Proceedings of TENCON 2018 - 2018 IEEE Region 10 Conference (Jeju, Korea, 28-31 October 2018)

Decoding ECoG Signal with Deep Learning Model Based on LSTM

Anming Du
Dept. of Computer Science and Technology
Nanjing University of Posts and Telecommunications
Nanjing, China
15005187213@163.com

Shuqin Yang
Dept. of Electronic and Optical Engineering
Nanjing University of Posts and Telecommunications
Nanjing, China
122662848@qq.com

Weijia Liu
Schol. of Bell Honors
Nanjing University of Posts and Telecommunications
Nanjing, China
1042514377@qq.com

Haiping Huang
Dept. of Computer Science and Technology
Nanjing University of Posts and Telecommunications
Nanjing, China
hhp@njupt.edu.cn

***Abstract*—Currently, brain-computer interface technology (BCI) has been widely used in brain disease diagnosis and motor disabilities recovery. In this paper, it proposes a novel scheme that using deep learning model based on Long Short-Term Memory (LSTM) to extract and classify ECoG signals. First, it preprocesses the ECoG and voltage signal when the subject's finger is bent. Second, according to the time characteristics of the ECoG, it designs a 6-layer deep learning model to classify ECoG signals directly, while avoiding the time-consuming of feature extraction. This method is applied to an open ECoG dataset and experimental results achieve 83.3% accuracy over 5 categorical ECoG data. The accuracy is significantly higher than traditional linear analysis and ordinary machine learning methods. To demonstrated the feasibility of this proposal, it is applied to real-time control of the mechanical arm.**

***Keywords*—brain-computer interface, deep learning, LSTM, recurrent neural network, motor disabilities recovery**

I. Introduction

In recent years, Brain Computer Interface (BCI) has applied widely to diagnose brain diseases and cure motor disabilities. BCI establishes a direct link between the human brain and external auxiliary equipment, which could be directly controlled by the mind. The emergence of brain-computer interface technology has brought new treatments and rehabilitation methods for patients with dyskinesia. There is no doubt that BCI technology has a significant medical value.

According to the different sources of signals, BCI systems are divided into implantable systems and non-implantable ones. The non-implantable system samples EEG signal by the way of placing the electrodes on the scalp of the brain. The implantable system obtains ECoG (electrocorticography) via laying electrode arrays in specific regions of the cerebral cortex. Comparatively speaking, EEG signal is easy to be collected and the failure risk is smaller, though whose signal quality is poor. ECoG can directly reflect the local electrical signals of the brain, and has excellent performance on signal-to-noise ratio. In view of the characteristics and advantages of the ECoG signal, the system based on ECoG is recognized as an optimal scheme in clinical rehabilitation field.

Recurrent Neural Network (RNN) is an extension of a conventional feedforward neural network, which is able to handle a variable-length sequence input. RNN handles the variable-length sequence by having a recurrent hidden state whose activation at each time is dependent on that of the previous time. RNN can use its internal memory to process input sequences of arbitrary timing, which can make it easier to handle handwriting recognition and speech recognition. Human brain signal is a typical time series signal, so that RNN can make full use of the temporal characteristics of ECoG signal during feature extraction.

This paper is conducted to use LSTM and deep learning model to extract features and classify the ECoG. The rest of this paper is organized as follows: Section 2 introduces our related work and Section 3 describes the LSTM-based ECoG classification method in detail, which includes data preprocessing, design and tuning of deep learning model, and comparisons with other methods. Section 4 provides an application case of our proposal and Section 5 is the conclusion of this paper.

II. Related work

The traditional ECoG classification method firstly extracts the time and frequency domain information by hand, then uses different kinds of machine learning methods to establish the mapping relationship between the ECoG and the finger bending motion. S. Samiee et al.[1] used Principal Component Analysis (PCA) and Scatter Matrix to extract features, and utilized support vector machine (SVM) and Fisher linear discrimination to classify ECoG signals. N.Y. Liang [2] used the linear regression method to parse the ECoG signal by dividing into some specific frequency bands. They made a criterion by the correlation coefficient between the prediction result and the actual value. Z. Wang [3] took advantage of prior supervised convolutional stacked auto-encoders (PCSA) to match the brain electrical signals and finger movements. A. Elgharabawy [4] obtained features of ECoG signals by Shift Invariant Wavelet Packet Decomposition and Multi-taper Time-Frequency Spectrum, and classified results by SVM. The correlation coefficient between the prediction results and the actual values of the model reaches 0.82. A. Marjaninejad [5] filtered three different frequency bandwidths of the ECoG signal and amplitude-modulated the raw signal, and then input the processed signal into linear regression model (LRM) and artificial neural network (ANN) to recognize and classify. Due to the separation of preprocessing, feature extraction and classification in traditional methods, the timeliness of signal processing is poor and the recognition rate is unsatisfactory. How to process EEG and ECoG signals faster and more accurately has become the focus of current research.

