PAPER

A novel deep learning approach for classification of EEG motor imagery signals

IOP Publishing Journal of Neural Engineering
J. Neural Eng. **14** (2017) 016003 (11pp) doi:10.1088/1741-2560/14/1/016003

A novel deep learning approach for classification of EEG motor imagery signals

Yousef Rezaei Tabar[1] **and Ugur Halici**[2]

[1] Biomedical Engineering Department, Middle East Technical University, Ankara, Turkey
[2] Department of Electrical and Electronics Engineering, Neuroscience and Neurotechnology, Middle East Technical University, Ankara, Turkey

E-mail: rezaeetabar@gmail.com and halici@metu.edu.tr

Received 16 March 2016, revised 3 November 2016
Accepted for publication 14 November 2016
Published 30 November 2016

Abstract

Objective. Signal classification is an important issue in brain computer interface (BCI) systems. Deep learning approaches have been used successfully in many recent studies to learn features and classify different types of data. However, the number of studies that employ these approaches on BCI applications is very limited. In this study we aim to use deep learning methods to improve classification performance of EEG motor imagery signals. *Approach*. In this study we investigate convolutional neural networks (CNN) and stacked autoencoders (SAE) to classify EEG Motor Imagery signals. A new form of input is introduced to combine time, frequency and location information extracted from EEG signal and it is used in CNN having one 1D convolutional and one max-pooling layers. We also proposed a new deep network by combining CNN and SAE. In this network, the features that are extracted in CNN are classified through the deep network SAE. *Main results*. The classification performance obtained by the proposed method on BCI competition IV dataset 2b in terms of kappa value is 0.547. Our approach yields 9% improvement over the winner algorithm of the competition. *Significance*. Our results show that deep learning methods provide better classification performance compared to other state of art approaches. These methods can be applied successfully to BCI systems where the amount of data is large due to daily recording.

Keywords: EEG, BCI, motor imagery, deep learning, convolutional neural networks, stacked autoencoders

(Some figures may appear in colour only in the online journal)

1. Introduction

A brain computer interface (BCI) system provides an alternative pathway between human brain and external devices [1]. This process starts by recording the brain activity and continues through the signal processing part to detect the user's intent. Then an appropriate signal is sent to the external device to control the device according to the detected signal. BCI systems are currently being studied for a wide range of applications from communication tools for locked-in state (CLIS) patients [2] to video gaming for healthy people [3].

Among different brain activity monitoring modalities, electroencephalography (EEG) technique provides an easy and inexpensive solution for BCI systems and is used in many non-invasive BCI studies [4].

Various types of EEG signals have been used in BCI studies to serve as BCI control signals. Among these signals, P300 evoked potentials, steady-state visual evoked potentials (SSVEP) and motor imagery (MI) are the most popular signals. In this paper we consider brain potentials related to motor imagery tasks, as control signals. MI refers to imagination of moving a body part without actual movement. Different MI tasks cause oscillatory activities observed in different regions in the sensorimotor cortex of the brain [5]. The goal is to classify these activities in order to recognize the underlying MI task performed. Imagination of movement of left hand, right hand, both feet and tongue are the MI tasks investigated in many BCI studies [6–10]

Pattern recognition is one of the most important parts of a BCI system. Many studies have investigated different feature

extraction and classification methods for MI task recognition. Common spatial patterns (CSP) [6, 9] is a popular method among MI studies for extracting features. CSP has been applied successfully in many MI task recognition studies [7–10]. Other well-known feature extraction and dimension reduction methods like independent component analysis (ICA) [11] and principal component analysis (PCA) [12] are also used frequently to improve the classification accuracy [13–15]. In the classification part, many traditional algorithms like support vector machine (SVM) [2, 16], linear discriminant analysis (LDA) [17] and Bayesian classifier [18] have been employed in different studies [8, 15, 19].

Continuous data recording in BCI systems provide a large amount of data that can be used for classifier training. On the other hand, deep learning methods are known to provide better classification performance by increasing the size of training data. This makes deep learning methods a great choice for signal processing step of BCI systems. However, the number of studies using deep learning methods in BCI is very limited compared to the huge applications of deep learning in other fields.

In a study by An *et al* [20], a deep belief network (DBN) model was applied for two class MI classification and DBN was shown more successful than the SVM method. DBN is used in several other EEG studies in order to perform tasks like anomaly measurement of EEG signals [21], classification of image RSVP events [22] and feature extraction [23].

Convolutional neural networks (CNN) are also used in some EEG studies. For instance, in [24], CNN was used to classify MI EEG signals. In a recent study [25], a new form of multi-dimensional features was proposed and employed in a recurrent-convolutional neural network architecture in order to model cognitive events from EEG data. A network with three autoencoders and two softmax layers was proposed in [26] for automatic emotion recognition from EEG signals.

Several studies use different methods to convert EEG signal to image representation before applying CNN. In [24], augmented common spatial pattern (ACSP) features were generated based on pair-wise projection matrices. This can be considered as extraction of the CSP features from a multi-level decomposition of different frequency ranges. In [25], the authors proposed a new form of features that preserve the spatial, spectral, and temporal structure of EEG. In this method, power spectrum of the signal from each electrode is estimated and sum of squared absolute values are computed for three selected frequency bands. Then polar projection method is used to map the electrode locations from 3D to 2D and construct the input images.

In this study, a new form of input that combines time, frequency and location information extracted from MI EEG signal is introduced. short time Fourier transform (STFT) method was used to convert EEG time series into 2D images. Unlike [25] where spectral measurements are aggregated for each band, in this study, spectral content of mu and beta frequency bands are used explicitly. This way we preserve activation patterns among different frequencies as well as time and location.

In our approach, input data is used in a 1D convolutional neural network (CNN) [27] to learn the activation patterns of different MI signals. 1D convolution is applied only in time axis, rather than frequency and location. Therefore, the shape of activation patterns (i.e. power values from different frequencies) and their location (i.e. EEG channel) are being learned in the convolutional layer. Max-pooling is applied after convolution in time axis, which makes CNN partially invariant to the time location of activation patterns. Then a stacked autoencoder (SAE) [28] with 6 hidden layers is used to improve the classification through a deep network. In order to compare the performance of the proposed CNN-SAE network, CNN and SAE networks are also used separately for classification.

The proposed approach is analyzed and evaluated by using BCI Competition IV dataset 2b [29]. The results of methods used in this study are compared with the results of current state of art algorithms in this field. To be fair, the experiments are performed in the same way as these studies and the results are presented considering classification accuracy and kappa value [30] metrics. In order to test the proposed methods on another dataset, we also applied these methods with the same testing protocol on BCI Competition II dataset III [31] and compared the results with current state of art studies.

The rest of the paper is organized as follows: Input data form and applied networks (CNN, SAE and combined CNN-SAE) are explained in section 2. Datasets and experiments as well as their results are presented and discussed in section 3. Finally, section 4 concludes the study.

