PROCEEDINGS OF SCIENCE

A Review of EEG Signal Classifier based on Deep Learning

Yao Lu[1]
Brain Cognitive Computing Lab, School of Information Engineering, Minzu University of China
Beijing, 100081, China
E-mail: `5206661314@163.com`

Huiping Jiang[2]
Brain Cognitive Computing Lab, School of Information Engineering, Minzu University of China
Beijing, 100081, China
E-mail:`jianghp@muc.edu.cn`

Wenqiang Liu[3]
Brain Cognitive Computing Lab, School of Information Engineering, Minzu University of China
Beijing, 100081, China
E-mail:`liuwqleo@163.com`

Electroencephalogram (EEG) signal recognition is an active research topic in the field of artificial intelligence and has been gaining extensive attention and engineering communities. This technology is an important basis of human computer interaction and many other fields. The deep learning theory has made remarkable achievements on feature extraction and gradually extended to the time sequences of EEG research. This paper reviews the traditional feature extraction of EEG recognition and discuss the traditional classification methods of EEG recognition such as linear discriminant analysis (LDA), support vector machine (SVM) and long short time memory (LSTM). Finally, this paper summarizes advantages and disadvantages of these methods. Through the feature extraction by the wavelet transform and LSTM classification, it has achieved 98% accuracy and verified that LSTM is suitable for EEG signal with time sequence feature.

ISCC 2017
16-17 December 2017
Guangzhou, China

[1] Speaker

[2] Correspongding Author Acknowledgements. Supported by National Natural Science Foundation of China Youth Project Funding (No: 61503423), the Leading Talent Program of State Ethnic Affairs Commission and Double First-class Special Funding of MUC

[3] The Second Author

1. Introduction

Electroencephalogram (EEG) is a non-stationary signal which has strong randomness with weak intensity and strong background noise. It needs some processing techniques to extract its features. There are many feature extraction methods of EEG includingthe most common methods such as time-frequency analysis, Fast Fourier transform (FFT), wavelet transform and Autoregressive model (AR) methods. The key to the Brain Computer Interface (BCI) system is to identify EEG features quickly and accurately. At present, there are several common EEG classification methods, such as Linear Discriminant Analysis (LDA), Support Vector Machine (SVM) and Softmax[1]. These methods do not utilize the temporal characteristics of EEG signals, so the effect of classification still needs to be improved.

In the deep learning field, the input of recurrent neural network (RNN) model not only includes the input of the current time, but also the input of the hidden layer data of the last moment. Therefore, RNN has the memory storage function, which can be used for time sequences analysis. The recurrent neural network (LSTM) model resolves the problem of gradient vanish and gradient explosion, and can be used for long-distance sequence information. At present, the LSTM type RNN model has been widely used in the fields of voice and video, but has not been widely used in the field of EEG. EEG signal is a kind of time-domain signal. The EEG feature extracted by wavelet transform retains its time-domain features. Experiments show that using LSTM classifier to classify EEG signal can make full use of the time-domain information of EEG features, and thus improve the accuracy of EEG classification which can achieve 98% accuracy and provide an effective idea for EEG recognition.

2. EEG Feature Extraction Method

EEG signal is very weak, and it has time-domain characteristics, frequency domain characteristics, time-varying sensitivity and large differences between each individual., It is a typical non-stationary time-varying signal. The key of BCI system is how to identify and classify EEG quickly and accurately, and the result of feature extraction will affect the classification result directly; therefore, the key to improve the accuracy of EEG classification is to select the feature extraction method. Currently, the common EEG feature extraction methods are shown as follows:

(1) Time Domain Analysis Method

The time domain analysis method mainly analyzes the geometric properties of the EEG wave form,, such as amplitude, mean value and variance, etc.. Common analysis methods include histogram analysis, analysis of variance, correlation analysis and AR model [2], etc..

