Measurement 93 (2016) 490–502

Contents lists available at ScienceDirect

Measurement

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

Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis

Xiaojie Guo, Liang Chen *, Changqing Shen

School of Mechanical and Electric Engineering, Soochow University, Suzhou 215021, PR China

ARTICLE INFO

Article history:
Received 5 April 2016
Received in revised form 19 June 2016
Accepted 15 July 2016
Available online 17 July 2016

Keywords:
Fault diagnosis
Feature extraction
Adaptive learning rate
Deep convolution network
Hierarchical structure

ABSTRACT

Traditional artificial methods and intelligence-based methods of classifying and diagnosing various mechanical faults with high accuracy by extracting effective features from vibration data, such as support vector machines and back propagation neural networks, have been widely investigated. However, the problems of extracting features automatically without significantly increasing the demand for machinery expertise and maximizing accuracy without overcomplicating machine structure have to date remained unsolved. Therefore, a novel hierarchical learning rate adaptive deep convolution neural network based on an improved algorithm was proposed in this study, and its use to diagnose bearing faults and determine their severity was investigated. To test the effectiveness of the proposed method, an experiment was conducted with bearing-fault data samples obtained from a test rig. The method achieved a satisfactory performance in terms of both fault-pattern recognition and fault-size evaluation. In addition, comparison revealed that the improved algorithm is well suited to the fault-diagnosis model, and that the proposed method is superior to other existing methods.

1. Introduction

Rolling-element bearings are important components of many heavy-duty machines used in auto-manufacturing, shipping, etc. Faults in rolling-element bearings may impair machine operation, resulting in large economic losses and even human casualties [1]. Therefore, effective fault diagnosis plays a highly significant role in increasing the safety and reliability of machinery and reducing operation and maintenance costs [2].

As vibration signals are highly accurate indicator of the health conditions of mechanical equipment, they are widely used in fault diagnosis based on artificial methods, such as multinomial logistic regression, support vector machines (SVMs) and the wavelet packet transform (WPT) [3–5]. Kankar et al. [6] used multinomial logistic regression combined with the WPT to diagnose faults, and demonstrated the effectiveness of two wavelet features in indicating faults: energy and kurtosis. Huang et al. [7] proposed a multi-fault diagnosis method based on an improved SVM with a radial kernel basis function and empirical mode decomposition. Konar et al. [8] combined continuous wavelet transform with an SVM to create an advanced signal-processing tool, which was used to analyze frame vibrations. Zhang et al. [9] improved an SVM diagnosis system by adding ensemble empirical mode decomposition. Wang et al. [10] combined a new approach to Gauss-Hermite integration with Bayesian inference to estimate the posterior distribution of wavelet parameters, and [11] proposed a superior health evaluation method based on wavelet decomposition that performed satisfactorily in diagnosing early faults in gears. Li et al. [12] constructed a novel feature-extraction model to diagnose faults in rolling-element bearings based on two-dimensional non-negative matrix factorization and the generalized S transform. The model was found to work well. Yang et al. [13] compared the effectiveness of capacity, correlation and information parameters in classifying various fault patterns in rolling-element bearings and evaluating their health conditions, and proved using an SVM that all of the fractal dimensions and their combinations achieved a satisfactory performance. Liu et al. [14] presented a novel data fusion method based on fuzzy measures and fuzzy integrals, which performed satisfactorily in motor-fault diagnosis. Vakharia et al. [15] proposed a tool of using multi-scale permutation entropy for feature selection, and combined the method with both unsupervised and supervised classification methods. Besides, a comparison of different classification methods combined with this tool were also conducted by the tenfold cross validation and the results proved the satisfactory performance [16]. These traditional methods have all been shown to successfully extract the desired fault features.

Nevertheless, the effective diagnosis of the health state of machinery based on vibration signals remains a challenge when

* Corresponding author.
E-mail address: ChenL@suda.edu.cn (L. Chen).

http://dx.doi.org/10.1016/j.measurement.2016.07.054
0263-2241/

systems are highly complex, and the choice of suitable feature functions requires considerable machinery expertise and a deep mathematical foundation [17]. It is important for mechanical equipment to automatically extract failure features from rolling element bearing vibration signals [18]. Therefore, intelligent methods such as back propagation (BP) neural networks and artificial neural networks have been widely investigated and used to diagnose faults in rotating machinery [19,20]. BP neural networks with wavelet packet decomposition were improved and used to identify faults in rolling-element bearings by Liang et al. [18]. Recently, deep-learning methods such as the combination of a deep-belief network with the WPT have been found to overcome obstacles to fault diagnosis in complex machines [21,22]. However, these methods still require artificial feature extraction for pre-training. Convolution neural networks (CNNs), which are specifically designed for use with variable and complex signals, have been shown to outperform all other techniques. LeCun et al. [23] first designed a CNN and optimized the model using an error-gradient algorithm. Due to their unique ability to maintain initial information regardless of shift, scale and distortion invariance—achieved through local receptive fields, shared weights and spatial subsampling—CNNs are widely used in image classification [24], large-scale speech tasks [25], traffic-sign recognition [26] and various other applications [27–33]. To increase training speed, many accelerators have been designed with improved hardware equipment, such as memory [33] or graphics-processing units [34]. However, few researchers have used CNNs for fault diagnosis, and the algorithms on which existing methods are based require improvement. Convolution algorithms can help to remove noise from vibration signals. As a deep architecture, ADCNN can extract features automatically without prior information to a high-level degree compared to shallow architectures which rely on the prior information mainly. ADCNN also holds the potential to deal with big data from rotating machinery, which also helps in extracting more meaningful information from mass samples. Therefore, a hierarchical learning rate adaptive deep CNN (ADCNN) is proposed in this study, and its application to bearing-fault diagnosis and fault-severity evaluation is examined. To more precisely reflect the adaptive process, a traditional deep CNN (DCNN) is improved by the addition of an adaptive learning rate and a momentum component, which prevent training failure caused by an unsuitable learning rate [35]. The merits of the proposed method are as follows: (1) the ability to extract deep fault features automatically and sensitively, without manual feature selection or extraction, (2) the use of ADCNNs rather than traditional DCNNs to choose a suitable learning rate, and (3) a higher degree of accuracy than traditional methods of fault diagnosis. The proposed model comprises two functional layers: the first responsible for fault-pattern recognition, classifying faults into four patterns; and the second responsible for fault-size evaluation, calculating actual fault sizes. Both layers have the same improved ADCNN structure.

The rest of this paper is organized as follows. In Section 2, the basic theory of deep convolution networks and the proposed ADCNN algorithm are detailed. In Section 3, the proposed hierarchical ADCNN structure is described, and the model is experimentally validated using a rolling element bearing dataset in Section 4. In Section 5, the effectiveness and superiority of the proposed method are demonstrated by comparison with the traditional DCNN method and an artificial support vector regression machine (SVRM) method. Conclusions are drawn in Section 6.

2. The propose of ADCNN and its background

DCNNs represent the state of the art in automatic feature extraction, due to their full supervision and basis in classical gradient descent. Typical DCNN models have two parts: a feature extractor and a multilayer perception (MLP) classifier. The feature extractor is composed of a convolution layer and a pooling layer. The convolution layer is responsible for extracting signal features, and the pooling layer further reduces computation time and gradually establishes the invariance of space and structure, while maintaining the basic characteristics of the original signals. Next, the extracted features are input into the MLP classifier. The MLP is constructed with hidden layers and output layers. All of the nodes in a DCNN have the same function, namely a sigmoid function. We improve the traditional DCNN model by adding adaptive training, resulting in the proposed ADCNN model. The theoretical basis of automatic feature extraction using the ADCNN method is detailed below.

2.1. Forward-transmission process

Consider the vibration-signal dataset $\{X, Y\}$, where X refers to the input signal vector and Y refers to the target value, namely the fault-pattern index. N is the number of training samples. Each input can be transformed into a matrix with several dimensions [36]. In a convolution layer, feature maps can be convoluted using filters (the function of kernels in a convolution layer), generating new feature maps [37]. Each output map is the result of convoluting multiple input features. The progress of feature maps in a convolution layer can be defined as follows:

$$O_j^l = f\left(\sum_{i\in M_j} \sigma_{ij} O_i^{l-1} * W_{ij}^l + b_j^l\right) \tag{1}$$

where O_i^{l-1} represents the output of the $l-1$th layer and the input into the following layer through the feature map i, and O_j^l represents the jth feature map of the $l-1$th layer. $f(o)$ refers to the sigmoid function, M_j represents a selection of input maps and W_{ij}^l is the weight of the kernel connecting the ith feature map of the $l-1$th layer with the jth feature map of the lth layer. $*$ denotes the computation of convolution. An additive bias b, is given to each output map. The input maps associated with a particular output map are convolved using several kernels. During the convolution process, weight distribution and size are the same for every output figure map, which considerably reduces the training parameters.

