Neural Networks 114 (2019) 67–77

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Learning joint space–time–frequency features for EEG decoding on small labeled data

Dongye Zhao [a,b,c], Fengzhen Tang [a,b,*], Bailu Si [a,b], Xisheng Feng [a,b]

[a] State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China
[b] Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110016, China
[c] University of Chinese Academy of Sciences, Beijing 100049, China

ARTICLE INFO

Article history:
Received 21 November 2018
Received in revised form 16 January 2019
Accepted 28 February 2019
Available online 11 March 2019

Keywords:
Brain–computer interfaces
Convolutional neural network
Joint space–time–frequency feature learning
Subject-to-subject weight transfer
Small labeled data

ABSTRACT

Brain–computer interfaces (BCIs), which control external equipment using cerebral activity, have received considerable attention recently. Translating brain activities measured by electroencephalography (EEG) into correct control commands is a critical problem in this field. Most existing EEG decoding methods separate feature extraction from classification and thus are not robust across different BCI users. In this paper, we propose to learn subject-specific features jointly with the classification rule. We develop a deep convolutional network (ConvNet) to decode EEG signals end-to-end by stacking time–frequency transformation, spatial filtering, and classification together. Our proposed ConvNet implements a joint space–time–frequency feature extraction scheme for EEG decoding. Morlet wavelet-like kernels used in our network significantly reduce the number of parameters compared with classical convolutional kernels and endow the features learned at the corresponding layer with a clear interpretation, i.e. spectral amplitude. We further utilize subject-to-subject weight transfer, which uses parameters of the networks trained for existing subjects to initialize the network for a new subject, to solve the dilemma between a large number of demanded data for training deep ConvNets and small labeled data collected in BCIs. The proposed approach is evaluated on three public data sets, obtaining superior classification performance compared with the state-of-the-art methods.

1. Introduction

Brain Computer Interfaces (BCIs) utilize brain signals to control external devices, providing an alternative pathway for human brain to communicate with the outside world. It is widely used for stroke rehabilitation (Meng, Lu, Man, Ma, & Gao, 2015) and other areas. Among many neuroimaging methods to capture the brain activities, electroencephalography (EEG) is by far the most widely used one, owing to its high temporal resolution, high portability, low cost, and few risks to the users (Nicolas-Alonso & Gomez-Gil, 2012). In the research field of EEG-based BCI, the core problem is how to decode EEG signals into correct instructions effectively, and is still an ongoing research question.

One type of frequently used methods to decode EEG signals is to extract time–frequency features (e.g. power spectral) through time–frequency transformation (e.g. wavelet transformation, Adeli, Zhou, & Dadmehr, 2003) and input the extracted features into a classifier (e.g. support vector machine, Kousarrizi, Ghanbari, Teshnehlab, Shorehdeli, & Gharaviri, 2009) to perform the final decoding. This type of methods only takes advantage of temporal and spectral information in EEG signals, ignoring the spatial information.

Another popular method is termed as filter bank common spatial pattern (FBCSP, Kai, Zheng, Zhang, & Guan, 2008), reaching great performance in multiple EEG signals decoding. FBCSP extracts features for each of frequency bands based on the spatial filtering method, but ignores correlations among different frequencies. Then Aghaei, Mahanta, and Plataniotis (2016) propose a separable common spatial–spectral patterns (SCSSP) method, which uses spectral power in multiple frequency bands and the spatial features of EEG signals. The performance of SCSSP may outperform the FBCSP if enough training data are provided. More importantly, the SCSSP requires significantly lower computations than the FBCSP. The work in Molina, Ebrahimi, and Vesin (2003) also develops a joint space–time–frequency method, which spatially decorrelates multivariate signals into univariate signals and then uses the quadratic transformation to represent each univariate representative data. This method obtains good classification performance for three-class BCI tasks. Therefore, jointly considering time, frequency, and space may provide better EEG decoding performance.

* Corresponding author at: State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China.
E-mail addresses: zhaodongye@sia.cn (D. Zhao), tangfengzhen@sia.cn (F. Tang), sibailu@sia.ac.cn (B. Si), fxs@sia.cn (X. Feng).

https://doi.org/10.1016/j.neunet.2019.02.009
0893-6080/

Above EEG decoding methods separate feature extraction from classification. The features are separately and manually designed according to experience. They are good for understanding the corresponding task but may not be optimal for classification. More importantly, manually designed features are not robust across subjects. For example, imagination of hand movement leads to event-related desynchronization (ERD) at μ rhythm, i.e. 8–12 Hz. When one particular subject may slightly shift ERD at a lower frequency, methods developed based on ERD at 8–12 Hz do not work well for this subject. Manually tuning the frequency range for the subject may solve the problem, but it is time-consuming.

Deep learning methods improve traditional signal processing methods by automatically learning subject-specific features guided by classification tasks. The method can be trained end-to-end, that is feeding raw EEG signals into the network could obtain the predicted label corresponding to the input in the end. Schirrmeister et al. (2017) build different architectures of convolutional neural networks (ConvNets) according to recent advances of the deep learning such as dropout. Disadvantages of ConvNets include that they are difficult to interpret, involve a large number of hyperparameters to learn, and require a large amount of training data. Although Schirrmeister et al. (2017) offset the first disadvantage by proposing a novel method to visualize extracted features, ConvNets still require to learn hundreds of parameters based on large training data sets. Our work not only makes the ConvNet more easy to interpret, but also solves the latter two problems.

In this paper, we stack time–frequency transformation, spatial filtering, and classification as a multiple layered neural network, implementing a joint space–time–frequency feature learning guided by classification performance. The method we propose is a deep convolutional network, termed as wavelet–spatial filters ConvNet (WaSF ConvNet). There are two convolutional layers in our network. The first convolution is designed to perform time–frequency transformation using adaptive wavelet kernels. The second convolution is designed to perform spatial filtering. Thus, our network is able to learn joint space–time–frequency features from the data and features in which frequency band useful for task-specific classification can be directly read from the first convolutional kernel. The proposed method competes closely with and even outperforms the state-of-the-art method on three public data sets.

Our main contributions are summarized as follows.

- We directly take spectral power modulations of EEG signals into consideration by using wavelet kernels. The wavelet central frequency corresponds to the used frequency of EEG signals for the task.
- We significantly reduce the number of hyperparameters in the learning process. For example, each wavelet kernel with 25 size only involves 2 learning parameters, while the similar kernel in Schirrmeister et al. (2017) requires 25 parameters.
- We propose a subject-to-subject transfer strategy to solve the overfitting problem caused by small training samples in deep learning algorithms. In other words, training samples required for the WaSF ConvNet may be in a small amount by using the proposed transfer strategy. For the BCIC IV 2a data set, the size of training data is decreased to 62% in average and 80% at least.

The rest of this paper is organized as follows. Related work is introduced in Section 2. The network architecture, network training strategies, and transfer learning strategy are described in Section 3. Validation of our network on three data sets from BCI competition IV is given in Section 4. Conclusion and main findings are provided in Section 5.

2. Related work

EEG decoding is one core issue in the EEG-based BCI systems. Existing approaches for EEG classification can be grouped into three categories: traditional signal processing approaches (Blankertz, Tomioka, Lemm, Kawanabe, & Muller, 2007), deep learning methods (Ma et al., 2016; Schirrmeister et al., 2017), and Riemannian geometry based approaches (Congedo, Barachant, & Bhatia, 2017). The Riemannian geometry based approaches represent EEG signals as covariance matrices, which live in a curved Riemannian space, and then perform classification in the Riemannian space. This type of approaches is not directly related to this paper. We only briefly mention it. If readers are interested in this type of approaches, please refer to Barachant, Bonnet, Congedo, and Jutten (2012), Congedo et al. (2017) and Yger, Berar, and Lotte (2017).

The most frequently used methods for EEG decoding in BCI systems are the traditional signal processing methods, which have been developed for a long time and thus have solid theoretical and empirical foundation. This kind of methods takes advantage of findings in brain science, i.e. timing, frequency and brain region characterizing an EEG signal triggered by a particular task, and manually designs features accordingly. The extracted features are then input to a separate classifier to perform the final decoding.

Many works utilize time–frequency transformation, e.g. wavelet transformation (Adeli et al., 2003), wavelet packet (Yen & Lin, 2000), and dual-tree complex wavelet transform (DTCWT, Kingsbury, 1998), to transform EEG signals from time domain to time–frequency domain and then extract features, e.g. energy, power spectral, and entropy (Meng et al., 2015), in the time–frequency domain. The band powers combining with the statistical features of wavelet coefficients are extracted from the wavelet transformed EEG signals to decode left–right hand motor imagery in work (Hong, Qin, Bai, Zhang, & Cheng, 2015).