(2) Frequency domain analysis method

The method mainly uses the Fourier Transformation (FT) and other frequency domain analysis methods to extract the frequency characteristics of EEG signals, which can make full use of the frequency characteristics of EEG signals. The commonly used methods include Fast Fourier Transformation (FFT) and spectral estimation, etc.;. however, studies have shown that EEG signals are non-stationary time-varying signals. The time-domain analysis and the frequency-domain analysis methods can’t extract the features of EEG signals effectively.

(3) Time-frequency domain analysis method

The method of mapping the time-domain signal or frequency-domain signal to the time-frequency two-dimensional signal is called the time-frequency domain analysis, which reveals the frequency distribution of the EEG signal and the law of each frequency component changing with time. This method can overcome the problem of non-stationary and non-linear EEG signals, but a single time-domain analysis or frequency-domain analysis can’t solve the problem . At present, this method is widely applied to the feature extraction of EEG signals, which has also achieved high classification accuracy. However, EEG has obvious individual differences. EEG signals of different people and EEG signals of different time periods of the same individual have great differences. This method cannot extract the features of EEG signals self-adaptive and it also has some limitations.

(4) Spatial domain analysis method

Different thinking activities of the brain lead to various neural electrical activities in different regions of the cerebral cortex. This method can take advantage of the spatial distribution of the brain to extract the spatial features of EEG, and is one of the general methods for extracting EEG features. Common spatial method refers to the Common Spatial Pattern (CSP) , etc.. However, this method cannot extract EEG features self-adaptive and requires a large number of leads at the same time, which limits its application in online EEG analysis.

3. EEG Classification Algorithm

In the general BCI systems, the classification of EEG features quickly and accurately is the key to the BCI system. There are several classification methods which have bee used widely.

3.1 Linear Discriminant Analysis(LDA)

Ronald A. Fisher proposed the linear discriminant analysis method in 1936 (the Use of Multiple Measurements in Taxonomic Problems), which is used to solve the classification problem. The original LDA was applied to the dichotomous problem and promoted as "multi-class linear discriminant analysis" or "multiple discriminant analysis" gradually. LDA is a classical statistical analysis method, which is a supervised linear classification method. In the pattern classification and machine learning practice, it is often used for dimension reduction steps in data preprocessing. This method maps the dataset to a lower dimension space under the premise of ensuring the category discrimination in order to reduce the computational cost and avoid over-fitting. Moreover, the calculation principle is simple, robust and has good generalization ability. However, this method is not suitable for non-linear and non-stationary EEG signal recognition because it is suitable for linear signal analysis.

3.2 Support Vector Machine(SVM)

The original SVM algorithm was invented by Vladimir N. Vapnik and Alexey Ya, the currently standardized SVM algorithm was proposed by Corinna Cortes and Vapnik in 1993. As its full name is “support vector machine”, called as SVM, it is a kind of dichotomous model,. The basic model is defined as the feature domain of the largest linear classifier and the learning strategy is larger intervals. Finally, it can be transformed into a convex quadratic programming problem.

Compared with other traditional pattern recognition algorithms, this method has obvious advantages in solving nonlinear, small sample and high-dimensional mode tasks, and it has good generalization and classification accuracy. SVM has become a research hotspot in the field of machine learning and is used widely in the recognition of EEG.But using SVM as EEG classifier cannot make full use of EEG's time-domain information. In 2016, Parvez et al. used the related potential of the 6-lead EEG as the eigenvector, and then classified it by SVM, which achieved an accuracy of 91.95% [3].

3.3 BP(Back Propagation,BP)Neural Network

In 1943, Psychologist Warren Mcculloch and mathematical logicist Walter Pitts proposed the concept of artificial neural network, and presented the mathematical model of artificial neuron in a collaborative paper[4], which initiated the research on human neural network era. Until 1957 Frank Rosenblatt firstly proposed a machine that modeled human perception and callsed it Perceptron [5]., whichn is a neural network with single-level computation unit consisting of linear components and threshold components. THe perceptron logic diagram is shown in Figure 1.