2. Deep learning framework

2.1. Input image form

The datasets that we used in this study include recordings from three electrodes (C3, Cz and C4) during left/right hand MI task. These electrodes are located on motor area of the brain.

It is shown in [5] that the energy in mu band (8–13 Hz) observed in motor cortex of the brain decreases by performing an MI task. This decrease is called event related desynchronization (ERD). An MI task also causes an energy increase in the beta band (13–30 Hz) that is called event related synchronization (ERS). Left and right hand movement MI tasks are said to cause ERD and ERS respectively in the right and left sides of the motor cortex affecting EEG signals at C4 and C3 electrodes. Cz is also affected by hand movement MI task. Considering these facts, we designed our network input in order to take advantage of time and frequency properties of the data.

Short time Fourier transform (STFT) was applied on the time series for each 2 s long trial. In case of 250 Hz signal this is corresponding to 500 samples. STFT was performed with window size equal to 64 and time lapses equal to 14. Starting from sample 1 toward sample 500, STFT is computed for 32 windows over 498 samples and the last 2 samples, remaining at the end, are simply ignored. This leads to a 257×32 image where 257 and 32 are the number of samples along the frequency and time axes respectively. Then we extracted mu and

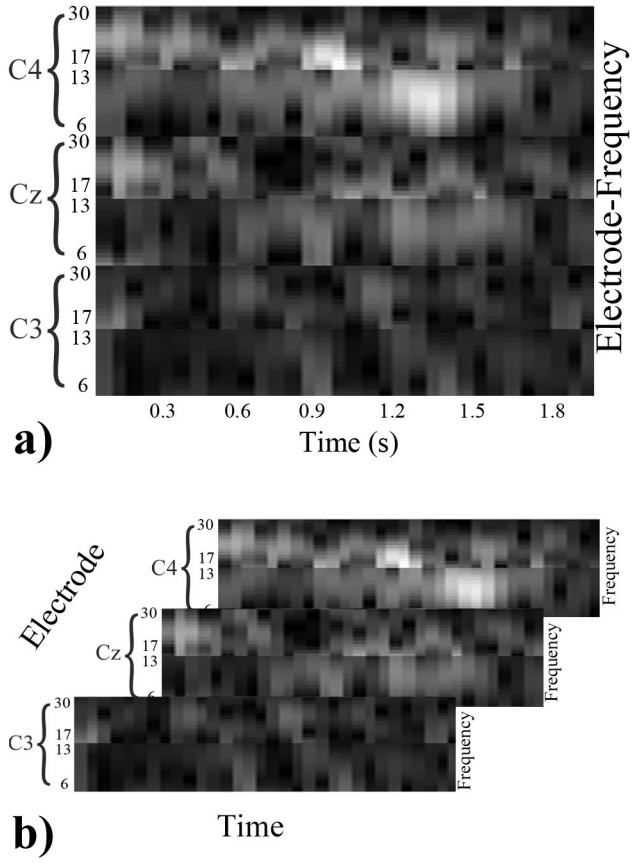

Figure 1. (a) A right hand sample input image including 2 frequency bands for each electrode C3, Cz and C4. (b) The same image as a 3D tensor.

beta frequency bands from the output spectrum. The frequency bands between 6–13 and 17–30 were considered to represent mu and beta bands. The frequency bands are slightly different than literature but they were resulted in a better data representation in our experiments. The size of extracted image for mu band was 16×32 where the size of extracted image for beta band was 23×32. In order to keep the effect of both bands similar, beta band was resized to 15×32 by using cubic interpolation method. Then these images were combined to make a $N_{\text{fr}} \times N_t$ image where $N_{\text{fr}} = 31$ and $N_t = 32$.

This process was repeated for $N_c = 3$ electrodes which are C4, Cz and C3. The results were combined in a way that the electrode neighboring information was preserved. The size of the resulting input image is $N_h \times N_t$ where $N_h = N_c * N_{\text{fr}} = 93$. A sample input image constructed for a right hand MI task trial is illustrated in figure 1. By using the proposed method, brain activations in right and left sides of the motor cortex of the brain, cause different activation patterns along the vertical cortex of the brain.

The ERD effect in C3 channel is clear in figure 1 for a sample that corresponds to a right hand MI task (darker 6–13 Hz band in C3 compared to C4). However the ERS effect in C3 electrode is not strong for this sample. Similar to right hand, for a sample collected for left hand MI task, such activation is expected to occur at the electrodes taking place at the opposite side. These images are constructed for each trial

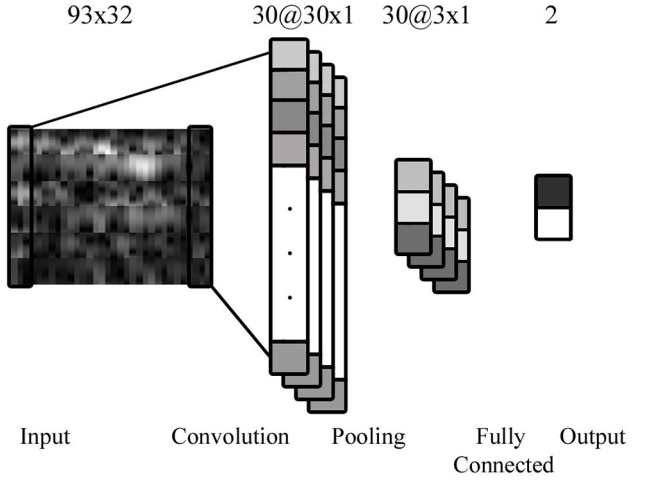

Figure 2. Proposed convolutional neural network model, $N_{\text{F}} = 30$.

sample and used as the input of CNN and SAE networks in the next stage.

2.2. Convolutional neural network (CNN)

CNNs are multi-layer neural networks with several convolution-pooling layer pairs and a fully connected layer at the output. Standard CNN [32] is designed to recognize shapes in images and is partially invariant to the location of the shapes. Input image is convolved with several 2D filters in the convolutional layer and subsampled to a smaller size in the pooling layer. Network weights and filters in the convolution layer are learned through back-propagation algorithm in order to decrease the classification error.

However, in our data, frequency, time and electrode location information are used together. Vertical location of the activation on the input image plays an important role in the classification performance while its horizontal location is not that critic. Therefore, instead of 2D filtering that is commonly used in literature, we introduced filters having the same height as the input and 1D filtering is applied only along the horizontal axis.

Notice that, instead of giving the input to CNN as a 2D image, it is also possible to give it as a 3D tensor as shown in figure 1(b), where the dimensions are time, frequency and electrodes. Since we are using 1D convolution only in time axis, this is not necessary in our case. For the studies considering the 2D locations of electrodes, 4D tensors may be used as input requiring adjustment of the filters and convolutions accordingly. Consequently we used a CNN that has only one convolutional layer with 1D filtering and one pooling layer. By using this network, we are able to train filters to respond differently to different activation patterns along the vertical axis. We also take advantage of tolerance of MI data to localization in time by sweeping filters along the time axis.