As the utilization of features extracted in the time–frequency domain does not provide sufficient high classification accuracy for some BCI systems, a spatial filtering method called common spatial patterns (CSP) is proposed to extract discriminative features (Blankertz et al., 2007). A simple description of the CSP method is that it maximizes the variance for one class, while minimizes the variance for the other class (Blankertz et al., 2007). Subsequently, many approaches based CSP are developed. A successful example is filter bank common spatial patterns (FBCSP) proposed by Kai et al. (2008). This method solves the limitation of CSP, where a frequency band of the EEG needs to be determined manually before CSP operates on. The FBCSP method first uses bandpass-filters to make the EEG measurements into multiple frequency bands, then extracts CSP features for each of these bands, and finally automatically selects discriminative pairs of frequency bands and corresponding CSP features using a feature selection algorithm. Even though the FBCSP method obtains relatively high classification performance, spatial filtering methods usually treat each frequency band independently, ignoring correlations between features obtained from different EEG rhythms, leading redundancy in extracted features and the high requirement of the computational power (Aghaei et al., 2016; Ang, Chin, Zhang, & Guan, 2012). Separable common spatial–spectral pattern (SCSSP, Aghaei et al., 2016), a more efficient method, is proposed to significantly reduce computational cost compared with FBCSP. The SCSSP method processes EEG signals in both spatial and spectral domains by using a heteroscedastic matrix-variate Gaussian model.

Other methods extracting time, spectral and spatial features together are also proposed for EEG classification. The work (Ferrante, Gavriel, & Faisal, 2015) combines a Morlet wavelet transformation and CSP for feature extraction of EEG signals, achieving

better classification performance than other methods only using wavelet transformation. A joint time–frequency–space classification method is developed to classify EEG signals, based on joint time–frequency–space decorrelation (Molina et al., 2003). The method first decomposes the multivariate signals coming from several electrodes into several univariate representative signals, then obtains quadratic time–frequency representation for each univariate representative signal, and finally performs multivariate classification through an ensemble of univariate signal classification. This method achieves good classification performance for simple BCI tasks.

Traditional signal processing methods rely on manually designed features guided by experience and thus are not robust across different subjects. Significant progress is made in computer vision and natural language processing with the development of deep learning. This biologically-inspired deep learning method learns subject-specific features guided by the classification task instead of the prior knowledge. There exists research work applying deep learning for EEG decoding end-to-end (Kumar, Sharma, Mamun, & Tsunoda, 2017; Lee & Kwon, 2016; Schirrmeister et al., 2017; Tang, Li, & Sun, 2017). Kumar et al. (2017) propose a CSP-DNN method, which extracts CSP features first and then feeds features to a deep neural network (DNN) for classification. The feature extraction and classification are further combined into a single process. For example, Tang et al. (2017) train a convolutional neural network with five layers end-to-end, but the method has to initially select the time period and the frequency band based on the ERD/ERS phenomenon. In this aspect, our method only requires little preprocessing. Besides, Schirrmeister et al. (2017) study different architectures of deep convolutional neural networks (ConvNets) and especially propose a novel method to visualize the learned features. The most important conclusion in Schirrmeister et al. (2017) is the ConvNets indeed use spectral amplitude in the alpha, beta, and gamma frequency bands. Under this condition, we design convolutional kernels according to the wavelet transformation and directly extract features in time–frequency domains. Then the spectral power modulations in which frequency could be quickly read from parameters of the convolutional layer.

3. The proposed method

The decoding of EEG signals can be formulated as a supervised classification problem. Our goal is to develop a methodology that is able to classify EEG signals with high accuracy, high robustness and small training data set. All event-related potentials are limited in duration and in frequency. And the majority of events activate distinct brain regions (Sanei & Chambers, 2007). Therefore, efficient classification of EEG signals exploits features incorporating the space, time and frequency dimensions of the EEG data. This kind of joint space–time–frequency classification of EEG data has been studied in BCIs (Molina et al., 2003). However, existing methods separate feature extraction and classification into two independent procedures. They manually design subject-specific features first and then classify the EEG data based on the extracted features. The separation of feature extraction and classification may lead to inferior classification performance. We thus present a novel EEG classification method based on convolutional neural networks, merging the feature extraction and classification into a single process. The proposed method also exploits joint space–time–frequency features of EEG data using trainable wavelet based time–frequency filters and spatial filters. The proposed method is called wavelet–spatial filters ConvNet (WaSF ConvNet). In this section, we will give a detailed description of our proposed network.

3.1. Problem definition

Suppose that we are given one EEG data set for each subject (denoted by s). Each data set contains a number of labeled trials. Each trial is a time-segment of the originally continued EEG recording with each belonging to one of several classes. The data sets we are given can be denoted by $D^s = \{(X_1^s, y_1^s), (X_2^s, y_2^s), \ldots, (X_{N_s}^s, y_{N_s}^s)\}$, where N_s represents the number of recorded trials for subject s. Here, $X_i^s \in \mathbb{R}^{E\times T}$ is the input matrix with E denoting the number of electrodes and T representing the number of sampled time steps in each trial, while y_i^s is the class label of the ith trial for subject s. It takes values from a set of C class labels L ($L = \{l_1, l_2, \ldots, l_C\}$) corresponding to a set of brain activities. For instance, for the BCI competition IV 2a data sets of 4 classes ($C = 4$), y_i^s can take class l_1, l_2, l_3, or l_4, meaning that during the ith trial, the subject s performed either imagined left-hand movement, right-hand movement, foot movement, or tongue movement.

The task is to find a decoder f trained on existing trials such that it can assign new unseen trials correct class labels. In this paper, we consider the parametric classifier $f(X_i;\theta) : \mathbb{R}^{E\times T} \to L$, parameterized by θ, which assigns label y_j to the trial X_j, i.e. $y_j = f(X_j;\theta)$. The decoder $f(X_j;\theta)$ of EEG signals jointly represents two parts which are separated in the traditional wavelet transform: (i) extracting a feature representation $\phi(X_j;\theta_\phi)$ with parameter θ_ϕ, which could be learned from the data; (ii) utilizing a classifier g parameterized by θ_g trained using previous features, specifically, $f(X_j;\theta) = g(\phi(X_j;\theta_\phi),\theta_g)$.

3.2. The proposed network architecture

Different brain activities may trigger different brain regions to emit potentials in different timings and different frequencies. The WaSF ConvNet thus involves wavelet kernels and spatial filters, as shown in Fig. 1. The network consists of 5 layers: two specific convolutional layers followed by a pooling layer, then a dropout layer added before the final dense output layer.

3.2.1. Convolutional layers

The network contains two convolutional layers, performing time–frequency convolution and spatial convolution respectively. The first convolutional layer contains time–frequency filters designed by real-valued wavelets motivated by Morlet wavelet. In this paper, we use 25 time–frequency filters, corresponding to 25 time–frequency convolutional units. Each unit will convolute with inputs by using the same kernel for all electrodes E. The wavelet kernels (convolutional kernels) shown in Fig. 2 are formulated as follows:

$$w_\eta(t) = e^{-\frac{a_\eta^2 t^2}{2}} \cos(2\pi b_\eta t) \tag{1}$$

where $\eta = 1, \ldots, 25$ and t denotes the sampling time steps. a_η and b_η are two free parameters. $1/a_\eta$ is the bandwidth of the Gaussian, controlling the active time window of the wavelet kernel. b_η represents the wavelet central frequency. In this paper, the width of each wavelet is set to be 0.36 s (−0.18 s, 0.18 s) with 25 sampling time points.[1] The inputs are convoluted with each kernel and then fed to linear units, i.e. $f(x) = x$, where x is the convoluted results of the inputs with kernels. In total, this convolutional layer needs to learn 25×2 free parameters, which are significantly smaller than a traditional convolutional layer with $25(units)\times 25(width)\times 1(height)$ parameters for all convolutional units. Another advantage of our filters over traditional convolutional filters is that our filters also extract the frequency

[1] Other choices were also tried. This one delivers good performance across all data sets.