RNN is widely used in various fields such as speech recognition, handwriting recognition and video analysis. RNN has memory and storage function which make full use of timing information. Traditional RNN model still has some problems such as gradient disappearance and gradient explosion in model training. LSTM [6] alleviates the above problems in a certain degree and now has been widely applied to the analysis of EEG signals. X. Zhang [7] used two LSTM layers to process a public EEG data set, and designed a deep learning model base on LSTM. Finally, they applied the model into home automation which would be helpful for the people with physically disabilities and senior citizens.

Deep learning [8] can simulate the internal relations which are hidden in the raw data. It has multiple information processing modules and captures the complex relationships through layered architecture and stacking. LSTM has more long-term correlation than traditional RNN, which shows better performance in deal with time series signal. The innovations of this paper can be summarized as follows:

- Introduce LSTM to handle ECoG.
- According to the characteristics of ECoG signals, a 6-layer deep learning model is specifically designed to extract features and realize classification recognition.
- The model is used to real-time control of robot hand, which verify the feasibility of our proposal.

III. ECOG CLASSIFICATION METHOD BASED ON LSTM

A. Experimental Data

In order to verify the validity of the method proposed in this paper, it is applied to the classification of public data sets (Data set of BCI Competition IV [9]). We will analyze the data set that can predict the Finger bending action by ECoG signal. The following briefly describes the data set.

The data set collected ECoG data from three patients with epilepsy. Each subject had a miniature electrode array placed on the cerebral cortex. When the subject performed a finger bending task, the data of finger bending motion and the corresponding ECoG signal were recorded synchronously. Electrode arrays is composed of 8*6 or 8*8 grids, and the number of electrodes on the three subjects' cortex was 62, 48, and 64, respectively.

During the experiments, the subject was asked to bend the corresponding finger according to the cue on the computer screen. Typically for each cue, the subjects flexed the finger 3~5 times lasting 2s followed by a rest period of 2s. There were 30 actions for each finger resulting in 600s records for each subject. The first 400s records were used as training set and the last 200s records were used as testing set.

In the experiments, the ECoG signal was recorded by the BCI2000 system. The sampling frequency was 1000 Hz, and band-pass filtered between 0.15 Hz and 200 Hz. The finger movements were recorded using a data glove and sampled at 25 Hz. Figure 1 provides an example of the visualization of the ECoG signals and the corresponding finger movement time course from subject 1.

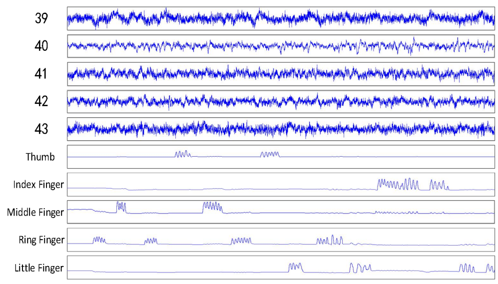

Figure 1: ECoG signals from channel 39 to 43 and the corresponding movement time courses for each finger

B. Data Preprocessing

Since the most useful information of ECoG signal is in the mid-to-high frequency band, so we firstly use FIR filter to extract 7 to 175 Hz information for each channel signal. Inspired by the rate coding approach used in spike train decoding, we will modulate the filtered data to achieve amplitude modulation at a high bandwidth. And then we get a band-specific AM as the descriptor for ECoG signal decoding, which is defined as the sum of the square voltage of the band-specific ECoG signals in a time window Δt. The formula is shown in (1):

$$x(t_n) = \sum_{t=0}^{\Delta t} v^2(t_n + t) \quad (1)$$

In (1), v represents raw ECoG amplitude signal, and x is the modulated signal. Where $\Delta t = t_{n-1} - t_n$ and we set Δt = 40 ms so that the resulting band-specific AM features have the same sampling rate as that of the data glove position measurements.

