A Deep Learning Scheme for Motor Imagery Classification based on Restricted Boltzmann Machines

Na Lu, Tengfei Li, Xiaodong Ren, and Hongyu Miao

***Abstract*— Motor imagery classification is an important topic in brain–computer interface (BCI) research that enables the recognition of a subject's intension to, e.g., implement prosthesis control. The brain dynamics of motor imagery are usually measured by electroencephalography (EEG) as nonstationary time series of low signal-to-noise ratio. Although a variety of methods have been previously developed to learn EEG signal features, the deep learning idea has rarely been explored to generate new representation of EEG features and achieve further performance improvement for motor imagery classification. In this study, a novel deep learning scheme based on restricted Boltzmann machine (RBM) is proposed. Specifically, frequency domain representations of EEG signals obtained via fast Fourier transform (FFT) and wavelet package decomposition (WPD) are obtained to train three RBMs. These RBMs are then stacked up with an extra output layer to form a four-layer neural network, which is named the frequential deep belief network (FDBN). The output layer employs the softmax regression to accomplish the classification task. Also, the conjugate gradient method and backpropagation are used to fine tune the FDBN. Extensive and systematic experiments have been performed on public benchmark datasets, and the results show that the performance improvement of FDBN over other selected state-of-the-art methods is statistically significant. Also, several findings that may be of significant interest to the BCI community are presented in this article.**

***Index Terms*— Brain–computer interface (BCI), deep learning, motor imagery, restricted Boltzman machine (RBM).**

I. Introduction

BRAIN–computer interface (BCI) has attracted increasing attention of a variety of research communities, including neuroscience, neuroimaging, rehabilitation medicine, pattern recognition, signal processing, machine learning, and so on [1]. One of the important goals of BCI research is to restore regular functions for people with severe neuromuscular disabilities or enhance certain functions for healthy person via a new signal pathway [2]. Motor imagery is an important research topic in the field of BCI that mentally simulates a given action, e.g., imaging the motions of the limbs [3]. Specifically, an exogenous event like visual cue is employed to trigger the mental simulation, which would induce event related synchronization (ERS) and event related desynchronization (ERD) simultaneously in the sensorimotor rhythms at different positions over the scalp. These phenomena could be experimentally observed through various brain activity measuring techniques. The most popular technologies to record such brain signals is electroencephalography (EEG) that is non-invasive and comparatively easy to operate [2]. However, the EEG recordings are usually of low signal-to-noise ratio (SNR) due to the volume-conduction effect, which makes it very challenging to accurately understand brain dynamics and classify different motor imageries [4]. Some interesting applications about motor imagery have been performed, e.g., 2D cursor control [5], wheelchair control [6], and quadcopter control [7], and so on, which are based on motor imagery classification. Therefore, to make efficient use of motor imagery, great efforts have been made for motor imagery feature extraction and classification in the past decade [8]–[11].

Existing motor imagery classification methods can be categorized into five types [8]: 1) linear classifiers with linear discriminant analysis (LDA) and support vector machine (SVM) being the representative ones; 2) nonlinear Bayesian classifiers such as Bayes quadratic and Hidden Markov Model (HMM); 3) nearest neighbor classifiers like k nearest neighbors (KNN); 4) neural network methods such as multilayer perceptron (MLP) [12] and Radial Basis Function (RBF) neural network [13]; 5) combinations of different classification methods by boosting or voting. For most of the classification methods above, feature extraction from EEG time series needs to be conducted first using methods like independent component analysis (ICA) [9], nonnegative matrix factorization (NMF) [14], [15], and empirical mode decomposition (EMD) [16], etc. However, such computing procedures may associate with a heavy computational burden, which could render themselves unsuitable for certain tasks (e.g., online processing). Neural network methods adopt a different paradigm by combining feature extraction and classification into one pipeline. The training stage of neural network may require a long period of time, but once the neural network has been

Manuscript received October 28, 2015; revised March 20, 2016 and June 16, 2016; accepted August 15, 2016. Date of publication August 17, 2016; date of current version June 18, 2017. This work is supported by Fundamental Research Funds for the Central Universities, National Natural Science Foundation of China under Grant 61105034 and Grant 61673312, Research Fund for the Doctoral Program of Higher Education of China under Grant 20100201120040, and the China Postdoctoral Science Foundation under Grant 20110491662, 2012T50805.

N. Lu, T. Li, and X. Ren are with the State Key Laboratory for Manufacturing Systems Engineering, Systems Engineering Institute, Xi'an Jiaotong University, Xi'an Shaanxi 710049, China (e-mail: lvna2009@mail.xjtu.edu.cn).

H. Miao is with the Department of Biostatistics, School of Public Health, University of Texas at Houston, Houston, TX 77030 USA.

Digital Object Identifier 10.1109/TNSRE.2016.2601240

trained, the resulted parameters could be directly applied to new data, which can be computationally more efficient and more suitable for certain problems (e.g., real-time learning of unlabeled EEG data).

For traditional neural network methods, the initial weights need be chosen carefully, which is one major obstacle of their broader application [17]. Specifically, large initial values of the weights could lead to poor local minima, while small values could make the multilayer network untrainable due to weight diffusion [17]. To address this problem and construct neural networks with high descriptive power, a new category of strategies and methods, called deep learning, has been recently developed and become prevailing in both academia and industry [18], [19]. In the deep learning scheme, restricted Boltzmann machine (RBM) and autoencoder are the basic building blocks [20], which are trained in a layer-by-layer manner and can be used to construct deep neural networks. Contrastive divergence (CD) is specifically developed to train a RBM based on the Gibbs sampling theory [17]. The features extracted by RBM or autoencoder in the pretraining stage are employed to initialize the multilayer neural network, which may achieve notable performance improvement as demonstrated by many previous studies [17], [18], [21], [22]. Based on the pretraining results, a fine-tuning of the weights in the deep neural network is performed via error backpropagation. In view of its success in many scientific fields [23], the deep learning strategy can also be a promising solution for motor imagery classification. Some efforts have been made to apply related methods into BCI research (see Section II for details). *However, the development and application of deep learning methods in the BCI field is still quite rare*.

In this study, a RBM based deep learning scheme for motor imagery is developed. Considering the frequency feature of sensorimotor rhythms (SMR) is the major characteristic of motor imagery related brain activity, frequency domain representation obtained through FFT or WPD is thus used to pretrain the RBMs and fine-tune the deep belief neural network. Only bandpass filtering and data normalization are employed to preprocess the EEG data, and no artifact noise removing is conducted. In the pretrainging stage, unsupervised training of each layer of RBM is performed by part or the whole dataset for performance comparison. Three layers of pretrained RBMs are stacked together with an output layer of softmax regression added to form the deep neural network, which is called Frequential Deep Belief Network (FDBN). The specific structure of each layer is determined through experiments. The training error is backpropagated across the multilayer network to fine-tune the connection weights. In addition, the fine-tuning stage has been further divided into two phases. In the first phase, only the output layer is trained; in the second phase, all the layers are fine-tuned. Stochastic binary neurons are employed in the pretraining stage, while deterministic and real-valued probabilities are used at the fine-tuning stage. Extensive experiments on multiple subjects, different parameter and structure settings of the neural networks have been conducted, which verify the effectivity of the proposed deep learning scheme for motor imagery classification. The contributions of the paper are summarized below.