A feature map of the sampling layer is produced using the following formula:

$$O_j^l = f\left(\delta_j^l S(O_j^{l-1}) + b_j^l\right) \tag{2}$$

where $S(o)$ represents the max-pooling function, which maximizes each group with a certain block size in the feature map, and δ_j^l represents the deviation in the multiplier for the jth feature map of the lth layer.

One of the advantages of ADCNN is the automatic extraction of features layer by layer. Each layer of ADCNN can be regarded as a feature map obtained from the previous layer, and the weights and bias combining the two layers can be regarded as feature extractors. The dimensionality of the final feature depends on the number of neural units of the last layer of ADCNN. Gradient descent is used to update the feature extractors of each layer (weights and bias) to obtain the best classification result.

2.2. Gradient descent during BP

The gradient of the loss function for all of the weights in all of the layers is calculated by BP. A squared-error loss function is used to address the objective function. This is a multiclass problem with m classes and N training examples. As an illustration, Formula (3) presents the loss function after training a single sample, n. The

overall loss function is calculated by summing the loss functions of the individual samples, as follows:

$$E^n = \frac{1}{2}\sum_{k=1}^{m}\left(l_k^n - y_k^n\right)^2 \tag{3}$$

where n is the sample index and k is the label index. Here, l_k^n is the kth corresponding label in sample n and y_k^n is the value of the output-layer unit. In multi-class classification problems, the output is usually expressed as a vector, and only the output node of the class corresponding to the input dimension is positive. The other class nodes are zero or negative, which depends on the activation function in the output layers: the sigmoid function is 0 and the tanh function is 1.

Next, the sensitivity of neuron ϕ_j^l from u neurons to v neurons in a layer is calculated, and the gradient of the kernel weights is computed by BP as follows:

$$grd = \frac{\partial E}{\partial W_{ij}^l} = \sum_{u,v}\left(\phi_j^l\right)_{u,v}\left(O_j^{l-1}W_{ij}^l\right) \tag{4}$$

To reduce the number of parameters and the complexity of the calculation, a sparse connection rather than a full connection is added to the DCNN. σ_{ij} refers to sparse weight, and the constraint function $\phi(\sigma_{ij})$ is introduced. The loss function and the gradient can then be calculated, respectively, as follows:

$$\widetilde{E}^n = E^n + \sum_{ij}|\sigma_{ij}| \tag{5}$$

$$grd = \frac{\partial \widetilde{E}}{\partial W_{ij}^l} = \frac{\partial E}{\partial W_{ij}^l} + \frac{\partial \varphi(\sigma_{ij})}{\partial W_{ij}^l} \tag{6}$$

2.3. Improved adaptive training

Using traditional methods, the learning rate is a global constant. However, a high learning rate increases loss error and results in excessive changes in weight. A low rate can prevent these problems, but increases convergence time. To extract the most typical features and obtain the best recognition results, we introduce an improved algorithm with an adaptive learning rate to give the proposed ADCNN model. For each W_{ij}^l update, the parameter α or β is used to update the learning rate, respectively, as follows:

$$\lambda = \begin{cases} 1 & \text{if} \quad \operatorname{sgn}(grd_{ij}^k grd_{ij}^{k-1}) = 1 \\ -1 & \text{if} \quad \operatorname{sgn}(grd_{ij}^k grd_{ij}^{k-1}) = 0 \end{cases} \tag{7}$$

$$r(k) = \alpha^{\lambda} r(k-1) \tag{8}$$

where grd_{ij}^k is the gradient of the weight after the kth training. As λ is -1 or 1, to ensure that α^{λ} falls into the range 0–1 or 1–2, respectively, the value of α should be as follows: $1 < \alpha < 2$. $sgn(o)$ denotes the function used to judge the orientation of the two neighboring gradients. Therefore, as the loss function decreases constantly, learning rate increases to accelerate convergence; If the loss function increases excessively, learning rate decreases. This method prevents the vanishing (or exploding) gradient problem [38], which explains why many deep-learning models fail when trained by BP.

The gradient-descent algorithm always updates weight, W_{ij}^k, toward a negative gradient orientation without considering past experience, i.e., the gradient orientation of the previous moment. This often leads to oscillation and slow convergence during training. Therefore, a momentum item is added. Accordingly, weight with respect to loss-function gradient is computed as follows:

$$W_{ij}^k = W_{ij}^{k-1} - r(k)\left[(1-\beta)\frac{\partial E}{\partial W_{ij}^k} + \beta\frac{\partial E}{\partial W_{ij}^{k-1}}\right] \tag{9}$$

where β should be in the range 0–1, according to the weight distribution of the last two training effects.

3. Proposed hierarchical ADCNN

Deep neural networks can adaptively capture the representation information from raw signals through multiple non-linear transformations and approximate complex non-linear functions with small error. Having a deep architecture, ADCNN combines three architectural ideas to ensure some degree of shift, scale, and distortion invariance: local receptive fields, shared weights and spatial sub-sampling [23]. This enables it to extract useful features layer by layer as well as dispose valueless information, such as interferential noises in the practical situations.

As previously mentioned, effective bearing-fault diagnosis requires not only fault patterns but fault sizes to be identified, enabling the health status of a bearing to be tracked. This facilitates timely maintenance [39]. Based on the theory outlined above, a hierarchically structured ADCNN model is presented here to fulfill this requirement. The model has two hierarchically arranged components: a fault pattern determination layer and a fault size evaluation layer. These layers are described in detail in the following sections. The proposed structure is presented in Fig. 1. Two sets of data are prepared for model verification: one consists of training samples with prior information, and the other consists of testing samples for subsequent labeling.

3.1. Fault pattern decision layer

As Fig. 1 shows, the first layer is responsible for fault-pattern identification, which entails pattern recognition or pattern classification. In this study, a new classification strategy known as an ADCNN is proposed. This model offers a highly effective means of extracting features automatically from a series of signals. Although bearing vibration signals contain a large amount of data, the ADCNN helps to accelerate convergence during training and prevents the vanishing-gradient problem associated with most deep-learning methods. The class labels are defined as $1, 2, \ldots, C$ for a problem with C classes, and the dataset $\{X_i, Y_i\}^N$ is constructed, where x_i is an input vibration signal vector, $y_i \in \{1, 2, \ldots, C\}$ represents each of the target values and N is the number of training samples. Next, training samples are put into the first ADCNN layer for fault-pattern recognition. The ADCNN model in the first layer is based on the classical LeNet5 models proposed by LeCun [21]. The architecture of the proposed ADCNN is shown in Fig. 2. Each plane is a feature map, with a set of units whose weights must be identified.

The first ADCNN structure has seven main layers. The first layer converts the signal-vector input into a matrix, which is the typical DCNN input format. The next three layers are ConvNet layers, each of which comprises a convolution layer and a max-pooling layer. There are 5, 10 and 10 feature filters in the first, second and third ConvNet layers, respectively. The next two layers are full connection layers, which prepare features for classification. The last layer is a logistic-regression layer, which uses the softmax method for classification. The weight in each layer is randomly initialed and trained for optimization. After training, the test samples are input into the logistic-regression layer, whose output comprises class labels corresponding to the samples.

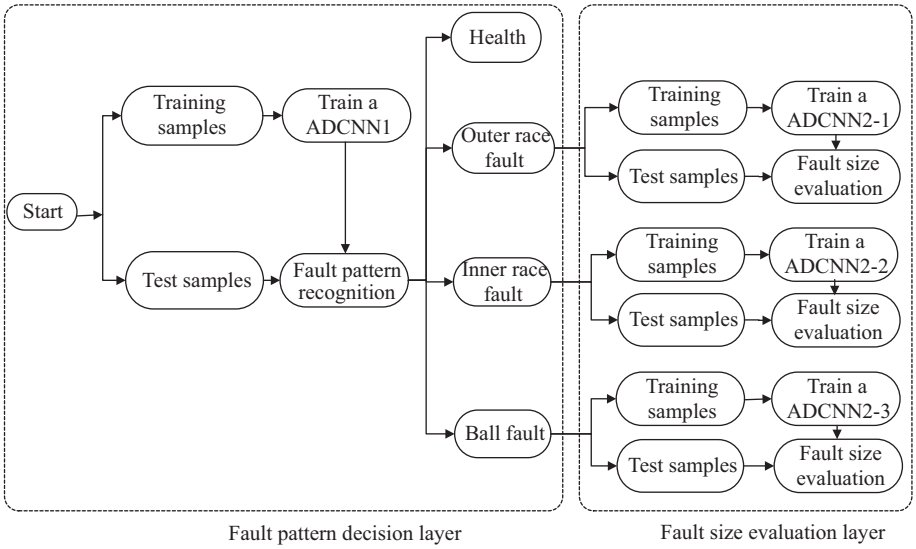

Fig. 1. Hierarchical framework of ADCNNs.