In this paper, finger's activity is divided into two different states: Active and Idle. Active state means the subject flexes the corresponding finger according to the cue on the screen, while idle state represents no any movement. In the experiment, only one finger can be active state at some time point. We use Threshold1 and Threshold2 to respectively determine whether the finger is active or idle:

$$\begin{aligned} &\textit{if} \quad Vi>=Threshold1 \\ &\quad y=i \\ &\textit{else if} \quad Vi<Threshold2 \\ &\quad y=0 \end{aligned} \tag{2}$$

Where i (i=1, 2, 3, 4, 5) denotes finger's number. For example, 1 denote a thumb, 2 represent index finger and so on. V_i is action voltage amplitude of the i-th finger. Threshold1 and Threshold2 are threshold values to determining a finger in active or idle state. In this paper, we set Threshold1 = 2mv and Threshold2 = 1mv.

After modulating raw ECoG data and labeling active finger, 15000 pieces of tagged data were obtained, which are selected to evaluate the model. Among the data, 13000 pieces of data are used as training sets and other 2000 pieces of data are treated as test sets. Taking subject 1 as an example, the training set is a 12000*63 , where 1 to 62 columns are modulated ECoG data, and 63 column is regarded as finger activity tag.

C. LSTM

LSTM is a special RNN, which can solve gradient disappearance in circulating neural networks. When dealing with long-term sequence signals, traditional RNN network structure is difficult to obtain valid information from previous nodes, which always resulting in poor prediction accuracy.

The main idea of LSTM cell is to create several gates to control the memory of special information. The gates (input gate, forget gate, and output gate) are used to control when and how the state of the memory cells is updated, where the memory cells are used to store state information.

The internal structure of the LSTM unit is shown in Figure 2. x_t denotes the input, and h_t and h_{t-1} are the outputs of time point at t and t-1 respectively. σ represents sigmoid function, and tanh denotes hyperbolic tangent activation function. C_t is the cell's storage state, and $\odot$ represents dot-multiply. Input gate is used to determine the information to be updated, and forget gate is used to control whether to save or discard the information from the previous moment. LSTM Unit updates the storage state C_t through the forget gate and the input gate. Output gate is used to determine information in the storage state C_t whether to flow the next unit.

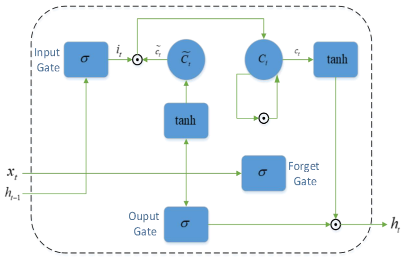

Figure 2: The structure of LSTM

LSTM calculation process is accomplished by the following formula:

$$\begin{cases} f_t = \sigma(W_{xf} \bullet x_t + W_{ht} \bullet h_{t-1} + b_f) \\ i_t = \sigma(W_{xi} \bullet x_t + W_{hi} \bullet h_{t-1} + b_i) \\ c_t = f_t \odot c_{t-1} + i_t \odot \widetilde{c_t} \\ \widetilde{c_t} = \tanh(W_{xc} \bullet x_t + W_{hc} \bullet h_{t-1} + b_c) \\ o_t = \sigma(W_{xo} \bullet x_t + W_{ho} \bullet h_{t-1} + b_o) \\ h_t = o_t \odot \tanh(c_t) \end{cases} \tag{3}$$

Where x is the input, W_x and b denotes the weight and biases of each gate, respectively. We defined h_{t-1} as previous moment's output and W_h is the weight between h and h-1. c_t is the output of the state C_t, and $\widetilde{c_t}$ is the output of temporary state $\widetilde{C_t}$.

D. Deep learning model base on LSTM

The deep learning model includes three parts: input layer, hidden layer and output layer. The input layer is the first layer of the model, and the hidden layer lies in the model's middle. The LSTM layer is the part of the hidden layer, and the output layer is the final layer of the model. Our learning model will be elaborated specifically as follows .