1) Frequency domain feature rather than time domain feature has been employed as the input to train a DBN for motor imagery classification, which has achieved a significant performance improvement compared with other state-of-the-art methods.
2) Experiments on BCI competition benchmark has been conducted, which has provided a feasibility study for application of deep learning method in the field of brain signal analysis.
3) Extensive and systematic experiments have been performed and some insightful observations have been made.

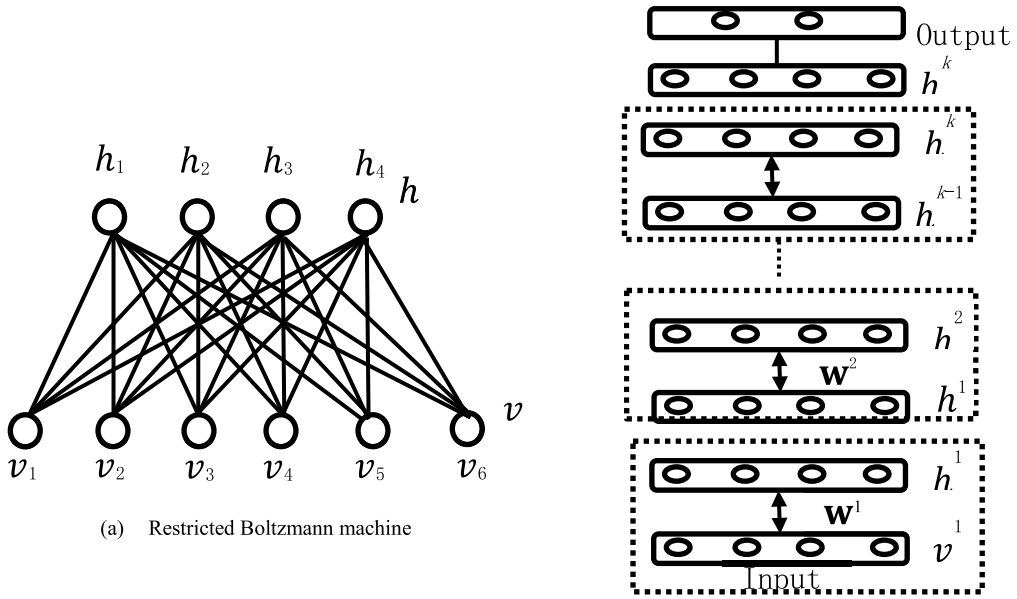
a) The experiment results suggest that session-to-session transfer of the EEG data for the same subject is quite effective, but subject-to-subject transfer is not.
b) It can be concluded that for motor imagery classification, the time localization of the frequency component is less efficient than the frequency component itself. Therefore, FFT turns out to be a better selection than WPD.
c) Experiments have also shown the robustness of FDBN to network structure change on about 20 times of scale.

The paper is organized as follows. Section I is the introduction. Section II describes the related works. Section III presents the FDBN scheme for motor imagery classification; experiments and discussions are given in Section IV. Conclusions can be found in Section V.

II. Related Works

The conventional methods for brain activity classification have been briefly described in Section I, so here mainly the related deep learning approaches are discussed in detail.

To the best knowledge of the authors, a very few studies have considered the deep learning approach for BCI research. Li et al. proposed to classify incomplete EEG [24] with denoising autoencoder based on spectral power features, which is of very low dimension (i.e., 16). However, no systematic experiments on algorithm performance with considering different network settings have been conducted, and no benchmark dataset has been evaluated in [24]. A deep learning method based on deep belief net (DBN) and Ada-boost was developed by An et al. [25], which trained a DBN for each EEG channel and combined these channels by boosting. Time domain data is fed to each DBN directly for training, and no benchmark dataset was tested in [25], too. Wang et al [26] developed a prior supervised convolutional stacked auto-encoder (PCSA) for ECoG classification, which employed the label information in the pretraining stage. Freudenburg et al. [27] employed DBN for ECoG data correlation analysis and verified its superiority to PCA. Jirayucharoensak et al [28] proposed an EEG-based emotion recognition method based on stacked autoencoders in combination with power spectral density and principle component analysis (PCA). A deep learning method for detecting target images in image rapid serial visual presentation (RSVP) task based on EEG measurement is developed in [29]. The input to the deep neural network is uncorrelated discriminant features obtained by linear discriminant analysis (LDA) and area under the ROC curve (AUC).



(a) Restricted Boltzmann machine

(b) Deep belief network

Fig. 1. Illustration of restricted Boltzmann machine and deep belief network.

Yang et al [30] have combined convolutional neural network and augmented common spatial filter for motor imagery classification. A better performance has been obtained as compared to filter-bank common spatial filter (FBCSP). A DBN solution using raw EEG data to detect epileptiform discharges and seizure-like activity was developed by Wulsin et al. [31]. It is claimed in [31] that raw data input is effective for EEG pattern recognition, which could achieve a classification performance comparable to the conventional approaches.

An interesting but unanswered question is thus how the deep learning idea can be adapted for motor imagery classification to achieve a notable performance improvement. Our study addresses this problem by introducing a novel deep learning strategy in the frequency domain, and our results do suggest the significant potential of deep learning in BCI.

III. Method

Deep Belief Network (DBN) is one of the representative deep learning methods that can be constructed by stacked autoencoders or restricted Boltzmann machines. Without loss of generality, RBM is selected to construct the DBN in this study. The training process of DBN consists of two phases, pretraining of each RBM and fine tuning of the stacked RBMs together with the softmax regression layer. During the pretraining stage, each unidirectional RBM is trained separately in an unsupervised manner. The pretraining aims at reconstructing the input at the output layer with no label information provided. The input to the RBM is the EEG time series or frequency feature obtained via FFT or WPD of the motor imageries. The final DBN is fine-tuned in a directional manner using the conjugate gradient method. Fine-tuning is conducted in a supervised way with labeled data. At the early stage of the fine-tuning, only the weights connected to the output layer will be adjusted; and after certain number of epochs, the weights of all layers will be tuned sequentially. The obtained parameters can be directly applied to the incoming new data, which enables efficient data classification and is thus suitable for online BCI.

A. Restricted Boltzmann Machine

Restricted Boltzmann machine (RBM) works in an unsupervised manner, which consists of one visible and one hidden layers [17], [20]. The input is fed to the visible layer, and the hidden layer aims at reconstructing the input as close as possible. The visible layer is connected to the hidden layer, while there is no connection between the neurons within the same layer. The neurons in both the visible and hidden layers are stochastic binary units. An illustration of an individual RBM is given in Fig. 1(a). The energy of certain joint configuration of the two layers is given by

$$E(\mathbf{v},\mathbf{h}) = -\sum_{i=1}^{m} a_i v_i - \sum_{j=1}^{n} b_j h_j - \sum_{i=1}^{m}\sum_{j=1}^{n} v_i h_j w_{ij} \quad (1)$$

where v_i and h_j are the binary state of the corresponding units, a_i and b_j are the biases, w_{ij} is the weight of the connection between visible unit i and hidden unit j. Based on the definition of the energy function, a probability of the joint presence of the visible and hidden layer is defined as

$$p(\mathbf{v},\mathbf{h}) = \frac{1}{Z} e^{-E(\mathbf{v},\mathbf{h})} \quad (2)$$

where $Z = \sum_{\boldsymbol{v},\mathbf{h}} e^{-E(\boldsymbol{v},\mathbf{h})}$ is the partition function. Accordingly, the probability of the visible vector being assigned as $\mathbf{v}$ can be obtained through summing over all possible hidden vectors as follows:

$$p(\mathbf{v}) = \frac{1}{Z}\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}. \quad (3)$$