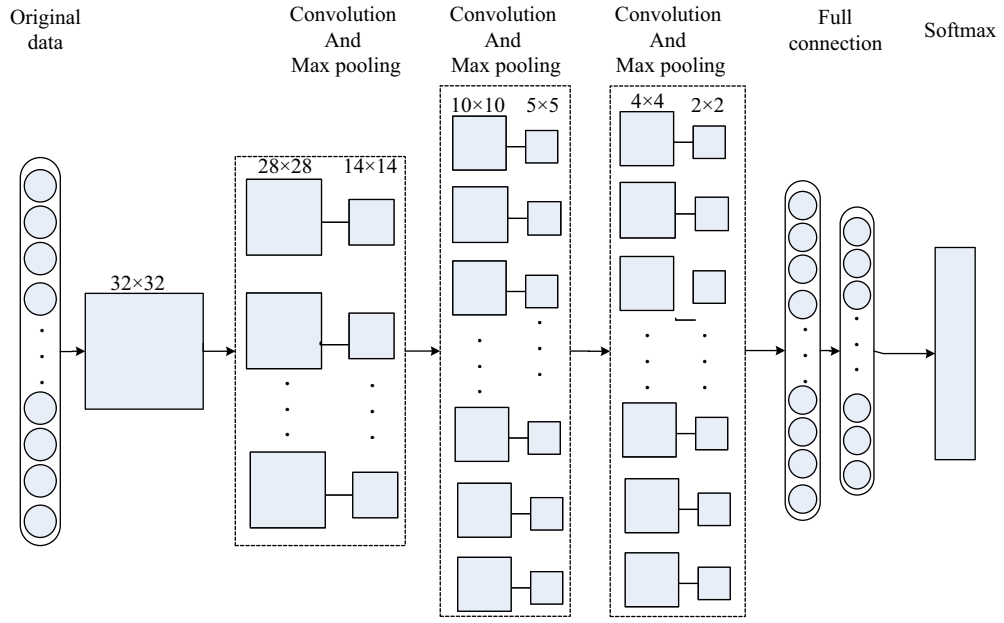

Fig. 2. Architectural hierarchy of ADCNNs.

3.2. Fault size evaluation layer

After fault-pattern recognition, three ADCNNs are constructed for each pattern. Each has the same structure as the ADCNN in the first layer, because the associated samples have the same dimensions. As described above, the weight in each layer is randomly initialed and trained for optimization. However, after training, the test samples are input into the layer, and the output is a probability vector indicating the probability that a given sample belongs to each class. Due to the requirement of size recognition, we propose a method of calculating the fault size of each sample rather than providing a simple label. Using the softmax method, the probability that each sample belongs to each class size is calculated as follows:

$$P(x_i) = \{p_1, \ldots, p_C\} \tag{10}$$

Accordingly, the fault size for each sample can be calculated as follows:

$$S_i = \sum_{j=1'}^{c} A_j p_j \tag{11}$$

where A_j is the typical fault size for the jth class and c is the number of classes. p_j refers to the probability that the ith sample belongs to the jth class. This enables the system to output a predicted size for each sample.

3.3. Selection and optimization of parameters for ADCNNs

To effectively train the ADCNNs in each layer, three key parameters must be properly determined: learning rate, batch size and number of kernels in each layer. The process of selecting individual parameters is discussed in the following sections.

3.3.1. Selection of learning-rate parameter

Learning rate, the coefficient of the gradient during stochastic gradient descent (SGD), is very important to the final result, as a

Table 1
Parameters of ADCNN during training.

Layer part	The first layer	The second layer		
Parameters	ADCNN1	ADCNN2-1	ADCNN 2-2	ADCNN 2–3
No. of layers	9	9	9	9
Learning rate	Adaptive	Adaptive	Adaptive	Adaptive
No. of epochs	8000	2000	2000	2000
Batch size	100	100	100	100
No. of kernels	[5, 10, 10]	[5, 10, 10]	[5, 10, 10]	[5, 10, 10]

high learning rate will prevent optimization and a small learning rate will yield a local optimum. To date, no theoretical strategies have been proposed for choosing a suitable learning rate. Many researchers have chosen learning rates based on experience.

First, based on the number of samples used in this study, a learning-rate range of 0.0002–0.02 is identified. Next, we test the learning rate at 0.0005 intervals in this range. The results indicate that 0.001 is the best choice; rates lower than 0.0001 causedivergence and rates higher than 0.01 result in a local optimum with a low classification accuracy. However, the use of a large range for data training is unwise, especially with a big dataset. Therefore, an adaptive method of selecting learning rate is proposed. The learning rate is updated in the direction of the gradient, which always ensures a suitable learning rate. Convergence is enhanced regardless of the initial rate chosen.

3.3.2. Selection of batch-size parameter

As the training samples make up a large dataset, we use mini-batch SGD to perform the training. First, the samples are packed into a batch and input into the ADCNN system. Next, the parameters are optimized according to the mean loss function of the whole batch. "Batch size" denotes the number of samples in a batch, which obviously affects the optimization performance and training rate of the model.

As the training dataset and the test dataset each comprise 500 samples, batch size should be a divisor of 500, such as 100, 50, 25, 5 or 1; otherwise, samples will be wasted and test accuracy may be impaired. If the batch is small, each sample will comprise few classes, leading to slow optimization. A large batch will result in a local optimum. Therefore, batch sizes are compared, and 50 is found to yield the best convergence and the highest accuracy.

3.3.3. Selection of number of kernels in convolution layer

The number of kernels determines the features extracted. As the model contains three convolution layers, three parameters must be decided. In theory, the more kernels used, the more features are extracted, which increases accuracy. However, the use of too many kernels increases the scale of the parameters, which complicates the computational process. As each sample comprises 1024 data units, each layer should have 5–10 kernels. We compare the performance of the three layers with different numbers of kernels, and set kernel number at 5, 10 and 10, respectively.

The number of epochs is based on the number of samples, and has little effect on the final result. The parameters chosen are listed in Table 1.

4. Validation of proposed method

The bearing-fault data used for experimental validation were provided by Case Western Reserve University [40]. The data were collected from a test motor driving system, as shown in Fig. 3. The main components of the experimental apparatus were a 2-hp motor (left-hand side of figure), a torque transducer and a dynamometer (right-hand side of figure). The motor shaft was supported by 6205-2RS JEM SKF bearings. The three bearing components under study are (1) the inner race (IR), (2) the outer race (OR) and (3) the rolling element, the ball (B). Each is artificially given a single-point fault using electrical-discharge machining. For the faults localized to the IR, the B rolling element and the OR, the accelerometers are arranged in the dead-end position at 12 o'clock, 6 o'clock and 3 o'clock, respectively, and used to sample vibration signals at 12 kHz. Fig. 4 shows the data samples obtained for the different bearing health conditions. A dataset comprising health signals and three fault-pattern signals is used for analysis. As Table 2 shows, in the first layer of the hierarchical ADCNN model, 1000 samples for each condition are used for training (500 samples) and testing (500 samples). In the second layer, 50 training samples and 50 testing samples are used for fault-size evaluation. In the first layer, the IR-fault, B-fault, OR-fault and bearing health target values are artificially set at 1, 2, 3 and 4, respectively; during the training in the second layer, actual fault size is given in terms of the target values [14].

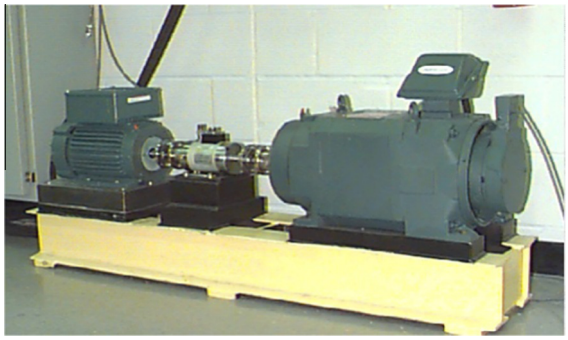

Fig. 3. Experimental platform for acquiring vibration signals from rolling bearings.

The bearing data are arranged based on the proposed hierarchical structure, as illustrated in Fig. 3. In the first layer of the two-layer ACDNN, 500 samples are input to train the system; suitable weights are determined and updated and the model is constructed. Next, another 500 samples are tested for fault-pattern recognition, with the class labels belonging to individual samples as the output. Fig. 5 shows the training results and testing results for IR, OR and B faults, and the health patterns obtained for the whole dataset.

