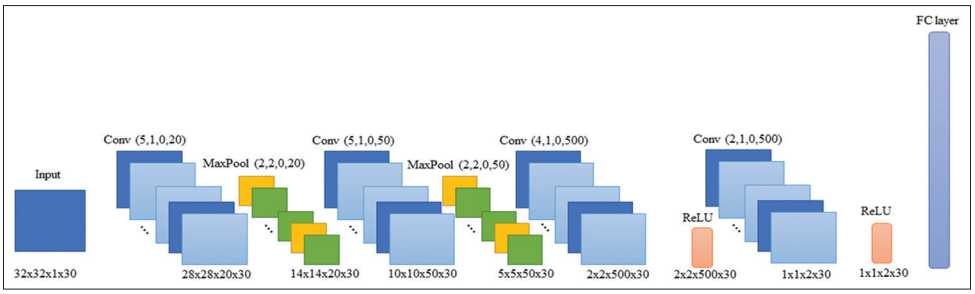

Figure 4: The architecture of convolutional neural network model for P300 signal detection. In the above of each layer, its name and structure are appeared. (Conv: Convolutional layer, MaxPool: Max pooling layer, FC layer: Fully connected layer and Conv/MaxPool [kernel, stride, padding, neuron]. The arguments in parentheses after the name of layer refer to the size of filter/kernel, the size of stride, the size of padding, and the number of neurons in the current layer, respectively). The below information in each layer indicates the size of the output of that layer

padding, and 20 neurons are used. Therefore, the output size of the current layer is 28 by 28 by 20 by 30 as indicated below of this layer. The phrase 28 × 28 × 20 × 30 in the bottom of layer mentioned that a linear product using the first layer is applied as the input of CNN. Therefore, the size of input changed to 28 by 28; furthermore, the value 20 refers to the number of neurons and 30 is the batch size in each epoch of learning.

Finally, to estimate how good they work, their performances were measured. For each subject, fourfold cross-validation was employed to evaluate true-positive ratio (TPR), true-negative ratio (TNR), false-positive ratio (FPR), false-negative ratio (FNR), and classification accuracy. The TP shows the number of correctly identified P300 signals. The TN shows the non-P300 signals which were rejected correctly. The FP is the number of false detections and the FN shows the number of missed P300 signals. The TPR is the ratio of correct detections and may be defined as:

$$\text{TPR} = \frac{\text{TP}}{\text{TP}+\text{FN}} \times 100 \quad (17)$$

Furthermore, The FNR refers to the ratio of missed P300 signals:

$$\text{FNR} = \frac{\text{FN}}{\text{FN}+\text{TP}} \times 100 \quad (18)$$

Moreover, the FPR, measured the non-P300 as P300 signals:

$$\text{FPR} = \frac{\text{FP}}{\text{FP}+\text{TN}} \times 100 \quad (19)$$

The TNR, measured the non-P300 signals which were classified correctly.

$$\text{TNR} = \frac{\text{TN}}{\text{TN}+\text{FP}} \times 100 \quad (20)$$

The accuracy is defined to measure the state of being correct detection as follows:

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FN}+\text{FP}} \times 100 \quad (21)$$

They were computed as evaluation parameters which illustrate the effectiveness of the examined approaches as shown in Table 2.