Considering that there are no direct connections between the hidden units, the conditional probability of the binary state of unit h_j being set to 1 given visible vector $\mathbf{v}$ can be calculated as

$$p\left(h_j = 1 \middle| \mathbf{v}\right) = \sigma\left(b_j + \sum_i v_i w_{ij}\right) \tag{4}$$

where $\sigma(\cdot)$ is the sigmoid function. Also, given a hidden vector $\mathbf{h}$, the probability of the visible unit being 1 could be obtained as

$$p\left(v_i = 1 | \mathbf{h}\right) = \sigma\left(a_i + \sum_j h_j w_{ij}\right). \tag{5}$$

To train a generative model based on training samples, minimization of a log-likelihood could be considered. Specifically, the log-likelihood could be defined as

$$\mathcal{L}\left(D_{train}\right) = \sum \log p\left(\boldsymbol{v}, \mathbf{h}\right) \tag{6}$$

where D_{train} is the training dataset. To maximize the co-occurrence probability of certain pair of $\mathbf{v}$ and $\mathbf{h}$, the gradient direction of the presence probability of a training vector with respect to the connection weight $\mathbf{w}$ need be calculated. According to the definition given in (2) and (6), and based on the theory of contrastive divergence that maximizing the log-likelihood of the data is equivalent to minimizing the Kullback-Leibler divergence between the data distribution and the equilibrium (model) distribution [20], the gradient of the log-likelihood with respect to $\mathbf{w}$ can be obtained equivalently as

$$\frac{\partial \log(p(\mathbf{v}, \mathbf{h}))}{\partial w_{ij}} = \left\langle \frac{\partial \log \mathcal{L}\left(D_{train}\right)}{\partial w_{ij}} \right\rangle_{data} - \left\langle \frac{\partial \log \mathcal{L}\left(D_{train}\right)}{\partial w_{ij}} \right\rangle_{model}. \tag{7}$$

Considering (1), (2), (6), and (7), it could be obtained that

$$\frac{\partial \log(p(\mathbf{v}, \mathbf{h}))}{\partial w_{ij}} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model} \tag{8}$$

where $\langle \cdot \rangle$ denotes the expectation under the distribution of the data or the model. However, exact maximum likelihood learning for this model is intractable. The time cost of exact computation of the data-dependent expectation is exponential in the number of hidden units, and that of the model expectation is exponential in the number of hidden and visible units. Therefore, contrastive divergence (CD) is proposed to approximate the expectations. Specifically, the unbiased expectation of $\langle v_i h_j \rangle_{data}$ can be simply approximated by $v_i h_j$ through CD_1 learning [18]. While the unbiased expectation of $\langle v_i h_j \rangle_{model}$ can be calculated using Gibbs sampling of the data dependent hidden vector and further reconstruction of the data, denoted by $\langle v_i h_j \rangle_{recon}$. Therefore, the learning rule, also know as the weight change during optimizing the RBM becomes

$$\Delta w_{ij} = \varepsilon(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{recon}) \tag{9}$$

where ε is the learning rate. Similarly, the learning rules for the bias terms are respectively

$$\Delta a_i = \varepsilon(\langle v_i \rangle_{data} - \langle v_i \rangle_{recon}) \tag{10}$$

and

$$\Delta b_j = \varepsilon(\langle h_j \rangle_{data} - \langle h_j \rangle_{recon}). \tag{11}$$

Following the learning rules in (9), (10), and (11), an individual RBM can be efficiently trained.

The procedures above describe the pretraining stage of the DBN, and the fine-tuning stage will be discussed in the next section. One should note that each pretrained RBM aims at reconstructing the input, and the pretraining stage is performed unsupervisedly.

B. Deep Learning Scheme Based on Restricted Boltzmann Machines

1) Deep Belief Network:: After training multiple RBMs individually, a deep belief network can be constructed by stacking the RBMs one by one. The state of the hidden layer in the lower RBM is used as the input to the visible layer of the upper RBM, as shown in Fig. 1(b). The input is fed to the bottom RBM. The state vector of the top hidden layer is the input to the softmax regression layer, i.e. the output layer.

As an extension to logistic regression, which usually deals with binary classification problems, softmax regression [32] addresses multiclass classification problem. Given a training set $\{(\boldsymbol{h}^{\boldsymbol{k}}_{(\mathbf{1})}, \boldsymbol{y}_{(\mathbf{1})}), (\boldsymbol{h}^{\boldsymbol{k}}_{(\mathbf{2})}, \boldsymbol{y}_{(\mathbf{2})}), \cdots, (\boldsymbol{h}^{\boldsymbol{k}}_{(\boldsymbol{M})}, \boldsymbol{y}_{(\boldsymbol{M})})\}$ where M is the number of training samples and $\{\boldsymbol{h}^{\boldsymbol{k}}_{(\mathbf{1})}, \cdots, \mathbf{h}^{\boldsymbol{k}}_{(\boldsymbol{M})}\}$ are the hidden vector of the top RBM. Denote all the parameters of the softmax regression as $\mathbf{w}$, the conditional probability of $p(y = j|\boldsymbol{h}^{\boldsymbol{k}})$ for each class $j = 1, \cdots, c$ can be calculated as

$$\begin{bmatrix} p\left(y_{(i)} = 1 \middle| \boldsymbol{h}_{(i)}; \mathbf{w}\right) \\ p\left(y_{(i)} = 2 \middle| \boldsymbol{h}_{(i)}; \mathbf{w}\right) \\ \vdots \\ p\left(y_{(i)} = c \middle| \boldsymbol{h}_{(i)}; \mathbf{w}\right) \end{bmatrix} = \frac{1}{\sum_{j=1}^{c} e^{\boldsymbol{w}_j^T \mathbf{h}_{(i)}}} \begin{bmatrix} e^{\boldsymbol{w}_1^T \mathbf{h}_{(i)}} \\ e^{\boldsymbol{w}_2^T \mathbf{h}_{(i)}} \\ \vdots \\ e^{\boldsymbol{w}_c^T \mathbf{h}_{(i)}} \end{bmatrix} \tag{12}$$

where $\mathbf{h}_{(i)} \in \mathbf{R}^{n+1}$ is the top hidden vector $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_c \in \mathbf{R}^{n+1}$ are the parameters connected to each unit in the output layer $\sum_{j=1}^{c} e^{\boldsymbol{w}_j^T \mathbf{h}_{(i)}}$ is the normalization term. When deal with binary classification problem, it will degenerate to logistic regression.

The cost function of softmax regression takes a form similar to that of logistic regression as

$$J(\mathbf{w}) = -\frac{1}{M}\left[\sum_{i=1}^{M}\sum_{j=1}^{c} 1\left\{y_{(i)} = j\right\} \log p\left(y_{(i)} = j \middle| \boldsymbol{h}^{\boldsymbol{k}}_{(\boldsymbol{i})}; \mathbf{w}\right)\right], \tag{13}$$

where $1\{\cdot\}$ is the indicator function, which takes 1 if the input statement is true, 0 otherwise.

