IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 8, 2018, accepted April 28, 2018, date of publication May 7, 2018, date of current version May 24, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2833746

Deep Convolution Neural Network and Autoencoders-Based Unsupervised Feature Learning of EEG Signals

TINGXI WEN AND ZHONGNAN ZHANG, (Member, IEEE)
Software School, Xiamen University, Xiamen 361005, China

Corresponding author: Zhongnan Zhang (zhongnan_zhang@xmu.edu.cn)

This work was supported by the Science and Technology Guiding Project of Fujian Province, China, under Grant 2015H0037 and Grant 2016H0035.

ABSTRACT Epilepsy is a health problem that seriously affects the quality of humans for many years. Therefore, it is important to accurately analyze and recognize epilepsy based on EEG signals, and for a long time, researchers have attempted to extract new features from the signals for epilepsy recognition. However, it is very difficult to select useful features from a large number of them in this diagnostic application. As the development of artificial intelligence progresses, unsupervised feature learning based on the deep learning model can obtain features that can better describe identified objects from unlabeled data. In this paper, the deep convolution network and autoencoders-based model, named as AE-CDNN, is constructed in order to perform unsupervised feature learning from EEG in epilepsy. We extract features by AE-CDNN model and classify the features based on two public EEG data sets. Experimental results showed that the classification results of features obtained by AE-CDNN are more optimal than features obtained by principal component analysis and sparse random projection. Using several common classifiers to classify features obtained by AE-CDNN model results in high accuracy and not inferior to the research results from most recent studies. The results also showed that the features of AE-CDNN model are clear, effective, and easy to learn. These features can speed up the convergence and reduce the training times of classifiers. Therefore, the AE-CDNN model can be effectively applied to feature extraction of EEG in epilepsy.

INDEX TERMS EEG, unsupervised learning, feature extraction, CNN, epileptic seizure.

I. INTRODUCTION

Epilepsy is a noninfectious chronic brain disease, which affects people of all ages. There are about 50 million epileptic patients at present, which results in becoming one of the most common neurological diseases in the world [1]. Seizures can cause cognitive dysfunction such as loss of consciousness, which leads to great physical harm to patients, e.g. fracture and injury. Moreover, patients may suffer from great mental pain because of shame and discrimination. Because epileptic seizures can cause irreversible damage to the brain and may result in unprovoked recurrent attack, it is of great significance to analyze epilepsy.

Electroencephalogram (EEG) is a measure of the voltage fluctuation generated by the ion current of neurons in the brain, which reflects the activity of the brain's bioelectricity and contains a large number of physiological and disease information [2]. Because the frequency and rhythm of brain activity can change during seizures, the EEG has become the most common used epilepsy diagnostic methods. After the first study using EEG to detect epileptic by Gotam [3], researchers have done many experiments on this technique [4]–[6]. The essence of EEG-based epileptic detection is the classification of patients' EEG signals. Other studies [7]–[9] mentioned the EEG during the presence and absence of epileptic seizures. Then, some studies [10]–[12] aimed at studying healthy persons, as well as patients with epileptic seizures during the onset and absence of the seizures. Their method follows the steps of data acquisition and preprocessing, feature extraction, classification model training and EEG signals classification. In data acquisition, researches focused on physiological signal sensor [13]–[15]. In data preprocessing, Sharma et al. [16] subtracted EEG signals of adjacent channels to reduce the effect of noise. Anindya et al. [17] filtered frequencies that are higher than 64HZ by means of a 6th order Butter-worth filter. Feature extraction has always been the focus of EEG classification, which greatly reduces the dimension of data. A small number of features can describe EEG data well and improve classification performance greatly. There are many EEG feature extraction methods, such as time-domain, frequency-domain, time-frequency analysis, and chaotic features. Zandi et al. [18] proposed an algorithm based on wavelet for real-time detection of epileptic seizures. Polat and Güneş [19] extracted features using fast Fourier transform and classified the EEG using a decision tree classifier. Acharya et al. [20] mentioned the Entropy-based feature extraction method in his review of the application of Entropy in epilepsy detection, such as Approximate Entropy, Sample Entropy, and Spectral Entropy.

The core of these common methods is to perform effective feature extraction for EEG. However, the process of designing new feature is complex and not easily verifiable. It is also extremely difficult to select a number of optimal characteristics from a large number of time-domain, frequency-domain, time-frequency analysis, and chaotic features. At the same time, some wavelet transform, empirical mode analysis and trend multifractal feature extraction process are complicated and time-consuming. In recent years, deep learning has become the research hotspot in machine learning and has achieved high efficiency in computer vision. In EEG classification, Tabar and Halici [21] used the short time Fourier transform to transform EEG into 2D images, and then performed independent feature learning and classification based on a deep learning method. Xun et al. [22] dissected the EEG into smaller sections through a small window and used a context-learning model for each small section in order to form various ‘‘EEG word’’, so as to compile them into an ‘‘EEG dictionary’’ and resulting in a new feature. Then, they used the original data to attain new features from the ‘‘EEG dictionary’’ and present them in the classifier by using two parts of these features. Deep learning can independently learn features from data, which greatly improves the performance of the classification model. Because of the powerful feature learning ability, deep convolution neural network (CNN) has become an important research hotspot in image field and has significant influences in EEG classification. Masci et al. [23] proposed a convolutional auto-encoder, which is an unsupervised learning method for features learning based on CNN. Chen et al. [24] presented several descriptors for feature-based matching using convolution and pooling auto-encoders. Noh et al. [25] proposed a novel semantic segmentation algorithm by implementing a deconvolution network. However, these unsupervised feature learning methods are mostly used in the image analysis field. This paper presents an unsupervised feature learning method based on convolution, deconvolution and autoencoders, and applies it on epilepsy detection.

Although some experienced experts can identify EEG in epilepsy heuristically, and scholars have performed research widely on EEG based epilepsy detection, there are still many challenges in automatic epilepsy detection. Feature design and selection is difficult, and whether these methods can be applied to new patients is still unknown. Compared with improving classification accuracy by existing techniques, we still need a model with independent feature learning ability. Facing this challenge, we propose an auto-coding framework- based deep network by combining it with convolution and deconvolution in order to perform unsupervised feature learning from EEG signals. This model can effectively learn low-dimensional features from high-dimensional EEG data in order to help classifier achieve a higher detection accuracy and faster speed.

The structure of this paper is as follows: in Section II, we will detail the deep learning-based unsupervised feature learning model. Experiment and results will be presented in Section III. In this section, we describe the source and structure of the datasets, and the classification result by applying the proposed model. Section IV is based on the discussion of features and classification results. Besides, we will compare our model with other well-established models in analyzing EEG. Finally, Section V concludes this paper.

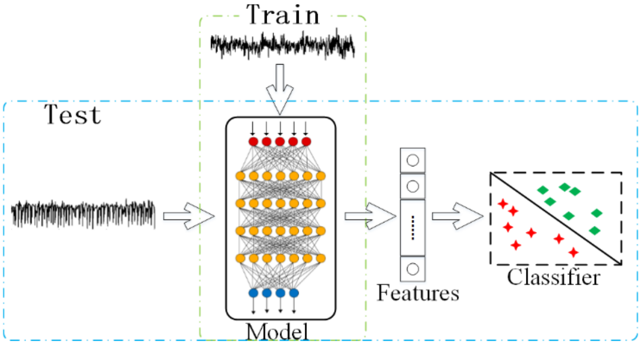

FIGURE 1. Flowchart for unsupervised learning method whereby the light green and the light blue dashed frames represent the training and testing phrases respectively.