Table 2: The average of parameters over subjects

Method	Subject	TPR	FPR	TNR	FNR	Accuracy
NHP method	1	70.74	8.86	91.13	29.25	80.93
	2	67.45	5.35	94.64	32.54	81.05
	3	77.18	2.14	97.85	22.81	87.51
	4	68.39	3.24	96.75	31.60	82.57
	6	76	5.14	94.85	23.99	83.41
	7	72.61	3.22	96.77	27.38	84.69
	8	78.85	2.23	97.77	21.14	88.31
	9	61.79	2.75	97.24	38.20	79.52
Adam	1	63.67	3.28	96.71	36.32	80.19
	2	65.84	4.93	95.06	34.15	80.45
	3	73.51	5.15	94.84	26.48	84.18
	4	72.31	2.58	97.41	27.68	84.86
	6	70.96	6.87	93.12	29.03	82.04
	7	72.11	3.51	96.48	27.88	84.30
	8	83.19	3.96	96.03	16.80	89.61
	9	75.42	17.06	82.93	24.57	79.17
RMSprop	1	57.80	2.72	97.27	42.19	77.53
	2	55.59	2.31	97.68	44.40	76.64
	3	79.19	5.93	94.06	20.80	86.62
	4	72.38	2.90	97.09	27.61	84.74
	6	70.28	6.22	93.77	29.71	82.02
	7	75.64	3.03	96.96	24.35	86.30
	8	80.55	2.98	97.01	19.44	88.78
	9	63.03	4.29	95.70	36.96	79.36
Proposed method	1	79.34	1.72	98.27	20.65	88.81
	2	81.77	2.09	97.90	18.22	89.84
	3	92.90	2.22	97.77	7.02	95.34
	4	85.59	2.92	97.07	14.40	91.33
	6	90.12	4.35	95.65	9.87	92.88

TPR: True positive rate, FPR: False positive rate, TNR: True negative rate, FNR: False negative rate, NHP: Naive hyperparameter tunning

As Table 2 shows, based on TPRs, the proposed method outperformed in all of the subjects. For example, the best TPR which has been obtained by this method was equal to 92.90%, over subject 3. However, the best TPRs which have been obtained using NHP, Adam, and RMSprop methods were 78.85%, 83.19%, and 80.55%, respectively, over subject 8. Furthermore, according to TNRs, the

proposed method has been better than alternatives. For instance, the proposed method has been obtained TNR of 99.22% over subject 8, which was the best among all TNRs. Whereas, the best TNRs which have been gained by NHP, Adam, and RMSprop methods were 97.85%, 97.41%, and 97.68% over subjects 3, 4, and 2, respectively.

Exploring FPR and FNR values also demonstrated the superiority of the proposed algorithm against its alternatives. By investigating the obtained FPRs, the proposed scheme has been achieved to the value of 0.77% over subject 8, which is the best among all obtained false-positive rates. On the other hand, the best values among alternatives were equal to 2.14%, 2.58%, and 2.31% over subjects 3, 4, and 2, respectively. Moreover, the best obtained FNR was equal to 7.02% over subject 3 using the proposed protocol. However, best FNRs which have been achieved using alternatives (i.e., NHP, Adam, and RMSprop) were equal to 21.14%, 16.80%, and 19.44% over subject 8.

Finally, the classification accuracy was confirmed better performance of our proposed method. The best accuracy has been obtained 95.34% using proposed method over subject 3. However, the best values which have been gained by the alternatives were equal to 88.31%, 89.61%, and 88.78%, over subject 8.

Discussion

The subjects were faced to a screen on which six images were sequentially displayed including a television, a telephone, a lamp, a door, a window, and a radio. The images were offered in random sequences with a stimulus interval of 400 ms.

Each subject was completed four recording sessions. The first two sessions were performed on one day and the last two sessions on another day. Each of the sessions consisted of six runs, containing one run for each of the six images. As illustrated in Figure 5, for eight subjects, the averaged data of each run was calculated over all of four sessions. As discussed before, each run is corresponding to each of six images in the specific random order. Hence, each column in presented bar charts refers to how correct an image was detected by special user over all sessions.

The given bar charts show the trend of classification accuracy in six runs over eight subjects. As is observed in the column graph, in all cases, the proposed has been better than the best among alternatives based on accuracy (i.e., NHP). The amount of this superiority was various from subjective point of view. At a glance over the proposed method, in subject 1, the significant rise of averaged classification accuracy in run 3 was 90.09%, besides, the lowest number, 87.85% was belonging to run 2. Similarly, the highest and lowest growth of accuracy have been observed in subject 2 in such way that they were equal to 90.70% and 88.55%, respectively. In the same token, in subject 3, the highest and lowest growth of accuracy were equal to 97.20% and 94.02%. Just as, in subject 4, they were equal to 93.81% and 89.89%. In the same way, in subject 6, the above parameters were obtained equal to 95.12% and 90.45%. For subject 7, they were equal to 95.73% and 92.19%. As same, for subject 8 and 9, the highest numbers were 94.66% and 91.31% and the lowest numbers were equal to 91.43% and 87.82% interval.