Totally $N_{\text{F}} = 30$ filters with size $N_h \times 3$ are trained via this network. The proposed CNN structure is presented in figure 2. At the convolution layer, the input image is convolved with trainable filters and put through the output function f to form

the output map [32]. The kth feature map at a given layer is obtained as

$$h_{ij}^{k} = f(a) = f\left((W^{k} * x)_{ij} + b_{k}\right) \tag{1}$$

where x is the input image, W^k is the weight matrix for filter k and b_k is the bias value, for $k = 1, 2, \ldots, N_F$. In our case with 1D filtering with filter size $N_h \times 3$, j is equal to 1 and $i = 1, 2, .., N_t - 2$. The output function f is selected as rectified linear unit (ReLU) function. ReLU is approximated by softplus function defined as

$$f(a) = \mathrm{ReLU}(a) = \ln(1 + e^{a}) \tag{2}$$

where a is as defined in equation (1).

The output of convolutional layer has N_F vectors with $(N_t - 2) \times 1$ dimension. At the max-pooling layer, sampling factor 10 and zero padding is applied. Therefore, the output maps from the previous layer are subsampled into N_F vectors with 3×1 dimension. The max-pooling layer is followed by a fully connected layer having two outputs representing left and right hand MI. Parameters of the CNN are learned by using back-propagation algorithm. In this method, the labeled training set is fed to the network and the error E is computed considering the difference between the network output and the desired output. Then, gradient descent method is used to minimize this error E by changing network parameters as shown in equations (3)–(4) [33].

$$W^{k} = W^{k} - \eta \frac{\partial E}{\partial W^{k}} \tag{3}$$

$$b_{k} = b_{k} - \eta \frac{\partial E}{\partial b_{k}} \tag{4}$$

Here η denotes the learning rate of the algorithm, while W^k is the weight matrix for filter k and b_k is the bias value as defined previously. Finally, the trained network is used for classification of the new samples in the test set.

2.3. Stacked autoencoder (SAE)

An autoencoder (AE) is a network with one input layer, one hidden layer and one output layer [28]. The number of neurons in the output layer is equal to the number of neurons in the input layer. During training, the input x is first mapped to the hidden layer to produce hidden output y. Then, y is mapped to the output layer to produce z values. These two steps can be formulated as

$$y = f\left(W_y x + b_y\right) \tag{5}$$

$$z = f(W_z y + b_z) \tag{6}$$

where f is the activation function which is selected as

$$f(a) = 1/(1 + \exp(-a)) \tag{7}$$

W_y and W_z are the weights from input to hidden and hidden to output layers. b_y and b_z are bias values of hidden and output layers. If we set $W_y = W_z'$ where W_z' is the transpose of W_z and denote it as W, the weights are said to be tied. This helps to obtain the model parameters by minimizing the cost function E shown in equation (8).

$$\arg\min_{W, b_y, b_z} [E(x, z)] \tag{8}$$

Here $E(x, z)$ is the reconstruction error when the network is trained to reconstruct the output values equal to the applied input values. Then the model parameters can be updated using equations (9)–(11) [34].

$$W = W - \eta \frac{\partial E(x, z)}{\partial W} \tag{9}$$

$$b_y = b_y - \eta \frac{\partial E(x, z)}{\partial b_y} \tag{10}$$

$$b_z = b_z - \eta \frac{\partial E(x, z)}{\partial b_z} \tag{11}$$

Here η denotes the learning rate of the algorithm.

After an AE is trained, the learned features in the hidden layer can be used for classification or as input of the higher layer in a deep AE network, which is called stacked auto encoder (SAE).

SAE is composed of an input layer, several AEs and an output layer. Each AE layer is trained separately in an unsupervised manner and the output of the hidden layer in the previous AE is used as input of the next layer in the deep network. After this unsupervised pre-training step, supervised fine-tuning step is applied to learn the whole network parameters by using back-propagation algorithm [28]. The SAE network used in this study is shown in SAE part of figure 3. This model is composed of 1 input layer, 6 hidden layers and one output layer. The number of nodes at each layer is indicated at the top of the figure.

2.4. Combined CNN-SAE

The amplitude of recorded EEG signal is very low. Therefore, the signal is very sensitive to external and internal noises [4]. Artifacts like eye blinking and muscle movement are another source of disturbance that cause irrelevant effects that corrupts the desired brain pattern. Furthermore, some subjects are unable to perform successful MI tasks in some trials. These issues cause the input data to vary slightly among the trials. In order to overcome these problems, we propose a new deep structure, which includes a CNN followed by a SAE. The proposed model is shown in figure 3.

In this deep structure, firstly the CNN structure explained in section 2.2 is employed over the input data and the filters and network parameters are learned. Then the output of the convolutional layer of CNN is used as input to the SAE network. Input layer of SAE has 900 neurons that is the output of 30 neurons in the convolutional layer for each of 30 filters trained in CNN. By using this model, we aim to take advantage of extracting features considering time, frequency and location information of EEG data by using a CNN. We also try to improve the classification accuracy through the deep network in SAE part.

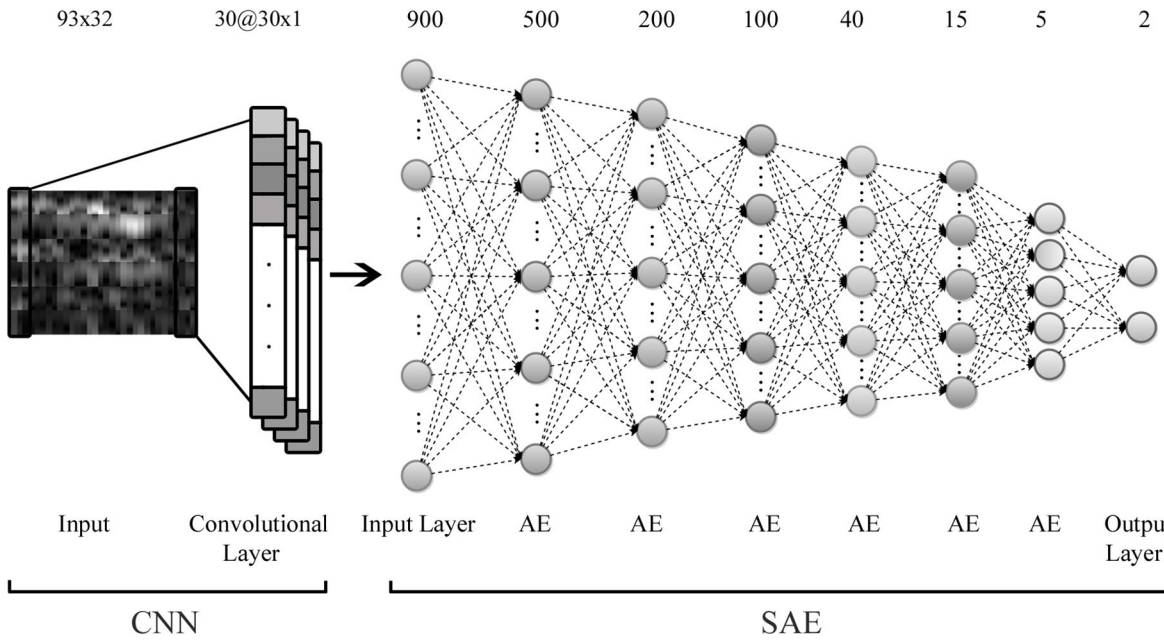

Figure 3. The proposed CNN-SAE network. Number of neurons in each layer is shown at top of the layer and the type of each layer is shown at the bottom.