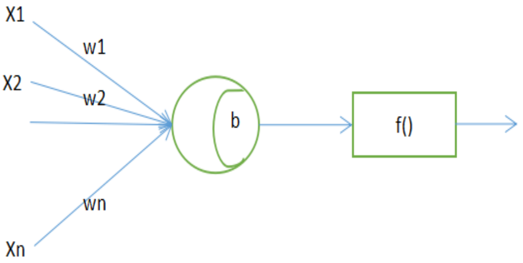


Figure 1: Perceptron Model

Although the perceptrons have good classification effects, but the perceptrons can not handle many pattern recognition problems. In 1969, Marvin Minsky and Seymour Papery[6] analyzed the computational limitations of single-layer perceptrons, proved that the perceptrons can not solve linearly inseparable problems such as simple exclusive-OR (XOR). Since the single-layer perceptron can not solve the classification problem, a hidden layer is added between the input layer and the output layer of the single-layer perceptronso that it can form a multi-layer perceptron (MLP). MLP structure is shown in Figure 2.

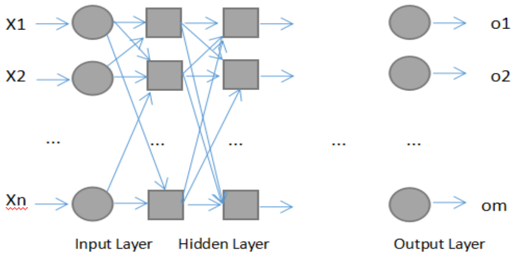


Figure 2: Multi-layer Perceptron Structure

Until 1982, the Hopfield Network is proposed by John J. Hopfield of the California, Rumelhart and McCelland's team proposed the "parallel distribution processing." Both of these achievements have rekindled the interest in the artificial neural networks, reopened the research on intelligent computers that imitate brain information processing. The latter analyzed the Error Back Propagation algorithm of MLP with non-linear continuous transformation function. It's a test of Minsky's assumption of multi-layer network. The error back propagation is called the back propagation algorithm (Back Propagation algorithm, BP) [7]. The BP algorithm was designed by this idea, shown as follows: the learning process consists of two processes: the signal forward propagation and the error back propagation.

1) In the forward propagation, the input samples are input from the input layer and processed by the hidden layer layer by layer, and then transmitted to the output layer. If the output layer of the actual output and the predict output do not match, then go to the error of the back propagation phase.

2) In the case of back propagation, the output is relayed layer by layer through the hidden layer to the input layer in some form, and the error is distributed to all cells on each layer to obtain the error signal of each layer of the unit, and the error signal is used as the correction the weight of each unit.

The neural network combined with BP algorithm is called BP neural network. The problem in BP neural network is that Gradient Diffusion easily occurs due to the adjustment of weights based on local gradient descent. The nonconvex objective cost function leads to the solution getting into the local optimum. Moreover, when the nembers of network layers increase, this situation will be more and more serious. The emergence of this problem restricts the development of neural networks. Experiments show that the eigen-values are extracted from the wavelet transform. With these eigen-values as training data are trained by the BP neural network. The results show that the classification accuracy of EEG signals by BP neural network can reach 93.2% [8].

4. Analysis based on Deep Learning

The concept of deep learning originates from the artificial neural network as a general term of learning algorithms based on deep neural network (DNN). It is a hot research topic in machine learning in recent years. In 2006, Geoffrey Hinton advanced the deep learning and the improvement of the model training method, which broke the bottleneck of BP neural network. Hinton [9] proposed a layer-by-layer pretraining algorithm in Science. He solved the problem thatit is hard to train the traditional BP algorithm when the number of layers increas. In this sense, he also opened the wave of deep learning research.

From MLP to neural network to deep learning, the development of deep learning is not easy. Until 2006, Geoffrey Hinton proposed a deep belief network [10]consisting of a series of Restricted Boltzmann Machine (RBM) [11]. It proposed an unsupervised greedy layer-by-layer training algorithm, followed by the Deep Boltzmann Machine (DBM) as proposed by Ruslan Salakhutdinov [12], which has rekindled the attention of the artificial intelligence community to the neural networks and the Boltzmann machines. According to the latest research progress, as long as the data is large enough and the hidden layer is deep enough, the deep learning can achieve good results even without pre-training. This reflects the intrinsic connection between big data and deep learning. In addition, although the advantages of deep learning lie in unsupervised learning, it can also be used in supervised situations. In fact, the supervised convolutional neural networks [13] are used widely and even beyond DBM.