The mentioned parameters showed that the lowest superiority has been achieved in subject 9 and the highest superiority belonged to subject 3. Contrary to the proposed method, in the NHP, the evaluated average accuracy based on different runs was not stable. It can be clearly seen that in subject 6, overall runs the NHP method achieved some accuracy in the range of 81.29% to 89.39%. Whereas, based on the proposed, the trend of accuracy in the same subject was in the small range approximately 4.67% which demonstrate that it is a more reliable method. The maximum variance of the proposed method based on average accuracy has been obtained 3.54% in subject 7, on the other hand, the maximum number based on NHP was 9.8% in subject 7 which shows approximately three times of variance against our proposed method.

In a similar way, the minimum variance of the proposed scheme according to average accuracy has been obtained 4.67% in subject 6, whereas, the minimum of the NHP method was 8.1% which was belonging to subject 6 and shows approximately two times of variety against our proposed method.

A main drawback of the proposed method is to compute several derivatives, which likely need to be worked analytically in advance of iterating to a solution. Therefore, the proposed method has to handle computational complexity in large problems with many variables or constraints.

Conclusion

In this article, a new adaptation method for hyperparameters in CNNs was introduced to improve the detecting of P300 signal in BCI applications. The proposed method has been based on quadratic optimization of Adadelta error which has been illustrated in the form of a set of qualities and inequalities.

To figure out the efficiency of the proposed method, it was compared with the NHP, Adam, and RMSprop techniques in terms of TPR, TNR, FPR, FNR, and accuracy. The obtained results showed that by performing the proposed optimization, a better well-fit model had been obtained for distinguishing P300 and non-P300 signals. The average accuracy of the proposed method had been at least 7.03%, 5.73%, and 6.56% better than the best among of NHP, Adam, and RMSprop methods, respectively.

Moreover, the achieved TPR was also confirmed the superiority of the proposed method in such way that it was 14.05%, 9.71%, and 12.35% better than