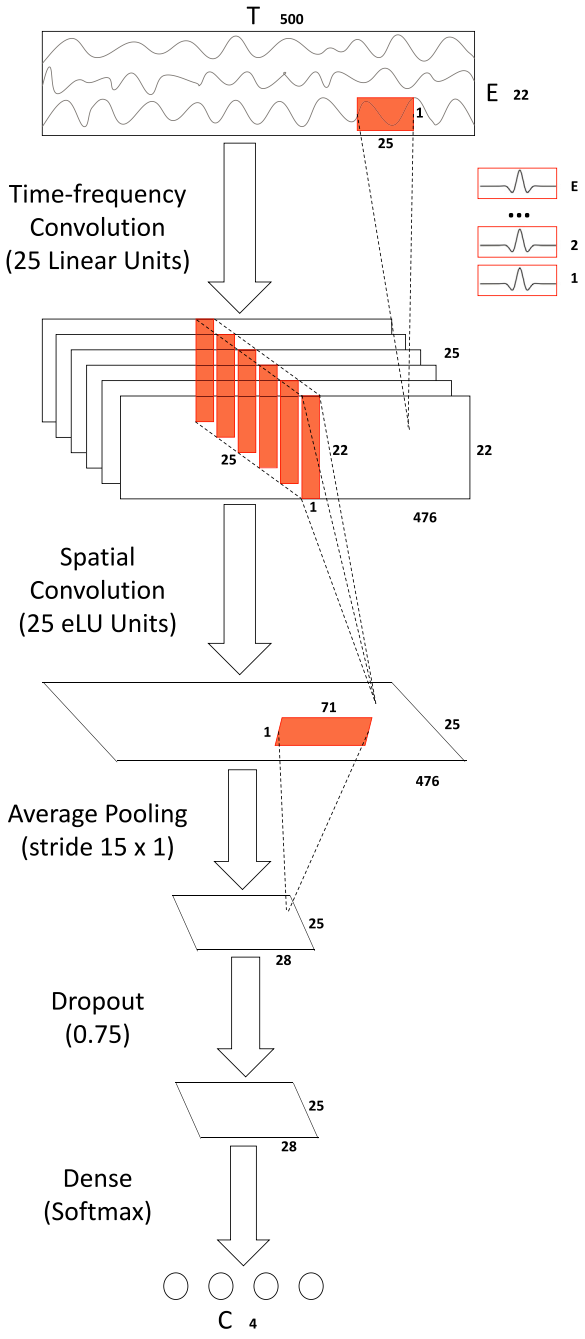

Fig. 1. Architecture. The network contains 5 layers: the time–frequency convolution (*25 linear units*) along the time dimension of the trial, each unit of which has the same convolutional kernel for all electrodes; the spatial convolution (*25 eLU units*) to collect mutual information among all electrodes; the pooling layer for coarser representations; the drop out layer to address overfitting; the dense layer with softmax non-linear activation for classification. *Red rectangles* represent kernels of convolutional and pooling layers.

information of the data besides the temporal information. The convolution process of this layer will change a two-dimensional EEG signal into a three-dimensional feature map.

The second convolutional layer consists of 25 spatial filters designed to extract the mutual information among all electrodes. Each spatial filter is designed by a kernel of size $1 \times E$ to convolve with each of the 25 feature maps obtained from the previous time–frequency convolution, respectively. Thus, in total, we need to learn $25(time - frequential\ units) \times 25(spatial\ units) \times$

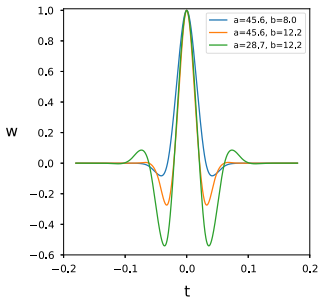

Fig. 2. Kernels of time–frequency convolutional units.

$1(width) \times E(height)$ parameters for this layer. Through the convolution process of this layer, the three-dimensional feature map obtained by the previous convolution process is changed into a two-dimensional representation. The activation functions of this layer are exponential linear units (eLUs), which are more robust to changeable inputs and more quickly convergent to the stable value than other activation functions (Clevert, Unterthiner, & Hochreiter, 2016):

$$f(x) = \begin{cases} x & x \geq 0 \\ e^x - 1 & x < 0 \end{cases} \tag{2}$$

3.2.2. Pooling and dropout layers

After the two convolution layers, a mean pooling layer and a dropout layer follow. The pooling layer processes a kernel of size 71×1 by moving a stride of size 15×1 each time. The pooling layer will create a coarser intermediate feature representation and make the network more translation invariant.

The dropout layer randomly sets some of the outputs from the pooling layer to zero in each update. The probability of each output to be set as zero is 0.75 in this paper. This technique is used to prevent co-adaption of different units and can be interpreted as analogous to training an ensemble of networks.

3.2.3. Final layer

The final layer is a dense layer with a full connection between the input units of this layer and the output units, performing softmax regression (multi-class logistic regression). The final layer contains C output units with feature representation extracted by previous layers being inputs.

The entire convolution neural network maps input data to one real number per class, i.e. $\boldsymbol{g}(X_i; \theta) : \mathbb{R}^{E \times T} \rightarrow \mathbb{R}^C$ where θ denotes the collection of all the parameters of the network, E is the number of electrodes, T represents the number of time steps, and C denotes the number of possible output labels. To obtain a classification result, the output is transformed to conditional probabilities of a label l_k given the input X_i using softmax function:

$$p(l_k|\boldsymbol{g}(X_i; \theta)) = \frac{e^{g_k(X_i;\theta)}}{\sum_{m=1}^{C} e^{g_m(X_i;\theta)}} \tag{3}$$

The softmax activation produces a conditional distribution over all possible output classes for each example. The entire network is trained to assign high probabilities to the correct labels by minimizing the sum of losses with respect to each training example:

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{N} L(y_i, \boldsymbol{g}(X_i; \theta)) \tag{4}$$

where

$$L(y_i, p(l_k|\boldsymbol{g}(X_i; \theta))) = \sum_{k=1}^{C} -\log(p(l_k|\boldsymbol{g}(X_i; \theta)))\delta(y_i = l_k) \tag{5}$$

is the negative log likelihood function, also known as the cross-entropy loss. Here $\delta(\cdot)$ is the indicator function, i.e. $\delta(true) = 1$ and $\delta(false) = 0$.

The final decoding of the EEG signal is to assign the example the label with maximum conditional probability, i.e.

$$f(X_i; \theta) = \arg\max_{l_k} p(l_k|\boldsymbol{g}(X_i; \theta)) \tag{6}$$

3.3. Network training

In order to obtain good generalization capability, we tried several training tricks popular in deep learning.

3.3.1. Cropped training

One drawback of deep neural networks is that they need a large number of examples to train the network. However, commonly, in BCIs, each subject or user only offers a small number of trials to train the corresponding EEG decoder. Fortunately, in each trial, we have recordings that last for several seconds. Here, instead of using the entire trials as input and per-trial labels as targets to train the network, we use a cropped training strategy (Schirrmeister et al., 2017).

A sliding time window of width T' time steps obtains multiple crops within the trial. The first crop starts from the t_c second of the trial. We slide the time window by t_c second each time to obtain another crop. All the crops extracted from the corresponding trial share the label of that trial. The cropped training strategy utilizes the obtained crops as input and the per-crop labels as targets to train the network. Crops extracted from the same trial are thus highly correlated to each other. Instead to minimize the loss function defined by Eq. (5), we need to minimize the regularized version defined as follows:

$$\begin{aligned} &L(y_i, p(l_k|\boldsymbol{g}(X_i^{t_c..t_c+T'}; \theta))) = \\ &\sum_{k=1}^{C} -log(p(l_k|\boldsymbol{g}(X_i^{t_c...t_c+T'}; \theta)))\delta(y_i = l_k) + \\ &\sum_{k=1}^{C} -log(p(l_k|\boldsymbol{g}(X_i^{t_c...t_c+T'}; \theta)))p(l_k|\boldsymbol{g}(X_i^{t_{c'}...t_{c'}+T'}; \theta)) \end{aligned} \tag{7}$$

where the cross-entropy of the predictions $p(\cdot)$ of two neighboring crops, which respectively starts from t_c and $t_{c'}$ within the EEG signal of the trial denoted by X_i, are added into the loss function. The regularized loss function gives penalization when neighboring crops are given different predictions. This will reduce the differences among features of the neighboring input crops, forcing the network to focus on features which are stable across several neighboring input crops.

In the training phase, we collect the first crop of each trial i to train the network (randomly initialized) by optimizing Eq. (4) with the loss function defined by Eq. (5). The trained network will give an intermediate prediction on the first crop of each trial i, denoted by $p_{\mu=1}(l_k|\boldsymbol{g}(X_i^{t_c...t_c+T'}; \theta))$. Then we collect the next crop of each trial to train the network again. At this time, the network is initialized with weights learned from the previous crop of each trial and trained to minimize the regularized losses defined by Eq. (7). This training procedure is repeated until the last crop, i.e. Pth crop, of each trial is used to train the network.