4.1 Convolutional Neural Networks Feature extraction

The image processing is the earliest application of deep learning. As early as in 1989, Yann Lecun proposed the Convolutional Neural Networks (CNN), which was a deep neural network model that contained convolutional layers. Usually a convolutional neural network architecture consists of two nonlinear convolutional layers that can be trained, two fixed sub-sampling layers and a fully connected layer. The number of hidden layers is generally at least five or more.

The deep convolutional neural network learns data features layer by layer through multiple serial convolutional and pooled layers. The network structure is shown in Figure 3. When the input data is a two-dimensional image, as the convolution operation can deal with the two-dimensional topological structure directly, the number of weights can be reduced, the feature extraction and the pattern classification can be facilitated. The outputs of the convolutional layers are often discretized and normalized, and are called feature maps, each of which corresponds to a feature map. The feature map inputs into the pooling layer which approximates to a spatial sub-sample. Processing by the pooling layer can reduce the resolution of the output feature map and reduce the sensitivity of the CNN to position changes of the object in the input image so that it has a certain degree of anti-distortion ability. The convolution neural network with its convolutional operation, convolution kernel sharing and subsampling structure in the image processing has inherent advantages: its strong robustness and self-learning ability to handle complex two-dimensional signal recognition problem.

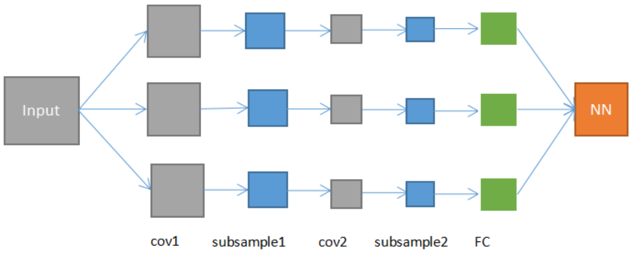


Figure 3: CNN Structure

Since CNN was proposed, there was no qualitative improvement and breakthrough in image recognition. Until in 2012, Hinton constructed a profound neural network to go to the astonishing achievement. Thanks to the improvement of the algorithm, the concept of weight attenuation has been introduced to the network training, which has thus effectively reduced the weight range and prevented the network from over fitting. The current deep learning network models have been able of understanding and recognizing general natural images. However, the EEG signal contains a large amount of time-frequency information. And the CNN model can hardly obtain a good classification result if the information can not be fully utilized. Experiments show that the CNN model solves the classification of EEG signals, but the classification result is not very satisfactory, and the recognition accuracy is only 88.75% [14].

4.2 LSTM Type RNN

4.2.1 Recurrent Neural Network

In the deep learning field, the traditional MLP perform well and has created records for many different tasks - including handwritten digit recognition and target classification. In spite of this, the MLP can still achieve very limited functions. However, when analyzing logically input sequences, the information sequences contain a large amount of content. The information has complex temporal correlation and the information length varies. In 1986 and 1990, Jordan and Elman proposed the Recurrent Neural Network (RNN) respectively. The input of each moment includes not only the input data of the current moment but also the hidden layer unit data of the previous moment. The key point is that the current hidden layer of the network will retain the previous input information, used to make the current network output. The RNN structure is shown in Figure 4.