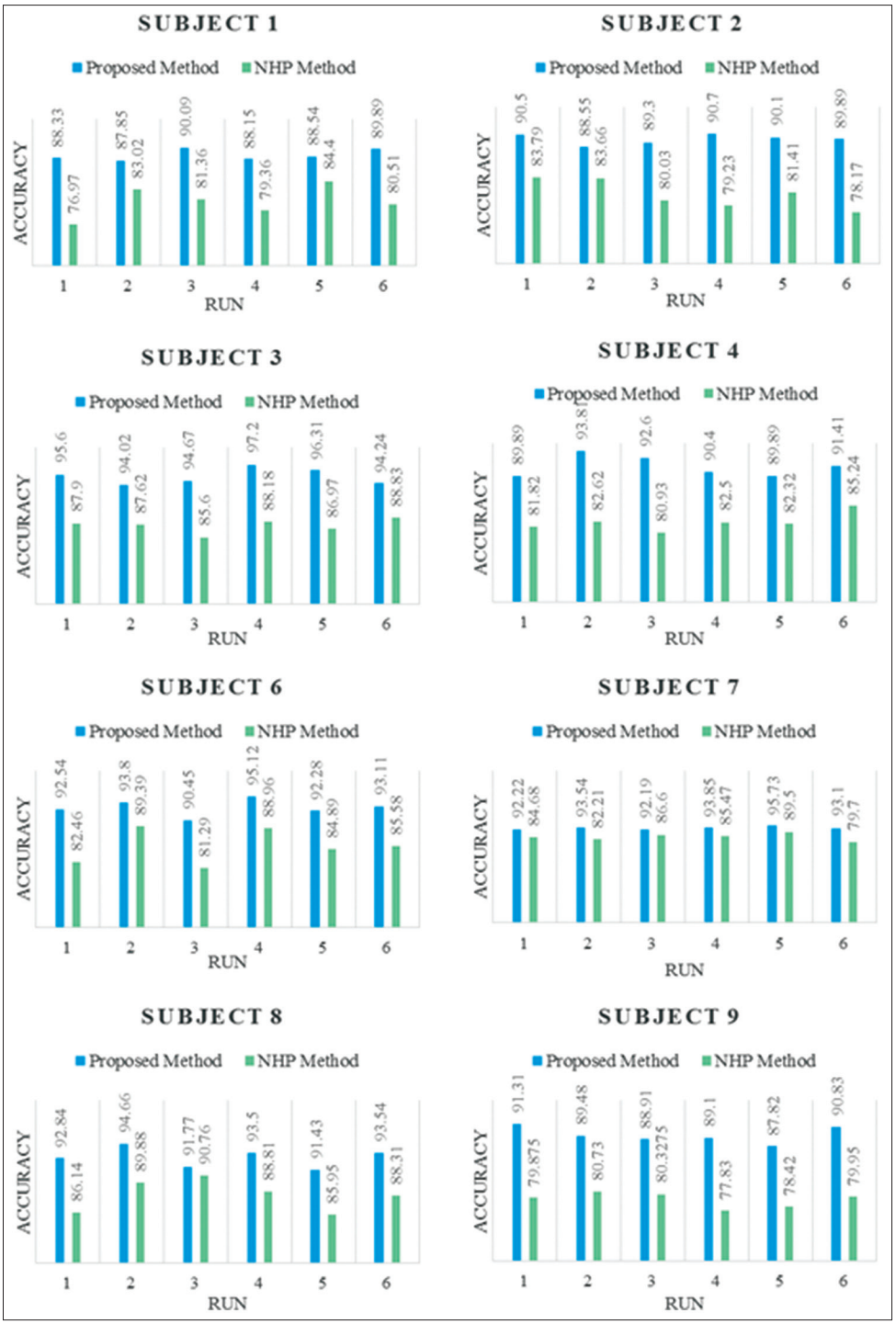

Figure 5: Empirical result of applying the NHP and the proposed methods on Adadelta algorithms for convolutional neural network which is trained on the EPFL dataset based on classification accuracy

alternatives (i.e., NHP, Adam, and RMSprop). Similarly, FPR, FNR, and TNR of the proposed method showed considerable superiorities (1.37, 1.81, 1.54), (14.12, 9.78, 12.42), and (1.37, 1.81, 1.54) percent, respectively, against the alternative methods. Furthermore, it was understood that the performance of the proposed method has had lower sensitivity against different stimulating patterns (i.e., runs) than the best among alternatives based on accuracy. The accuracies which have been obtained from proposed method against several runs showed the variances approximately 2–3 times lower than the same variances which had been obtained for its alternative.

Based on the mentioned results, it may be concluded that the proposed method has a considerable potential to improve P300 detection in BCI applications.

Financial support and sponsorship

This research was supported in part by Iranian Research Organization for Science and Technology.

Conflicts of interest

There are no conflicts of interest.

