Hindawi
Shock and Vibration
Volume 2020, Article ID 1949863, 13 pages
https://doi.org/10.1155/2020/1949863

Research Article

1DCNN Fault Diagnosis Based on Cubic Spline Interpolation Pooling

Shuzhan Huang,[1] **Jian Tang**,[2] **Juying Dai**,[2] **Yangyang Wang**,[1] **and Junjie Dong**[1]

[1]*School of Graduate, Army Engineering University of PLA, Nanjing 210000, China*
[2]*School of Field Engineering, Army Engineering University of PLA, Nanjing 210000, China*

Correspondence should be addressed to Jian Tang; lgdx_tj@163.com

Received 12 October 2019; Accepted 28 December 2019; Published 28 February 2020

Academic Editor: Xavier Chiementin

The conventional pooling method for processing one-dimensional vibration signals may lead to certain issues, such as weakening and loss of feature information. The present study proposes the cubic spline interpolation pooling method. The method is appropriate for processing one-dimensional signals. The proposed method can transform the pooling problem into a linear fitting problem, use the cubic spline interpolation method with outstanding fitting effects, and calculate the fitting function of the input signals. Moreover, the values of the interpolation points are sequentially taken as the feature value output. Furthermore, the network using the conventional pooling method and the pooling network model proposed in the present study are compared, tested, and analyzed on the constructed simulation signals and the measured bearing dataset. It is concluded that the proposed pooling method can reduce the data dimension while improving the network feature extraction capability and is more appropriate for pooling one-dimensional signals.

1. Introduction

With the increasing development of the deep learning in the last few decades, it has extensively attracted many communities so that remarkable achievements have been achieved in many fields [1–3]. Convolutional neural network (CNN) is an important branch of the deep learning. Because of reasonable features of the CNN in the adaptive learning ability and feature abstraction ability, it is widely used in numerous applications such as speech and image recognition [4–6]. For example, Krizhevsky and Sutskever [7] used the deep-expanded CNN to achieve the best online classification effect in the ImageNet large-scale visual recognition challenge (LSVRC). Many scholars have applied the deep-expanded CNN to the fault diagnosis and achieved reasonable results. For example, Wen et al. [8] applied the bearing dataset of the Case Western Reserve University (CWRU) on images as inputs to diagnose the bearing fault through the CNN, where the diagnostic accuracy higher than 95% is obtained. Ince et al. [9] used the 1DCNN for real-time condition monitoring and fault diagnosis of TV sets. Moreover, Peng and Liu [10] applied the 1DCNN to diagnose the HSTs' wheel-to-bearing vibration signals and achieved reasonable results.

Reviewing the literature indicates that the pooling is an important part of the CNN architecture [11]. Its core idea originates from the pioneering research of Hubel and Wiesel [12] on the structure of mammalian visual cortex models and the principle of the local correlation of signals proposed by Koenderink and Doorn [13]. The pooling principle is to combine outputs of several near feature detectors into a local or global "feature package" to remove irrelevant details while retaining the task-related information. Furthermore, the pooling is normally used to achieve the signal conversion in variance, a more compact representation, and the improved robustness against the noise and clutter [14]. Studies show that the data-processing load can be reduced so that useful information can be retained through pooling operations. In fact, pooling operations are of significant importance in the CNN structure and can have great impacts on the network performance. Therefore, it is necessary to investigate the pooling method comprehensively. Fukushima and Miyake [15] applied pooling operations in the signal pattern recognition. Lecun et al. [16, 17] successfully trained the error-based CNN through pooling. Moreover, Jarrett et al. [18] used the max pooling to achieve the best results on the Caltech-101 and MNIST datasets. Recently, Matthew [19, 20] used a simple and effective stochastic pooling method to prevent the overfitting during the CNN training and achieved reasonable results. Moreover, Yu et al. [21] combined the conventional max pooling and mean pooling methods and proposed a hybrid pooling method to replace deterministic pooling operations with the stochastic process. Most recently, Hai and Xiao [22] proposed a probability weighting pool and compared it with the max pooling, mean pooling, and stochastic pooling. They showed that the proposed method improves the recognition rate. Lee et al. [23] added the intermap pooling (IMP) layer in the CNN and proposed the IMP-CNN scheme. Then, they verified the performance of the proposed IMP-CNN structure.