Fig. 5 shows that five testing samples are misjudged and that the other 495 samples are properly classified. Two OR fault samples are identified incorrectly as a B fault and one OR fault sample is identified incorrectly as an IR fault. Two B fault samples are misjudged as a health pattern and an OR fault, respectively. Therefore, the first layer of ADCNN model has an accuracy of 99%. Although the concrete expression of features is invisible, the ADCNN model automatically and almost perfectly carries out feature extraction and classification. Another validation method, tenfold validation, is adopted to test the proposed method further. The samples are divided into ten parts for cross validation and the mean validation accuracy is 97.9%.

In the second layer of the two-layer ADCNN, fault size is evaluated. Samples representing three levels of fault severity for each fault pattern, as shown in Table 2, are used to train and test the model. Table 3 shows the accuracy of fault-size evaluation for each health condition.

As shown in Table 3, the classification result indicates that the fault-size evaluation conducted in the second layer is highly accurate: in most cases reaching 100% accuracy. Concrete fault-size calculation is more important than classification to the overall process of fault-size evaluation. We thus use the result of fault-size calculation to further verify the proposed method.

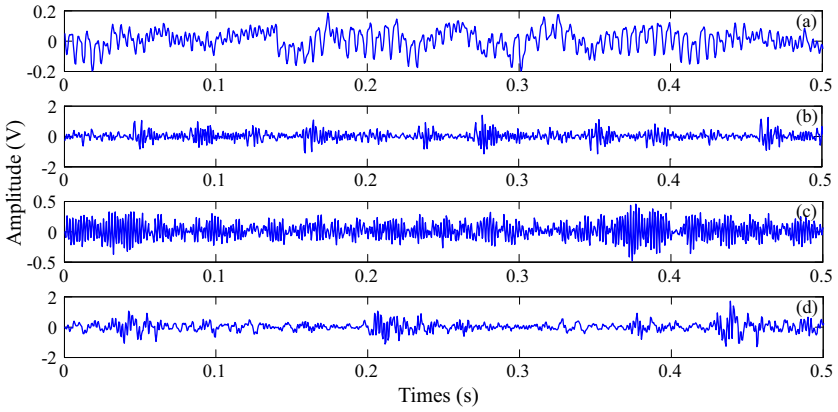

Fig. 4. Data samples obtained for different bearing health conditions: (a) normal signal, (b) IR-fault signal, (c) OR-fault signal and (d) B-fault signal.

Table 2
Parameters of rolling bearings.

Items	Health	Fault 1			Fault 2			Fault 3		
Fault location	None	OR	OR	OR	IR	IR	IR	BA	BA	BA
Fault size	0	0.007	0.014	0.024	0.007	0.014	0.021	0.007	0.014	0.021
Testing samples no.	50	50	50	50	50	50	50	50	50	50
Training samples no.	50	50	50	50	50	50	50	50	50	50

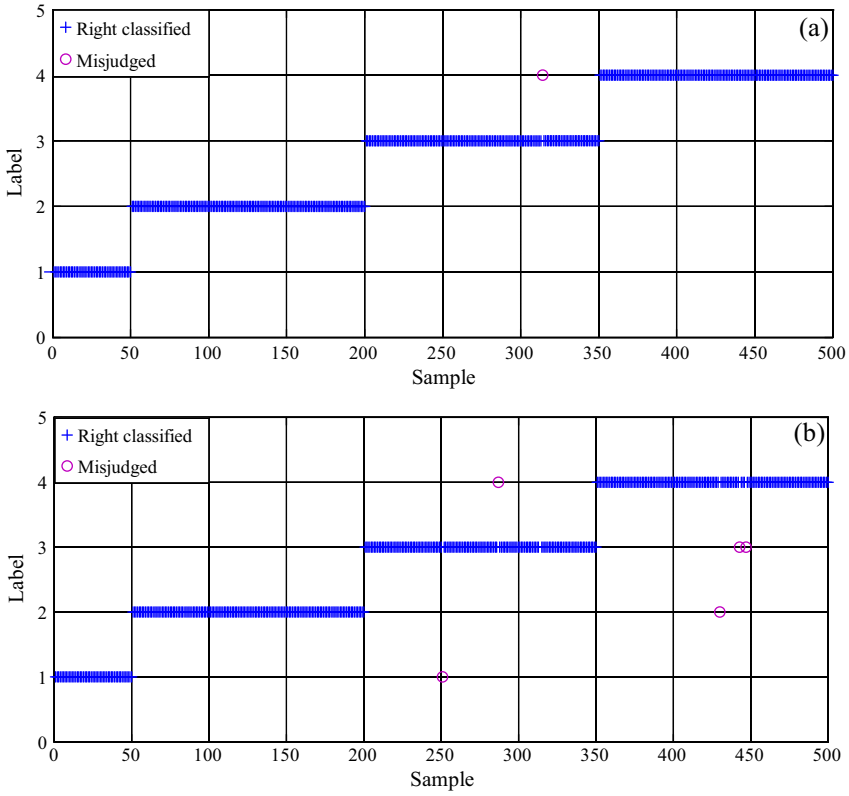

Fig. 5. (a) Training results and (b) testing results for first layer of hierarchical ADCNN.

Fig. 6 shows the training and testing results for the IR-fault samples obtained using the fault size calculation function shown in Formula (11). The maximum error in the predicted result is less than 0.0004 in. for testing samples.

Fig. 7 presents the training and testing results of B-fault size prediction. As B-faults are the most complicated of the three fault patterns, due to the stochastic motion of a defective B, it is difficult to precisely determine the relationship between the signal vector and actual fault size. However, in the second layer of the proposed two-layer ADCNN model, the maximum error in predicted B fault size is less than 0.004 in., although predicted B-fault size shows greater fluctuation than predicted IR-fault or OR-fault size.

Table 3
Fault size evaluation results.

Health condition	Testing accuracy (tenfold cross validation)	Training accuracy
Inner race fault	100% (100%)	100% (100%)
Ball fault	99.3% (94.4%)	100% (100%)
Outer race fault	100% (100%)	100% (100%)
Overall accuracy	99.7% (98.1%)	100% (100%)

Similarly, Fig. 8 illustrates the training and testing results for OR-fault size prediction. The predicted fault sizes match well with actual fault size. The maximum error in both the training and the testing results is 0.0002 in.

As described in Section 3, the proposed two-layer ADCNN model provides a systematic method of bearing-fault diagnosis that can be used to identify both fault size and fault severity in a bearing. This method is experimentally validated using a bearing dataset and found to yield satisfactory results.

5. Comparison with traditional DCNN and SVRM methods

The proposed hierarchical ADCNN model provides a systematic and accurate method of diagnosing bearing faults, and is shown to solve the problem of choosing a suitable learning rate. To further demonstrate the superiority of the proposed method to existing methods, its performance is compared with that of the typical DCNN method and an SVRM method, respectively.

5.1. Comparison with traditional DCNN

We compare the convergence speed and training error of the proposed model with those of a typical DCNN model, LeNet5. As the two layers of the ADCNN model have the same theoretical basis and structure, the comparison is conducted using only the first layer. To choose the learning-rate parameter for comparison, we first test a wide range of learning rates, namely 0.0005, 0.005 and 0.05. Fig. 9 shows the convergence achieved by the ADCNN and DCNN models with the same initial learning rate of 0.0005. As shown in Fig. 9, the ADCNN method achieves more rapid convergence than the traditional DCNN method during the first 1000 epochs, with an approximately 10% smaller validation error.

Fig. 10 shows the convergence achieved by the DCNN and ADCNN models with the same initial learning rate of 0.005. 0.005 is the best rate for both the ADCNN method and the DCNN method. This rate also yields a higher initial performance with the ADCNN model than the traditional DCNN model, although both eventually achieved the same validation error.

As previously discussed, the use of a learning rate as large as 0.05 for training yields fairly low recognition result, as shown in Fig. 11. The proposed ADCNN method helps to reduce the validation error, although this level of error is still undesirable.

Comparison of the global constant learning rate with the adaptive learning rate method reveals that the ADCNN model provides the most rapid convergence and the highest accuracy. Despite improvement on the global learning rate, the initial learning rate still has little effect on the outcome, as shown in Figs. 9 and 10. There is an 8% gap in the accuracy of the ADCNN model between the0.005initial learning rate and the 0.05 initial learning rate. Learning rate is only detrimental if very large, when it produces a small gap.