References

1. Prasad G, Herman P, Coyle D, McDonough S, Crosbie J. Using Motor Imagery Based Brain-Computer Interface for Post-Stroke Rehabilitation. International IEEE/EMBS Conference on Neural Engineering. Antalya, Turkey. 2009. p. 258-62.
2. Chowdhury A, Raza H, Meena YK, Dutta A, Prasad G. Online Covariate Shift Detection Based Adaptive Brain-Computer Interface to Trigger Hand Exoskeleton Feedback for Neuro-Rehabilitation. IEEE Transactions on Cognitive and Developmental Systems; 2017.
3. Mubeen MA, Knuth KH. Evidence-Based Filters for Signal Detection: Application to Evoked Brain Responses. arXiv preprint arXiv 2011;1107:1257.
4. Vareka L, Mautner P. Using the Windowed Means Paradigm for Single Trial P300 Detection. In: Telecommunications and Signal Processing (TSP), 2015 38th International Conference. Prague, Czech Republic: IEEE; 2015. p. 1-4.
5. Morales C, Held CM, Estevez PA, Perez CA, Reyes S, Peirano P, *et al*. Single trial P300 detection in children using expert knowledge and SOM. InEngineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE; 2014 Aug 26. p. 3801-3804.
6. Hutagalung SS, Turnip A, Munandar A. P300 Detection Based on Extraction and Classification in Online BCI. In: Instrumentation Control and Automation (ICA), 2013 3rd International Conference. Bali, Indonesia: IEEE; 2013. p. 35-8.
7. Sobhani A. P300 Classification Using Deep Belief Nets (Doctoral Dissertation, Colorado State University); 2014.
8. Haghighatpanah N, Amirfattahi R, Abootalebi V, Nazari B. A Single Channel-Single Trial P300 Detection Algorithm. In: Electrical Engineering (ICEE), 2013 21st Iranian Conference. Mashhad, Iran: IEEE; 2013. p. 1-5.
9. Chen SW, Lai YC. A signal-processing-based technique for P300 evoked potential detection with the applications into automated character recognition. EURASIP J Adv Signal Process 2014;2014:152.
10. Lazar AM, Ursulean R. The P300 Event-Related Potential Detection-A Morphological Approach. In: E-Health and Bioengineering Conference (EHB). Iasi, Romania: IEEE; 2013. p. 1-4.
11. Hoffmann U, Garcia G, Vesin JM, Diserens K, Ebrahimi T. A Boosting Approach to P300 Detection with Application to Brain-Computer Interfaces. In: Neural Engineering, 2005. Conference Proceedings. 2nd International IEEE EMBS Conference. Washington D.C., USA: IEEE; 2005. p. 97-100.
12. Daubigney L, Pietquin O. Single-Trial P300 Detection with Kalman Ltering and SVMs. In: 19th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESSAN) 2011. Bruges, Belgium: ISBN 978-2-87419-044-5; 2011. p. 399-404.
13. Duvinage M, Castermans T, Petieau M, Hoellinger T, Cheron G, Dutoit T, *et al*. Performance of the emotiv epoc headset for P300-based applications. Biomed Eng Online 2013;12:56.
14. Rakotomamonjy A, Guigue V. BCI competition III: Dataset II- ensemble of SVMs for BCI P300 speller. IEEE Trans Biomed Eng 2008;55:1147-54.
15. Magee R, Givigi S. A Genetic Algorithm for Single-Trial P300 Detection with a Low-Cost EEG Headset. In: Systems Conference (SysCon), 2015 9th Annual IEEE International. Vancouver, BC, Canada: IEEE; 2015. p. 230-4.
16. Zhang JC, Xu YQ, Yao L. P300 Detection Using Boosting Neural Networks with Application to BCI. In: Complex Medical Engineering, 2007. CME 2007. IEEE/ICME International Conference. Beijing, China: IEEE; 2007. p. 1526-30.
17. Cecotti H, Gräser A. Convolutional neural networks for P300 detection with application to brain-computer interfaces. IEEE Trans Pattern Anal Mach Intell 2011;33:433-45.
18. Maddula RK, Stivers J, Mousavi M, Ravindran S, de Sa VR. Deep Recurrent Convolutional Neural Networks for Classifying P300 BCI Signals. In: Proceedings of the Graz BCI Conference; 2017.
19. Bashivan P, Rish I, Yeasin M, Codella N. Learning representations from EEG with deep recurrent-convolutional neural networks. arXiv preprint arXiv 2015;1511:06448.
20. Carabez E, Sugi M, Nambu I, Wada Y. Identifying single trial event-related potentials in an earphone-based auditory brain-computer interface. Appl Sci 2017;7:1197.
21. Lawhern VJ, Solon AJ, Waytowich NR, Gordon SM, Hung CP, Lance BJ. Eegnet: A compact convolutional network for eeg-based brain-computer interfaces. arXiv preprint arXiv 2016;1611:08024.
22. Kawaguchi K. Deep Learning Without Poor Local Minima. In: Advances in Neural Information Processing Systems; 2016. p. 586-94.
23. Dauphin YN, Pascanu R, Gulcehre C, Cho K, Ganguli S, Bengio Y. Identifying and Attacking the Saddle Point Problem in High-Dimensional Non-Convex Optimization. In: Advances in Neural Information Processing Systems. Montreal, Quebec, Canada; 2014;4:2933-41.
24. Pascanu R, Dauphin YN, Ganguli S, Bengio Y. On the saddle point problem for non-convex optimization. arXiv preprint arXiv 2014;1405:4604.
25. Ge R, Huang F, Jin C, Yuan Y. Escaping from Saddle Points-Online Stochastic Gradient for Tensor Decomposition. In: Conference on Learning Theory. Paris, France; 2015;40:797-842.
26. Wang Z, Oates T, Lo J. Adaptive Normalized Risk-Averting Training for Deep Neural Networks. In: 30th Association for the Advancement of Artificial Intelligence (AAAI). Arizona; USA; 2016;16:2201-7.
27. Morse G, Stanley KO. Simple Evolutionary Optimization Can Rival Stochastic Gradient Descent in Neural Networks. In: Proceedings of the Genetic and Evolutionary Computation Conference. Berlin, Germany: Association for Computing Machinery (ACM); 2016. p. 477-84.
28. Wu X. A Study of Stability in Data Privacy. The University of Wisconsin-Madison; 2016.
29. Cong G, Peng WC, Zhang WE, Li C, Sun A, editors. Advanced Data Mining and Applications: 13th International Conference, ADMA 2017, Singapore, November 5–6, 2017, Proceedings. Singapore:Springer, 2017;10604.
30. Hoffmann U, Vesin JM, Ebrahimi T, Diserens K. An efficient P300-based brain-computer interface for disabled subjects. J Neurosci Methods 2008;167:115-25.
31. Ruder S. An overview of gradient descent optimization algorithms. arXiv preprint arXiv 2016;1609:04747.
32. Zeiler MD. ADADELTA: An adaptive learning rate method. arXiv preprint arXiv 2012;1212:5701.
33. Lessard L, Recht B, Packard A. Analysis and design of optimization algorithms via integral quadratic constraints. SIAM