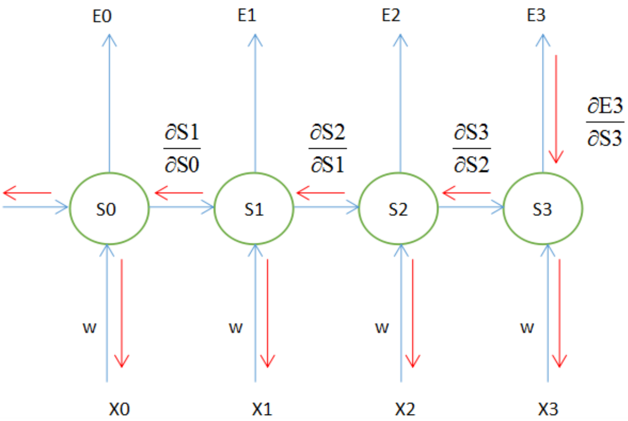


Figure 4:RNN structure

The process of RNN forward propagation is as follows：

$$S_t = \tanh(Wx + US_{(t-1)}) \tag{4.1}$$

$$y_t = Softmax(VS_t) \tag{4.2}$$

S is the incentive for the intermediate layer to be saved at a certain moment， W is the input excitation parameter, U is the state at $\hat{y}_t$ different moments, who is the predicted value at time t. The loss function of RNN is calculated as follows：

$$E_t(y_t, \check{y}_t) = -y_t log \check{y}_t \tag{4.3}$$

$$E(y, \check{y}_t) = \Sigma\, E_t(y_t, \check{y}_t) = -\Sigma\, y_t \log \check{y}_t \tag{4.4}$$

E_t is the loss function (cross-entropy) at the time t, and the loss function is calculated to back propagate the errors along the time direction to optimize the neural network parameters.

With the traditional neural network, the gradient descent method is used to optimize the parameters and calculate the derivative of the loss function on the parameters. Since each output has an impact on the parameters, the derivative of the parameters is the sum of the output of each output parameter derivative：

$$\partial E / \partial w = \Sigma\, \partial E_t / \partial w \tag{4.5}$$

There are many loss functions. Taking E3 as an example, E3 is affected by time t0-t3, and the chain rule is used to forward the gradient. Therefore, the back propagation of E3 is optimized for the parameters t0-t3. The optimization process is shown in Figure 4：

$$gradient\ \Delta S3 = \partial E3 / \partial S3 \tag{4.6}$$

Since E3 is jointly determined by x, w at the time t0-t1, the determination of Δw takes into account the derivative of w at each time for E3：

$$\partial E3 / \partial w = \Sigma\, (\partial S_i / \partial S_{(i-1)})(\partial S_k / \partial w) \Delta S3 \tag{4.7}$$

Let $U = \partial S_i / \partial S_{(i-1)}$, U>1 will face the problem of gradient explosion, U<1 will lead to the problem of gradient vanish, which make the expansion of RNN training of too much stratospheric instability, so RNN losses the ability of long-term memory.

4.2.2 Long Short-Term Memory

In order to solve the problem that long-term memory can not be achieved due to gradient disappearance or gradient explosion in RNN network. In 1997, Hochreiter & Schimidhuber proposed the Long Short-Term Memory (LSTM) to improve the traditional RNN. A LSTM unit is shown in Figure 5:

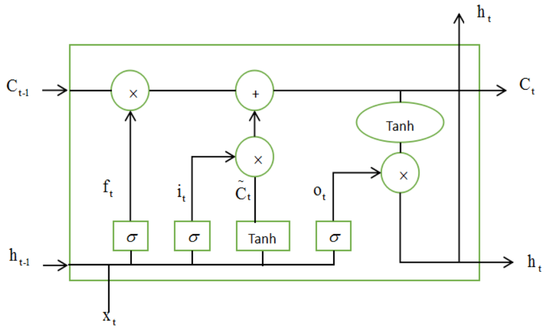


Figure 5:A LSTM Unit

The LSTM unit calculation is shown as follows:

5.Forget Gate

The state $h_{(t-1)}$ before new input x_t determines the part of the information C can be discarded, f_t and $C_{(t-1)}$ were operated, removal part of the information. σ operator represents the sigmoid operation，0 represents discard，1 represents save， used to determine parameter $C_{(t-1)}$ changes. The formula is shown as follows:

$$f_t = \sigma\left(w_{fx} x_t + w_{fh} h_{(t-1)} + w_{fc} C_{(t-1)} + b_f\right) \qquad (5.1)$$

w_{fx}, w_{fh}, w_{fc} respectively correspond to the forget gate, the last moment LSTM unit forget gate and the weight of last moment forget gate unit, b is the bias.