II. METHODOLOGY

The paper adopts unsupervised learning method for the EEG of epilepsy patients so as to automatically obtain features that can better describe the recognized objects. In other words, it extracts a number of features from the original high-dimensional data by means of the dimension reduction algorithms. The light green dashed frame in Fig.1 is the training stage of the model, which uses unlabeled EEG signals to train the deep convolution network. The light blue dashed frame represents the testing phrase, in which the test data is input into the model to obtain the relevant features. After this stage, we use different common classifiers to verify the validity of the test data features. The classifier validation will be described in the experimental results in Section 3 and to be discussed in Section 4. Then, Section 2.1 describes the Autoencoders framework used in the model, and the structure and detail of the model will be shown in Section 2.2.

A. AUTOENCODERS FRAMEWORK

Autoencoders [26] is a special neural network structure, which has input layer, output layer and hidden layer. It adjusts the weights of hidden layer by training to enable the input and output value to be as close to each other as possible. Therefore, the hidden layer has important features of original signal in order to realize unsupervised feature extraction. Autoencoders is similar to the Principal Component Analysis (PCA), which can reduce data dimension [27].

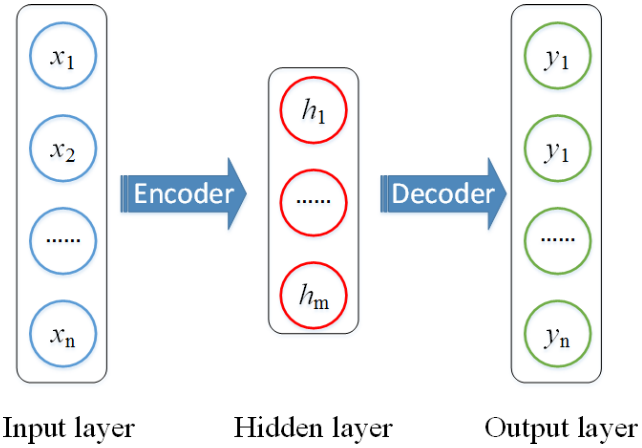

FIGURE 2. Autoencoders framework that includes input layer ($x_1, x_2, \ldots, x_n$), hidden layer ($h_1, h_2, \ldots, h_m$), and output layer ($y_1, y_2, \ldots, y_n$) whereby the weights of hidden layer represent features of the input signal.

Fig.2 illustrates the basic framework of autoencoders, which has two data process phases, encoding and decoding. During encoding, the original input signal data is $\mathrm{x} \in [0, 1]^n$. Then we obtain the hidden layer h by encoding function $h = \mathrm{encoder}(x)(h \in [0, 1]^m)$, whereby the encoding function is defined as follows:

$$h = \mathrm{encoder}\,(x) = \mathrm{g}(\mathrm{W} \cdot x + b). \tag{1}$$

In the function, $\mathrm{W} \in R^{m\times n}$ is the weight matrix connecting input layer and hidden layer. $\mathrm{b} \in [0, 1]^m$ is the bias vector, and g is activation function. In decoding phase, hidden layer h is the input of the decoding function $\mathrm{y} = \mathrm{decoder(h)}$ in order to obtain the output layer y. The decoding function is:

$$y = \mathrm{decoder}\,(\mathrm{x}) = \mathrm{g}\left(W' \cdot x + b'\right). \tag{2}$$

Here, the weight matrix between hidden layer and output layer is $W' \in R^{n\times m}$, and the bias vector is $b' \in [0, 1]^n$. For the model training process, we let each output signal $y^{(i)}$ to be as close in value as possible to the original input signal $x^{(i)}$. Then, the object function of the model is given byčž

$$\min \sum \left| y^{(i)} - x^{(i)} \right|. \tag{3}$$

B. DEEP NETWORK BASED UNSUPERVISED FEATURE LEARNING MODEL

Because it is easy to directly copy the input vector to output vector during the traditional autoencoders' training process, the model has inferior performance. When the test samples and training samples do not meet the same distribution, the prediction result of the model will decrease dramatically [28]. However, the EEG signal sample is high-dimensional and their dimensions are not independent. Therefore, autoencoders is difficult to extract effective features from EEG signals. On the other hand, CNN can perceive adjacent dimension of the signal (i.e. local perception) to attain local features by receptive field and parameter sharing. This process is based on the kernel's convolution. Using multiple convolution kernels to construct the sample signals can allow us to obtain a variety of local features. At the same time, the features of the model can be reduced by down-sampling. Therefore, the final features of the sample signals can be extracted by iterative convolution and down-sampling. Our autoencoders framework-based unsupervised feature learning is described as follows. In the encoder part, we extract features by CNN, which constantly iterates multiple convolution kernels' convolution and down-sampling to reduce the number of features to our preset. Then, in the decoder part, we use the extracted features to reconstruct sample signals by deconvolution and up-sampling. This means that we will carry out deconvolution first, and then iterate up-sampling and deconvolution so as to restore the signal.

We presents a fusion model based on the deep convolution network and autoencoders-based (AE-CDNN). The structure of the AE-CDNN model is as shown in Fig.3. The model mainly has two stages: 1) The encoder stage has sample input, convolution layer, pooling layer (down-sampling layer), reshape operation, full connection layer, and the feature coding; 2) The decoder stage includes feature coding input, full connection layer, reshape operation, deconvolution layer, up-sampling layer and the reconstruction samples. The following presents a specific description of each layer of the model.

Assume that we input a one-dimension EEG data into the model. Let x represent the input data, and the convolution layer is the feature extractor. It uses multiple convolution kernels to perform convolution calculation of x (multiple convolution kernels perceive x locally), so as to attain more feature maps, which can maintain the main components of the input samples. The k-th feature map fm_k in the convolution layer is calculated as follows:

$$\mathrm{fm}_k = \mathrm{g}\,(\mathrm{w\_c}_k * \mathrm{x} + \mathrm{b\_c}_k). \tag{4}$$

In the k-th convolution kernel of the convolution layer, $\mathrm{w\_c}_k$ and $\mathrm{b\_c}_k$ represent filters and biases of the convolution kernel, and $*$ is convolution computation. The pooling layer is a down-sampling process, which samples upper layer's feature maps to obtain pooled feature maps for reducing data dimension. The pooling operation uses a window of length l to allow sliding and extraction of the sample feature maps. Each sampling interval does not overlap each other, and we sample the maximum value within the window to obtain the pooled feature maps. The calculation process is as follows:

$$\mathrm{pm}_k = \mathrm{Maxpooling}\,(\mathrm{fm}_k, l). \tag{5}$$

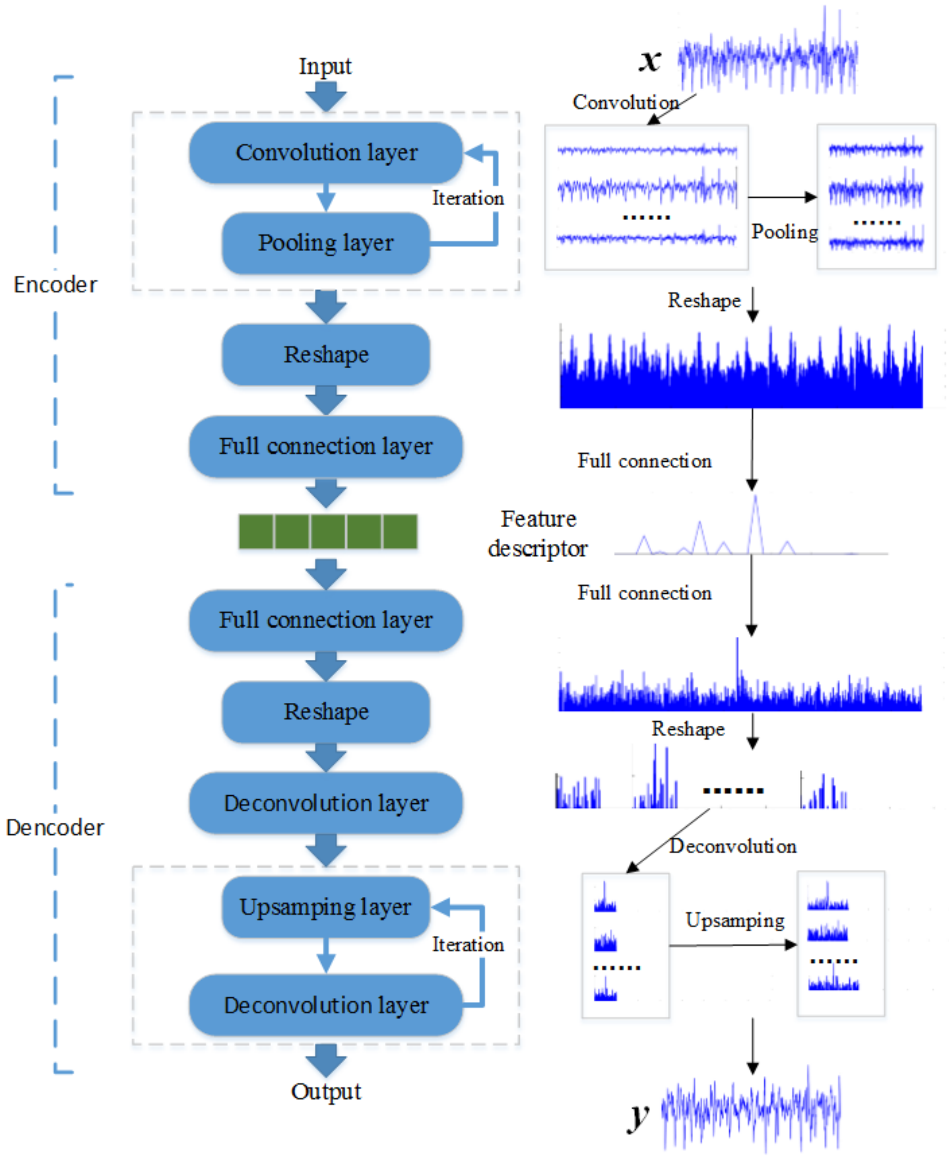

FIGURE 3. Structure of the AE-CDNN model that consists of the encoder and decoder stages whereby the left and right show processing flow and data flow respectively.

Here, we can iterate multiple processes of convolution and pooling to reduce the number and dimension of the pooled feature maps to result in m pooled feature maps. Reshape operation reflects pooled feature maps to one dimensional vector and then achieve feature coding by the operation of full connection layer, which synthesize information of all pooled feature maps. The reshape operation makes all pooled feature maps to generate a one-dimension vector v of length r. Therefore, after the calculation of full connection layer, the vector v becomes feature coding as follows:

$$\mathrm{c} = \mathrm{g}\left(\mathrm{w_v} * \mathrm{v} + \mathrm{b_f}\right). \tag{6}$$

In the above equation, $\mathrm{w_v}$ and $\mathrm{b_f}$ represent the weight and bias of full connection layer respectively. Here, c is the achieved feature. The decoding (signal reconstruction) starts after the encoding. Firstly, the feature coding becomes a one-dimension vector v' of length r by the calculation of the second full connection layer. The associated equation is as follows:

$$v' = \mathrm{g}\left(\mathrm{w}'_{\mathrm{v}} * c\right). \tag{7}$$

The weight of the full connection layer is w'_{v}. Because we need to ensure that in the decoding process, all information is from the feature coding, there will be no bias in this layer. The second reshape operation cuts v' into m pooled feature maps, which corresponds to the first reshape operation. The k-th pooled feature map is pm'_k. In the up-sampling layer, we use the original sampling window to insert the same value as the previous sample to attain fm'_k as follows:

$$\mathrm{fm}'_k = \mathrm{upsampling}\left(\mathrm{pm}'_k, l\right). \tag{8}$$

The deconvolution is:

$$y = \sum \mathrm{g}\left(\sum \mathrm{w\_c}'_k \circ \mathrm{fm}'_k + \mathrm{b\_c}'_k\right). \tag{9}$$

In this case, the deconvolution kernel $\mathrm{w\_c}'_k$ is equal to the shape of the transpose of $\mathrm{w\_c}_k$ and $\mathrm{b\_c}'_k$ is bias. Here, $\circ$ denotes deconvolution. Next, the reconstructed signal y is achieved. In the network, the output layer activation function adopts the sigmoid function and all others are relu activation function.

Assume that there are N training samples and $x^{(i)}$ is a sample, to get $y^{(i)}$ from $x^{(i)}$ by calculation, we need to generate the minimized loss function such as the target function in the autoencoders model (based on Equation (3)). The calculation is as follows:

$$Loss1 = \sum_{i=1}^{N} \left| x^{(i)} - y^{(i)} \right| \Big/ N. \tag{10}$$

However, the signal is serial and its features are not independent. There are differences between each sample in terms of magnitude and Equation (10) has great impact on large size samples. We need to adopt a new loss calculation method:

$$Loss2 = \sum_{i=1}^{N} \left| (x^{(i)} - y^{(i)}) \Big/ avg(x^{(i)}) \right| \Big/ N. \tag{11}$$

Here, $avg(x^{(i)})$ is the average of sample $x^{(i)}$. The loss function of this network is optimized using the Adam optimizer [22].

III. EXPERIMENT AND RESULTS

This paper focuses on the unsupervised feature learning of EEG signal. The research process is as follows: 1) Perform unsupervised feature learning of two data sets to obtain features. 2) After that, we use a variety of common classifiers to classify these features and verify effect of features by the classification accuracy. In this section, we explain the source of experimental data and the preprocessing. Next, we describe the network structure and parameters of the AE-CDNN model. Then, we list several common classifiers to validate AE-CDNN model. Finally, we explain the experimental results.

A. DATA PREPROCESSING

1) DATASET 1

The first dataset is based on the online public dataset that was published by Andrzejak et al. [30]. This dataset consists of five subsets (expressed as A-E) and each subset has 100 EEG signals. Each EEG signal has a duration of 23.6s and the length is 4096, which includes records of health and epilepsy patients. The subsets A and B contain EEG records of 5 healthy volunteers, which were recorded based on the standard international 10-20 electrode placement program. The subsets C and D describe EEG signals during the absence of 5 epileptic patients. And subset E includes the EEG signals during seizure of epileptic patients. The C, D and E subsets came from the skull. The analysis of subsets C, D and E are shown in Fig.4.

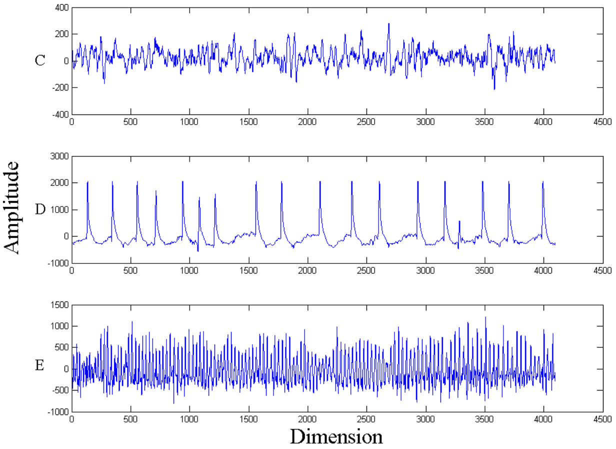

FIGURE 4. Sample signals of dataset 1 based on the intracranial signal whereby subsets C and D pertain to the EEG signals during the absence of epileptic patient and subset E pertains to the EEG signal during seizure.