The final decoding $f(X_i; \theta)$ of each trial i is calculated by

$$f(X_i; \theta) = \arg\max_{l_k} \sum_{\mu=1}^{P} p_\mu(l_k|\boldsymbol{g}(X_i^{\mu t_c...\mu t_c+T'}; \theta)) \tag{8}$$

3.3.2. Optimization and early stopping

Mini-batch gradient descent algorithm implemented with Adam optimizer is used to solve the optimization problem Eq. (4) with losses defined by Eq. (5) or Eq. (7). The learning rate η follows exponential decay function:

$$\eta(t) = t_0 + (t_1 - t_0)\exp(-t/\tau) \tag{9}$$

where t_0 and t_1 are the minimal and maximal allowed learning rates, respectively, and τ is the exponential decay factor. Here we allow the learning rate η decay from 0.003 to 0.0001 by a decay factor of 2000.

Early stopping strategy is also used for training. Corresponding training data set is divided into two folds: one for training and the other for validation. The optimization procedure is performed through two stages. At the first stage, we train the WaSF ConvNet for 100 iterations on the training fold only and test it on the validation fold. The weights with best validation accuracy are selected as the initial weights of the network at the second stage. The network is then trained on the whole training set (including both training and validation folds) until the loss drops to the same value as the training loss associated with best validation accuracy at the first stage.

3.3.3. Weight transfer

In the application of BCIs, collecting a large number of qualitative training examples for each BCI user is generally impractical. The main reason is that the training (calibration) session is so boring that the user will lose attention and produce an inaccurate cerebral response to the required action (Tu & Sun, 2012). But a short calibration session means only a few training examples of the user are available, which may lead classifiers, especially deep neural networks, to suboptimal or overfitting. Thus, how to solve this dilemma is a hot topic in BCI researches.

Besides the cropped training strategy, we also consider the transfer learning (Pan & Yang, 2010) to solve the problem of training deep neural networks with small labeled data sets. The transfer technique used in this paper is called subject transfer (Samek, Meinecke, & Muller, 2013), which is to use samples collected from other subjects (source subjects) to assist the subject whose brain signals will be classified in the test session (target subject). This naturally leads to a critical problem of how to use samples from source subjects to aid the target subject to train his specific model. Pretraining is utilized. We borrow parameters of several layers in source networks (networks that have been successfully trained using samples of source subjects) to initialize the target network (network that will be trained on the target subject for solving similar tasks).

The realization of the parameter transfer is divided into four stages. The first stage is to train a WasSF ConvNet per source subject using samples of the corresponding source subject. At this step, cropped training and early stopping strategies are used. The second stage is to initialize the WasSF ConvNet for the target subject. Assume there exist M source subjects w^s, the nth layer of the target network (denoted by w_n^t) is initialized by the weighed average of nth layers of M source networks:

$$w_n^t = \sum_{m=1}^{M} \rho_m w_{mn}^s \tag{10}$$

where w_{mn}^s denotes the connecting weights of the nth layer to the next layer in the source network m. Here $\sum_{m=1}^{M} \rho_m = 1$, where ρ_m represents the strength of the source subject network m contributing to the initialization of the target network. The third stage is to finetune the target ConvNet initialized above using the training samples of the target subject, calculating the optimal feature extractor and classifier for the target subject. The final

stage is to test the finetuned target network using the samples of the target subject collected during the test session. The parameter transfer is supposed to improve the classification performance on the small training data sets and boost the convergence speed of the deep network.

4. Experiments

In this section, we evaluated our proposed EEG decoder implemented by the deep convolution network on three public data sets for the motor imagery (MI) paradigm. Weights of the time–frequency convolution a and b are randomly initialized by uniform distribution in $U(1, 10)$ and $U(3, 30)$, respectively.

4.1. Data sets

BCIC IV 2a. BCI competition IV data set 2a (Brunner, Leeb, Muller-Putz, Schlogl, & Pfurtscheller, 2008) consists of EEG signals from 9 healthy subjects who were performing four different motor imagery tasks, i.e. imagination of the movement of the left hand, right hand, both feet and tongue. The signals were recorded by placing 22 electrodes distributed over sensorimotor area of the subject at a sampling rate of 250 Hz. The signals were bandpass-filtered between 0.5 Hz and 100 Hz. For each subject, two sessions were recorded on two different days, each containing 288 trials with 72 trials per class. At each trial, a cue was given in the form of an arrow pointing either to the left, right, down or up, corresponding to one of the four classes, to prompt the subject to perform the corresponding motor imagery task. Each trial lasted 4 s from the presence of cue till the end of motor imagery task. The first 3 seconds' data was extracted for further processing in this paper. In other words, in the following experiments, our WaSF ConvNet was trained using 288 trials. Each trial is a 22 dimensional time series containing 750 sampling points. The trained network was then tested by the remaining 288 trials.

BCIC IV 2b. BCI competition IV 2b (Leeb, Brunner, Muller-Putz, Schlogl, & Pfurtscheller, 2008) contains EEG signals from 9 subjects. The brain activities of the subject during motor imagery of either the left hand movement or the right hand movement were recorded using 3 electrodes at a sampling rate of 250 Hz. The signals were bandpass-filtered between 0.5 Hz and 100 Hz. For each subject, 5 sessions were recorded in five different days. For the first two sessions, 120 trials each were recorded, with 60 trials each class. At each trial, the subject performed the imagination of the cue-indicated hand movement over a period of 4 s, starting from the presence of the cue, without feedback. For the other three sessions, 160 trials per session were recorded with 80 trials each class. At each trial this time, the subject performed the imagination of the corresponding hand movement over a period of 4.5 s, starting from the presence of the cue, with online smiley feedback. Again, the time interval of the processed data was restricted to the time segment comprised between 0 s and 3 s starting from the cue. The first three sessions, containing 400 trials in total, were used as training sets, while the remaining two sessions, containing 320 trials, were used for test. Each trial is a 3 dimensional time series containing 750 sampling points.

Upper limb movement. This data set (Ofner, Schwarz, Pereira, Muller-Putz, & Zhang, 2017) consists of EEG data from 15 healthy subjects. Each subject was measured for 2 sessions on different days, performing the motor execution (ME) and the motor imagination (MI) tasks respectively. The signals were recorded using 61 electrodes, and were sampled with 512 Hz and bandpass-filtered between 0.01 Hz to 200 Hz. In this paper, only MI data is used to verify our proposed model. Each subject was cued to imagine either of the six movement types, including elbow flexion or extension, forearm supination or pronation, hand open or close. For each subject, 360 trials (60 trials per class) were recorded. The imagination of each movement lasted for 3 s, starting from the presence of the cue. The data of the first 1.5 s starting from the cue was used for the classification, since the work (Ofner et al., 2017) indicated that the MI classification becomes significant at $t < 0.81$ s. Thus the experiments were performed using the 360 trials, with each trial being a 61 dimensional time series containing 768 sampling points.

Table 1
Comparison of our method with baseline method in terms of κ value for BCIC IV 2a data set.

Subject	WaSF ConvNet (no weight transfer)	Baseline
s1	**0.62 ± 0.006**	0.41
s2	**0.32 ± 0.010**	0.09
s3	**0.71 ± 0.026**	0.61
s4	**0.40 ± 0.028**	0.29
s5	**0.59 ± 0.009**	0.11
s6	**0.33 ± 0.009**	0.27
s7	**0.66 ± 0.013**	0.43
s8	**0.72 ± 0.009**	0.44
s9	**0.69 ± 0.005**	0.63
mean	**0.56 ± 0.013**	0.36

4.2. Classification performance

Minimal pre-processing were performed on the data sets so that the convolution neural network can learn any transformations itself. The raw input signals, on both BCIC IV 2a and 2b data sets, were only bandpass-filtered with a third-order Butterworth filter using cutoff frequencies of 0.5 Hz and 38 Hz. The signals on the upper limb movement data set were also bandpass-filtered but using cut frequencies of 0.01 Hz and 38 Hz.

The three data sets only contain a small amount of training trials. However, each trial contains a large number of sampling points. Thus, many training examples can be obtained through cropping strategy. For both BCIC IV 2a and 2b data sets, we set $t_c = 0.25$ s (signals were considered as starting at 0 s), meaning three crops ($P = 3$) were extracted per input. For the upper limb movement data set, t_c was set as 0.13 s[2] and three consecutive crops ($P = 3$) were also obtained by sliding the time window 0.13 s each time within the input trial.