Input Gate：

The state $h_{(t-1)}$ before new input x_t is to determine the saving of the information C, and the input gate i_t is calculated as per the formula as follows：

$$i_t = \sigma\left(w_{ih} h_{(t-1)} + w_{ix} x_t + w_{ic} C_{(t-1)} + b_i\right) \qquad (5.2)$$

i_t is the control parameter of the coefficient C_t when the new information is added and used to update C; w_{ix}, w_{ih}, w_{ic} respectively correspond to the input gate, the last time LSTM unit input gate and the previous time input gate memory unit weight, b is bias.

Update Control Parameters：

Based on the old control parameter $c_{(t-1)}$ and the newly generated control parameter, the final control parameter is：

$$C_t = f_t * C_{(t-1)} + i_t * C_t \qquad (5.3)$$

In Formula 4-10, the memory unit update depends on its own state $C_{(t-1)}$ and the current candidate memory cell value C_t . It is adjusted by both the input gate and the forget gate.

Output Gate：

The LSTM output is generated according to the control parameter C_t ：

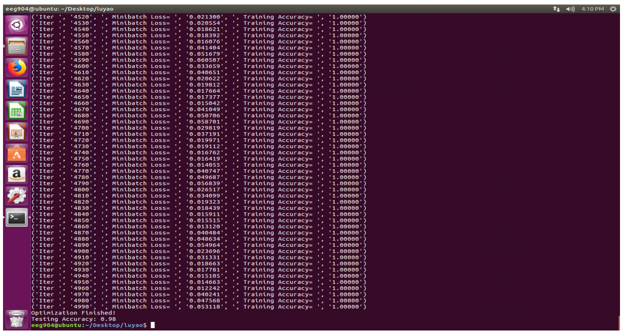
$$O_t = \sigma\left(w_{xo} x_t + w_{ho} h_{(t-1)} + w_{co} C_{(t-1)} + b_o\right) \qquad (5.4)$$

O_t is the state value of the control memory unit , w_{ox}, w_{oh}, w_{oc} represent the corresponding output gate, the last moment LSTM unit output gate and the last moment output gate memory unit weight, b is the bias.

$$h_t = o_t * \tanh(C_t) \qquad (5.5)$$

Through using the gated design, LSTM can effectively mitigate the problem of gradient disappearance of RNN, which makes the RNN model to be effectively applied to long-distance sequence information.