5.2. Comparison with SVRM

The SVRM model has a two-layer structure with statistical parameters for tracing fault patterns and fault size, respectively. It is developed based on the principle of SVM but with higher accuracy, for which it is chosen to make a comparison. First, the statistical parameters are extracted to compress the original

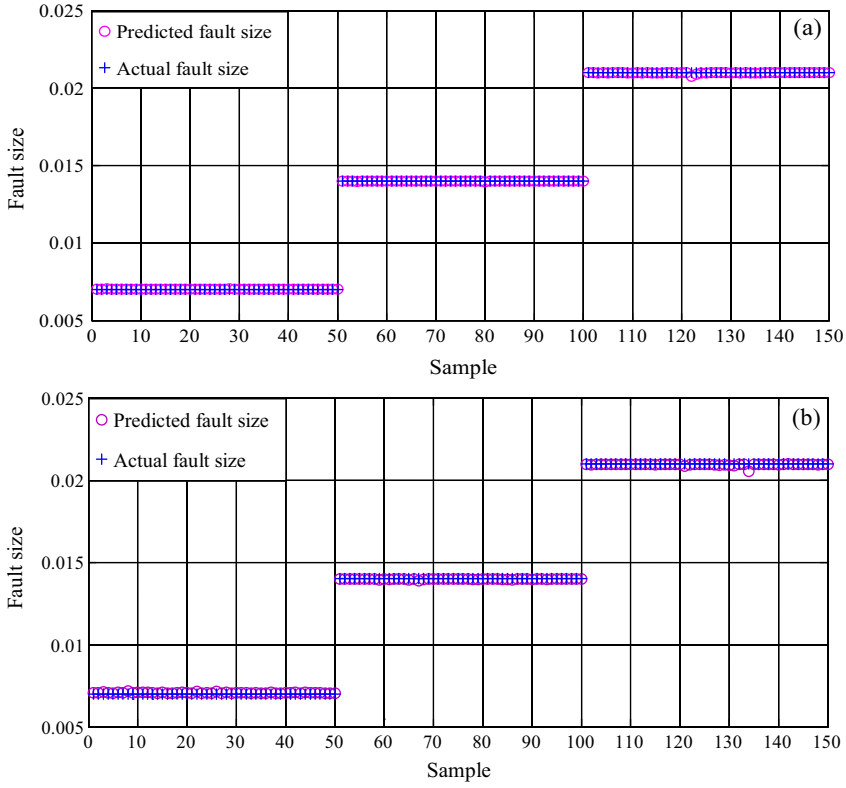

Fig. 6. Predicted IR-fault sizes for (a) training samples and (b) testing samples.

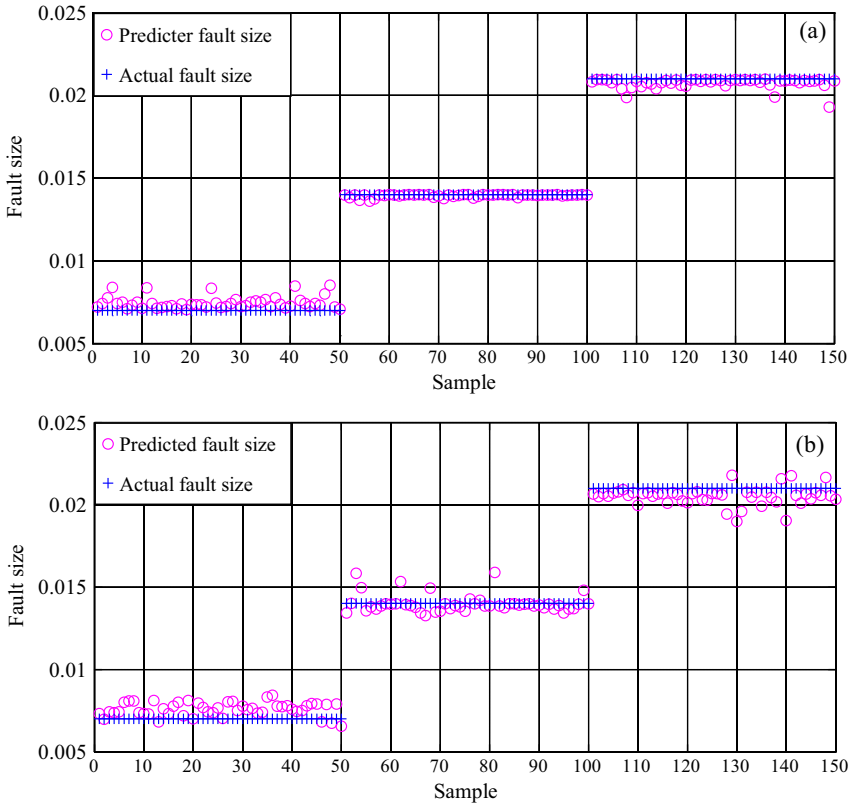

Fig. 7. Predicted B-fault sizes for (a) training samples and (b) testing samples.

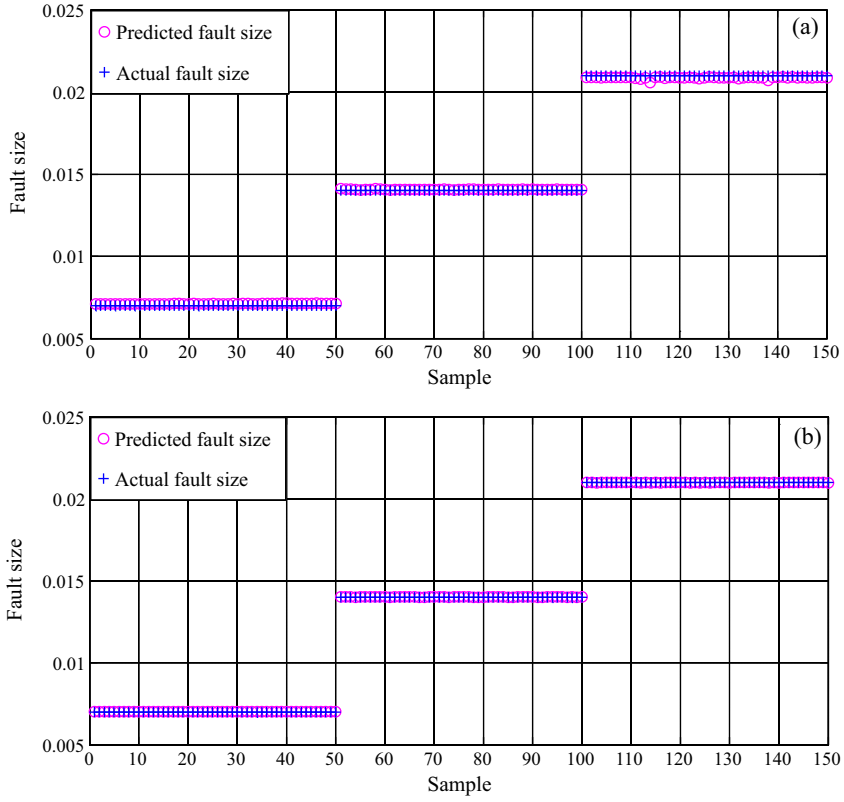

Fig. 8. Predicted OR-fault sizes for (a) training samples and (b) testing samples.

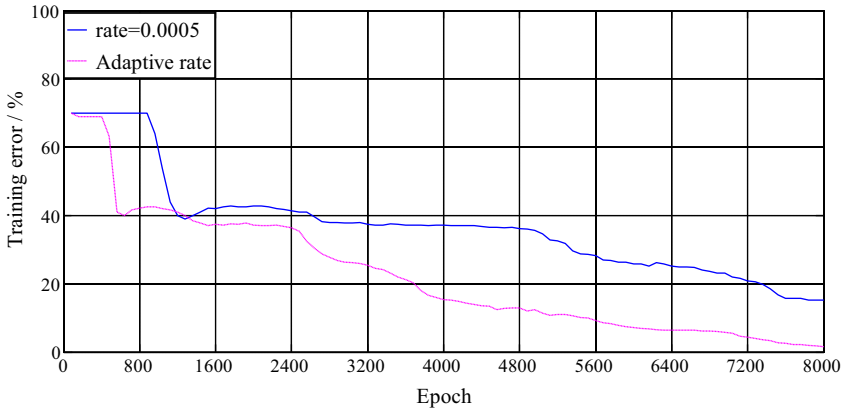

Fig. 9. Convergence of traditional DCNN and ADCNN at learning rate of 0.0005.