In this paper, kappa value $\kappa = (P_a - P_c)/(1 - P_c)$ were used as one of the evaluation metrics to assess the performance of the classifiers, where P_a is the proportion of the successful classification (identical to accuracy) and P_c is the proportion of random classification.

4.2.1. Comparison with the baseline method

We compared our WaSF ConvNet (without weight transfer) with the baseline method (Ref. Appendix). For the BCIC IV 2a and 2b data sets, training and test folds were provided separately. However, for the upper limb movement data set, training fold and test fold were not separated and thus we used a 10-fold cross validation to estimate the generalization performance.

For the BCIC IV 2a data set, as shown in Table 1, the performance of our proposed model is significantly different from the baseline method ($P = 0.008$, Wilcoxon signed-rank test). The kappa value of our method for all the 9 subjects is higher than that of the baseline method (increased about 55.6% in mean kappa). The maximal kappa discrepancy between our method and

[2] $t_c = 0.25$ s was also tried, but performs inferior average accuracy than 0.13 s does. Ofner et al. (2017) indicates that the MI classification becomes significant at $t < 0.81$ s.

Table 2
Comparison of our method with baseline method in terms of κ value for BCIC IV 2b data set.

Subject	WaSF ConvNet (no weight transfer)	Baseline
s1	**0.47 ± 0.026**	0.43
s2	**0.27 ± 0.031**	0.16
s3	**0.72 ± 0.015**	0.17
s4	**0.95 ± 0.008**	0.94
s5	**0.73 ± 0.027**	0.65
s6	0.45 ± 0.026	0.65
s7	**0.77 ± 0.012**	0.46
s8	**0.86 ± 0.014**	0.84
s9	0.62 ± 0.017	0.64
mean	**0.65 ± 0.020**	0.55

Table 3
Comparison of our method with baseline method in terms of κ value for the upper limb movement data set.

Subject	WaSF ConvNet (no weight transfer)	Baseline
s1	**0.17 ± 0.038**	0.06 ± 0.058
s2	**0.17 ± 0.044**	0.10 ± 0.079
s3	**0.17 ± 0.039**	0.09 ± 0.072
s4	**0.17 ± 0.054**	0.08 ± 0.072
s5	**0.16 ± 0.028**	0.07 ± 0.055
s6	**0.16 ± 0.038**	0.03 ± 0.084
s7	**0.17 ± 0.032**	−0.00 ± 0.051
s8	**0.17 ± 0.036**	0.10 ± 0.064
s9	**0.17 ± 0.034**	0.02 ± 0.071
s10	**0.18 ± 0.055**	0.01 ± 0.066
s11	**0.18 ± 0.042**	0.04 ± 0.026
s12	**0.16 ± 0.024**	0.02 ± 0.055
s13	**0.16 ± 0.028**	0.02 ± 0.050
s14	**0.17 ± 0.043**	0.01 ± 0.077
s15	**0.17 ± 0.040**	0.08 ± 0.065
mean	**0.17 ± 0.038**	0.05 ± 0.063

the baseline method is achieved for subject 5, with our method reaching as high as 0.59, 4.4 times higher than that of the baseline method.

For the BCIC IV 2b data set, as shown in Table 2, our method is not statistically different from the baseline method ($P = 0.123$, Wilcoxon signed-rank test). Our method beats the baseline over 7 out of 9 subjects and obtains significantly higher mean kappa value over the 9 subjects by 18.2% than that of the baseline method.

The upper limb movement data set is used to test the performance of our model on the multiclass classification of imagery movements. Table 3 shows the results. The mean κ value of our method is 0.17, significantly (2.4 times) higher than that of the baseline method. Note that the baseline method obtained κ values close to 0. For some subjects, the kappa values were even slightly negative, e.g. −0.0033 mean κ for subject 7. This implies the performance of the baseline method on the upper limb movement data set is more or less a random guess. Instead, the performance of our method on all subjects reached positive kappa, indicating the efficiency of our model. The performance of our model improved significantly comparing with the baseline method; nevertheless, it was still unsatisfying. One possible reason is that the classification task of this data set is rather difficult. Six types of motor imagery movements are involved in this data set. Among these six types of motor imagery movements, each two movements, i.e. elbow flexion and extension, forearm supination and pronation, and hand open and close, involve the same joints (i.e. muscle groups), making the classification task even more difficult (Ofner et al., 2017). Moreover, EEG signals of this data set contain 61 channels, much greater than the other two data sets. Thus the performance of our model on this data set is much worse than its performance on the other two data sets.

The experimental results show that out WaSF ConvNet significantly outperforms the baseline method. Moreover, the performance of our WaSF ConvNet is rather stable, i.e. the standard derivation is very small. Therefore the combination of time–frequency transformation, spatial filtering, and classification through deep convolutional neural network is a successful design for EEG signal decoding.

4.2.2. Feature analysis

The idea of deep ConvNets is usually attacked on their interpretability. However, our proposed deep ConvNet is different. The first valid layer of our ConvNet is designed to perform a time–frequency transformation while the second valid layer is designed to extract spatial information in the EEG signals. Consequently, our ConvNet implements an automatic joint space–time–frequency transformation to extract the features of EEG signals.

In this part, we delineate the extracted features of our proposed WaSF ConvNet. The EEG signals of subject 8 on BCIC IV 2a were used. No pre-processing and weight transferring were performed on the signals. We prepared the WaSF ConvNet as follows. We first divided the initialization range of the central frequencies b_η of the time–frequency convolutional kernels given by Eq. (1) into three bands, namely α-band (7–13 Hz), β-band (13–31 Hz), and γ-band (71–91 Hz). We then randomly divided the 25 time–frequency convolution units into three groups and associated each group with a frequency band by initializing the units of this group with random numbers within the corresponding frequency band. In other words, 9 units were initialized by random values within α-band (7–13 Hz), 8 units within β-band (13–31 Hz), while the rest 8 units within γ-band (71–91 Hz). The time–frequency convolution units will collect spectral amplitudes for different frequencies. Finally the ConvNet was trained using cropped training and early stopping strategies described in Section 3.3. The parameters a_η and b_η would be updated accordingly.

First, features extracted at the time–frequency convolutional layer are analyzed. We computed mean output of each time–frequency convolution unit across all trials and reassociated each unit with corresponding bands, according to the updated b_η. The result is given by Fig. 3(a). We can see that the grouping structure of time–frequency convolutional units is reserved in the successfully trained network. We then averaged the mean outputs of units within each band to obtain Fig. 3(b). From Fig. 3(a) and (b), we can clearly see that the firing rate of units in α-band and high γ-band is significantly higher than that in β-band. Finally, we separated the average output of units within each band according to their tasks (Fig. 3(c)). To achieve this, we grouped the trials according to their tasks, computed mean output of each unit over trials within the corresponding task group, and then averaged the mean outputs of units within each band. Fig. 3(c) indicates the outputs associated with α-band and high γ-band contribute greatly to the motor imagery decoding, confirming the findings in Schirrmeister et al. (2017).

We then analyze the features extracted at the spatial convolutional layer. For each trial, we averaged the output of each spatial convolution unit over time and selected three important units, which are associated with highest mean outputs. Then we counted the frequency of these selected units for trials within each class. Five most frequently selected units per class were chosen for visualization, given by Fig. 3(d). From Fig. 3(d), we can see that 6 out of 25 units were frequently activated for the four types of motor tasks. Among the 6 frequently activated units, the four types of motor tasks activated 4 units simultaneously but with different activation orders, indicating varied firing rates of units encode different types of movement imagination.