The abovementioned pooling methods mostly focus on two-dimensional inputs and solve problems such as the image recognition. The input is a two-dimensional matrix, such as pixel values of the image. However, in the field of fault diagnosing, most of the measured results through sensors, including the vibration and pressure signals, are one-dimensional signals [24–26]. Hence, one-dimensional convolutional neural networks (1DCNNs) are often used for these problems. Unlike the image recognition, each sample point of vibrational signals indicates the magnitude of the amplitude, while the order of the points indicates the time sequence. It should be indicated that the time sequence may contain periodic or short-term pulse characteristics of the signal in the segment. These features are often important features of the state signal, while they may not exist in two-dimensional images. Therefore, when the one-dimensional state signal is pooled, the commonly used pooling algorithms in the image recognition are applied, which may lead to weakening or even losing certain important features of the signal information. Therefore, the present article proposes a pooling algorithm for the feature extraction of one-dimensional state signals.

The cubic spline interpolation pooling is compared with the max pooling and mean pooling commonly used in the CNN to verify the effectiveness of cubic spline interpolation pooling.

The main contents and research ideas of the present study are organized as follows: The network structure of the 1DCNN is introduced in Section 2. Then, three commonly used pooling methods and the cubic spline interpolation pooling method are introduced in Section 3. Moreover, typical simulation signals are constructed and problems dealing with one-dimensional vibration signals are analyzed in Section 4. Then, the feasibility of the proposed pooling algorithm is verified through the actual mechanical fault signals. Finally, main achievements and conclusions are presented in Section 5.

2. Introduction of 1DCNN

2.1. Network Structure. Figure 1 shows the constructed 1DCNN structure. It indicates that the time-domain status signal is input from the input layer, while the layer-by-layer feature extraction and sparse processing are performed through several convolutional layers and pooling layers. Then, signals are classified through the fully connected layer and the output layer by the Softmax classifier.

Various layers of the CNN are introduced below.

The convolutional layer uses a one-dimensional convolution kernel to perform the convolution calculation for the local region of the input signal to produce the corresponding one-dimensional feature map, and different convolution kernels extract different features in the input signals [27]. Each convolution kernel detects specific features at all locations on input feature maps to achieve the weight sharing on the same input feature map. Characteristics of the local connection and the weight sharing can effectively reduce the network complexity and subsequently reduce the number of training parameters.

If the l^{th} layer is a convolutional layer, the one-dimensional convolutional layer can be expressed as

$$x_j^i = f\left(\sum_{i=1}^{M} x_i^{l-1} * k_{ij}^l + b_j^l\right), \tag{1}$$

where k represents the convolution kernels, j denotes the number of kernels, M represents the channel number of the input x^{l-1}, b is the bias corresponding to the kernel, f is the activation function, and $*$ is the convolution operator.

After passing through convolutional layers, the number of feature maps increases, resulting in an expansion of the data dimension, which is not conducive to the calculation. Therefore, each feature map should be processed at the pooling layer. It should be indicated that the 2DCNN usually employs the mean pooling or the max pooling. The mean pooling calculates the mean value of the parameter within the range in accordance with the predetermined pooling window size, while the max pooling selects the largest parameter within the predetermined window range as the output value.

All the neuron nodes of the fully connected layer are connected to all neuron nodes in the feature map output by the previous layer. It should be indicated that if the last pooling layer is the $(l+1)^{\text{th}}$ layer, and it is passed as an input to the fully connected layer, then its output is described as follows:

$$h(x) = f\left(w^{l+1} \cdot x^{l+1} + b^{l+1}\right), \tag{2}$$

where w denotes the weight and b denotes the bias.

2.2. Network Training. During the network training process, the network initializes the parameters such as weights and thresholds initially. The input data are propagated through the convolutional layers, the pooling layers, and the fully connected layer to obtain output values. Then, the errors between the output values of the network and the expected