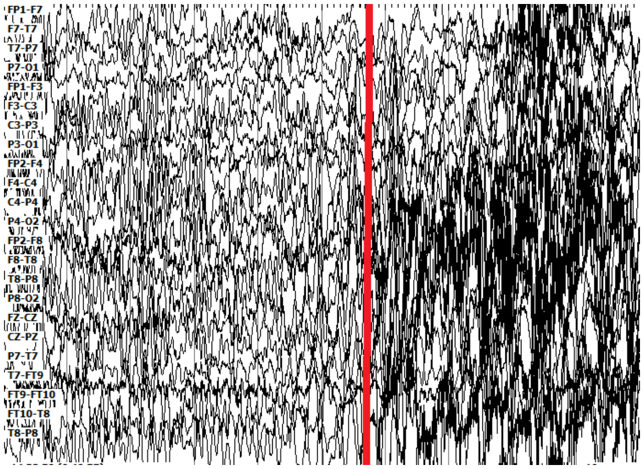

FIGURE 5. Sample signal of dataset 2 based on brain scalp signal whereby the left and the right of the red line are non-epilepsy and epileptic seizures respectively.

2) DATASET 2

The second dataset includes the scalp EEG signals collected from a children hospital in Boston [31]. It was based on measuring electrical activity in the brain to obtain EEG signals by connecting multiple electrodes to the patients' scalp. This dataset is composed by EEG signals of 23 children with refractory epilepsy. All the signals were recorded at 256 Hz with 16-bit resolution. Each sample has 23 channels and the length of each channel is about 921600. Some of the samples contain epileptic EEG signals. Each channel has its own name, i.e. the first channel is FP1-F7 (see Fig.5). We selected one channel from the 23 channels to implement a study. The larger the variance of the signal is, the greater the fluctuation range is. When epilepsy occurred, the EEG signals fluctuate significantly.

Therefore, we chose channel based on variance [32]. Our method is as follows: 1) calculate the variance of each channel in each sample, and select the channel with the maximum

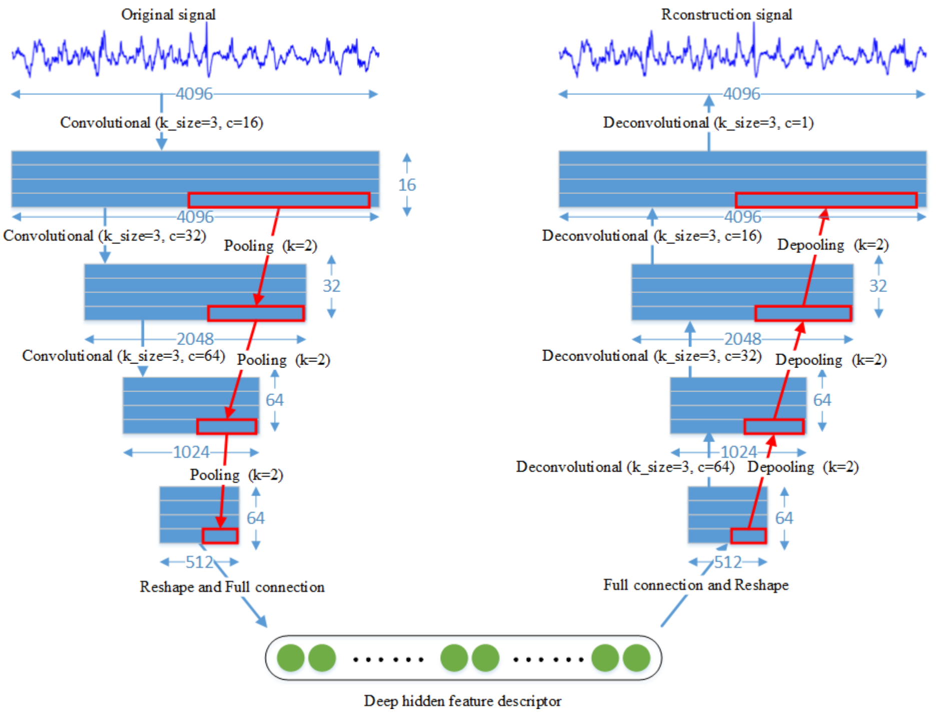
Original signal
4096
Convolutional (k_size=3, c=16)
16
4096
Convolutional (k_size=3, c=32)
Pooling (k=2)
32
2048
Convolutional (k_size=3, c=64)
Pooling (k=2)
64
1024
Pooling (k=2)
64
512
Reshape and Full connection
Rconstruction signal
4096
Deconvolutional (k_size=3, c=1)
16
4096
Deconvolutional (k_size=3, c=16)
Depooling (k=2)
32
2048
Deconvolutional (k_size=3, c=32)
Depooling (k=2)
64
1024
Deconvolutional (k_size=3, c=64)
Depooling (k=2)
64
512
Full connection and Reshape
Deep hidden feature descriptor

FIGURE 6. Network structure was constructed based on AE-CDNN model in the experiment whereby the left is encoding network while the right is decoding network.

variance for each sample; 2) count these channels. In the first 10 patients, the "FT9-FT10" channel appeared most. So, we chose "FT9-FT10" channel as sample data for classification. For the EEG data of the first 10 patients, we randomly extracted 200 epileptic seizure EEG signal samples and 200 non-epileptic seizure EEG signal samples on channel "FT9-FT10". Note that the length of each sample is 4096.

The activation function of AE-CDNN is sigmoid function and the output range is (0, 1). To ensure that the loss functions (10) and (11) are available, the range of each dimension of the input layer should be set at (0, 1). We used 0-1 normalization to map the sample data to [0, 1]. The transformation function was as follows:

$$\text{dist} = max\left(x^{(k)}\right) - min\left(x^{(k)}\right), \quad k \in \{1, \ldots N\} \quad (12)$$

$$tranfun = (d - \min\left(x^{(k)}\right))/\text{dist}. \quad (13)$$

In the above equation, d is the dimension value of the input sample x. Then, $max\left(x^{(k)}\right)$ is the maximum value of samples in all dimensions, and $max\left(x^{(k)}\right)$ is the minimum of samples in all dimensions.

TABLE 1. Parameters used in each classifiers.

Method	Parameters details
k-NN	*k*=3
SVM1	kernel="linear"
SVM2	kernel='rbf'
DT	max_depth=5
RF	max_depth=5, n_estimators=10, max_features=1
MLP	--
ADB	--
GaussianNB	--

B. NETWORK PARAMETERS AND CLASSIFIERS PARAMETERS

The AE-CDNN model-based deep network is shown in Fig. 6. Although the increase of network depth can enhance learning ability of the model, it may also cause gradient to disappear while training, or overfitting. In the deep network, we fixed the network depth, encoding and decoding process. In our experiment, we analyzed learning ability of the model by setting different values for feature encoding length m (feature quantity).

In Fig. 6, the *Convolutional* ($k\_szie = 3, c = 16$) represents a convolution layer, where $k\_size = 3$ means that the convolution kernel is 3 and $c = 16$ means the number of output channels of this layer is 16. *Pooling* ($k = 2$) is a pooling layer. k is down-sampling factor whose value is 2 and *stride* is 1, *Deconvolutional* ($k\_szie = 3, c = 1$) is a deconvolution layer whose number of deconvolution kernel is 3 and the number of output channels is 16. *Depooling* ($k = 2$) is up-sampling layer. The original input data was a one-dimension vector of length 4096. After *Convolutional* ($k\_szie = 3, c = 16$), it became a matrix of 4096∗16 (16 channels, the length of each channel is 4096). Therefore, *Convolutional*($k\_szie = 3, c = 32$) changed the matrix of upper layer into 4096∗32. And then, the matrix became 2048∗32 after *Pooling*($k = 2$).