Conjugate gradient method can be employed to minimize the cost function in (13). The error term obtained will be back propagated through the multilayer DBN to fine tune the parameters, which is thus called the fine-tuning stage. The fine-tuning stage can be further divided into two phases. In the first phase, only the weight $\mathbf{w}$including the corresponding bias term connected to the output layer is tuned; after an empirical number of epochs (e.g., 6 or 10), all the parameters throughout

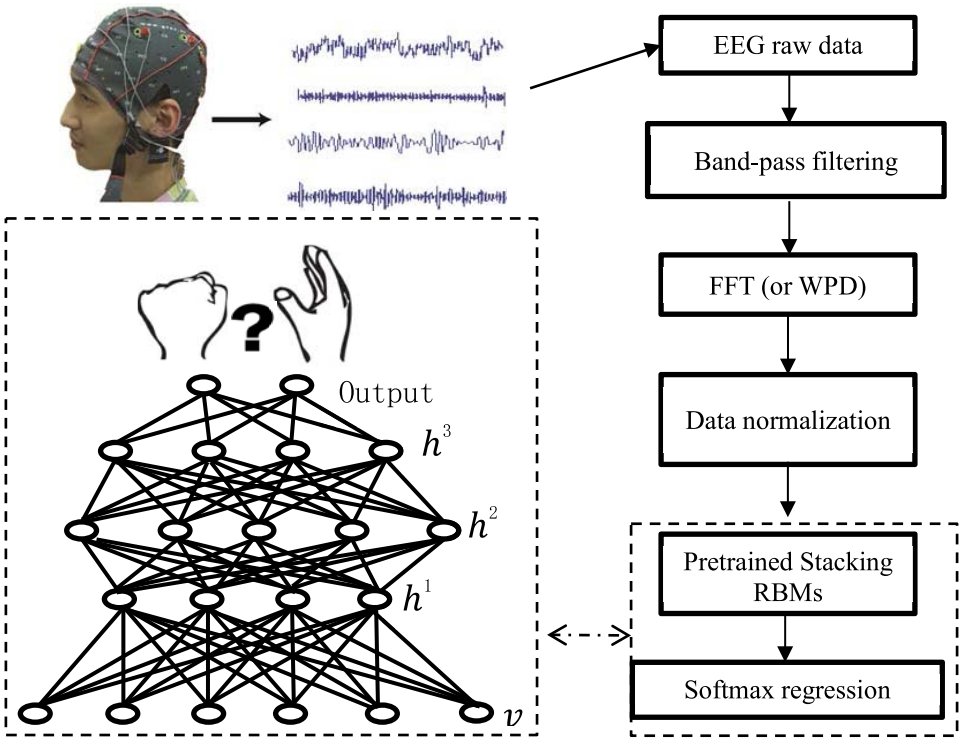


Fig. 2. Deep belief network scheme for motor imagery classification.

the network will be tuned. Without loss of generality, the updating rule of the weight in the top layer can be written as

$$\mathbf{w}_{ij} = \alpha \mathbf{w}_{ij} - \varepsilon \nabla_{\mathbf{w}_{ij}} J(\mathbf{w}) \tag{14}$$

where α is the momentum, and ε is the learning rate. The update of the weights in the other layers follows the back-propagation rules [33].

In practical applications, different constraints can be incorporated into (13) as regularization terms. Two most widely-used constraints are weight decay and sparsity constraint. A cost function with weight decay is formulated as

$$\hat{J}(\mathbf{w}) = J(\mathbf{w}) + \frac{\lambda}{2} \sum_l \sum_i \sum_j (\mathbf{w}_{ij}^l)^2 \tag{15}$$

where the second term is called weight decay that could suppress the magnitude of the weight and help to avoid overfitting. In addition, sparsity constraint on the weight is incorporated through restricting the value of the activations at each unit [34]–[36]. Denote the activation (the weighted output with sigmoid function employed) of unit j in layer l in response to input $\mathbf{h}$ with $a_j^l(\mathbf{h})$, and the average activation of unit j over the M training samples can be computed as

$$\hat{\rho}_j = \frac{1}{M} \sum_{i=1}^{M} a_j^l(\mathbf{h}_{(i)}) \tag{16}$$

where $a_j^l(\mathbf{h}_{(i)}) = \sigma((\mathbf{w}_j^l)^T \mathbf{h}_{(i)})$. $\mathbf{h}_{(i)}$ is sample i, $\sigma(\cdot)$ is the sigmoid function and $\mathbf{w}_j^l$ is the weight that link unit j in layer l to all the units in layer $l-1$.

To enforce the sparsity constraint, the following cost function can be formulated based on (16)

$$\widehat{J}(\mathbf{w}) = J(\mathbf{w}) + \beta \sum_j KL\left(\rho \,\|\, \hat{\rho}_j\right) \tag{17}$$

where $KL(\cdot)$ is the KL-divergence, and ρ is called the sparsity parameter (i.e., a predefined small value close to zero) [37]. Minimizing (17) would make the average activation $\hat{\rho}_j$ as close as to ρ, which means that most of the activations may become zero and thus incorporates sparsity.

2) FDBN for Motor Imagery Classification:: Similar training procedure of DBN as used for other types of data like images can also be employed for motor imagery classification. However, our experiments have shown that even though the EEG time series data can be directly used as the input to DBN and achieve a performance comparable with the classic methods like SVM, the frequency domain input can lead to a much better performance. Therefore, frequency domain data obtained via FFT or WPD are fed to train the DBN for motor imagery classification in this study. The proposed scheme is illustrated in Fig. 2, and given the name frequential DBN (FDBN) as it operates in the frequency domain.

It should be stressed that three key preprocessing procedures are employed in the pipeline, including bandpass filtering of the raw time series, FFT or WPD, and normalization. Previous research on neurophysiology has demonstrated that the activities of the brain in response to motor imagery focus on five frequency bands, including Alpha (8–13 Hz), Sigma (11–15 Hz), low Beta (18–23 Hz), high Beta (21–26 Hz) and low Gamma (25–35 Hz). In addition, electrooculography (EOG, lower than 4 Hz) and electromyography (EMG, higher

TABLE I
COMPARISON OF FDBN AND OTHER STATE-OF-THE-ART METHODS AND TOP THREE METHODS IN BCI COMPETITION IV IN TERMS OF CLASSIFICATION ACCURACY

Subject	Chin	Gan	Coyle	Bispectrum	CSP	FBCSP MIBIF	FBCSP MIRSR	FDBN
B01	0.70	0.71	0.60	0.77	0.66	0.68	0.70	**0.81**
B02	0.61	0.61	0.56	**0.65**	0.62	0.59	0.61	**0.65**
B03	0.61	0.57	0.56	0.61	0.57	0.59	0.61	**0.66**
B04	0.98	0.97	0.89	0.97	0.97	**0.98**	0.98	**0.98**
B05	**0.93**	0.86	0.79	0.82	0.77	**0.93**	**0.93**	**0.93**
B06	0.81	0.81	0.75	0.85	0.75	0.80	0.81	**0.88**
B07	0.78	0.81	0.69	0.75	0.77	0.78	0.78	**0.82**
B08	0.93	0.92	0.93	0.91	0.93	0.93	0.93	**0.94**
B09	0.87	0.89	0.81	0.87	0.83	0.88	0.87	**0.91**
Avg.	0.80	0.79	0.73	0.80	0.76	0.80	0.80	**0.84**

than 35 Hz) may contaminate true signals and thus make it difficult for motor imagery discrimination. Therefore, bandpass filtering is applied to exclude such noises and only leave the most discriminative frequency bands. Specifically, the data has been bandpass filtered between 8 Hz and 35 Hz with a fifth order Butterworth filter.