Once training is completed, the computational complexity of CNN-SAE for testing is $O(N_h \times N_t \times N_F \times N_{Fs}) + O((N_t \times N_F)^2 \times N_L)$. The first term is the computational complexity of CNN part of the network, where $N_h \times N_t$ is the input image size, N_F is the number of convolutional filters and N_{Fs} is the filter size. The second term is the computational complexity of SAE part, where N_L is the number of layers; there is a square term in the complexity since SAE layers are fully connected.

Table 1. Dataset properties.

Dataset	Subjects	Channels	Trials	Rate (Hz)
Competition II dataset III	1	C3, Cz, C4	280	128
Competition IV dataset 2b	9	C3, Cz, C4	400	250

3. Datasets, results and discussion

In this section the datasets that are used in our experiments are explained and the results of the proposed approaches on these datasets are presented.

3.1. Datasets

To evaluate our method we used dataset III from BCI Competition II [29] and training set of dataset 2b from BCI Competition IV [31].

Both datasets include MI task experiments for right hand and left hand movements. The common goal of these competitions was to classify these MI tasks by using EEG signals recorded at C3, Cz and C4 channels. These datasets are very well known in this field and have been used in many similar studies. Properties of the datasets are summarized in table 1.

The BCI Competition IV dataset 2b includes three sessions in the training set. In the first two sessions, each trials starts with a short acoustic stimulus and a fixation cross. Then, at $t = 3$ s an arrow indicates the MI task. The arrow is displayed for 1.25 s. Then the subjects have 4 s to imagine the task. The third session includes online feedback with a smiley face indicating task's success. The timing of this session is similar to the previous sessions. In this study, the training set mentioned above is considered with 10×10 fold cross validation to measure the performance of each method.

During acquisition of the EEG signals in the BCI Competition II dataset III, at $t = 2$ s an acoustic stimulus indicating the beginning of the trial was used and a cross '+' was displayed for 1 s. Then, at $t = 3$ s, the subject was asked to perform the related MI task by displaying an arrow (left or right). Visual feedback was provided during the MI task.

3.2. Results

In this study, the experiments were conducted in Matlab environment on an Intel 4.00 GHz Core i7 PC with 16 GB of RAM. Matlab deep learning toolbox [35] was used for designing and testing the proposed networks. The toolbox was slightly changed in order to perform 1D convolution and max-pooling in CNN network. The algorithms were evaluated by using BCI Competition IV dataset 2b and BCI Competition II dataset III.

In BCI Competition IV dataset 2b, the classifiers were trained and tested separately for each subject. The performance of the proposed method was evaluated by accuracy and mean kappa value [32] metrics using 10×10 fold cross-validation. In this way, in each session, 90% of 400 trials were selected randomly as the training set and the remaining 10%

Table 2. BCI Competition IV dataset 2b accuracy (%) and standard deviation (std. dev) results for CNN, SAE and CNN-SAE methods.

	Accuracy % (mean ± std. dev.)			
Subjects	CNN	SAE	CNN-SAE	SVM
1	74.5 ± 4.6	61.0 ± 12.3	**76.0 ± 2.7**	71.8 ± 5.2
2	64.3 ± 2.0	47.5 ± 5.9	**65.8 ± 1.9**	64.5 ± 4.2
3	71.8 ± 1.6	45.0 ± 3.2	**75.3 ± 1.8**	69.3 ± 7.0
4	94.5 ± 0.2	81.3 ± 0.5	**95.3 ± 0.4**	93.0 ± 5.4
5	79.5 ± 2.5	62.8 ± 9.7	**83.0 ± 1.4**	77.5 ± 7.2
6	75.0 ± 2.4	47.8 ± 2.9	**79.5 ± 2.5**	72.5 ± 7.4
7	70.5 ± 2.3	53.8 ± 5.4	**74.5 ± 1.8**	68.0 ± 8.4
8	71.8 ± 4.1	63.8 ± 2.0	**75.3 ± 2.6**	69.8 ± 6.0
9	71.0 ± 1.1	56.5 ± 7.9	**73.3 ± 3.6**	65.0 ± 6.1
Average	74.8 ± 2.3	57.7 ± 5.5	**77.6 ± 2.1**	72.4 ± 5.7
Inter subject std. dev.	8.4	11.2	8.1	8.7

were selected as the test set. This process was repeated for 10 times. Similar to [8] for each EEG trial, we extracted the time interval between 0.5 and 2.5 s after the cue was displayed. Then STFT was applied to the extracted signal and the input was constructed as explained in section 2.1.

In CNN method, we used the network shown in figure 2 and described section 2.2, which has one convolutional and one max-pooling layers. The network was trained by using batch training method with batch size equal to 50 for 300 epochs. In SAE method, the input image shown in figure 1 was down-sampled and turned to a 900×1 vector in order to be used as input to the SAE network. A network with 6 hidden layers was used for classification as described in section 2.3. Each Autoencoder in SAE was trained for 200 epochs with batch size 20. Then fine tuning was applied for 200 epochs with batch size 40.

Filters and other network parameters (i.e. neuron bias values) learned by CNN method were used to compute the output values of the convolutional layer of the CNN network. Computed output values were then converted to vectors as explained in section 2.4 and used as the input of SAE network. The structure of the SAE part of the network and the training configuration for this method were the same as SAE method.

The accuracy results of CNN, SAE and CNN-SAE methods are presented in table 2. We also applied SVM classifier to the same input images in order to investigate the role of deep networks in classification performance. As you can see in table 2, the average accuracy results of CNN and CNN-SAE methods are higher than SVM with the same input data. Inter subject standard deviation of CNN-SAE method is also higher than SVM. The average accuracy of CNN-SAE is also higher than SAE method which emphasizes the role of CNN in extracting features.

Kappa results of our method is compared to current state of art studies in table 3. The kappa value is a measure for classification performance removing the effect of accuracy of random classification. Kappa is calculated as

$$\text{kappa} = \frac{\text{acc} - \text{rand}}{1 - \text{rand}} \tag{12}$$

In equation (12), acc is the classification accuracy and rand is the result of random classification which is 0.5 for two class classification. Mean kappa value of CNN and CNN-SAE methods are compared to FBCSP [8] (winner algorithm of the competition) and Twin SVM [36] methods in the left side of table 3. On the average, the FBCSP and Twin SVM approaches obtain Kappa values of 0.502 and 0.526 respectively whereas the average Kappa value is 0.547 for our CNN-SAE approach. This indicates that our method provides 9.0% improvement with respect to FBCSP and 4.0% improvement with respect to Twin SVM in terms of average Kappa value.