TABLE 2. Classification results of AE-CDNN-L1 for dataset 1.

m	*k*-NN	SVM1	SVM2	DT	RF	MLP	ADB	GNB	AVG
2	0.73336	0.6667	0.72668	0.75002	0.75002	0.69002	0.75334	0.51998	0.698765
4	0.76336	0.74668	0.78002	0.79002	0.79336	0.76336	0.78668	0.65666	0.7600175
8	0.76	0.81334	0.83336	0.82336	0.83334	0.82336	0.83668	0.67998	0.8004275
16	0.86666	0.87666	0.89334	0.89334	0.85668	0.88334	0.88668	0.80332	0.8700025
32	0.88002	0.93	0.94332	0.93998	0.93666	0.96336	0.92534	0.93334	0.92875
64	0.8394	0.94334	0.92	0.91332	0.89664	0.92666	0.92668	0.92334	0.91074
128	0.78	0.93666	0.90866	0.86666	0.88334	0.94666	0.92664	0.92	0.8960775

TABLE 3. Classification results of AE-CDNN-L2 for dataset 1.

m	*k*-NN	SVM1	SVM2	DT	RF	MLP	ADB	GNB	AVG
2	0.6	0.555	0.6	0.6025	0.6075	0.5875	0.6025	0.58	0.591875
4	0.5575	0.545	0.555	0.5475	0.5525	0.54	0.555	0.5575	0.55125
8	0.805	0.8375	0.83	0.8325	0.8675	0.855	0.8525	0.84	0.84
16	0.67	0.725	0.6975	0.725	0.7625	0.75	0.7125	0.75	0.7240625
32	0.7225	0.8725	0.8575	0.88	0.8725	0.865	0.87	0.8625	0.8503125
64	0.6725	0.9025	0.9025	0.8675	0.8825	0.895	0.8625	0.84	0.843125
128	0.5975	0.9125	0.855	0.845	0.92	0.9175	0.8825	0.9	0.85375

TABLE 4. Classification results of AE-CDNN-L1 for dataset 2.

m	*k*-NN	SVM1	SVM2	DT	RF	MLP	ADB	GNB	AVG
2	0.6	0.555	0.6	0.6025	0.6075	0.5875	0.6025	0.58	0.591875
4	0.5575	0.545	0.555	0.5475	0.5525	0.54	0.555	0.5575	0.55125
8	0.805	0.8375	0.83	0.8325	0.8675	0.855	0.8525	0.84	0.84
16	0.67	0.725	0.6975	0.725	0.7625	0.75	0.7125	0.75	0.7240625
32	0.7225	0.8725	0.8575	0.88	0.8725	0.865	0.87	0.8625	0.8503125
64	0.6725	0.9025	0.9025	0.8675	0.8825	0.895	0.8625	0.84	0.843125
128	0.5975	0.9125	0.855	0.845	0.92	0.9175	0.8825	0.9	0.85375

In this paper, we used several common classifiers, including k-NN, support vector machine (linear kernel and radial basis kernel), decision tree, random forests, multilayer neural network, AdaBoost algorithm, and Guass Bayesian classification, to classify features attained by unsupervised algorithm in order to verify the effectiveness of AE-CDNN model. All these classifiers are from the scikit-learn library [33] and the parameters in the classifiers are based on the default parameters in the library, which were shown in Table 1. In this table, "–" indicates that the parameters were set to default value.

C. EXPERIMENTAL RESULTS

In our experiment, we used the training set to train model and the testing set to extract features by the trained model. And then, we used several common classifiers to classify the extracted features, so that to verify the effectiveness of the features obtained by the unsupervised learning method. For AE-CDNN model, we proposed two different loss functions (10) and (11). Here, we treat the method that uses function (10) as AE-CDNN-L1, and the method that used function (11) as AE-CDNN-L2. Table 2 to 5 represent the classification results of different classifiers for features learned by AE-CDNN-L1 and AE-CDNN-L2, respectively of the two datasets. We used a 5-fold cross validation to calculate each classifier's accuracy. In the first column, m is the number of features that the model learnt. For example, in Table 2, $m = 2$ means that the model need to learn two features, and after 5-fold cross validation the average accuracy of the k-NN classifier is 73.336%. In the last column, AVG means the average accuracy for the listed classifiers.

Fig.7 shows the average accuracy's change of AE-CDNN-L1 and AE-CDNN-L2 based on different feature dimensions of the two datasets. When the number of features is greater than 8, the average performance of the above classifiers is acceptable. In general, the accuracy and stability of AE-CDNN-L2 are better than those of AE-CDNN-L1.

IV. DISCUSSION

In this section, we analyze the effectiveness of features extracted by AE-CDNN. Then, we compared the features of the model with other unsupervised feature extraction models. Finally, we performed comparison of our classification results with those of other related studies.

TABLE 5. Classification results of AE-CDNN-L2 for dataset 2.

m	*k*-NN	SVM1	SVM2	DT	RF	MLP	ADB	GNB	AVG
2	0.5525	0.5525	0.56	0.5525	0.5525	0.4975	0.555	0.55	0.5465625
4	0.7175	0.73	0.7325	0.7	0.73	0.73	0.73	0.725	0.724375
8	0.89	0.88	0.885	0.8725	0.8875	0.8825	0.885	0.87	0.8815625
16	0.7925	0.82	0.7925	0.7975	0.815	0.8175	0.8	0.7725	0.8009375
32	0.8275	0.905	0.85	0.88	0.9075	0.9125	0.9	0.8675	0.88125
64	0.805	0.8875	0.86	0.8825	0.91	0.73	0.88	0.8825	0.8546875
128	0.6825	0.84	0.84	0.835	0.8975	0.8725	0.8675	0.8825	0.8453125

TABLE 6. Classification results of AE-CDNN-L1 for dataset 1.

5-fold	*k*-NN	SVM1	SVM2	DT	RF	MLP	ADB	GNB
1	0.9	0.6667	0.85	0.9167	0.9333	0.75	0.9333	0.85
2	0.6667	0.6667	0. 6667	0.6667	0.6667	0.6667	0.6667	0.3333
3	0.6667	0.6667	0. 6667	0.6667	0.6667	0.6667	0.6667	0.3333
4	0.7667	0.6667	0.7833	0.8333	0.8167	0.7	0.8333	0.75
5	0.6667	0.6667	0. 6667	0.6667	0.6667	0.6667	0.6667	0.3333

TABLE 7. Classification results of AE-CDNN-L2 for dataset 1.

5-fold	*k*-NN	SVM1	SVM2	DT	RF	MLP	ADB	GNB
1	0.6667	0.6667	0.6667	0.6667	0.6667	0.6667	0.6667	0.3333
2	0.8667	0.85	0.8667	0.8833	0.8667	0.8667	0.8667	0.8667
3	0.7333	0.7167	0.7167	0.7167	0.7167	0.7167	0.7167	0.7333
4	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
5	0.7333	0.7333	0.7333	0.7333	0.7333	0.7333	0.7333	0.7333

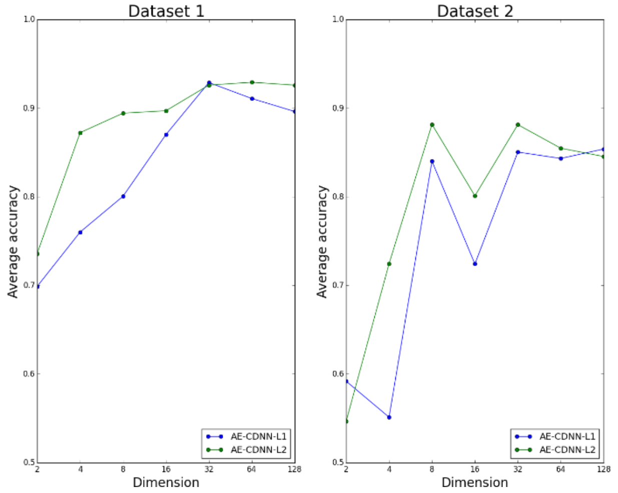

FIGURE 7. Average accuracy of AE-CDNN-L1 and AE-CDNN-L2 based on different feature dimensions of the two datasets.