IV. EXPERIMENTS

A. Datasets and Comparison Criteria

To verify the performance of the proposed scheme for motor imagery classification, extensive and systematic experiments have been conducted on BCI competition IV data set 2b [38]. The dataset includes nine subjects each with five sessions of motor imagery experiments, among which the first two sessions are recorded without feedback and the other three sessions have incorporated online feedback. The EEG measurement was recorded at three channels (C3, Cz, and C4) with a sampling frequency of 250 Hz, and two classes of motor imageries were included: left hand and right hand. Each of the first two sessions has 120 trials, and each of the rest three sessions includes 160 trials. During recording, the EEG sequences have been bandpass filtered between 0.5 Hz and 100 Hz, and a notch filter at 50 Hz was applied to remove the influence from the power line. According to the knowledge on motor imagery related rhythms [39] as mentioned in Section III-B2), the data has been further bandpass filtered between 8 Hz and 35 Hz.

The motor imagery classification task aims at predicting the label of the trials in sessions 4 and 5 for each subject, given the supervising information of sessions 1 to 3. Considering the different generation principle for sessions 1, 2 and sessions 3 to 5, the trials in session 3 are the most similar ones to those in sessions 4 and 5. To evaluate the influence of incorporating sessions 1 and 2 for training or not, experiments on different size of training set have also been performed. To quantitatively compare the performance of different approaches, classification accuracy is employed for evaluation.

B. Comparison of FDBN With State-of-the-Art Methods

To evaluate the performance of FDBN for motor imagery classification, comparison experiments have been conducted for FDBN and other state-of-the-art methods, including Bispectrum-based approach [40], common spatial filter (CSP) [41], filter bank common spatial pattern (FBCSP) in combination with mutual information-based best individual feature (MIBIF) [42], FBCSP in combination with mutual information-based rough set reduction (MIRSR) [42]. The top three methods in BCI competition IV, respectively referred as Chin, Gan and Coyle, have also been compared.

Considering that the brain activities reflecting the imaginary limb motion mainly exist in the contralateral region of the brain, the data from channel Cz is excluded in our experiments as done in [39]. The three hidden layers in FDBN have 70, 60, and 50 units respectively, which are selected through experiments. 80 epochs of pretraining have been employed for each RBM. The learning rate for both the weights and the bias terms is set as 0.1. The momentum is set as 0.5 in the first five training epochs and changed to 0.9 afterwards. Both FFT and WPD have been considered for experiments, and the performance of FFT turned out to be superior to that of WPD. Therefore, the results reported in this section are based on FFT. Specifically, each trial of the experiments includes 1000 data points per channel and a 1024 point FFT is applied separately. Then in the frequency domain, data from 8 Hz to 35 Hz has been segmented as the input to train the network, which resulted in 113 data points per channel. Therefore, the final input to train the RBM is a number of 226-dimensional vectors.

The classification accuracy of the experiments is given in Table I and the algorithm(s) performing the best for each subject is highlighted in bold face. The average performance of each method over the nine subjects is also reported. From these results, it can be clearly seen that the FDBN method is superior to other methods for all the subjects with utmost 8.2% (subject B03) better than the second best. On an average level, the classification accuracy has been improved about 5% compared with other methods. In addition, to verify whether the performance difference between the proposed FDBN method and other methods is statistically significant, paired t-test has been conducted between the results of FDBN and other methods. The obtained p-values are given in Table II. It can be seen that all the obtained p-values are less than 0.01, which

TABLE II
PAIRED t-TEST BETWEEN FDBN AND OTHER METHODS

	Chin	Gan	Coyle	Bispectrum	CSP	FBCSP MIBIF	FBCSP MIRSR
FDBN	0.0076	0.0028	0.0002	0.0033	0.0030	0.0098	0.0071

TABLE III
CLASSIFICATION PERFORMANCE OF FDBN WITH SESSION 3 AS THE TRAINING SET

	Accuracy	sub1	sub2	sub3	sub4	sub5	sub6	sub7	sub8	sub9
Session 4	Testing set	0.77	0.68	0.58	0.99	0.81	0.81	0.82	0.93	0.91
Session 5	Testing set	0.67	0.63	0.62	0.95	0.92	0.86	0.83	0.94	0.85

TABLE IV
CLASSIFICATION PERFORMANCE OF FDBN WITH SESSIONS 1, 2, 3 AS THE TRAINING SET

	Accuracy	sub1	sub2	sub3	sub4	sub5	sub6	sub7	sub8	sub9
Session 4	Testing set	0.86	0.69	0.66	1.00	0.85	0.88	0.83	0.92	0.91
Session 5	Testing set	0.75	0.63	0.64	0.96	0.95	0.88	0.85	0.96	0.88

TABLE V
AVERAGE PERFORMANCE OF FDBN WITH DIFFERENT TRAINING SET

Testing set	Training set	Sessions 1,2,3	Session 3
Session 4	Avg. of training accuracy	0.99	0.98
	Avg. of testing accuracy	0.84	0.81
Session 5	Avg. of training accuracy	0.99	0.98
	Avg. of testing accuracy	0.83	0.81

TABLE VI
AVERAGE PERFORMANCE OF FDBN ACROSS ALL SUBJECTS

Accuracy	**Session1, 2, 3 for training Session 4 for testing**	**Session 3 for training Session 4 for testing**
Training set	0.98	0.96
Testing set	0.72	0.70
Accuracy	**Session1, 2, 3 for training Session 5 for testing**	**Session 3 for training Session 5 for testing**
Training set	0.97	0.99
Testing set	0.71	0.70

suggests that the performance improvement of FDBN over other methods is statistically significant.

C. Comparison of DBM on Different Size of Training Sets

According to the experiment setting of EEG recording, session 3 followed the same generation mechanism as that of sessions 4 and 5, which involved feedback. It is natural to use the data in session 3 as the training set to predict the labels of the trials in sessions 4 and 5. However, even though the generation mechanism of sessions 1 and 2 is different (without feedback), it would be interesting to find out whether they are helpful to train the deep neural network. In this section, results of experiments on different size of training set are reported.

Specifically, two kinds of experiment settings have been employed for comparison, i.e., only using session 3 as the training set, or using sessions 1, 2, and 3 as the training set. The training set is employed for both pretraining and fine-tuning. The results in terms of classification accuracy are given in Table III and IV, respectively, where the accuracy on the testing set is presented. The average performance is given in Table V. It can be seen from these results that including sessions 1 and 2 for training can obviously improve the classification performance, with an accuracy increase of 3.7% and 2.5% on session 4 and 5, respectively. It suggests that the brain activities related to certain motor imagery may share common features no matter with or without feedback, and FDBN can extract these features automatically. It would be helpful to collect more data for each subject in the future research. The results also verified the effectivity of session-to-session transfer for the same subject.

To test the potential of subject-to-subject transfer, all the training data from all the subjects have been pulled together as the training set. The corresponding results are given in Table VI. These results suggest that the classification accuracy on the testing set actually decreased when compared with the average performance in Table V. It clearly indicated that subject-to-subject transfer is less efficient than session-to-session transfer.