CNN-SAE method has the best kappa values for 5 out of 9 subjects. Our approach outperforms FBCSP [8] and Twin SVM [36] methods both for 6 out of 9 subjects. Standard deviation of kappa values for 10 sessions in 10×10 cross validation process is presented for CNN-SAE method in tables 2 and 3. The average standard deviation for kappa value of CNN-SAE method is 0.083. Even though this value is larger than average standard deviation of FBCSP (0.014), it is still a small value showing that CNN-SAE method is quite robust to changes in EEG recordings of the same subjects from session to session. Inter subject standard deviation values for CNN, CNN-SAE, FBCSP and twin SVM methods are also presented in table 3. Our proposed CNN-SAE method has the lowest inter subject standard deviation value among these methods. The corresponding standard deviation values are 0.0213 for FBCSP and 0.177 for Twin SVM, whereas it is 0.0161 for CNN-SAE, which demonstrates 24.4% improvement with respect to FBCSP and 9.0% improvement with respect to Twin SVM on the standard deviation of Kappa values across subjects. This is showing that CNN-SAE method is more robust to subject dependent differences than other methods. All these results show that CNN-SAE method provides more reliable classification with higher accuracy.

Some other studies only provide best case kappa values for dataset 2b. To compare our approach with these studies we also provide best case kappa values for CNN-SAE method. As it is presented in the right side of table 3, the average of best kappa value of our approach is higher than DDFBS [37], Bi-Spectrum [38] and RQNN [39] approaches. Our approach outperforms DDFBS and Bi-spectrum methods both for 6 out of 9 and RQNN for 5 out of 9 subjects. We also performed session-to-session classification on Competition IV dataset 2b, where sessions 1 and 2 with 240 trials were used to train the network and session 3 with 160 trials was used to test the network. Accuracy results of session-to-session classification are presented in table 4.

The average accuracy value for all subjects was 72.4% for CNN and 75.1% for CNN-SAE methods respectively. The average result of session-to-session classification is 2.3% lower for CNN and 2.5% lower for CNN than those given in table 3. Since the experimental protocol was different between first two sessions and session 3, reduction in the accuracy was expected. However, the results of session-to-session classification is not necessarily lower for all subjects. As you can see in table 4, the accuracy of CNN-SAE method for subject 9 is higher when using session-to-session classification. This may be due to large session-to-session variation in performance of subject 9. It is well known that, beside across subjects, BCI

Table 3. Kappa value and standard deviation (std. dev.) results of CNN and CNN-SAE methods compared with FBCSP [8], TWIN SVM [36], DDFBS [37], BI-SPECTRUM [38] and RQNN [39] methods.

	Mean kappa value (mean ± std. dev.)				Best kappa value			
Subject	CNN	CNN-SAE	FBCSP	Twin SVM	CNN-SAE	DDFBS	Bi-spectrum	RQNN
1	0.488 ± 0.158	0.517 ± 0.095	**0.546** ± 0.017	0.494	**0.738**	0.710	0.600	0.640
2	0.289 ± 0.068	0.324 ± 0.065	0.208 ± 0.028	**0.416**	0.458	0.310	0.310	**0.590**
3	0.427 ± 0.071	**0.494 ± 0.084**	0.244 ± 0.023	0.322	**0.845**	0.750	0.300	0.650
4	0.888 ± 0.008	**0.905 ± 0.017**	0.888 ± 0.003	0.897	**1.000**	0.470	0.980	0.990
5	0.593 ± 0.083	0.655 ± 0.060	0.692 ± 0.005	**0.722**	**0.750**	0.190	0.660	0.460
6	0.495 ± 0.073	**0.579 ± 0.099**	0.534 ± 0.012	0.405	**0.796**	0.200	0.610	0.510
7	0.409 ± 0.079	**0.488 ± 0.065**	0.409 ± 0.013	0.466	0.699	0.780	0.750	**0.810**
8	0.443 ± 0.133	**0.494 ± 0.106**	0.413 ± 0.013	0.477	0.751	0.770	**0.800**	**0.800**
9	0.415 ± 0.050	0.463 ± 0.152	**0.583** ± 0.010	0.503	0.550	0.730	0.760	**0.770**
Average	0.494 ± 0.080	**0.547 ± 0.083**	0.502 ± 0.014	0.526	**0.732**	0.546	0.641	0.691
Inter subject std. dev.	0.169	**0.161**	0.213	0.177				

Table 4. Accuracy results of session-to-session classification for BCI competition IV dataset 2b.

	Accuracy %		
Subjects	CNN	SAE	CNN-SAE
1	76.3	57.5	78.1
2	60.0	58.1	63.1
3	56.3	50.6	60.6
4	95.6	94.4	95.6
5	79.4	75.0	78.1
6	65.6	67.5	73.8
7	65.6	76.2	70.0
8	70.6	75.6	71.3
9	82.5	78.1	85.0
Average	72.4	70.3	75.1

performance is also inconsistent within subjects and fluctuates greatly over time, which is a critical problem in BCI studies [40]. We tried to remove the effect of within subject variation on our results by using 10 × 10 cross validation in the previous test.

In order to evaluate our methods on another dataset, we used the same networks described before to classify data from BCI Competition II dataset III. The input images were generated as explained in section 2.1 with window size equal to 32 and time lapses equal to 7. Networks were trained with 140 trials in the training set and tested on 140 trials in the test set. These training and test sets were fixed and provided by the competition organizers.

The results for BCI Competition II dataset III are shown in table 5. The accuracy of the winner algorithm of the competition is 89.3% [41]. The accuracy performance of CNN-SAE algorithm was obtained as 90.0%, which is better than the winner algorithm of the competition. Since the winner algorithm was applied long time ago, we also compared our results to a recent study [23] where a deep learning network is used for feature extraction. The accuracy result of study [23] is 88.2%. The accuracy results of CNN and SAE methods are 89.3% and 60% respectively.

Table 5. BCI competition II dataset III accuracy (%) results for CNN, SAE, CNN-SAE, the winner algorithm [41] and deep network [23].

Method	CNN	SAE	CNN-SAE	[41]	[23]
Accuracy %	89.3	60.0	90.0	89.3	88.2
Kappa	0.786	0.200	0.800	0.783	0.764

3.3. Discussion

As described in [40, 42], certain subjects may face difficulties in performing MI task, where others perform it successfully. This causes a large difference between classification performances of subjects.

The difference of kappa value between best subject (subject 4, kappa = 0.905) and worst subject (subject 2, kappa = 0.324) is 0.581 when CNN-SAE method is used. This value is 0.680 for the method proposed in the winner algorithm (FBCSP) of the competition [8]. Furthermore, inter subject standard deviation value for CNN-SAE is 0.161 which is lower than FBCSP and Twin SVM. This is showing that CNN-SAE overcomes difficulties related to subject dependent disturbances by providing subject specific training, which is an important issue in BCI applications.