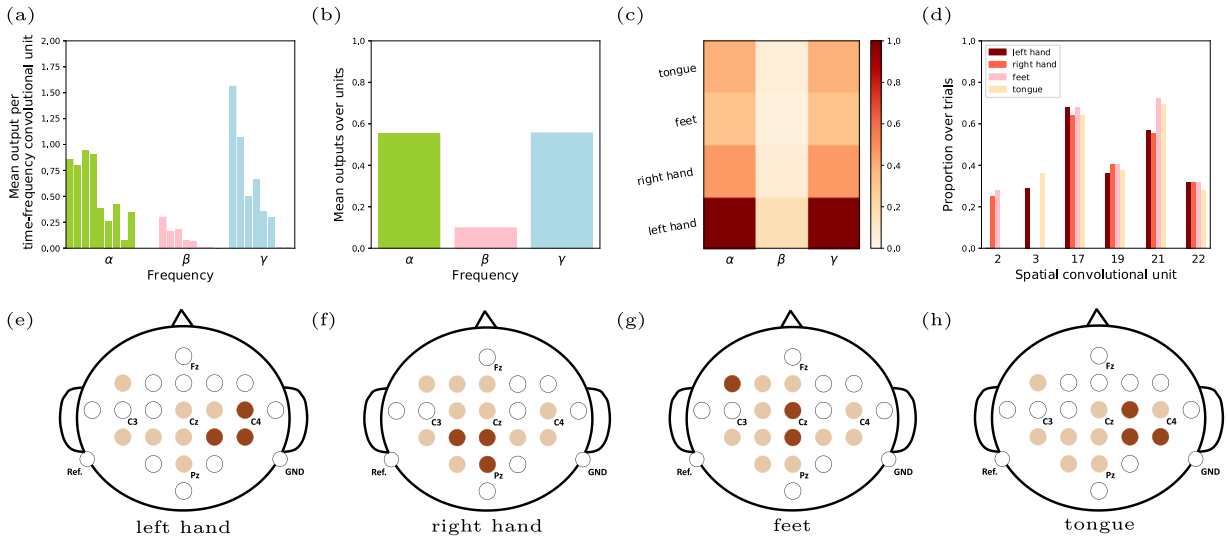

Fig. 3. Feature analysis on samples of subject 8 on BCIC IV 2a data set. (a) Mean output per time–frequency convolution unit across all trials is associated with either of α, β and high γ frequencies, according to the updated center frequency of this unit; (b) Average the mean outputs over all units within each band are plotted. The firing rate of units in α and γ rhythm is significantly higher than that in β; (c) separates the average outputs of units within each band according to different tasks. The left hand wins the highest correlation with frequency bands; (d) visualizes five most frequently selected units in spatial convolution layer for trials within each class. Varied firing rates of spatial convolution units encode different types of movement imagination; (e–h) show scalp maps, spatially mapping the learned features in the α frequency band for four classification tasks. Colorful circles denote the active electrodes for the corresponding classification task, in which dark red circles show the critical ones.

We further analyze the spatial distribution of class-related band power features by computing differences of synaptic connections from time–frequency convolutional units to spatial convolutional units between the trained and untrained models. For each task, the temporal units associated with α-band and five most frequently activated spatial units were selected for visualization. Each time–frequency convolutional unit is connected to one spatial convolutional unit through a convolutional kernel with size $1 \times E$. Each parameter within the kernel corresponds to one EEG electrode. The positive weight (kernel parameters) change of each connection comparing the trained network with untrained network indicates the corresponding electrode is class-related for the corresponding band. For each connection between the selected spatial unit and one temporal unit, we collected three most important electrodes (with the largest positive changes). Then we counted the frequency of the selected electrodes for all the connections from the temporal units within α-band to this spatial unit. For each spatial unit, we collected three most frequently appeared electrodes. Important class-related electrodes corresponding to all spatial units were marked in the scalp map and three most important electrodes were marked in dark red, given by Fig. 3(e–h), respectively.

The results given by Fig. 3(e–h) suggest that the four types of mental tasks activate different areas of the brain to emit α rhythm. The imagination of the left hand movement mainly activated the right part of the primary motor cortex around C4 (Fig. 3(e)), while the right hand imagery movement mainly activated the left part (Fig. 3(f)). The feet imagery movement activated the central primary motor cortex, which is specific around Cz (Fig. 3(d)). The area below Cz is called the primary somatosensory cortex, representing the brain activity triggered by the imagination of the tongue movement. In general, our network can extract α rhythm at channels that directly related to the task performed by the subject.

4.2.3. Weight transfer

We evaluated the weight transfer learning strategy presented in Section 3.3.3 under different choices, trying to identify an optimal choice that can provide positive transfer. For a data set, the subject to be tested is the target subject. Only training trials of this subject were involved in the learning process of the network. The test trials were reserved for test only. Other subjects within this data set are source subjects. Both training trials and test trials can be used to assist the learning process of the target network. The performance of the target network will be compared with that of the network without weight transfer reported in Section 4.2.1.

Choice of the transferred layers on the BCIC IV 2a. There are three layers in the WaSF ConvNet that can be transferred. To identify which layers to transfer, we performed the following experiments on BCIC IV 2a data set. Each subject was treated as the target subject except subject 8, using subject 8 as the source subject[3] since the WaSF ConvNet on subject 8 obtains the best performance (see Table 1). Reusing the weights of the best source network is supposed to improve the performance of the target subject.

Table 4 gives the experimental results of different transferring choices. That is to initialize the layer (layers) of the target network with the transferred layer (layers) of the source subject, while keeping other layers of the target network randomly initialized. After initialization, the target network was finetuned on the training trials only and tested on the separate test trials of the corresponding subject. From Table 4, we can see that transferring time–frequency convolution layer and spatial convolution layer at the same time improves 2.7% kappa value over the WaSF ConvNet without transfer, significantly more than other choices. This confirms that the combination of temporal, spectral and spatial features leads to improved classification performance of EEG signals. Transferring weights from the time–frequency convolution layer and spatial convolution layer was selected for further study.

Choice of the strengths of source subjects on the BCIC IV 2a. Source subjects contribute to the initialization of the target network based on specific strengths (ρ in Eq. (10)). We explored two

[3] Transferring weights from the next best performance subjects (i.e. subject 3 or 9) does not reach superior performance to subject 8.

Table 4
Transferring the source network with different layers on the BCIC IV 2a data set.

Subject	No Transfer		Time–frequency layer		Spatial layer		Dense layer		Time–frequency& spatial layers	
	κ	acc	κ	acc	κ	acc	κ	acc	κ	acc
s1	0.62	0.71	**0.62**	**0.71**	0.59	0.69	0.57	0.68	0.61	**0.71**
s2	0.32	0.49	**0.34**	**0.51**	0.31	0.48	0.31	**0.49**	**0.32**	**0.49**
s3	0.71	0.78	**0.74**	**0.81**	**0.71**	**0.78**	0.68	0.76	**0.72**	**0.79**
s4	0.40	0.55	**0.40**	**0.55**	0.38	0.54	0.39	0.54	**0.48**	**0.61**
s5	0.59	0.69	**0.60**	**0.70**	0.57	0.68	**0.59**	**0.69**	0.56	0.67
s6	0.33	0.50	**0.35**	**0.51**	0.32	0.49	**0.35**	**0.51**	**0.36**	**0.52**
s7	0.66	0.74	**0.66**	**0.75**	**0.66**	**0.74**	**0.66**	**0.74**	**0.68**	**0.76**
s8	0.72	0.79	–	–	–	–	–	–	–	–
s9	0.69	0.77	0.67	0.75	**0.69**	**0.77**	0.66	0.74	**0.71**	**0.78**
mean	0.558	0.668	**0.567**	**0.676**	0.550	0.662	0.548	0.660	**0.573**	**0.680**

Table 5
Transferring the source network with different strengths on the BCIC IV 2a data set.

Subject	Test trials		Validation trials	
	κ	acc	κ	acc
s1	0.61	0.71	**0.63**	**0.72**
s2	0.32	0.49	**0.32**	**0.49**
s3	0.72	0.79	**0.75**	**0.82**
s4	0.48	0.61	0.44	0.58
s5	0.56	0.67	**0.60**	**0.70**
s6	0.36	0.52	**0.38**	**0.54**
s7	0.68	0.76	**0.69**	**0.77**
s8	0.72	0.79	0.71	**0.79**
s9	0.71	0.78	**0.73**	**0.80**
Mean	0.573	0.680	**0.583**	**0.690**

options to obtain the strength for each source subject. Simulations transferred the time–frequency and spatial convolutional layers of the best source network. For the first option, termed as "test trials", we used the kappa values on test trials to find the best source network as we did in Table 4. For the second option, termed as "validation trials", we used the kappa values on validation set to find the best source network. Table 5 shows that the use of validation kappa values as strengths obtains better performance. One possible reason is that the validation set contains trials from both training session and test session of the source subject, providing better overall evaluation of the source network.

Consequently, in the following experiments, we applied two rules: (1) transferring the time–frequency and spatial convolution layers together; (2) obtaining the strength ρ by calculating the kappa values on the validation set randomly drawn from both training and test sessions.

Choice of transferring strategies on the BCIC IV 2a. Besides transferring the parameters of the best source network to the target network (top1), we also explored other choices, including transferring the weighted mean parameters of the best three source networks to the target network (top3), and transferring the mean (mean) or weighed mean (weighted) parameters of all the source networks to the target network. Note that the weights are given by the validation kappa values and properly normalized (divided by their sum). Table 6 shows the results.