D. Convergence and Computational Complexity of FDBN

To observe the convergence process of FDBN at the fine-tuning stage, the error change curves of four subjects along with epochs (iterations) for session 4 are given in Fig. 3, which measures the percentage of the incorrectly classified samples in the training set. For most of the subjects, the training process converges within about 100 epochs, or at most 200 epochs. For insurance, 500 epochs were employed for each subject.

At the pretraining stage, each RBM has been trained with 80 epochs, which is selected through experiments. For better illustration, the energy change of the first hidden layer RBMs for subject 1 and 2 based on training set session 4

TABLE VII
CLASSIFICATION ACCURACY OF DBN IN TIME DOMAIN

	Accuracy	sub1	sub2	sub3	sub4	sub5	sub6	sub7	sub8	sub9	Avg.
Session 4	Training set	0.98	0.97	0.90	0.96	0.96	0.94	1.00	0.92	0.95	0.95
	Testing set	0.54	0.52	0.52	0.63	0.82	0.58	0.91	0.63	0.58	0.65
Session 5	Training set	0.98	0.97	0.91	0.95	0.96	0.89	1.00	0.90	0.98	0.95
	Testing set	0.59	0.65	0.75	0.59	0.83	0.60	0.83	0.61	0.63	0.67

TABLE VIII
CLASSIFICATION PERFORMANCE COMPARISON OF FFT AND WPD ON SESSION 5

	sub1	sub2	sub3	sub4	sub5	sub6	sub7	sub8	sub9	Avg.
WPD	0.65	0.63	0.58	0.95	0.66	0.76	0.78	0.94	0.84	0.75
FFT	0.75	0.63	0.64	0.96	0.95	0.88	0.85	0.96	0.85	0.83

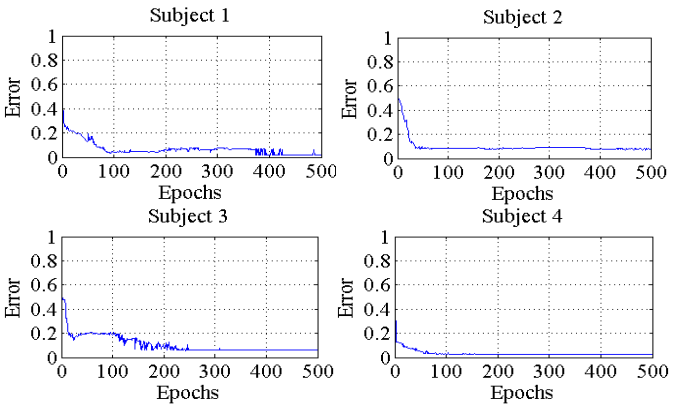


Fig. 3. Convergence curve of the training error on session 4.

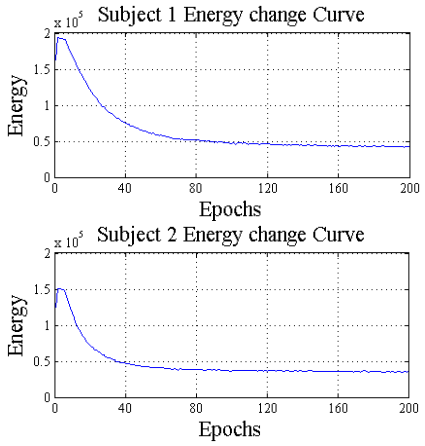


Fig. 4. Energy change curves of the first hidden layer RBM.

is shown in Fig. 4 with additional training epochs (200), where obvious convergence could be observed. These results have verified the efficiency of FDBN for motor imagery classification.

The structure of the FDBN employed in the experiments here is $226 \times 70 \times 60 \times 50 \times 2$ as mentioned in Section IV-B, where 226 is the dimension of the input layer, 70, 60, and 50 are respectively the size of the three hidden layers, and the output layer has two units. The experiments are conducted on a desktop with dual core processor i7-3700 of 3.4 GHz and 12 GB memory. Matlab is used to implement the algorithm. There are about 300 samples in the training set for each subject and about 100 samples in the testing set. For each traning set on average, the preprocesing including FFT transformation took about 1.39 s; the pretraing took about 12.38 s; the traing stage took about 98.84 s; the testing took about 2.04 s. Limit number of training samples assured fast training. However, our experiments suggest when the size of the training set is under about 100, overfitting is likely to happen.

E. Comparison of DBN in Time Domain and Frequency Domain

It has been mentioned in Section IV-B that the frequency domain input to the DBN leads to better performance than the raw data in time domain for motor imagery classification. For a formal comparison, Table VII lists the performance of DBN using the band-pass filtered time series as the input. Sessions 1, 2, and 3 have all be employed as the training set. Comparing the results in Table VII and Table IV, it can be seen that the frequency domain input has improved the classification accuracy by 27.69% and 23.88% for session 4 and 5, respectively, which suggests that frequency domain input works much better than the time domain counterpart for motor imagery classification.

In addition, two methods FFT and WPD (8 level WPD with Sym6 wavelet) have been employed to transform the data from time domain to frequency domain. The comparison results in classification accuracy are presented in Table VIII, which suggests FFT has enabled better performance. A possible explanation is that the time localization of the featured components related to specific motor imagery may varies along different trials, while the frequency features are relatively steady.

F. Effect of Different Preprocessing and Constraint

Different preprocessing method and constraint are compared here, including sliding window [40] in both time domain and frequency domain, whitening of the data, weight decay regularization and sparsity constraint. Sliding window can be adopted to segment each trial into overlapped patches,

TABLE IX
CLASSIFICATION PERFORMANCE OF SLIDING WINDOW IN TIME DOMAIN

	Accuracy	sub1	sub2	sub3	sub4	sub5	sub6	sub7	sub8	sub9	Avg.
Session 4	Training set	0.99	0.98	0.97	0.96	0.96	0.97	1.00	0.98	0.99	0.98
	Testing set	0.62	0.72	0.77	0.65	0.91	0.62	0.88	0.62	0.59	0.71
Session 5	Training set	0.98	0.98	0.96	0.95	0.97	0.93	1.00	0.97	0.99	0.97
	Testing set	0.59	0.56	0.90	0.57	0.68	0.62	0.78	0.64	0.58	0.66

TABLE X
CLASSIFICATION PERFORMANCE OF SLIDING WINDOW IN TIME DOMAIN ON SESSIONS 4 AND 5

Accuracy	sub1	sub2	sub3	sub4	sub5	sub6	sub7	sub8	sub9	Avg.
Training set	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Testing set	0.65	0.56	0.50	0.73	0.62	0.68	0.78	0.76	0.67	0.66

TABLE XI
CLASSIFICATION PERFORMANCE OF WHITENING WITH WPD ON SESSIONS 4 AND 5

	Accuracy	sub1	sub2	sub3	sub4	sub5	sub6	sub7	sub8	sub9	Avg.
Without whitening	Training set	0.93	0.97	0.87	0.96	0.92	0.97	0.86	0.92	0.93	0.93
	Testing set	0.61	0.63	0.58	0.95	0.73	0.83	0.83	0.97	0.74	0.76
With whitening	Training set	0.92	0.96	0.90	0.96	0.94	0.95	0.96	0.93	0.97	0.94
	Testing set	0.68	0.61	0.62	0.95	0.78	0.82	0.81	0.96	0.82	0.78

TABLE XII
CLASSIFICATION PERFORMANCE WITH REGULARIZATION AND SPARSITY CONSTRAINT ON SESSION 5

	sub1	sub2	sub3	sub4	sub5	sub6	sub7	sub8	sub9	Avg.
Training set	0.97	0.97	0.89	0.95	0.97	0.88	0.94	0.92	0.95	0.94
Testing set	0.70	0.59	0.59	0.95	0.73	0.79	0.82	0.97	0.78	0.77

which could then be used for the classification as a population through voting. In our experiments, each trial is segmented into 500 time point sequences with 90% of overlapping. These segments were directly used to train the DBN after bandpass filtered between 8 Hz and 35 Hz, which gave the classification accuracy results in time domain as in Table IV. FFT has been applied to these bandpass filtered segments to obtain frequency domain input. The corresponding results are given in Table X. The classification accuracy with sliding window employed has reduced on most of the cases, which suggests sliding window is not an optimal choice in this scenario.