The average of input images of right and left hand MI classes are shown in figure 4 for subjects 2 and 4. The difference between these two classes is quite clear on these average images for subject 4. While there is a decrease of activation in C4 area and increase in C3 area for left hand MI and it is just reverse for the right hand MI. However, average images of these classes are not that different for subject 2. Input images are designed to represent time, frequency and location information from EEG signals. By using 1D convolution (in the horizontal axis) in CNN method, filters are trained to learn the location of the activation in the vertical axis.

The set of filters of size $N_h \times 3$ learned in the convolution layer of CNN is presented in figure 5. For each filter, there are $N_h = 93$ components along vertical axis while there are

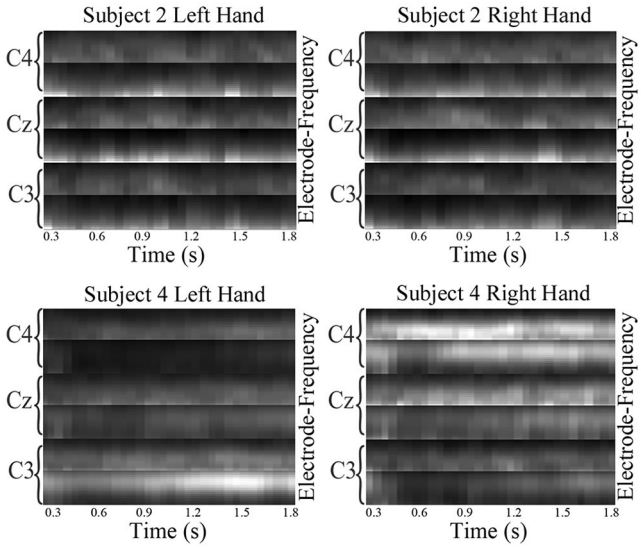

Figure 4. Average of input images for right and left hand classes and subjects 2 and 4.

3 components along horizontal axis. These filters are computed in a training session for subject 4. Although, these filter weights do not necessarily have any semantic meaning by themselves, multiple filters may be thought as the basis vectors of some space that represents image patches. As it can be seen in figure 5, some filters such as 7, 9, 12 and 14 are learned to detect activation in specific regions along the vertical axis. This corresponds to difference in activation of C3 and C4 electrodes related to the left and right parts of the motor cortex of the brain. Some other filters such as 3, 8 and 10 are learned to detect more detailed information.

In the pooling layer, max-pooling is resulted in a considerable performance improvement compared to average pooling. In our application, activation of any location in image (higher intensity value) is an important factor for classification. Averaging the outputs of convolutional layer may decrease the effect of these activations.

The effect of epoch size on performance (kappa) is shown in figure 6 together with the average training time (for one training set with 360 trials). As you can see in the figure, the performance is higher than others for 300 epochs with an appropriate computation time (1157 s). Number of epochs is selected as 300 in this study. The size of filters in the convolutional layer of CNN also change the performance of classification. A comparison between performances of different filter sizes is presented in table 6. Since the performance decreases considerably by increasing the filter size, the results of larger filters are not investigated. Notice that N_h is fixed due to 1D filtering and it has the value which is the height of the input images. As it is shown in table 6, a convolutional layer with filter size equal to $N_h \times 3$ obtains best classification performance. The performance of $N_h \times 2$ filter is very close to $N_h \times 3$ filter. However, in our experiments, the performance of CNN-SAE network was higher with $N_h \times 3$ filter. This means that the features are presented better by using this filter size.

The sampling factor at the max-pooling layer was selected as 10 for all experiments and zero-padding method is applied. We saw that a high sampling factor (i.e. 10) results in better

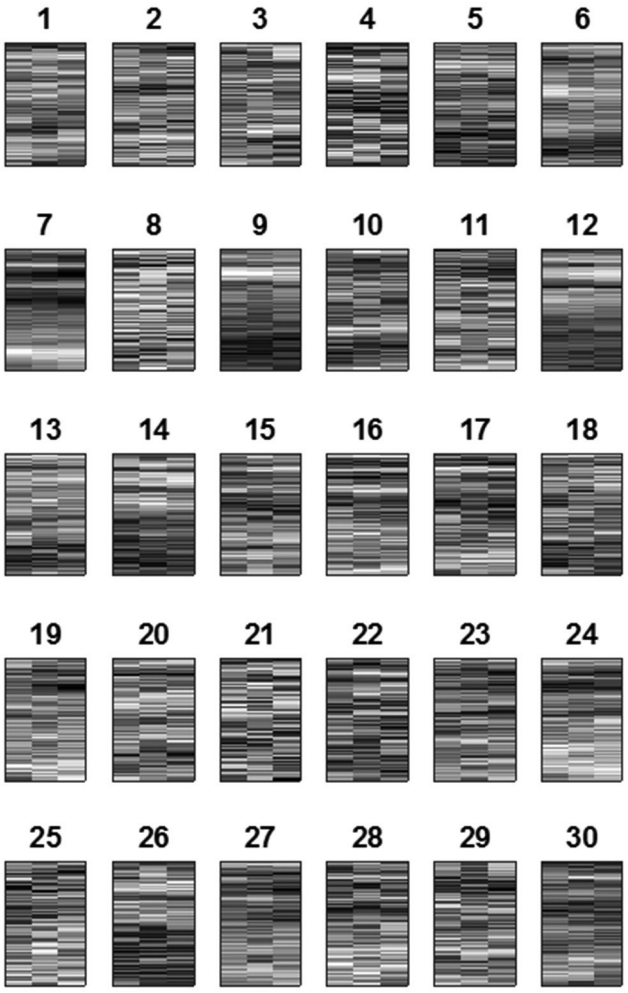

Figure 5. Weight values W^k $k = 1, 2, \ldots, N_{F0} = 30$, of filters of size $N_h \times 3$ learned by CNN for subject 4. For each filter there are $N_h = 93$ components along vertical axis while there are 3 components along horizontal axis.

classification. This means that the effect of time information (horizontal axis) is reduced and the effect of location information (vertical axis) is emphasized in our input data. Input vectors of SAE method are vectorised versions of our input images. Therefore using this method loses the neighborhood information. This causes a great amount of performance decrease compared to CNN method. The results of SAE were presented in table 2 in order to be compared with CNN-SAE method and investigate the role of CNN part of that method.

The proposed CNN_SAE framework takes advantage of multi domain (time, frequency and electrode location) feature extraction characteristic of CNN model as well as deep learning benefit of SAE method. In SAE part, the size of hidden layers are selected to be less than the size of the previous layer but not less than 25%. In our experiments, the performance was not changing considerably after 200 epochs for pre-training and fine-tuning.