A. NETWORK MODEL FEATURE EXTRACTION ANALYSIS

From Fig.7, when $m = 2$, the network models were difficult to learn effective features. Table 6 and 7 present the calculation results of 5-fold cross validation for dataset 1 with AE-CDNN-L1 and AE-CDNN-L2.

In Table 6 and 7, the accuracy with the value of 0.667 appeared multiple times. It is due to the initial weight of the network being 0 and the gradient of the autoencoder model disappeared quickly. It is very difficult to train multiple hidden layers (such as 2-4 layers) [34], which resulted in the hidden layer (including feature coding) not being able to receive training and the weights are 0. In addition, because the ratio of positive and negative samples in the testing set is 2:1, the accuracy is 0.667 when all samples were deemed as positive and the accuracy is 0.333 when all samples were deemed as negative. We can see that the hidden layers of network correspond to the 2nd, 3rd and 5th lines in Table 6, and the 1st line in Table 7 were not trained. It meant that the network has not learned any features. The problem also existed in dataset 2, whereby the ratio of positive and negative samples is 1:1. Therefore, when the model did not learn any features, the accuracy becomes 0.5, which is shown in the 2nd, 3rd, 4th and 5th lines of Table 8.

We increased the coding length of features. When m = 4, the model can learn features easier. Fig.8 shows the feature distribution of the two models in the first test of dataset 1. Here, f_1, f_2, f_3 and f_4 represents the four features. Note that f_2 and f_3 of AE-CDNN-L1 pertain to 0, which meant that the model did not learn any feature. From f_1 and f_4, we can see that the features obtained by AE-CDNN-L2 are more optimal than those of AE-CDNN-L1 in terms of feature

TABLE 8. Classification results of AE-CDNN-L1 for dataset 2.

5-fold	*k*-NN	SVM1	SVM2	DT	RF	MLP	ADB	GNB
1	0.7625	0.7625	0.8	0.7625	0.7625	0.4875	0.775	0.75
2	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
3	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
4	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5

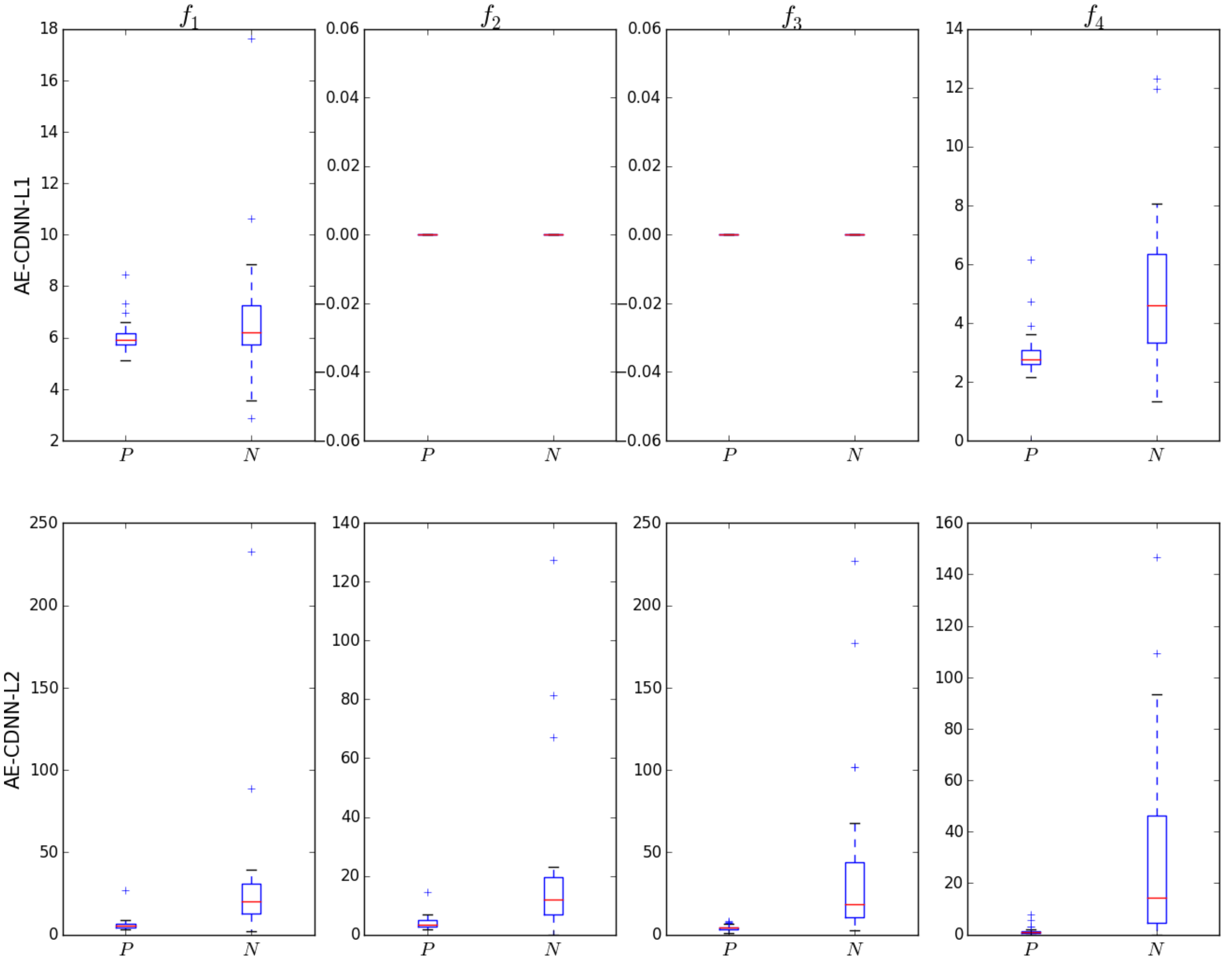

FIGURE 8. Feature distribution of AE-CDNN-L1 and AE-CDNN-L2 models in the first test of dataset 1 while feature dimension was 4.

separation degree. It also indicated that AE-CDNN-L2 can learn better features.

In Fig. 8, f_1 was the first feature of the learned feature coding array. f_1 and f_2 were independent of each other and the value of f_1 of the two models were independent and had no corresponding relationship. P, N represented the positive and negative samples. In the process of deep learning, the change of loss function value can reflect the learning situation of the deep network. Fig.9 shows the training of AE-CDNN-L1 and AE-CDNN-L2. We find that based on the testing set loss function of the two methods converged after 2000 epoch. From the convergence results of the two methods, we find that AE-CDNN-L2 could be a better choice.

B. COMPARISON OF DIMENSION REDUCTION METHODS

In last section, we discussed the feature learning ability of AE-CDNN model. We will compare the model with the existing main dimension reduction methods in this section. PCA is a very important linear dimension reduction method, which can obtain low dimensional features by linear transformation of high dimensional data [27]. Random projection (RP) is a powerful method used to construct Lipschitz mappings so as to realize dimension reduction with a high

TABLE 9. Classification results of AE-CDNN-L1 for dataset 1 (10-fold cross validation).

m	*k*-NN	SVM1	SVM2	DT	RF	MLP	ADB	GNB
16	0.94	0.93	0.94667	0.94333	0.93667	0.8766	0.9333	0.92
32	0.92333	0.93667	0.88667	0.93	0.93667	0.93	**0.95**	0.91333

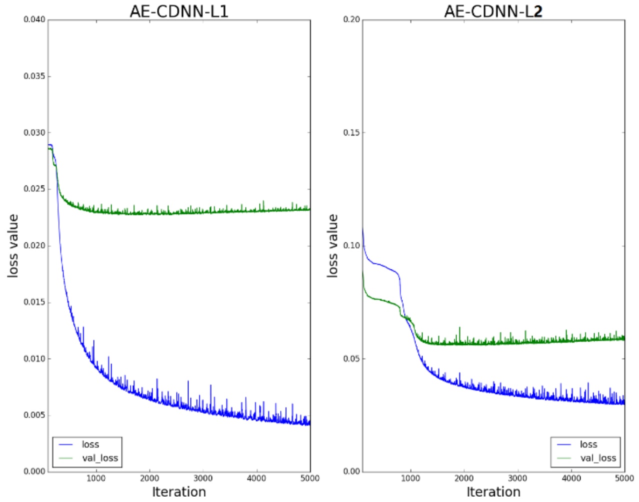

FIGURE 9. Change of loss function of AE-CDNN-L1 and AE-CDNN-L2 during training whereby the blue and green lines represent the change of loss function of the training and testing sets respectively.