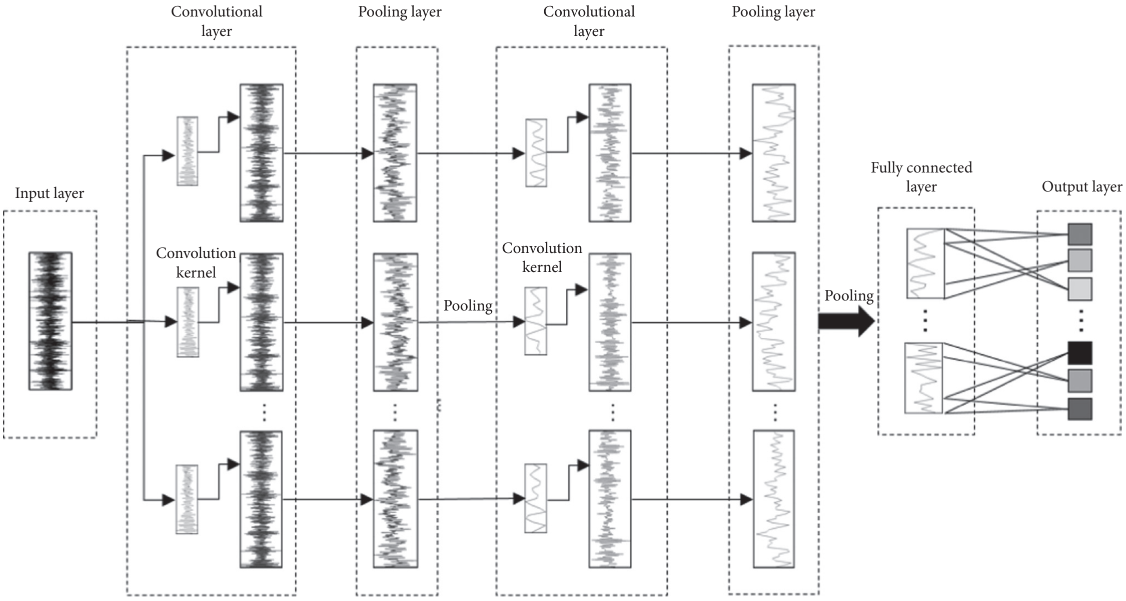


FIGURE 1: Structure of the 1DCNN.

values are obtained. The errors are reversed back to the network, and the errors of the fully connected layer, the pooled layer, and the convolutional layer are obtained successively. Moreover, the error gradient is calculated and the weight and threshold are updated until the tolerance condition of the error is satisfied to complete the training. Figure 2 shows the abovementioned process.

3. Pooling Methods in CNN

After the input signal is convoluted, it requires performing secondary feature extraction and data dimension reduction through the pooling layer. When calculating the feature value of a certain part of the input signal, it is necessary to analyze and count the features of the signal and to use the new feature to represent the total features of the signal. This segment signal is called the pooling domain, and the process is called pooling. The use of pooling operations can improve the expressive ability of features and reduce the data dimension, which effectively avoids the overfitting phenomenon caused by excessive parameters and complicated structure in the network training. Conventional pooling methods include mean pooling, max pooling, and stochastic pooling.

3.1. Conventional Pooling Methods

3.1.1. The Mean Pooling. Mean pooling means that all values in the pooling domain are summed and the arithmetic mean is used as the postpooling eigenvalue during the pooling process. In the 1DCNN pooling calculation, X, b_2, S, and a indicate the input signal, the offset, the obtained pooling feature vector, and the pooling process moving step, respectively. Moreover, the subsampling pooling domain is a $1 \times a$ vector. Then, the algorithm expression is described as follows:

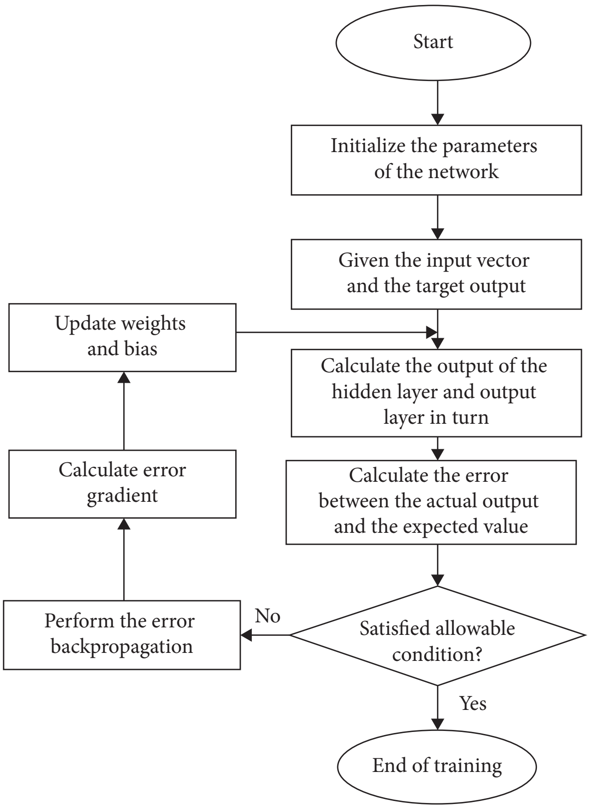

Start
Initialize the parameters of the network
Given the input vector and the target output
Update weights and bias
Calculate the output of the hidden layer and output layer in turn
Calculate error gradient
Calculate the error between the actual output and the expected value
Perform the error backpropagation
No
Satisfied allowable condition?
Yes
End of training