One training session with 360 trials takes about 1157 s for CNN-SAE method. However, testing is almost immediate with about 400 ms duration. The computational time for training the same amount of trials using FBCSP [8] method was 246 s in our application. Also the time needed for testing the same

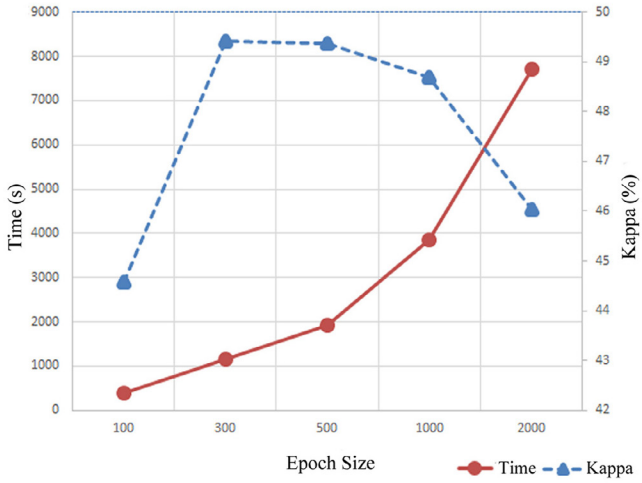

Figure 6. Effect of number of epoch on kappa value and training time.

Table 6. Effect of filter size on CNN performance. $N_h = 93$.

Filter size	Accuracy (%)	Kappa (%)	Time (s)
$N_h \times 1$	73.4	46.5	785
$N_h \times 2$	74.8	49.2	972
$\boldsymbol{N_h \times 3}$	**74.8**	**49.4**	1157
$N_h \times 4$	70.9	41.1	1401
$N_h \times 5$	70.4	40.4	1653

amount of trials was 508 ms. Our system performs faster than FBCSP in testing state where the training time of our method is higher. Since BCI systems are usually being trained prior to operation and perform testing in real time, testing time is much more critical than the training time for these systems.

4. Conclusions

In this study, three different neural network models, one of which is proposed in this study, are used to classify EEG motor imagery (MI) signals into right and left hand movement classes. Convolutional neural network (CNN), stacked autoencoder (SAE) and combined CNN-SAE methods are applied to two different MI datasets. Dataset III from BCI Competition II and training set of dataset 2b from BCI Competition IV are used in this study.

We designed a new input form by using time, frequency and location information of EEG signals. These inputs are used in a CNN network with one 1D convolutional and one max-pooling layers. The filters in the convolutional layer of CNN are being trained to learn the location of activation patterns as well as the frequencies while they are partially invariant to placement in time axis due to max-pooling. In the experiments, the effect of filter size was investigated and $N_h \times 3$ filter yielded the best performance. We also investigated epoch size for CNN training and the best value for epoch size was found as 300.

Our experiments showed that, performance of SAE method is poor by using this input form. This poor result is because of losing neighborhood information in SAE input layer.

A new combined network is proposed in this study by using CNN and SAE models. In this network, features of input EEG signal are first extracted by training the filters in convolutional layer of CNN. Then these features are used in a deep SAE network to perform classification. We showed that, by using this network, the average kappa value is 0.547. The kappa value of the winner algorithm of competition IV was 0.502, so kappa value is improved about 9% by using our approach. We also compared our results to state of art studies with different evaluation methods. To our knowledge, our approach yields the best accuracy performance on BCI competition IV dataset 2b.

The accuracy result of CNN-SAE method on dataset III is 90.0% that is higher than the winner algorithm of competition III. Our proposed network also obtains better performance results than the deep network introduced in [23] for feature extraction.

Notice that, the filters learned in CNN takes into account the activation in the neighboring regions but CNN does not provide any information on which filter contributes to classification performance more than the others. Instead, pooling is applied on the next layer. On the other hand, the neighboring information is lost in the SAE, but the weight values provide information to the network on the importance of the input elements. However, when the outputs of the filters trained by CNN are given as input to SAE, the advantages of each network are unified and the performance increases.

Another point that should be noticed is the high performance increase for the subjects having poor performance in the winner algorithm [8] of BCI competition IV dataset 2b. This is showing that the proposed algorithm provides subject specific training, which is an important issue in BCI applications.

The proposed network performs fast classification with a higher performance than other methods in literature by a training even using only a few samples, which is 360 trials in this study. The time required for classification of a single trial is only 0.4 s even in Matlab. This is an important advantage for BCI applications requiring real-time classifications for daily life use.

Lack of large amount of data for medical images makes deep learning methods less preferable for medical applications. However, in BCI applications this problem can be solved by using data gathered through daily life use and the system may be trained continuously. The result of our proposed system can be improved by using large datasets in such BCI system.

In this study we used a CNN having only one convolution and one pooling layer. The performance of the network may be increased if further convolution-pooling layers are used. However, for this purpose 1D filters special to these further layers should be designed.

The proposed method was evaluated by using datasets with $N_c = 3$ electrodes. However it can be extended to use larger number of electrodes. The input image can be constructed by aligning data from any number of electrodes. In this case, the size of the filters in the convolutional layer of the network should be changed in order to fit the input image. For any number of N_c, the input image would become $(N_c * 31) \times 32$ as described in section 2.1. Since convolution is applied at only

one dimension (time axis with 32 sample points), the filter size for the first level would be $(N_c * 31) \times 3$ as explained in section 2.2. It should be noted that, by increasing the size of filters, the number of variables in the network also increases and larger amount of data may be needed in order to train the network.

Currently, the amount of studies employing deep learning methods for BCI applications is very limited. We believe that further BCI studies using methods similar to our proposed method, will make deep learning methods a proper choice for BCI applications.

Acknowledgments

This study was partially supported by projects METU BAP-03-01-2015-001 and METU BAP-03-01-2016-003. The authors would like to thank Prof Dr Canan Kalaycıoğlu from Ankara University Brain Research Center for helpful discussion and suggestions for improvement of the manuscript.