J Optim 2016;26:57-95.
34. Hansen EL, Volcker C. Numerical Algorithms for Sequential Quadratic Optimization. Kongens Lyngby, Technical University of Denmark Informatics and Mathematical Modelling; 2007.
35. Kinga D, Adam JB. A Method for Stochastic Optimization. In: International Conference on Learning Representations (ICLR); 2015.
36. Tieleman T, Hinton G. Divide the Gradient by a Running Average of its Recent Magnitude. COURSERA: Neural Networks for Machine Learning. Technical Report. Available from: https://www.zh.coursera.org/learn/neuralnetworks/lecture/YQHki/rmsprop-divide-the-gradient-by-a-running-average-of-its-recent-magnitude. [Last accessed on 2017 Apr 21].

BIOGRAPHIES

Seyed Vahab Shojaedini has received his BSc in the field of Communication Engineering from Amirkabir University of Technology, Tehran, Iran in 1998. He also received his MSc and PhD in Bioelectrics from University of Tarbiat Modares, Tehran, Iran in 2001 and 2006 respectively. Since 2010 he has a position at Iranian Research Organization for Science and Technology (IROST) and at the moment he is associate professor in biomedical engineering and deputy of Institute of Electrical Engineering and Information Technology of IROST. He is interested in Signal and Image Processing, Machine Learning and Stochastic Process.
Email: shojadini@irost.ir

Sajedeh Morabbi has received her BSc in the fileld of Softwear Engineering from Non-profit-NGOs Shahrod University, Shahrood, Iran in 2011. She also received her MSc in Artificial Intelligence from Alzahra University, Tehran, Iran in 2017. She is interested in Machine Learning, Optimization and Signal Processing.
Email: report.classic@gmail.com

Mohamadreza Keyvanpour is Associate Professor of Software Engineering at Alzahra University, Tehran, Iran. He received his BSc in Software Engineering from Iran University of Science and Technology, Tehran, Iran. He received his MSc and PhD in Software Engineering from Tarbiat Modares University, Tehran, Iran. His research interests include Image Retrieval and Data Mining.
Email: keyvanpour@alzahra.ac.ir