Among all the choices, transferring the best source network, termed as top1 achieves the largest improvements, obtaining significantly ($P = 0.024$, Wilcoxon signed-rank test) better performance than the WaSF ConvNet without weights transfer. By employing the top1 subject transfer strategy, 8 out of 9 subjects obtain superior performance to their original methods. The corresponding mean kappa values over all subjects reaches 0.583, 4.5% higher than the original results.

We also validated the top1 subject transfer on the BCIC IV 2b data set, as shown in Table 7. Results with the weight transfer are not significantly different from the original one ($P =$ 0.433, Wilcoxon signed-rank test). WaSF ConvNet reaches a mean kappa value of 0.657, slightly greater than the mean value of no transferring process (0.649). This may be because the amount of data per class is big enough to train the network, such that transferring weights from the source to the target does not obtain good performance on BCIC IV 2b.

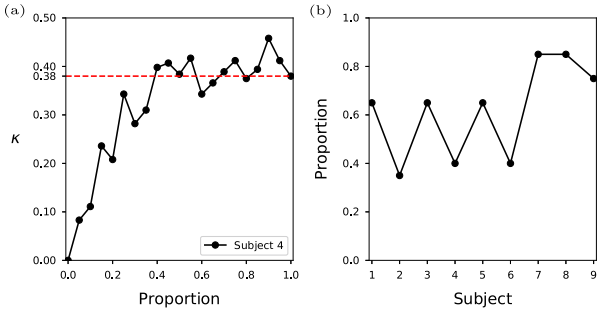

Fig. 4. The minimum amount of demanded training trials on the BCIC IV 2a data set. (a) When giving different proportion of trials from subject 4 to train the WaSF ConvNet, the corresponding kappa coefficients are calculated in the test session. The subject 4 reaches the benchmark (*red line*), obtained from fully trained model without the weight transfer, at about 40% point; (b) shows the minimal training sets of all subjects.

All above experiments show that transferring weights of the time–frequency and spatial convolutional layers of the best source network evaluated on a validation set randomly drawn from the whole data set is a successful transfer strategy.

4.2.4. Reduced size of demanded training trials

By using transfer strategies we may reduce the number of training trials without jeopardizing the performance. In the field of BCI, it is advantageous to use the reduced size of training trials, since this would reduce the calibration time of the BCI system.

We took the subject 4 of the BCIC IV 2a to demonstrate how the classification performance changes when we finetune the target network using only part of the training trials provided by the target subject. The results are given by Fig. 4(a). As the number of trials used to train the network increases, the classification performance of the WaSF ConvNet is improved. The performance of the weight-transferred network reaches the same performance as the original fully trained network (randomly initialized and trained using the training trials) only using about 40% of the provided training trials. Fig. 4(b) gives the minimal training trials of all subjects in the BCIC IV 2a data set that allow the weight-transferred network to reach the performance of the original fully trained network. As one can see in Fig. 4(b), the trials that are used to train the weight-transferred network can be reduced by 20% at least and to 62% in average. This indicates that for a new subject, by using weight transfer, the calibration time of the BCI system can be shortened by 20%, significantly increasing its practicability.

Table 6
Transferring weights with different strategies in BCIC IV 2a data set.

Subject	No Transfer		Top1		Top3		Weighted		Mean	
	κ	acc	κ	acc	κ	acc	κ	acc	κ	acc
s1	0.62	0.71	**0.63**	**0.72**	0.61	**0.71**	**0.62**	**0.71**	**0.64**	**0.73**
s2	0.32	0.49	**0.32**	**0.49**	0.30	0.47	0.31	**0.49**	0.30	0.48
s3	0.71	0.78	**0.75**	**0.82**	**0.71**	**0.79**	0.70	0.77	**0.73**	**0.80**
s4	0.40	0.55	**0.44**	**0.58**	0.38	0.53	0.35	0.51	0.34	0.51
s5	0.59	0.69	**0.60**	**0.70**	0.58	0.68	0.58	0.68	**0.59**	**0.69**
s6	0.33	0.50	**0.38**	**0.54**	**0.36**	**0.52**	**0.33**	**0.50**	**0.34**	**0.51**
s7	0.66	0.74	**0.69**	**0.77**	**0.69**	**0.76**	**0.69**	**0.77**	**0.69**	**0.76**
s8	0.72	0.79	0.71	**0.79**	**0.72**	**0.79**	0.69	0.77	0.70	0.77
s9	0.69	0.77	**0.73**	**0.80**	**0.73**	**0.80**	**0.69**	0.76	**0.69**	**0.77**
Mean	0.558	0.668	**0.583**	**0.690**	**0.564**	**0.672**	0.551	0.662	**0.558**	**0.669**

Table 7
Transferring weights on the BCIC IV 2b data set.

Subject	No transfer		Top1	
	κ	acc	κ	acc
s1	0.47	0.74	**0.51**	**0.75**
s2	0.27	0.64	**0.27**	**0.64**
s3	0.72	0.86	**0.74**	**0.87**
s4	0.95	0.98	**0.96**	**0.98**
s5	0.73	0.86	0.71	**0.86**
s6	0.45	0.73	0.41	0.70
s7	0.77	0.89	**0.81**	**0.90**
s8	0.86	0.93	0.84	0.92
s9	0.62	0.81	**0.66**	**0.83**
Mean	0.649	0.827	**0.657**	**0.828**

Table 8
Comparison of the ultimate mean performance of the WaSF ConvNet with State-of-the-art Results.

Data sets	Models	κ	acc
BCIC IV 2a	WaSF ConvNet	0.58	0.69
	FBCSP	0.57	0.67
	OSTP	0.60	0.72
	ConvNets	–	**0.74**
BCIC IV 2b	WaSF ConvNet	**0.66**	**0.83**
	FBCSP	0.60	0.79
	OSTP	0.60	0.78
	ConvNets	0.63	–
Upper limb	WaSF ConvNet	**0.17**	**0.31**
	DSP	–	0.27

4.2.5. Comparison with the state-of-the-art results

In Table 8, we compared the final classification performance of our proposed method with excellent performance of the state-of-the-art methods,[4] such as FBCSP (Kai et al., 2008), OSTP (Ang et al., 2012), ConvNets (Schirrmeister et al., 2017), and DSP (Ofner et al., 2017).

For the BCI competition IV data set 2a, our approach obtains accuracy in a very similar range as FBCSP, whereas slightly worse than ConvNets with 0.74 mean accuracy. However on the two additional data sets, the BCI competition IV data set 2b and the upper limb movement data set, WaSF ConvNet both reach better performance than the state-of-the-art results (about 4.8% mean kappa and 14.8% mean accuracy higher, respectively).

5. Conclusions

In this paper, we have proposed a convolutional network (ConvNet) combining wavelet transformation with spatial filtering to decode EEG signals end-to-end. Inspired by wavelet transformation, we design Morlet wavelet-like kernels for the convolution process in our deep network. Each wavelet kernel only has two free parameters to learn, i.e. the bandwidth of the Gaussian time window and the center frequency, significantly reducing the number of parameters compared with classical convolutional kernels and thereby decreasing the risk of overfitting. The utilization of wavelet kernels also endows the features learned at the corresponding layer with a clear interpretation (spectral amplitude). The features learned in our network have been shown matching the neuronal activity of sensorimotor areas for motor imagery EEG data. Experimental results on three public data sets reveal that our convolutional network significantly outperforms the method using manually designed wavelet spectral amplitudes with a separate classifier.

We further solve the contradiction between large amount of demanded data for training deep ConvNets and small labeled data collected in the BCI experiments by subject-to-subject weight transfer, which borrows weights from existing subjects to initialize the network for a new subject. The proposed strategy has been verified to be with beneficiary transfer. The results suggest that, through using the proposed transfer learning strategy, a BCI system is able to adapt to a new subject with shorter training sessions. This will make the BCI system more user friendly, improving the applicability of the BCIs. Additionally, with the help of weight transfer, our approach has obtained superior classification performance to the state-of-the-art methods, indicating that jointly learning features in space–time–frequency domains and the classifying will be a promising attempt for EEG decoding in BCI.

There still exist several points to be improved. For example, learning the network requires quite a long time comparing with the traditional process (i.e. classifying based on handcrafted features). We will try to reduce the training time of the network using paralleling training strategies in the future. Moreover the evaluation of our WaSF ConvNet has been limited to motor imagery EEG data. Our future work will extend the current network to a generic classification method for multivariate time series data.