References

[1] Graimann B, Allison B and Pfurtscheller G 2010 *Brain-Computer Interfaces: A Gentle Introduction* (Berlin: Springer) pp 1–27
[2] Kübler A, Furdea A, Halder S, Hammer E M, Nijboer F and Kotchoubey B 2009 A brain-computer interface controlled auditory event-related potential (P300) spelling system for locked-in patients *Ann. N. Y. Acad. Sci.* **1157** 90–100
[3] Krepki R, Blankertz B, Curio G and Müller K-R 2007 The Berlin brain-computer interface (BBCI)—towards a new communication channel for online control in gaming applications *Multimed. Tools Appl.* **33** 73–90
[4] Nicolas-Alonso L F and Gomez-Gil J 2012 Brain computer interfaces, a review *Sensors* **12** 1211–79
[5] Pfurtscheller G and Da Silva F L 1999 Event-related EEG/MEG synchronization and desynchronization: basic principles *Clin. Neurophysiol.* **110** 1842–57
[6] Müller-Gerking J, Pfurtscheller G and Flyvbjerg H 1999 Designing optimal spatial filters for single-trial EEG classification in a movement task *Clin. Neurophysiol.* **110** 787–98
[7] Grosse-Wentrup M and Buss M 2008 Multiclass common spatial patterns and information theoretic feature extraction *IEEE Trans. Biomed. Eng.* **55** 1991–2000
[8] Ang K K, Chin Z Y, Wang C, Guan C and Zhang H 2012 Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b *Front. Neurosci.* **6** 39
[9] Ramoser H, Muller-Gerking J and Pfurtscheller G 2000 Optimal spatial filtering of single trial EEG during imagined hand movement *IEEE Trans. Rehabil. Eng.* **8** 441–6
[10] Mousavi E A, Maller J J, Fitzgerald P B and Lithgow B J 2011 Wavelet common spatial pattern in asynchronous offline brain computer interfaces *Biomed. Signal Process. Control* **6** 121–8
[11] Comon P 1994 Independent component analysis, a new concept? *Signal Process.* **36** 287–314
[12] Jolliffe I 2002 *Principal Component Analysis* (New York: Wiley) (doi: 10.1002/9781118445112.stat06472)
[13] Guo X, Wang L, Wu X and Zhang D 2008 *Dynamic Analysis of Motor Imagery EEG Using Kurtosis Based Independent Component Analysis. Advances in Cognitive Neurodynamics ICCN 2007* (Amsterdam: Springer) pp 381–5
[14] Talukdar M T F, Sakib S K, Pathan N S and Fattah S A 2014 Motor imagery EEG signal classification scheme based on autoregressive reflection coefficients *2014 Int. Conf. on Informatics, Electronics & Vision (ICIEV)* (IEEE) pp 1–4
[15] Ince N F, Arica S and Tewfik A 2006 Classification of single trial motor imagery EEG recordings with subject adapted non-dyadic arbitrary time-frequency tilings *J. Neural Eng.* **3** 235
[16] Cortes C and Vapnik V 1995 Support-vector networks *Mach. Learn.* **20** 273–97
[17] Fukunaga K 2013 *Introduction to Statistical Pattern Recognition* (New York: Academic)
[18] Jensen F V 2001 *Bayesian Networks and Decision Graphs* (New York: Springer) p 34
[19] Schlögl A, Lee F, Bischof H and Pfurtscheller G 2005 Characterization of four-class motor imagery EEG data for the BCI-competition *J. Neural Eng.* **2** L14
[20] An X, Kuang D, Guo X, Zhao Y and He L 2014 A deep learning method for classification of EEG data based on motor imagery *Intelligent Computing in Bioinformatics* (Berlin: Springer) pp 203–10
[21] Wulsin D, Gupta J, Mani R, Blanco J and Litt B 2011 Modeling electroencephalography waveforms with semi-supervised deep belief nets: fast classification and anomaly measurement *J. Neural Eng.* **8** 036015
[22] Ahmed S, Merino L M, Mao Z, Meng J, Robbins K and Huang Y 2013 A deep learning method for classification of images rsvp events with eeg data *Global Conf. on Signal and Information Processing (GlobalSIP), 2013 IEEE* (IEEE) pp 33–6
[23] Ren Y and Wu Y 2014 Convolutional deep belief networks for feature extraction of EEG signal *2014 Int. Joint Conf. on Neural Networks (IJCNN)* (IEEE) pp 2850–3
[24] Yang H, Sakhavi S, Ang K K and Guan C 2015 On the use of convolutional neural networks and augmented CSP features for multi-class motor imagery of EEG signals classification *2015 37th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society (EMBC)* (IEEE) pp 2620–3
[25] Bashivan P, Rish I, Yeasin M and Codella N 2015 Learning representations from EEG with deep recurrent-convolutional neural networks *ICLR 2016* (arXiv:1511.06448)
[26] Jirayucharoensak S, Pan-Ngum S and Israsena P 2014 EEG-based emotion recognition using deep learning network with principal component based covariate shift adaptation *Sci. World J.* **2014** 627892
[27] LeCun Y, Bottou L, Bengio Y and Haffner P 1998 Gradient-based learning applied to document recognition *Proc. IEEE* **86** 2278–324
[28] Bengio Y, Lamblin P, Popovici D and Larochelle H 2007 Greedy layer-wise training of deep networks *Adv. Neural Inform. Process. Syst.* **19** 153
[29] Leeb R, Lee F, Keinrath C, Scherer R, Bischof H and Pfurtscheller G 2007 Brain-computer communication: motivation, aim, and impact of exploring a virtual apartment *IEEE Trans. Neural Syst. Rehabil. Eng.* **15** 473–82
[30] Cohen J 1960 A coefficient of agreement for nominal scales *Educ. Psychol. Meas.* **20** 37–46
[31] Schlögl A 2003 *Outcome of the BCI-Competition 2003 on the Graz Data Set* (Berlin: Graz University of Technology)
[32] Le Cun B B, Denker J S, Henderson D, Howard R E, Hubbard W and Jackel L D 1990 Handwritten digit recognition with a back-propagation network *Advances in Neural Information Processing Systems* (San Francisco, CA: Morgan Kaufmann Publishers Inc.)
[33] LeCun Y A, Bottou L, Orr G B and Müller K-R 2012 Müller K-R 2012 Efficient BackProp *Neural Networks: Tricks of the Trade* (Berlin: Springer) pp 9–48
[34] Ng A 2011 Sparse autoencoder *CS294A Lect. Notes Stanford University* **72** 1–19

[35] Palm R B 2012 *Prediction as a Candidate for Learning Deep Hierarchical Models of Data MS Thesis* Technical University of Denmark
[36] Soman S 2015 High performance EEG signal classification using classifiability and the Twin SVM *Appl. Soft Comput.* **30** 305–18
[37] Suk H-I and Lee S-W 2011 Data-driven frequency bands selection in EEG-based brain-computer interface *2011 Int. Workshop on Pattern Recognition in NeuroImaging (PRNI)* (IEEE) pp 25–8
[38] Shahid S, Sinha R K and Prasad G 2010 A bispectrum approach to feature extraction for a motor imagery based brain-computer interfacing system *Proc. 18th European Signal Processing Conf. EUSIPCO* pp 1831–5
[39] Gandhi V, Arora V, Behera L, Prasad G, Coyle D and McGinnity T M 2011 EEG denoising with a recurrent quantum neural network for a brain-computer interface *The 2011 Int. Joint Conf. on Neural Networks (IJCNN)* (IEEE) pp 1583–90
[40] Ahn M and Jun S C 2015 Performance variation in motor imagery brain-computer interface: a brief review *J. Neurosci. Methods* **243** 103–10
[41] Lemm S, Schäfer C and Curio G 2004 BCI competition 2003-data set III: probabilistic modeling of sensorimotor μ rhythms for classification of imaginary hand movements *IEEE Trans. Biomed. Eng.* **51** 1077–80
[42] García-Laencina P J, Rodríguez-Bermudez G and Roca-Dorda J 2014 Exploring dimensionality reduction of EEG features in motor imagery task classification *Expert Syst. Appl.* **41** 5285–95