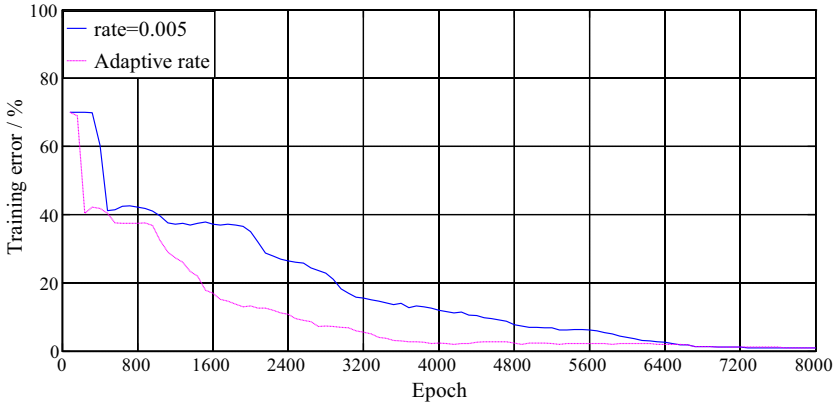

Fig. 10. Convergence of traditional DCNN and ADCNN at learning rate of 0.005.

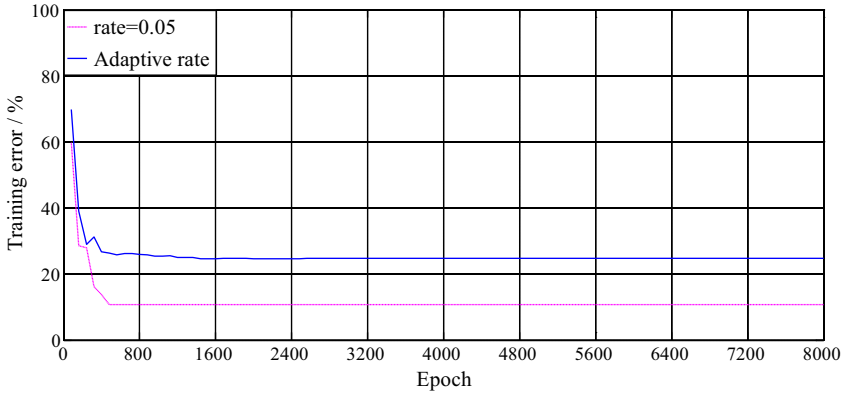

Fig. 11. Convergence of traditional DCNN and ADCNN at learning rate of 0.05.

vibration signals into nine compact features, which are then used to train an SVRM model. Table 4 shows the nine statistical parameters.

Then the SVRM optimization problem to be solved is defined as:

$$\min \frac{1}{2}|w|^2 + C\sum_{i=1}^{n}\left(\xi_i + \xi_i^*\right) \tag{12}$$

$$\begin{cases} y_i - w \cdot x_i - b \leqslant \varepsilon + \xi_i \\ w \cdot x_i + b - y_i \leqslant\ll \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* > 0 \end{cases} \tag{13}$$

Table 4
The nine statistical feature parameters.

Kurtosis: $\frac{1}{N}\sum_{i=1}^{N} x_i^4$	Clearance factor: $\frac{\max(\lvert x_i\rvert)}{\left(\frac{1}{N}\sum_{i=1}^{N}\sqrt{\lvert x_i\rvert}\right)^2}$
Skewness: $\frac{1}{N}\sum_{i=1}^{N} x_i^3$	Impulse indicator: $\frac{\max(\lvert x_i\rvert)}{\frac{1}{N}\sum_{i=1}^{N}\lvert x_i\rvert}$
Crest factor: $\frac{\max(\lvert x_i\rvert)}{\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}}$	Square root amplitude value: $\left(\frac{1}{N}\sum_{i=1}^{N}\sqrt{\lvert x_i\rvert}\right)^2$
Shape factor: $\frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}}{\frac{1}{N}\sum_{i=1}^{N}\lvert x_i\rvert}$	Variance: $\frac{1}{N}\sum_{i=1}^{N} x_i^2$
Absolute mean amplitude value: $\frac{1}{N}\sum_{i=1}^{N}\lvert x_i\rvert$	

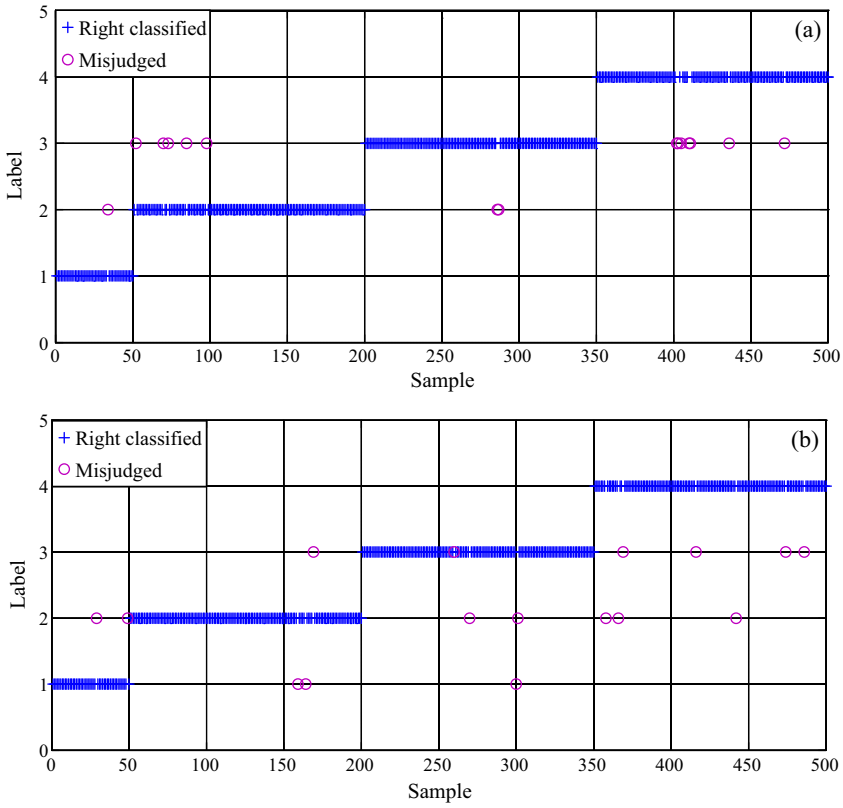

Fig. 12. SVRM diagnosis of fault patterns in (a) training samples and (b) testing samples.

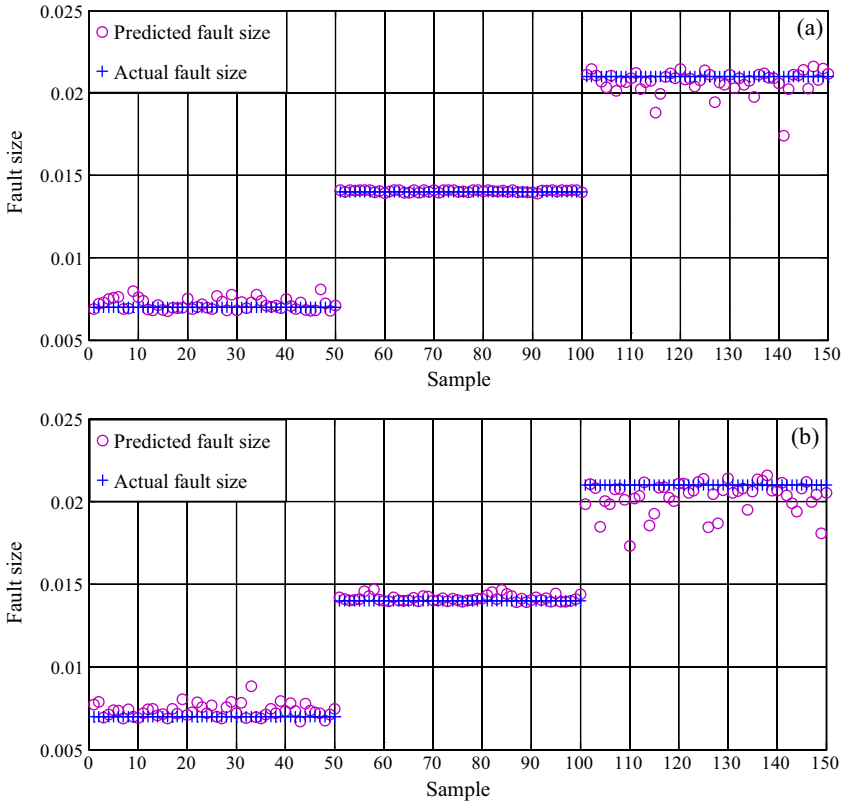

Fig. 13. SVRM predictions of IR-fault size for (a) training samples and (b) testing samples.