In addition, ZCA whitening is employed to reduce the redundancy within the data. The corresponding results are given in Table XI. Regularization term (weight decay with parameter $\lambda = 0.01$) and sparsity constraint (with parameter $\rho = 0.1$) have also been applied. Table XII gives the related results. The parameters were selected through experiments and WPD was employed to transform the data to frequency domain. These results show that such preprocessing procedures can improve the performance, but only to a trivial degree.

G. Comparison of FDBN With Different Structures

To test the sensitivity of FDBN to the changes of the DBN structure, different numbers of units have been employed for the three hidden layers. Specifically, the number of the units in one hidden layer was changed over a range, and the numbers of the units in the rest two hidden layers were kept fixed. The number of units in the first hidden layer takes value from [30 50 70 100 200 500] while the other two layers remain unchanged, respectively with 60 and 50 units. The corresponding performances of these six settings of different structures is given in Table XIII. Table XIV reports the results with hidden layer 1 and 3 of 70 and 50 units, and hidden layer 2 varying in [20 40 50 60 80 150 400]; Table XV presents the results with hidden layer 1 and 2 of 70 and 60 units when the number of units in hidden layer 3 takes value from [10 30 50 80 120 180 270].

TABLE XIII
CLASSIFICATION PERFORMANCE OF DIFFERENT NUMBER OF UNITS IN THE FIRST HIDDEN LAYER

	sub1	sub2	sub3	sub4	sub5	sub6	sub7	sub8	sub9	Avg
1	0.69	0.61	0.58	0.95	0.88	0.85	0.77	0.94	0.79	0.78
2	0.67	0.61	0.59	0.95	0.88	0.82	0.77	0.93	0.83	0.78
3	0.68	0.63	0.64	0.96	0.95	0.88	0.85	0.94	0.83	0.82
4	0.68	0.63	0.61	0.94	0.93	0.88	0.83	0.93	0.80	0.80
5	0.78	0.61	0.60	0.95	0.88	0.83	0.76	0.95	0.78	0.79
6	0.65	0.59	0.59	0.95	0.87	0.87	0.78	0.94	0.83	0.79

Parameter tuning can be very tricky for neural networks. However, when the number of units in each hidden layer changes more than 20 times, the change in the classification

TABLE XIV
CLASSIFICATION PERFORMANCE OF DIFFERENT NUMBER OF UNITS IN THE SECOND HIDDEN LAYER

	sub1	sub2	sub3	sub4	sub5	sub6	sub7	sub8	sub9	Avg.
1	0.68	0.63	0.58	0.95	0.88	0.85	0.79	0.94	0.79	0.79
2	0.66	0.60	0.61	0.95	0.88	0.83	0.77	0.95	0.85	0.79
3	0.68	0.63	0.60	0.95	0.88	0.84	0.75	0.94	0.80	0.79
4	0.69	0.63	0.65	0.96	0.89	0.88	0.83	0.95	0.81	0.81
5	0.68	0.63	0.59	0.95	0.87	0.83	0.77	0.95	0.85	0.79
6	0.69	0.61	0.60	0.95	0.87	0.83	0.77	0.95	0.79	0.79
7	0.68	0.62	0.59	0.95	0.88	0.84	0.78	0.95	0.82	0.78

TABLE XV
CLASSIFICATION PERFORMANCE OF DIFFERENT NUMBER OF UNITS IN THE THIRD HIDDEN LAYER

	sub1	sub2	sub3	sub4	sub5	sub6	sub7	sub8	sub9	Avg.
1	0.67	0.59	0.62	0.94	0.90	0.86	0.80	0.89	0.82	0.79
2	0.66	0.59	0.59	0.95	0.89	0.85	0.76	0.93	0.79	0.78
3	0.74	0.64	0.64	0.96	0.95	0.88	0.80	0.96	0.88	0.83
4	0.69	0.64	0.63	0.95	0.91	0.86	0.80	0.95	0.85	0.81
5	0.67	0.63	0.58	0.95	0.89	0.82	0.78	0.95	0.84	0.79
6	0.66	0.60	0.60	0.95	0.89	0.84	0.80	0.94	0.81	0.79
7	0.69	0.60	0.61	0.95	0.89	0.83	0.81	0.94	0.78	0.79

accuracy is about 5% for motor imagery classification task. These results have shown the robustness of the FDBN method against network structure variation.

V. CONCLUSION

A deep learning scheme based on restricted Boltzmann machine and FFT for motor imagery classification is developed in this paper. Extensive and systematic experiments on public available benchmark datasets have been performed. Some valuable suggestions have been generated based on our results. First, frequency domain input to DBN can lead to much improved performance on motor imagery classification than the raw time series data, as suggested by the comparisons with the state-of-the-art methods. The performance improvement has been verified to be statistically significant with a p-value less than 0.01. Second, session-to-session knowledge transfer for the same subject turns out to be effective even under different data generation mechanisms, while subject-to-subject transfer is inefficient. Third, time localization of the frequency components for motor imagery trials is not as discriminative as the typical frequency information. Therefore, the use of FFT has led to a better performance than that of WPD. Finally, the proposed FDBN scheme is relatively robust to the network structure. This research has thus made certain encouraging attempts for the application of deep learning in motor imagery classification, and our results can be of significant interest to the BCI community.

REFERENCES

[1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain–computer interfaces for communication and control," *Clin. Neurophysiol.*, vol. 113, no. 6, pp. 767–791, 2002.

[2] Y. Han and H. Bin, "Brain–computer interfaces using sensorimotor rhythms: Current state and future perspectives," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 5, pp. 1425–1435, May 2014.

[3] J. Decety and D. H. Ingvar, "Brain structures participating in mental simulation of motor behavior: A neuropsychological interpretation," *Acta Psychol.*, vol. 73, no. 1, pp. 13–34, 1990.

[4] B. Blankertz, S. Lemm, M. Treder, S. Haufe, and K.-R. Müller, "Single-trial analysis and classification of ERP components—A tutorial," *NeuroImage*, vol. 56, pp. 814–825, May 2011.

[5] Y. Li *et al.*, "An EEG-based BCI system for 2-D cursor control by combining Mu/Beta rhythm and P300 potential," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 10, pp. 2495–2505, Oct. 2010.

[6] J. Long, Y. Li, H. Wang, T. Yu, J. Pan, and F. Li, "A hybrid brain computer interface to control the direction and speed of a simulated or real wheelchair," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 20, no. 5, pp. 720–729, Sep. 2012.

[7] L. Karl, C. Kaitlin, D. Alexander, S. Kaleb, R. Eitan, and H. Bin, "Quadcopter control in three-dimensional space using a noninvasive motor imagery-based brain–computer interface," *J. Neural Eng.*, vol. 10, no. 4, p. 046003, 2013.