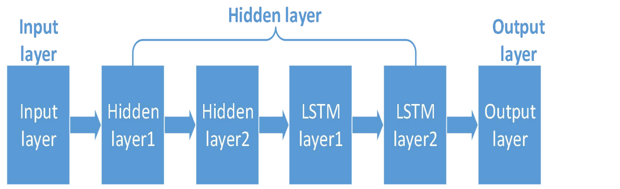

Figure 3: Deep learning model

(1) Input layer. The input of each sample is a matrix of [b_s*n_w, n_l], where b_s is the batch size that means all training samples need to be divided into b_s parts. A reasonable b_s can improve memory utilization and the parallelization efficiency of the large matrix multiplication. In a certain range, the larger the b_s is, the more accurate it is to determine the descending direction, and the smaller the training oscillation is. n_w denotes the width of the input sample, that is, the number of rows is input by a single sample. n_l denotes the length of the input sample, which means that each row of the sample has n_l input values.

(2) Hidden layer. In this paper, deep learning model has four hidden layers, among which H1~H2 are common hidden layers, and L1 and L2 are LSTM layers. The hidden layer is responsible for the decomposition and processing of the data, and integrates the final result into the output layer. During the training, the parameters such as weights and biases between the hidden layers are continuously adjusted and updated, and gradually approach the final result.

In this paper, the number of learning model layers is denoted by J. X_j represents the data of the jth layer ($j \in \{1, 2, \dots, J\}$), and the number of neuron nodes of the j_{th} layer is

n_j. $W_{j,j+1}$ denotes the weight between the jth and (j+1)th ($j \in \{1, 2, ..., J-1\}$) layers. b_j is the bias of the j_{th} layer. The relationship of the jth and (j+1)th layers can be obtained:

$$X_{j+1} = X_j * W_{j,j+1} + b_j,\ j \in 1,2,\cdots,(J-1) \quad (4)$$

The output of this model uses one hot label, which means each tag is represented by a list of binary sequences and only one bit of them is valid at any time. There are 5 types of labels in the preprocessed data. We tag thumb bending motion as [0, 1, 0, 0, 0, 0], and the corresponding classifications of the bending motion of index finger, middle finger, ring finger, and little finger are expressed as [0,0,1,0,0,0], [0,0,0,1,0,0], [0,0,0,0,1,0] and [0,0,0,0,0,1], respectively. In order to make the difference between different classifications more obvious, the result X_j is processed using the Sofmax function. The formula is shown in (5):

$$X'_{Jmp} = \frac{X_{Jmp}}{\sum_{m=1}^{n_{label}} X_{Jmp}}, m \in 1,2,\cdots,n_{label} \quad (5)$$

Where X'_{Jmp} represents the mth value in the output of the pth sample, and n_{label} represents the number of tags contained in the data set. In the training iteration process, we aim at continuously reducing the value of the cost function . In addition, we add the $L2$ regularization term to prevent overfitting of the model training, as shown in formula (6):

$$cost = -\frac{1}{b_s}\sum_{p=1}^{b_s}\sum_{m=1}^{n_{label}}(y_{mp}\log(X'_{Jmp}) + (1-y_{mp})\log(1-X'_{Jmp})) + L2 \quad (6)$$

Where y_{mp} refers to the mth value in the pth sample output truth table, and regularization term $L2$ is calculated using formula (7):

$$L2 = \lambda \sum_{k_1=1}^{n_{k_1}} \frac{\sum_{k_2=1}^{n_{k_2}} v_{h_2}^2}{2} \quad (7)$$

λ is a regularization term coefficient. v represents a variable that can be trained in the model. n_{k1} is the number of tensor variables in the model, and n_{k2} represents the number of medians of a particular tensor. The model uses the Adam (Adaptive Moment Estimation) optimizer to optimize the cost function. The optimizer dynamically adjusts the learning rate of each parameter using the first moment estimation and the second moment estimation of gradient . The advantage is that after the bias correction, each of the iterative learning rate has a definite range, making the parameters more stable.