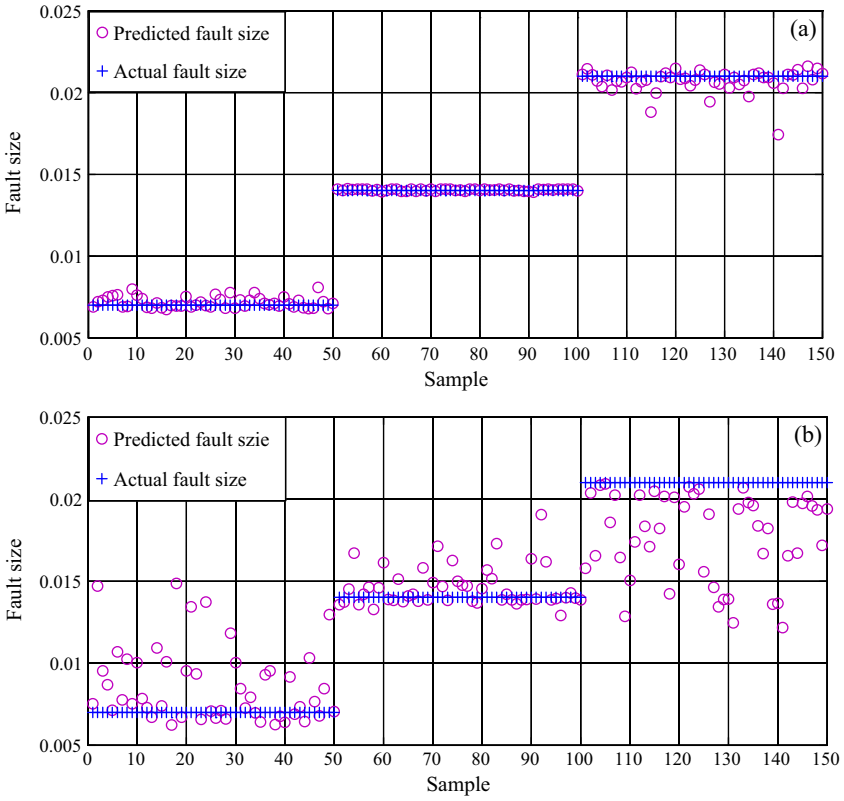

Fig. 14. SVRM predictions of B-fault size for (a) training samples and (b) testing samples.

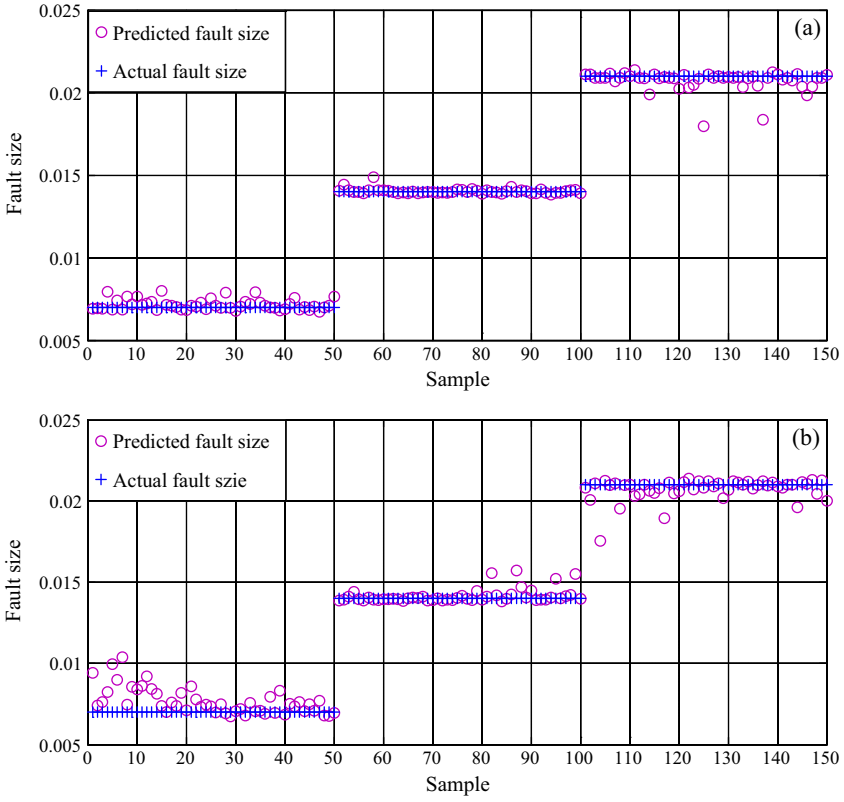

Fig. 15. SVRM predictions of OR-fault size for (a) training samples and (b) testing samples.

Table 5
Comparison of fault size evaluation performance of ADCNN and SVRM.

Fault pattern		Inner race fault		Ball fault		Outer race fault	
Method		Hierarchical ADCNNs	SVRM based	Hierarchical ADCNNs	SVRM based	Hierarchical ADCNNs	SVRM based
Maximum error	Testing result	0.0004	0.005	0.004	0.008	0.0002	0.002
	Training result	0.0002	0.005	0.004	0.006	0.0002	0.001
Classification accuracy	Testing result (10 fold cross validation)	100% (100%)	98.6%	99.3% (94.4%)	84.6%	100% (100%)	99.3%
	Training result (10 fold cross validation)	100% (100%)	100%	100% (100%)	99.3%	100% (100%)	100%

where ξ_i and ξ_i^* denote the stacked variable, C is a positive constant which penalises the errors larger than $\pm\varepsilon$ using ε – insensitive loss function as:

$$|\sigma|_\varepsilon = \begin{cases} 0, & \text{if } |\sigma| < \varepsilon \\ |\sigma| - \varepsilon, & \text{otherwise} \end{cases} \tag{14}$$

To process the non-linear problem, the popular radial basis function (RBF) is adopted to map the input vector into high dimensional feature space and its mathematical formula is given as:

$$K(x_i, x) = e^{\frac{-\|x_i - x\|^2}{2\sigma^2}} \tag{15}$$

Then the testing samples belong to class m if m satisfies the following decision-making function:

$$\arg\min_{m=1,2,\ldots,M} |m - (\sum_{i=1}^{n} (\alpha_i - \alpha_i^*) K(x_i, x) + b)| \tag{16}$$

where α_i and α_i^* are the Lagrange multipliers, M is the condition number. After a parameter grid search, C is 1, ε is 0.01, σ is 0.25. The model was trained by cross-validation method. The optimal SVRM model is then compared with the proposed model based on the same dataset, and the latter is found to be superior.

Fig. 12 presents the classification results of the SVRM method for fault-pattern recognition. The testing accuracy is 96.8%, lower than the equivalent result obtained using the ADCNN method. Comparison of Figs. 5 and 12, reveals that the fault-pattern recognition in the first layer of the proposed model is more accurate than that using the SVRM method.

Figs. 13–15 show the bearing fault size prediction results obtained using the SVRM method, which can be compared with the previously reported results of the proposed ADCNN method. The second layer of the proposed two-layer ADCNN model clearly performs better than the SVRM method.

To further analyze the evaluation performance of the two methods, a statistical indicator is used to quantify the accuracy of the second layer of the proposed system. The maximum error, which denotes the maximum deviation from actual fault size, is defined as follows:

$$\text{Maximum error} = \max(|f_{\text{actual}} - f_{\text{predicted}}|) \tag{17}$$

The formula above is used to calculate the maximum error achieved using the ADCNN and SVRM methods, and the results are listed in Table 5 for comparison. The performance of the proposed two-layer ADCNN model is approximately 10% better than that of the SVRM method.

Compared with methods of artificially choosing feature functions based on expertise, which are significantly limited, ADCNN offers a much more effective mean of automatically extracting features, especially when complex machinery and signals are involved. The results collectively confirm the superiority of the proposed hierarchical ADCNN model.

6. Conclusion

In this paper, a novel hierarchical ADCNN model is proposed, and its application to fault-pattern recognition and fault-size evaluation is evaluated.

First, a traditional DCNN model is improved by adding an adaptive learning rate and a momentum component to the process of weight updating, producing an ADCNN capable of extracting features automatically from vibration signals. Second, the improved method is hierarchically organized to give a two-layer ADCNN model: in the first layer, fault patterns are diagnosed, with fault-pattern indices as the output; in the second layer, fault size is evaluated with reference to three fault-size classes and concrete size. Third, samples from a bearing-fault dataset are collected for two experimental purposes. Training samples are used to build the model and testing samples are used to validate the model.