Finally, the LSTM type RNN classifier is used to classify the EEG signals after wavelet transform. Experiments show that this method can make full use of EEG signal and EEG features in time domain, which has improved the accuracy of EEG classification and achieved a classification accuracy of 98% [15]. It provides a feasible idea for EEG recognition.



```
('Iter ', '4520', ', Minibatch Loss= ', '0.021300', ', Training Accuracy= ', '1.00000')
('Iter ', '4530', ', Minibatch Loss= ', '0.020554', ', Training Accuracy= ', '1.00000')
('Iter ', '4540', ', Minibatch Loss= ', '0.018621', ', Training Accuracy= ', '1.00000')
('Iter ', '4550', ', Minibatch Loss= ', '0.018392', ', Training Accuracy= ', '1.00000')
('Iter ', '4560', ', Minibatch Loss= ', '0.016076', ', Training Accuracy= ', '1.00000')
('Iter ', '4570', ', Minibatch Loss= ', '0.041404', ', Training Accuracy= ', '1.00000')
('Iter ', '4580', ', Minibatch Loss= ', '0.051679', ', Training Accuracy= ', '1.00000')
('Iter ', '4590', ', Minibatch Loss= ', '0.060507', ', Training Accuracy= ', '1.00000')
('Iter ', '4600', ', Minibatch Loss= ', '0.033659', ', Training Accuracy= ', '1.00000')
('Iter ', '4610', ', Minibatch Loss= ', '0.040651', ', Training Accuracy= ', '1.00000')
('Iter ', '4620', ', Minibatch Loss= ', '0.020622', ', Training Accuracy= ', '1.00000')
('Iter ', '4630', ', Minibatch Loss= ', '0.019812', ', Training Accuracy= ', '1.00000')
('Iter ', '4640', ', Minibatch Loss= ', '0.017664', ', Training Accuracy= ', '1.00000')
('Iter ', '4650', ', Minibatch Loss= ', '0.017377', ', Training Accuracy= ', '1.00000')
('Iter ', '4660', ', Minibatch Loss= ', '0.015042', ', Training Accuracy= ', '1.00000')
('Iter ', '4670', ', Minibatch Loss= ', '0.041049', ', Training Accuracy= ', '1.00000')
('Iter ', '4680', ', Minibatch Loss= ', '0.050706', ', Training Accuracy= ', '1.00000')
('Iter ', '4690', ', Minibatch Loss= ', '0.058701', ', Training Accuracy= ', '1.00000')
('Iter ', '4700', ', Minibatch Loss= ', '0.029819', ', Training Accuracy= ', '1.00000')
('Iter ', '4710', ', Minibatch Loss= ', '0.037191', ', Training Accuracy= ', '1.00000')
('Iter ', '4720', ', Minibatch Loss= ', '0.019971', ', Training Accuracy= ', '1.00000')
('Iter ', '4730', ', Minibatch Loss= ', '0.019112', ', Training Accuracy= ', '1.00000')
('Iter ', '4740', ', Minibatch Loss= ', '0.016762', ', Training Accuracy= ', '1.00000')
('Iter ', '4750', ', Minibatch Loss= ', '0.016419', ', Training Accuracy= ', '1.00000')
('Iter ', '4760', ', Minibatch Loss= ', '0.014055', ', Training Accuracy= ', '1.00000')
('Iter ', '4770', ', Minibatch Loss= ', '0.040747', ', Training Accuracy= ', '1.00000')
('Iter ', '4780', ', Minibatch Loss= ', '0.049687', ', Training Accuracy= ', '1.00000')
('Iter ', '4790', ', Minibatch Loss= ', '0.056839', ', Training Accuracy= ', '1.00000')
('Iter ', '4800', ', Minibatch Loss= ', '0.026517', ', Training Accuracy= ', '1.00000')
('Iter ', '4810', ', Minibatch Loss= ', '0.034099', ', Training Accuracy= ', '1.00000')
('Iter ', '4820', ', Minibatch Loss= ', '0.019323', ', Training Accuracy= ', '1.00000')
('Iter ', '4830', ', Minibatch Loss= ', '0.018439', ', Training Accuracy= ', '1.00000')
('Iter ', '4840', ', Minibatch Loss= ', '0.015911', ', Training Accuracy= ', '1.00000')
('Iter ', '4850', ', Minibatch Loss= ', '0.015515', ', Training Accuracy= ', '1.00000')
('Iter ', '4860', ', Minibatch Loss= ', '0.013120', ', Training Accuracy= ', '1.00000')
('Iter ', '4870', ', Minibatch Loss= ', '0.040484', ', Training Accuracy= ', '1.00000')
('Iter ', '4880', ', Minibatch Loss= ', '0.048634', ', Training Accuracy= ', '1.00000')
('Iter ', '4890', ', Minibatch Loss= ', '0.054964', ', Training Accuracy= ', '1.00000')
('Iter ', '4900', ', Minibatch Loss= ', '0.023696', ', Training Accuracy= ', '1.00000')
('Iter ', '4910', ', Minibatch Loss= ', '0.031331', ', Training Accuracy= ', '1.00000')
('Iter ', '4920', ', Minibatch Loss= ', '0.018663', ', Training Accuracy= ', '1.00000')
('Iter ', '4930', ', Minibatch Loss= ', '0.017781', ', Training Accuracy= ', '1.00000')
('Iter ', '4940', ', Minibatch Loss= ', '0.015105', ', Training Accuracy= ', '1.00000')
('Iter ', '4950', ', Minibatch Loss= ', '0.014663', ', Training Accuracy= ', '1.00000')
('Iter ', '4960', ', Minibatch Loss= ', '0.012242', ', Training Accuracy= ', '1.00000')
('Iter ', '4970', ', Minibatch Loss= ', '0.040241', ', Training Accuracy= ', '1.00000')
('Iter ', '4980', ', Minibatch Loss= ', '0.047568', ', Training Accuracy= ', '1.00000')
('Iter ', '4990', ', Minibatch Loss= ', '0.053118', ', Training Accuracy= ', '1.00000')
Optimization Finished!
Testing Accuracy: 0.98
eeg904@ubuntu:~/Desktop/luyao$
```