(3) In this paper, the Sofmax function is applied to the multi-classification of the output layer. It maps the output of multiple neurons to the (0,1) interval. For example, we suppose that the final classification result of a sample is [0.067, 0.144.512, 0.156, 0.073, 0.128]. We use the argmax function to find the position of the maximum value in the classification result. The maximum value is 0.512. Its index position is 2 (starting from 0), so it can be considered that the classification result at this time is that the index finger of the subject is active. We use Y_p to represent the prediction tag value of the pth sample:

$$Y_p = \text{argmax}(X'_{Jmp}), m \in 1,2,\cdots,n_{label} \quad (8)$$

After obtaining the predictive tag value of each sample, we use the truth table to count the prediction accuracy rate of each tag class. The calculation formulas are as shown in (9) and (10):

$$accuracy = \frac{\sum_{p=1}^{b_s} u_p}{b_s} \quad (9)$$

$$u_p = \begin{cases} 1, Y_p = True\_label_p \\ 0, other \end{cases} \quad (10)$$

Among them, $True\_label_p$ is the real tag of the p_{th} sample. If the predicted tag Y_p is the same as the $True\_label_p$, the corresponding position in the truth table is incremented by one, and if not, zero is added.

E. Parameter Adjustment

Hyperparameter refers to the parameters before the model starts to learn, rather than the parameters obtained through training. Selecting a set of optimal hyperparameters for the deep learning model is a prerequisite for the model to be trained successfully. Some parameters can be initialized to random values. In this paper, the initial weight and the model bias are set to 0 and 0.9, respectively, and the number of LSTM layers is set to 2. Other hyperparameters need to be set by specific experimental results. We have selected the five most important hyperparameters: lr, n_{nodes}, λ, b_s, and J, where lr is the learning rate of the model, n_{nodes} is the number of neuron nodes per layer, and λ is the coefficient of the regularization term, b_s and J represent the batch size and the number of layers of the model, respectively. This paper divides the five parameters into five levels for verification, as shown in Table 1:

Table 1:Model training parameter table

	l_r	n_{nodes}	λ	b_s	J
$level1$	0.005	16	0.002	1000	5
$level2$	0.01	32	0.004	2000	6
$level3$	0.015	48	0.006	3000	7
$level4$	0.02	64	0.008	4000	8
$level5$	0.025	96	0.01	5000	9

Through a large number of experiments, the performance of the learning model is optimal when lr=0.005, n_{nodes}=64, λ=0.008, b_s=2000, and J=6. In the training phase of the model, we use the training set to construct the deep learning model for the training of parameters. The test set evaluates the accuracy of the trained model. Accuracy rate refers to the ratio of the number of samples correctly classified by the classifier to the total number of samples. In addition, we adopt precision and recall to evaluate the classification performance. The precision rate refers to the ratio of the number of correct prediction labels to the number of samples classified as corresponding labels. The recall rate refers to the ratio of the correct number of labels to all true labels.

$$accurary = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (11)$$

$$precision = \frac{TP}{TP+FP} \quad (12)$$

$$recall = \frac{TP}{TP+FN} \quad (13)$$

among them: TP = true positives, TN = true negatives, FP = false positives, FN = false negatives.

The test data is input into the model, and the obtained average accuracy rate of the final classification is 0.858. Table 2 is the confusion matrix generated by testing data of the subject 1.

Table 2: Confusion matrix

		Authentic Label					
		1	2	3	4	5	Precision rate
Predictive label	1	303	46	31	47	43	0.644
	2	41	264	39	29	22	0.668
	3	27	59	250	11	35	0.654
	4	19	41	47	231	38	0.614
	5	22	24	31	58	242	0.642
	Total	412	434	398	376	380	
	Recall rate	0.735	0.608	0.628	0.614	0.639	
	Accuracy rate	0.862	0.849	0.86	0.855	0.863	

F. Performance Comparison

In order to compare with other existing methods, we use the other two methods to establish the classification model for the same training set, and meanwhile comparisons and tests are also executed on the same test set.

(1) Data is input into Hidden Markov Models for feature extraction, and then SVM is selected as a classifier for classification.

(2) Shift Invariant Wavelet Packet Decomposition and Multitaper Time-Frequency Spectrum method is used for feature extraction, and then SVM is selected as a classifier for classification.

In order to compare the deep learning method proposed in this paper with the above two methods, we input the data of three subjects into different models, and the test results shown in Table 3. From Table 3, it can be intuitively found that the accuracy of the LSTM-based deep learning method (0.833) is improved by 7.5% and 12.3% compared with the HMM+SVM method (0.758) and SIWPD+MTTFS+SVM method (0.71) respectively, so it demonstrates the superiority of our proposal.