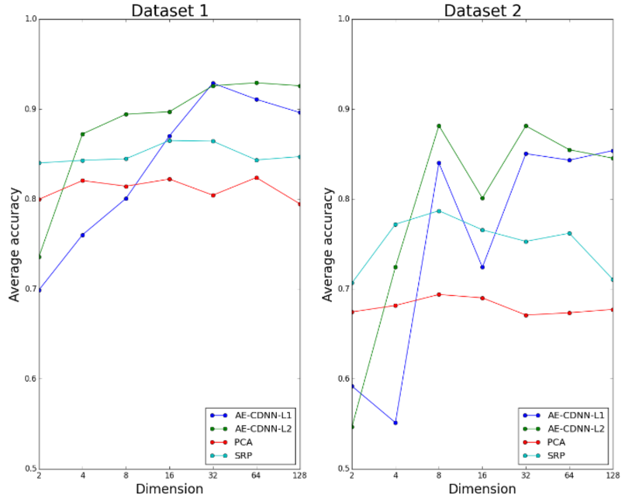

FIGURE 10. Comparison of average classification accuracy of the two datasets, obtained by AE-CDNN-L1, AE-CDNN-L2, PCA and SRP in each classifier.

probability [35]. Sparse random projection (SRP) can reduce the dimensionality by projecting the original input space using a sparse random matrix. Fig.10 presents the comparison of average classification accuracy of the two datasets that are obtained by AE-CDNN-L1, AE-CDNN-L2, PCA and SRP in each classifier. We find that the classification accuracy of features obtained by PCA and SRP are more stable, which mean that the number of effective features is not increased with the increase of total features. However, AE-CDNN-L1 and AE-CDNN-L2 can obtain new features as the increase of total features. When the number of features is greater than 4, the classification accuracy of features obtained by these two network performed better than those pertaining to PCA and SRP.

C. COMPARISON OF CLASSIFICATION APPLICATION

There are many classification researches on dataset 1. However, they all focused on designing new features based on the combination of existing features or searching useful features from a large number of features. For example, Pachori and Patidar [7] (2014) designed new features based on the empirical mode decomposition (EMD) and the second-order difference plot (SODP), and classified them using neural network. The best classification accuracy was 97.75%. Besides, Sharma and Pachori [8] (2015) combined the previous method and the phase space representation to obtain new features. They used SVM to classify and the best classification accuracy can reach up to 98.67%. Wen et al. used optimization algorithm to search features on the frequency of signal [36], which achieved an accuracy of 99%. It is worthwhile noting that high classification accuracy is often limited by specific data processing method, feature design, classification model, parameters, etc. When these constraints change, fluctuation of the accuracy occurs. Therefore, we need a model with autonomous learning ability that is not so limited by the classifier. Table 9 presents the classification results of AE-CDNN-L1 for dataset 1 after 10-fold cross validation. When m = 16 or 32, most of classifiers without parameter tuning can achieve good classification results.

Table 10 shows the 10-fold cross validation results of neural network (NN-2) with two hidden layers (the number of nodes in the hidden layer are 22 and 12) for features obtained by AE-CDNN-L1 in Table 9. The number 1-10 represent 10 independent experiments. Note that AVG is the average classification accuracy of the 10 experiments. The results show that the best classification accuracy can reach up to 100%. Fig.11 shows the change of loss function values of the NN-2 training process in Table 10. We find that based on the features obtained by AE-CDNN-L2, NN-2 can converge well at approximately 500 epoch during the learning process. It means that features attained by AE-CDNN-L2 are clear, effective and easy to learn.

Most studies related with the classification of EEG in epilepsy only focused on one dataset. But the application of AE-CDNN model in dataset 2 also gives good classification results. Because dataset 2 has multiple channels,

TABLE 10. Results of 10-fold cross validation based on NN-2.

m	1	2	3	4	5	6	7	8	9	10	AVG
16	0.9667	0.9333	0.9667	0.9333	0.9000	0.9333	1.0	0.9000	0.9000	1.0	0.9433
32	0.9667	0.9667	0.9000	0.9667	0.9667	0.9000	0.9667	0.8667	0.9333	1.0	0.9433

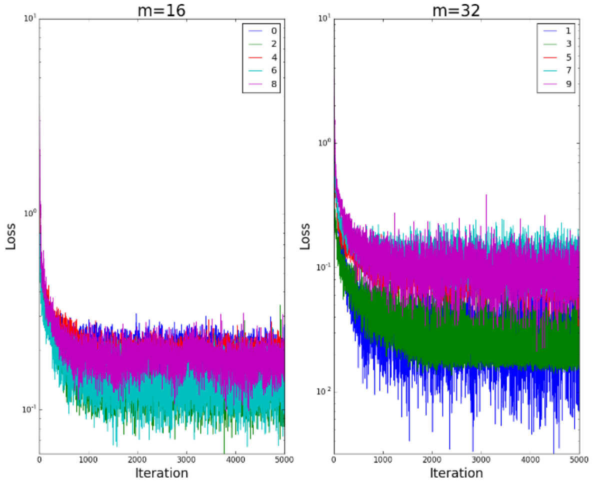

FIGURE 11. Change of loss function values of the NN-2 training process in Table 10.

Xun et al. used a window with the length of 5 secs to carry out single channel samples for classification, with an error rate of 22.93% [22]. Although there are some differences in the preprocessing, Table 5 shows that our method can achieve better classification accuracy.

V. CONCLUSION

In this paper, we attempt to use the deep convolution network and autoencoders to perform unsupervised feature learning of EEG in epilepsy, and to construct the AE-CDNN model. Our model can extract features from unlabeled EEG signals, which greatly reduce data dimension and achieve good classification accuracy. In our experiments, we have used two public EEG datasets (one is the intracranial EEG signals and the other is the brain epidermis EEG signals) to learn features and classify. Only from the classification results, we could see that using multiple classifiers without parameter tuning, the average classification accuracy of the obtained features could reach more than 92% when feature dimension was greater than 16. And the new method is more optimal than PCA and SRP in feature effectiveness after dimension reduction. However, the model performed badly when the dimension of features is lower. We find that it is the initial weight of the network that caused the problem. Therefore, our next research object is to discover how to pre-train the initial weight effectively. Our method is not inferior in terms of classification accuracy as compared to the other techniques. Because the traditional epilepsy classifications are based on single dataset, our method is less constrained. In future, we plan to make a meaningful visualization of the features extracted by the deep convolution network, which can be applied to feature recognition at all steps of epileptic seizure.

ACKNOWLEDGMENT

The findings achieved herein are solely the responsibility of the authors.

REFERENCES

[1] WHO. (2012). *Epilepsy*. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs999/zh/

[2] M. Z. Koubeissi, *Niedermeyer's Electroencephalography, Basic Principles, Clinical Applications, and Related Fields*, 6th ed. Philadelphia, PA, USA: Lippincott Williams & Wilkins, 2010.