The results of experiments with bearing data demonstrate the superiority of the proposed ADCNN model to other fault-diagnosis methods, such as traditional DCNNs. The proposed model achieves a high degree of accuracy and offers an automatic feature extraction procedure which is practical and convenient for use in rotating machine fault diagnosis. It is worth noting that some ingenious shallow learning models are also competitive in terms of fast training speed and high accuracy based on well-designed features extraction. It is meaningful to learn from the merits of the shallow architectures to accelerate the training speed and make clearer feature explanation. The authors will investigate this topic in the future work.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 51505311), the Natural Science Foundation of Jiangsu Province (No. BK20150339), and the China Postdoctoral Science Foundation funded project (2015M580457, 2016T90490). The authors would like to thank Professor K.A. Loparo of Case Western Reserve University for his kind permission to use their bearing data. The authors also would like to appreciate two anonymous reviewers for their constructive comments and suggestions.

References

[1] X.M. Zhao, Q.H. Hu, Y.G. Lei, et al., Vibration-based fault diagnosis of slurry pump impellers using neighbourhood rough set models, Proceed. Inst. Mech. Eng. Part C J. Mech. Eng. Sci. 1989–1996 (vols 203–210) 224 (4) (2010) 995–1006.
[2] Y.G. Lei, H.E. Zheng-Jia, Advances in applications of hybrid intelligent fault diagnosis and prognosis technique, J. Vibrat. Shock 30 (9) (2011) 129–135.
[3] Q. He, Time–frequency manifold for nonlinear feature extraction in machinery fault diagnosis, Mech. Syst. Signal Process. 35 (1–2) (2013) 200–218.
[4] Z.K. Peng, F.L. Chu, P.W. Tse, Detection of the rubbing-caused impacts for rotor–stator fault diagnosis using reassigned scalogram, Mech. Syst. Signal Process. 19 (2) (2005) 391–409.
[5] R. Yan, R.X. Gao, X. Chen, Wavelets for fault diagnosis of rotary machines: a review with applications, Signal Process. 96 (5) (2014) 1–15.

[6] P.K. Kankar, S.C. Sharma, S.P. Harsha, Rolling element bearing fault diagnosis using wavelet transform, Neurocomputing 74 (10) (2011) 1638–1645.
[7] J. Huang, X. Hu, F. Yang, Support vector machine with genetic algorithm for machinery fault diagnosis of high voltage circuit breaker, Measurement 44 (6) (2011) 1018–1027.
[8] P. Konar, P. Chattopadhyay, Bearing fault detection of induction motor using wavelet and Support Vector Machines (SVMs), Appl. Soft Comput. 11 (6) (2011) 4203–4211.
[9] X. Zhang, J. Zhou, Multi-fault diagnosis for rolling element bearings based on ensemble empirical mode decomposition and optimized support vector machines, Mech. Syst. Signal Process. 41 (s1–2) (2013) 127–140.
[10] D. Wang, K.L. Tsui, Q. Zhou, Novel Gauss-Hermite integration based Bayesian inference on optimal wavelet parameters for bearing fault diagnosis, Mech. Syst. Signal Process. 72–73 (2016) 80–91.
[11] D. Wang, Q. Miao, R. Kang, Robust health evaluation of gearbox subject to tooth failure with wavelet decomposition, J. Sound Vib. 324 (3–5) (2009) 1141–1157.
[12] B. Li, P.L. Zhang, D.S. Liu, et al., Feature extraction for rolling element bearing fault diagnosis utilizing generalized S transform and two-dimensional non-negative matrix factorization, J. Sound Vib. 330 (10) (2011) 2388–2399.
[13] J. Yang, Y. Zhang, Y. Zhu, Intelligent fault diagnosis of rolling element bearing based on SVMs and fractal dimension, Mech. Syst. Signal Process. 8 (11) (2013) 2012–2024.
[14] X. Liu, L. Ma, J. Mathew, Machinery fault diagnosis based on fuzzy measure and fuzzy integral data fusion techniques, Mech. Syst. Signal Process. 23 (3) (2009) 690–700.
[15] V. Vakharia, V.K. Gupta, P.K. Kankar, Ball bearing fault diagnosis using supervised and unsupervised machine learning methods, Int. J. Acoust. Vibr. 20 (4) (2015) 244–250.
[16] V. Vakharia, V.K. Gupta, P.K. Kankar, A comparison of feature ranking techniques for fault diagnosis of ball bearing, Soft. Comput. (2015) 1–19.
[17] C.Q. Shen, D. Wang, F.R. Kong, P.W. Tse, Fault diagnosis of rotating machinery based on the statistical parameters of wavelet packet paving and a generic support vector regressive classifier, Measurement 46 (2013) 1551–1564.
[18] Liangpei, Huang, Chaowei, et al., Fault pattern recognition of rolling bearing based on wavelet packet decomposition and BP network, Sci. J. Inform. Eng. 67 (1) (2015) 7–13.
[19] F. Jia, Y. Lei, J. Lin, et al., Deep neural networks: a promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data, Mech. Syst. Signal Process. 72–73 (2015) 303–315.
[20] G.F. Bin, J.J. Gao, X.J. Li, et al., Early fault diagnosis of rotating machinery based on wavelet packets—empirical mode decomposition feature extraction and neural network, Mech. Syst. Signal Process. 27 (1) (2012) 696–711.
[21] M. Gan, C. Wang, C. Zhu, Construction of hierarchical diagnosis network based on deep learning and its application in the fault pattern recognition of rolling element bearings, Mech. Syst. Signal Process. 72–73 (2015) 92–104.
[22] V.T. Tran, F. Althobiani, A. Ball, An approach to fault diagnosis of reciprocating compressor valves using Teager-Kaiser energy operator and deep belief networks, Expert Syst. Appl. 41 (9) (2014) 4113–4122.
[23] Y.L. Lecun, L. Bottou, Y. Bengio, et al., Gradient-based learning applied to document recognition., Proc. IEEE 86 (11) (1998) 2278–2324.
[24] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Adv. Neural Inform. Process. Syst. 25 (2) (2012) 1106–1114.
[25] T.N. Sainath, B. Kingsbury, G. Saon, et al., Deep convolutional neural networks for large-scale speech tasks, Neural Networks 64 (2015) 39–48.
[26] P. Sermanet, Y. Lecun, Traffic sign recognition with multi-scale convolutional networks, in: Proceedings of International Joint Conference on Neural Networks (IJCNN'II), 2011, pp. 2809–2813.
[27] H. Lee, P.T. Pham, L. Yan, et al., Unsupervised feature learning for audio classification using convolutional deep belief networks, Adv. Neural Inform. Process. Syst. (2009) 1096–1104.
[28] T.N. Sainath, A.R. Mohamed, B. Kingsbury, et al., Deep Convolutional Neural Networks for LVCSR, ICASSP, Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on. IEEE, 2013, pp. 315–320.
[29] X. Chen, S. Xiang, C.L. Liu, et al., Vehicle detection in satellite images by hybrid deep convolutional neural networks, IEEE Geosci. Remote Sens. Lett. 11 (2014).
[30] Y. Lecun, U. Muller, J. Ben, et al., Off-road obstacle avoidance through end-to-end learning, Nips (2005) 739–746.
[31] L. Deng, O. Abdel-Hamid, D. Yu, A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion, Bove Davis' Div. Med. (Fourth Ed.) 46 (2004) 575.
[32] P. Sermanet, S. Chintala, Y. Lecun, Convolutional neural networks applied to house numbers digit classification, in: International Conference on Pattern Recognition, IEEE, 2012, pp. 3288–3291.
[33] M. Peemen, A.A.A. Setio, B. Mesman, et al., Memory-centric accelerator design for Convolutional Neural Networks, in: 2013 IEEE 31st International Conference on Computer Design (ICCD), 2013, pp. 13–19.
[34] C. Farabet, B. Martini, P. Akselrod, et al., Hardware accelerated convolutional neural networks for synthetic vision systems, in: Circuits and Systems (ISCAS). Proceedings of 2010 IEEE International Symposium on, IEEE, 2010, pp. 257–260.
[35] Y. Zhang, D. Zhao, J. Sun, et al., Adaptive convolutional neural network and it's application in face recognition, Neural Process. Lett. (2015) 1–11.
[36] S. Zhou, Q. Chen, X. Wang, Convolutional deep networks for visual data classification, Neural Process. Lett. 38 (1) (2013) 17–27.
[37] J. Bouvrie, Notes on convolutional neural networks, Neural Nets. (2006) 38–44. MIT CBCL Tech Report.
[38] Y. Bengio, Learning deep architectures for AI, Foundat. Trends® Mach. Learn. 2 (1) (2009) 1–127.
[39] C. Shen, D. Wang, F. Kong, et al., Recognition of rolling bearing fault patterns and sizes based on two-layer support vector regression machines, Smart Struct. Syst. 13 (3) (2014) 453–471.
[40] K.A. Loparo, Case Western Reserve University Bearing Data Center, 2012, <http://csegroups.case.edu/bearingdatacenter/home> (last visit, January 25, 2015).