Figure 6: Experiment Result by Using LSTM

6. Conclusion

In this paper, we summarize the feature extraction and classification methods used widely for EEG signal recognition, and introduce the LSTM classifier. Compared with the traditional LDA, SVM and CNN, the LSTM classifier is superior to the traditional classifiers in terms of accuracy and generalization ability. The recognition accuracy of LSTM classifier can achieve 98%.

Reference

[1] SHIM Hyeon-min , LEE Sangmin. *Multi-channel electromyography pattern classification using deep belief networks for enhanced user experience.* Journal of Central South University, 2015, 22(5).

[2] Lu-Qiang Xu, Guang-Can Xiao. *Motor Imagery EEG Fuzzy Fusion of Multiple Classification.* Journal of Electronic Science and Technology of China, 2017, 15(1).

[3] Fei Hu,Li Li,Zi-Li Zhang. *Emphasizing Essential Words for Sentiment Classification Based on Recurrent Neural Networks.* Journal of Computer Science ＆ Technology, 2017, 32(4).

[4] McCulloch W S, Pitts W. *A logical calculus of the ideas immanent in nervous activity*[J]. The bulletin of mathematical biophysics, 1943, 5(4): 115-133.

[5] Mohamed A, Dahl G, Hinton G. *Deep belief networks for phone recognition*[C]//Nips workshop on deep learning for speech recognition and related applications. 2009, 1(9): 39.

[6] Minsky M, *Papert S.* Perceptrons[J]. 1969.

[7] Rumelhart D E, Hinton G E, Williams R J. *Learning representations by back-propagating errors*[J]. Cognitive modeling, 1988, 5(3): 1.

[8] SHE Qing-shan, MA Yu-liang, MENG Ming. *Noise-assisted MEMD based relevant IMFs identification and EEG classification. J*ournal of Central South University, 2017, 24(3).

[9] Hinton G E, Osindero S, Teh Y W. *A fast learning algorithm for deep belief nets*[J]. Neural computation, 2006, 18(7): 1527-1554.

[10] Salakhutdinov R, Mnih A, Hinton G. *Restricted Boltzmann machines for collaborative filtering*[C]//Proceedings of the 24th international conference on Machine learning. ACM, 2007: 791-798.

[11] Salakhutdinov R, Hinton G E. *Deep Boltzmann Machines*[C]//AISTATS. 2009, 1: 3.

[12] Ackley D H, Hinton G E, Sejnowski T J. *A learning algorithm for Boltzmann machine*s[J]. Cognitive science, 1985, 9(1): 147-169.

[13] LeCun Y, Bottou L, Bengio Y, et al. *Gradient-based learning applied to document recognition*[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[14] Bendong Zhao, Huanzhang Lu, Shangfeng Chen. *Convolutional neural networks for time series classification.* Journal of Systems Engineering and Electronics, 2017, 28(1).

[15] Mohammad Z P,Manoranjan *P.Epileptic seizure prediction by exploiting spatiotemporal relationship of eeg signals using phase correlation*[J].IEEE Trans on Neural Systems and Rehabilitation Engineering,2016,24(1):158-168.