[3] J. Gotman, "Automatic recognition of epileptic seizures in the EEG," *Electroencephalogr. Clin. Neurophysiol.*, vol. 54, no. 5, pp. 530–540, 1982.

[4] Y. Zhang, G. Xu, J. Wang, and L. Liang, "An automatic patient-specific seizure onset detection method in intracranial EEG based on incremental nonlinear dimensionalityreduction," *Comput. Biol. Med.*, vol. 40, nos. 11–12, pp. 889–899, 2010.

[5] A. Shoeb, A. Kharbouch, J. Soegaard, S. Schachter, and J. Guttag, "A machine-learning algorithm for detecting seizure termination in scalp EEG," *Epilepsy Behav.*, vol. 22, no. 1, pp. S36–S43, 2011.

[6] M. Qaraqe, M. Ismail, and E. Serpedin, "Band-sensitive seizure onset detection via CSP-enhanced EEG features," *Epilepsy Behav.*, vol. 50, pp. 77–87, Sep. 2015.

[7] R. B. Pachori and S. Patidar, "Epileptic seizure classification in EEG signals using second-order difference plot of intrinsic mode functions," *Comput. Methods Programs Biomed.*, vol. 113, no. 2, pp. 494–502, 2014.

[8] R. Sharma and R. B. Pachori, "Classification of epileptic seizures in EEG signals based on phase space representation of intrinsic mode functions," *Expert Syst. Appl.*, vol. 42, no. 3, pp. 1106–1117, 2015.

[9] G. Zhu, Y. Li, and P. Wen, "Epileptic seizure detection in EEGs signals using a fast weighted horizontal visibility algorithm," *Comput. Methods Programs Biomed.*, vol. 115, no. 2, pp. 64–75, Jul. 2014.

[10] A. T. Tzallas, M. G. Tsipouras, and D. I. Fotiadis, "Automatic seizure detection based on time-frequency analysis and artificial neural networks," *Comput. Intell. Neurosci.*, vol. 2007, Oct. 2007, Art. no. 80510.

[11] E. D. Übeyli, "Least squares support vector machine employing model-based methods coefficients for analysis of EEG signals," *Expert Syst. Appl.*, vol. 37, no. 1, pp. 233–239, 2010.

[12] A. S. M. Murugavel and S. Ramakrishnan, "Hierarchical multi-class SVM with ELM kernel for epileptic EEG signal classification," *Med. Biol. Eng. Comput.*, vol. 54, no. 1, pp. 149–161, 2016.

[13] X. Xiao, S. Pirbhulal, and K. Dong, "Performance evaluation of plain weave and honeycomb weave electrodes for human ECG monitoring," *J. Sensors*, vol. 2017, Jul. 2017, Art. no. 7539840.

[14] S. Pirbhulal, H. Zhang, and S. C. Mukhopadhyay, "An efficient biometric-based algorithm using heart rate variability for securing body sensor networks," *Sensors*, vol. 15, no. 7, pp. 15067–15089, 2015.

[15] W. Wu, H. Zhang, S. Pirbhulal, S. C. Mukhopadhyay, and Y. T. Zhang, "Assessment of biofeedback training for emotion management through wearable textile physiological monitoring system," *IEEE Sensors J.*, vol. 15, no. 12, pp. 7087–7095, Dec. 2015.

[16] R. Sharma, R. B. Pachori, and S. Gautam, "Empirical mode decomposition based classification of focal and non-focal seizure EEG signals," in *Proc. Int. Conf. Med. Biometrics*, May 2014, pp. 135–140.

[17] A. B. Das and M. I. H. Bhuiyan, "Discrimination and classification of focal and non-focal EEG signals using entropy-based features in the EMD-DWT domain," *Biomed. Signal Process. Control*, vol. 29, pp. 11–21, Aug. 2016.

[18] A. S. Zandi, M. Javidan, G. A. Dumont, and R. Tafreshi, "Automated real-time epileptic seizure detection in scalp EEG recordings using an algorithm based on wavelet packet transform," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 7, pp. 1639–1651, Jul. 2010.

[19] K. Polat and S. Güneş, "Classification of epileptiform EEG using a hybrid system based on decision tree classifier and fast Fourier transform," *Appl. Math. Comput.*, vol. 187, no. 2, pp. 1017–1026, 2007.

[20] U. R. Acharya, H. Fujita, V. K. Sudarshan, S. Bhat, and J. E. W. Koh, "Application of entropies for automated diagnosis of epilepsy using EEG signals: A review," *Knowl.-Based Syst.*, vol. 88, pp. 85–96, Nov. 2015.

[21] Y. R. Tabar and U. Halici, "A novel deep learning approach for classification of EEG motor imagery signals," *J. Neural Eng.*, vol. 14, no. 1, p. 016003, 2017.

[22] G. Xun, X. Jia, and A. Zhang, "Detecting epileptic seizures with electroencephalogram via a context-learning model," *BMC Med. Inform. Decision Making*, vol. 16, no. 2, p. 70, 2016.

[23] J. Masci, U. Meier, C. Dan, C. Dan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Proc. Int. Conf. Artif. Neural Netw.*, 2011, pp. 52–59.

[24] L. Chen, F. Rottensteiner, and C. Heipke, "Feature descriptor by convolution and pooling autoencoders," *Int. Arch. Photogram., Remote Sens. Spatial Inf. Sci.*, vol. 3, no. 3, pp. 31–38, 2015.

[25] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. IEEE Comput. Soc.*, May 2015, pp. 1520–1528.

[26] G. E. Hinton and R. S. Zemel, "Autoencoders, minimum description length and Helmholtz free energy," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 1993, pp. 3–10.

[27] T. Ahmad, R. A. Fairuz, F. Zakaria, and H. Lsa, "Selection of a subset of EEG channels of epileptic patient during seizure using PCA," in *Proc. World Sci. Eng. Acad. Soc. (WSEAS)*, 2008, pp. 270–273.

[28] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, Dec. 2010.

[29] D. P. Kingma and J. Ba. (2014). "Adam: A method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980

[30] R. G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. E. Elger, "Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 64, no. 6, p. 061907, 2001.

[31] A. H. Shoeb, "Application of machine learning to epileptic seizure onset detection and treatment," Dept. Electr. Med. Eng., Massachusetts Inst. Technol., Cambridge, MA, USA, 2009.

[32] T. Alotaiby, F. E. A. El-Samie, and S. A. Alshebeili, "A review of channel selection algorithms for EEG signal processing," *EURASIP J. Adv. Signal Process.*, vol. 2015, p. 66, Dec. 2015.

[33] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

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

[35] J. Wang, *Geometric Structure of High-Dimensional Data and Dimensionality Reduction*. Beijing, China: Higher Education Press, 2012.

[36] T. Wen and Z. Zhang, "Effective and extensible feature extraction method using genetic algorithm-based frequency-domain feature search for epileptic EEG multiclassification," *Medicine*, vol. 96, no. 19, p. e6879, 2017.

TINGXI WEN was born in Fujian, China. He received the M.S. degree in software engineering from Xiamen University. He has published widely in the field of robotic control based on EMG using various data classification methods. He has also developed a novel multi-objective optimization technique based on spark platform applied for optimizing routing. His research interests include data mining, machine learning, and cloud computing.

ZHONGNAN ZHANG (M'15) received the B.E. and M.E. degrees in computer science and technology from Southeast University, Nanjing, China, in 1999 and 2001, respectively, and the Ph.D. degree in computer science from The University of Texas at Dallas, TX, USA, in 2008.

Since 2017, he has been a Full Professor with the Software School, Xiamen University, Xiamen, China, where he was an Assistant Professor from 2009 to 2012 and an Associate Professor from 2012 to 2017. His research interests include big data analysis, data mining, machine learning, and bioinformatics.