[8] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi, "A review of classification algorithms for EEG-based brain–computer interfaces," *J. Neural Eng.*, vol. 4, no. 2, p. R1, 2007.

[9] A. X. Stewart, A. Nuthmann, and G. Sanguinetti, "Single-trial classification of EEG in a visual object task using ICA and machine learning," *J. Neurosci. Methods*, vol. 228, pp. 1–14, May 2014.

[10] H.-I. Suk and S.-W. Lee, "A novel Bayesian framework for discriminative feature extraction in brain-computer interfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 286–299, Feb. 2013.

[11] S. Lemm, B. Blankertz, T. Dickhaus, and K.-R. Müller, "Introduction to machine learning for brain imaging," *NeuroImage*, vol. 56, no. 2, pp. 387–399, 2011.

[12] C. M. Bishop, *Neural Networks for Pattern Recognition*. New York, NY, USA: Oxford Univ. Press, 1996.

[13] M. Hamedi, S.-H. Salleh, A. M. Noor, and I. Mohammad-Rezazadeh, "Neural network-based three-class motor imagery classification using time-domain features for BCI applications," presented at the IEEE Region 10 Symp., 2014.

[14] N. Lu and T. Yin, "Motor imagery classification via combinatory decomposition of ERP and ERSP using sparse nonnegative matrix factorization," *J. Neurosci. Methods*, vol. 249, pp. 41–49, Jul. 2015.

[15] N. Lu, T. Li, J. Pan, X. Ren, Z. Feng, and H. Miao, "Structure constrained semi-nonnegative matrix factorization for EEG-based motor imagery classification," *Comput. Biol. Med.*, vol. 60, pp. 32–39, May 2015.

[16] C.-H. Wu *et al.*, "Frequency recognition in an SSVEP-based brain computer interface using empirical mode decomposition and refined generalized zero-crossing," *J. Neurosci. Methods*, vol. 196, pp. 170–181, Mar. 2011.

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

[18] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.

[19] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

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

[21] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[22] H. Larochelle and Y. Bengio, "Classification using discriminative restricted Boltzmann machines," presented at the 25th Int. Conf. Mach. Learn., Helsinki, Finland, 2008.

[23] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[24] J. Li, Z. Struzik, L. Zhang, and A. Cichocki, "Feature learning from incomplete EEG with denoising autoencoder," *Neurocomputing*, vol. 165, pp. 23–31, Oct. 2015.

[25] X. An, D. Kuang, X. Guo, Y. Zhao, and L. He, "A deep learning method for classification of EEG data based on motor imagery," in *Intelligent Computing in Bioinformatics*, vol. 8590, D.-S. Huang, K. Han, and M. Gromiha, Eds. Springer, 2014, pp. 203–210.

[26] Z. Wang, S. Lyu, G. Schalk, and Q. Ji, "Deep feature learning using target priors with applications in ECoG signal decoding for BCI," presented at the 23rd Int. Joint Conf. Artif. Intell., 2013.
[27] Z. V. Freudenburg, N. F. Ramsey, M. Wronkiewicz, W. D. Smart, R. Pless, and E. C. Leuthardt, "Real-time naive learning of neural correlates in ECoG electrophysiology," *Int. J. Mach. Learn. Comput.*, vol. 1, no. 3, p. 269, 2011.
[28] S. Jirayucharoensak, S. Pan-Ngum, and P. Israsena, "EEG-based emotion recognition using deep learning network with principal component based covariate shift adaptation," *Sci. World J.*, vol. 2014, Sep. 2014, Art. no. 627892.
[29] S. Ahmed, L. M. Merino, Z. Mao, J. Meng, K. Robbins, and Y. Huang, "A deep learning method for classification of images RSVP events with EEG data," in *Proc. Global Conf. Signal Inf. Process. (GlobalSIP)*, 2013, pp. 33–36.
[30] H. Yang, S. Sakhavi, K. K. Ang, and C. Guan, "On the use of convolutional neural networks and augmented CSP features for multi-class motor imagery of EEG signals classification," in *Proc. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2015, pp. 2620–2623.
[31] D. F. Wulsin, J. R. Gupta, R. Mani, J. A. Blanco, and B. Litt, "Modeling electroencephalography waveforms with semi-supervised deep belief nets: Fast classification and anomaly measurement," *J. Neural Eng.*, vol. 8, no. 3, p. 036015, 2011.
[32] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
[33] R. Salakhutdinov and G. Hinton, "An efficient learning procedure for deep Boltzmann machines," *Neural Comput.*, vol. 24, no. 8, pp. 1967–2006, 2012.
[34] H. Lee, E. Chaitanya, and A. Y. Ng, "Sparse deep belief net model for visual area V2," presented at the Adv. Neural Inf. Process. Syst., 2008.
[35] K. Cho, "Simple sparsification improves sparse denoising autoencoders in denoising highly noisy images," presented at the Int. Conf. Mach. Learn., 2013.
[36] V. Nair and G. E. Hinton, "3D object recognition with deep belief nets," presented at the Adv. Neural Inf. Process. Syst., 2009.
[37] J. M. Tomczak and A. Gonczarek, *Sparse Hidden Units Activation in Restricted Boltzmann Machine*. New York: Springer, 2015.
[38] R. Leeb, F. Lee, C. Keinrath, R. Scherer, H. Bischof, and G. Pfurtscheller, "Brain–computer communication: Motivation, aim, and impact of exploring a virtual apartment," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 15, no. 4, pp. 473–482, Dec. 2007.
[39] J. F. Saa and M. Çetin, "A latent discriminative model-based approach for classification of imaginary motor tasks from EEG data," *J. Neural Eng.*, vol. 9, no. 2, p. 026020, 2012.
[40] S. Shahid and G. Prasad, "Bispectrum-based feature extraction technique for devising a practical brain-computer interface," *J. Neural Eng.*, vol. 8, no. 2, p. 025014, 2011.
[41] H. Wang, Q. Tang, and W. Zheng, "L1-norm-based common spatial patterns," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 3, pp. 653–662, Mar. 2012.
[42] K. K. Ang, Z. Y. Chin, C. Wang, C. Guan, and H. Zhang, "Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b," *Front. Neurosci.*, vol. 6, Mar. 2012.

Na Lu received the B.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, Shaanxi, China, in 2002 and 2008, respectively.

Currently, she is an Associate Professor at Xi'an Jiaotong University, Xi'an, Shaanxi, China. Her research interests include statistical image analysis, machine learning, cognitive science and robotics.

Tengfei Li received the B.S. degree from Zhengzhou University, Zhengzhou, Henan, China, in 2013. He is an M.S. degree student at Xi'an Jiaotong University, Xi'an, Shaanxi, China.

His research interests include brain–computer interface and machine learning.

Xiaodong Ren received the Ph.D. degree from Xi'an Jiaotong University, Xi'an, Shaanxi, China, in 2010.

Currently, he is an Assistant Professor at Xi'an Jiaotong University, Xi'an, Shaanxi, China. His research interests include brain–computer interface and robotics.

Hongyu Miao received the M.S. and Ph.D. degrees from University of Rochester, Rochester, NY, USA, in 2004 and 2007, respectively.

He is currently Associate Professor at University of Texas Health Science Center, Houston, TX, USA. His research interests include mathematical modeling, computational biology, and biological applications.