Acknowledgments

This work was supported by the National Key Research and Development Program of China [grant number 2016YFC0801808]; the Frontier Science research project of the Chinese Academy of Sciences [grant number QYZDY-SSW-JSC005]; CAS Pioneer Hundred Talents Program, China [grant number Y8F1160101]; and the State Key Laboratory of Robotics, China [grant number Y7C120E101].

[4] Some methods only report either of evaluation measures (κ or accuracy) in the literatures, thus we replace the absent value with — in Table 8.

Appendix. EEG decoding model based on morlet wavelet transformation

The baseline method decomposes the signals into several frequency bands of time–frequency representation using complex Morlet wavelet transformation, and then extracts the mean spectral amplitude (Rotermund et al., 2013) in the corresponding frequency band as input features to a SVM classifier.

Example EEG Data is first band-pass filtered between 8 Hz and 30 Hz using a 5th order Butterworth filter. The preprocessed signals are denoted by $x_{m,n}(t)$, where m denotes the trial, n the electrode, and t the time. The preprocessed signals are then convolved with complex Morlet wavelets $w(t, f_0)$ to obtain the wavelet coefficients $a_{m,n}(t, f_0)$.

$$a_{m,n}(t, f_0) = \int_{-\infty}^{\infty} w(\tau, f_0) x_{m,n}(t - \tau) d\tau \tag{11}$$

The central frequency f_0 needs to be manually determined. In this paper, we choose 8 frequencies, i.e. {8, 10, 12, 13, 15, 20, 25, 30}. With one specified central frequency, we can obtain one series of complex wavelet coefficients denoted by $a(t) = \alpha(t) \exp^{i\phi(t)}$, where α denotes the amplitude and ϕ represents the phase. Average spectral amplitude is then computed by $A = \frac{1}{t_1 - t_0} \sum_{t=t_0}^{t_1} \alpha(t)$ and chosen as the feature.

Features are finally input to a support vector machine (SVM) for classification. The regularization parameter was manually selected by exhaustive search using 5-fold cross-validation. Candidates of the regularization parameter are the set of {0.0001, 0.005, 0.01, 0.05, 0.1, 1, 5, 10, 20, 50, 100, 500, 600, 1000}.

References

Adeli, H., Zhou, Z., & Dadmehr, N. (2003). Analysis of EEG records in an epileptic patient using wavelet transform. *Journal of Neuroscience Methods*, *123*(1), 69–87.

Aghaei, A. S., Mahanta, M. S., & Plataniotis, K. N. (2016). Separable common spatio-spectral patterns for motor imagery BCI systems. *IEEE Transactions on Biomedical Engineering*, *63*(1), 15–29.

Ang, K. K., Chin, Z. Y., Zhang, H., & Guan, C. (2012). Mutual information-based selection of optimal spatial-temporal patterns for single-trial EEG-based BCIs. *Pattern Recognition*, *45*(6), 2137–2144.

Barachant, A., Bonnet, S., Congedo, M., & Jutten, C. (2012). Multiclass brain-computer interface classification by Riemannian geometry. *IEEE Transactions on Biomedical Engineering*, *59*(4), 920–927.

Blankertz, B., Tomioka, R., Lemm, S., Kawanabe, M., & Muller, K. R. (2007). Optimizing spatial filters for robust EEG single-trial analysis. *IEEE Signal Processing Magazine*, *25*(1), 41–56.

Brunner, C., Leeb, R., Muller-Putz, G. R., Schlogl, A., & Pfurtscheller, G. (2008). *BCI Competition 2008-Graz Data Set A* (pp. 136–142). Institute for Knowledge Discovery, Graz University of Technology.

Clevert, D., Unterthiner, T., & Hochreiter, S. (2016). Fast and accurate deep network learning by exponential linear units (ELUs). In *International conference on learning representations*.

Congedo, M., Barachant, A., & Bhatia, R. (2017). Riemannian geometry for EEG-based brain-computer interfaces; a primer and a review. *Brain-Computer Interfaces*, *4*(3), 155–174.

Ferrante, A., Gavriel, C., & Faisal, A. (2015). Data-efficient hand motor imagery decoding in EEG-BCI by using morlet wavelets and common spatial pattern algorithms. In *International IEEE/EMBS conference on neural engineering* (pp. 948–951).

Hong, J., Qin, X., Bai, J., Zhang, P., & Cheng, Y. (2015). A combined feature extraction method for left-right hand motor imagery in BCI. In *IEEE international conference on mechatronics and automation* (pp. 2621–2625).

Kai, K. A., Zheng, Y. C., Zhang, H., & Guan, C. (2008). Filter bank common spatial pattern (FBCSP) in brain-computer interface. In *IEEE international joint conference on neural networks* (pp. 2390–2397).

Kingsbury, N. (1998). The dual-tree complex wavelet transform: A new technique for shift invariance and directional filters. *Image Processing*, 319–322.

Kousarrizi, M. R. N., Ghanbari, A. R. A., Teshnehlab, M., Shorehdeli, M. A., & Gharaviri, A. (2009). Feature extraction and classification of EEG signals using wavelet transform, SVM and artificial neural networks for brain computer interfaces. In *International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing* (pp. 352–355).

Kumar, S., Sharma, A., Mamun, K., & Tsunoda, T. (2017). A deep learning approach for motor imagery EEG signal classification. In *3rd Asia-Pacific world congress on computer science and engineering* (pp. 34–39).

Lee, H., & Kwon, H. (2016). Single-trial EEG RSVP classification using convolutional neural networks. In *SPIE Defense + Security* (p. 983622).

Leeb, R., Brunner, C., Muller-Putz, G. R., Schlogl, A., & Pfurtscheller, G. (2008). *BCI Competition 2008-Graz data set B*. Institute for Knowledge Discovery, Graz University of Technology.

Ma, T., Li, H., Yang, H., Lv, X., Li, P., Liu, T., Yao, D., & Xu, P. (2016). The extraction of motion-onset VEP BCI features based on deep learning and compressed sensing. *Journal of Neuroscience Methods*, *275*, 80–92.

Meng, M., Lu, S., Man, H., Ma, Y., & Gao, Y. (2015). Feature extraction method of motor imagery EEG based on DTCWT sample entropy. In *Control conference* (pp. 3964–3968).

Molina, G. N. G., Ebrahimi, T., & Vesin, J. M. (2003). Joint time-frequency-space classification of EEG in a brain-computer interface application. *Eurasip Journal on Advances in Signal Processing*, *2003*(7), 713–729.

Nicolas-Alonso, L. F., & Gomez-Gil, J. (2012). Brain computer interfaces, a review. *Sensors*, *12*(2), 1211–1279.

Ofner, P., Schwarz, A., Pereira, J. L., Muller-Putz, G. R., & Zhang, D. (2017). Upper limb movements can be decoded from the time-domain of low-frequency EEG. *PLoS One*, *12*(8), e0182578.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359.

Rotermund, D., Ernst, U. A., Mandon, S., Taylor, K., Smiyukha, Y., Kreiter, A. K., & Pawelzik, K. R. (2013). Toward high performance, weakly invasive brain computer interfaces using selective visual attention. *Journal of Neuroscience the Official Journal of the Society for Neuroscience*, *33*(14), 6001–6011.

Samek, W., Meinecke, F. C., & Muller, K. (2013). Transferring subspaces between subjects in brain–computer interfacing. *IEEE Transactions on Biomedical Engineering*, *60*(8), 2289–2298.

Sanei, S., & Chambers, J. A. (2007). *EEG signal processing*. Wiley-Blackwell.

Schirrmeister, R. T., Springenberg, J. T., Fiederer, L. D. J., Glasstetter, M., Eggensperger, K., Tangermann, M., Hutter, F., Burgard, W., & Ball, T. (2017). Deep learning with convolutional neural networks for EEG decoding and visualization. *Human Brain Mapping*, *38*, 5391–5420.

Tang, Z., Li, C., & Sun, S. (2017). Single-trial EEG classification of motor imagery using deep convolutional neural networks. *Optik - International Journal for Light and Electron Optics*, *130*, 11–18.

Tu, W., & Sun, S. (2012). A subject transfer framework for EEG classification. *Neurocomputing*, *82*, 109–116.

Yen, G. G., & Lin, K. (2000). Wavelet packet feature extraction for vibration monitoring. *IEEE Transactions on Industrial Electronics*, *47*(3), 650–667.

Yger, F., Berar, M., & Lotte, F. (2017). Riemannian approaches in brain-computer interfaces: a review. *IEEE Transactions on Neural Systems & Rehabilitation Engineering*, *25*(10), 1753–1762.