Table 3: Performance comparison

Subject	Deep Learning Model	HMM+SVM	SIWPD+ MTTFS+SVM
1	0.858	0.77	0.73
2	0.81	0.753	0.68
3	0.832	0.75	0.712
Average	0.833	0.758	0.71

IV. Method Application

In order to verify the validity of our proposal, we conduct the training based on deep learning model for off-line ECoG signals of each subject, and then apply the trained model to finger movement and robotic arm real-time control. As shown in Fig. 4, the training set data is input into the deep learning model firstly with the initial parameters set. After many rounds of training and continuous updating of the parameters, an optimal model is achieved. And then the real-time ECoG data is input into the deep learning model to obtain the real-time classification result, and finally the action of the corresponding finger of the manipulator is controlled through different classification results. This shows that the control of brain electricity thoughts has great potential, and then it will be widely used in the future for physical rehabilitation of disabled people and real-time control of various types of equipment.

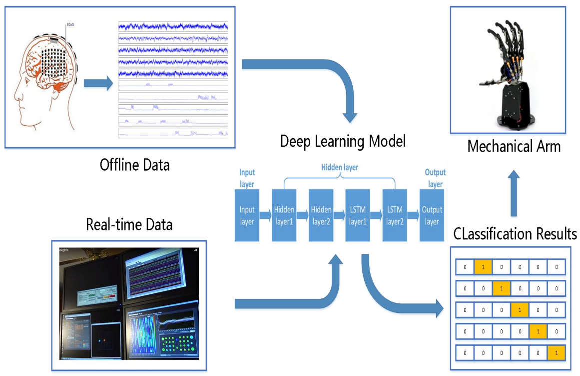

Figure 4: Application Diagram

V. Conclusion

In this paper, we use the deep learning model base on LSTM to extract and classify ECoG signals. Our proposal abandons the traditional manual extraction of features, and does not need a separate classifier to classify the results, which greatly saves time costs. In comparison experiments, the classification accuracy of our proposal is higher than that of the other two methods (HMM+SVM and

SIWPD+MTTFS+SVM). Finally, we apply the optimized training model to finger movement training and robotic arm real-time control of the handicapped, for the purpose of verifying the effectiveness of the proposed method. In the future work, we will further investigate the relevance between finger motion and various ECoG signals in the brain. Furthermore, according to the difference in relevance, we will adjust and improve the deep learning model for better classification results.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of P. R. China (No. 61672297), the Key Research and Development Program of Jiangsu Province (Social Development Program, No.BE2017742), the Postgraduate Research & Practice Innovation Program of Jiangsu Province (No. SJCX17_0235) and the Sixth Talent Peaks Project of Jiangsu Province (No. DZXX-017).

REFERENCES

[1] Samiee S, Hajipour S, Shamsollahi M B. Five-class finger flexion classification using ECoG signals[C]// International Conference on Intelligent and Advanced Systems. IEEE, 2010:1-4.

[2] Liang N, Laurent B. Decoding Finger Flexion from Band-Specific ECoG Signals in Humans[J]. Frontiers in Neuroscience, 2012, 6(91): 91.

[3] Wang Z, Lyu S, Schalk G, et al. Deep feature learning using target priors with applications in ECoG signal decoding for BCI[C]// International Joint Conference on Artificial Intelligence. 2013:1785-1791.

[4] Elgharabawy A, Wahed M A. Decoding of finger movement using kinematic model classification and regression model switching[C]// Biomedical Engineering Conference. IEEE, 2017:1-7.

[5] Marjaninejad A, Taherian B, Valerocuevas F J. Finger movements are mainly represented by a linear transformation of energy in band-specific ECoG signals[C]// International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE, 2017: 986-989.

[6] Sak H, Senior A, Beaufays F. Long short-term memory recurrent neural network architectures for large scale acoustic modeling [J]. Computer Science, 2014: 338-342.

[7] Zhang X, Yao L, Huang C, et al. Enhancing Mind Controlled Smart Living Through Recurrent Neural Networks [J]. 2017:1-10.

[8] Yann L C, Yoshua B, Geoffrey H. Deep learning[J]. Nature, 2015, 521(7553): 436-44.

[9] Tangermann M, Müller K R, Aertsen A, et a1． Review of the BCI competition IV [J]. Front Neurosci, 2012, 6(55): 1-31