FIGURE 2: Process of network training.

$$S_{1i} = \frac{1}{a} \sum_{i=1}^{a} X_{1i} + b_2. \tag{3}$$

The input signal pooling process with a length of 1×12 and a pooling step of 4 is taken as an example. Figure 3 shows the pooling calculation process.

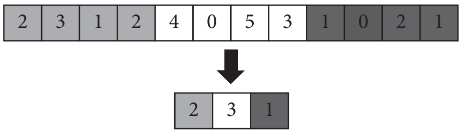


Figure 3: Process of mean pooling.

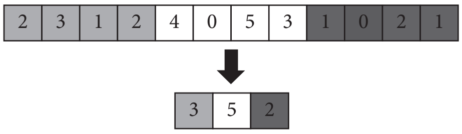


Figure 4: Process of max pooling.

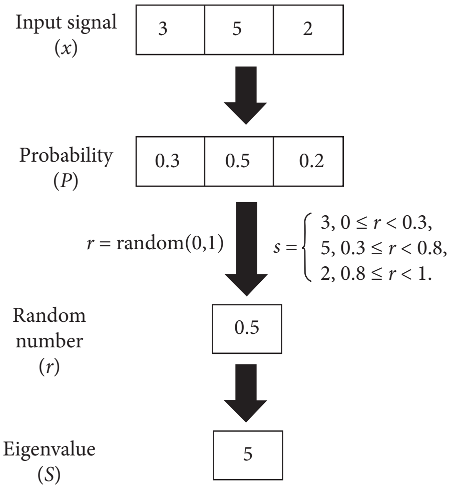


Figure 5: Process of stochastic pooling.

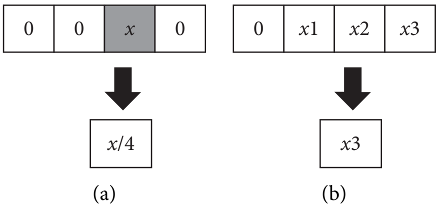


Figure 6: Two typical cases of pooling.

3.1.2. The Max Pooling. Max pooling uses the maximum value in the pooling domain as the pooling eigenvalue, and its algorithm is described as follows:

$$S_{1i} = \max_{i=1}^{a}\left(x_{1i}\right) + b_2, \tag{4}$$

where x, b_2, and S denote the input signal, the offset, and the pooling eigenvector, respectively. Moreover, the pooling domain is a $1 \times a$ vector. The pooling process has a moving step of a. It should be indicated that the input signal pooling process with a length of 1×12 and a pooling step of 4 is taken as an example. Figure 4 illustrate the pooling calculation process.

3.1.3. The Stochastic Pooling. The stochastic pooling method was proposed by Zeiler and has been widely used in recent years. Different from max pooling and mean pooling, the eigenvectors obtained by the stochastic pooling of the same input signal are not necessarily identical. This method can reduce the overfitting phenomenon that occurs during the training. In the stochastic pooling process, the output value by pooling is selected according to the probability distribution of the eigenvalues of the pooling domain. Moreover, the probability of being selected with a large eigenvalue is also large. The pooling steps are as follows: If the input signal is set as x, the probability value P is calculated for each feature selected, initially. It should be indicated that the probability value of each feature is distributed between 0 and 1. Then, a random number r is taken between 0 and 1. Moreover, r is in the probability interval of a certain eigenvalue. Then, this feature is selected as the pooling eigenvalue S. Figure 5 shows an example of this process.

During the pooling process of the input signals, it is assumed that the pooling domains of the two different cases shown in Figure 6 appear. The shaded part in Figure 6(a) represents that the input value is x, and the remaining input values are 0. It is observed that the feature information in Figure 6(a) is concentrated in the shadow x. If the mean pooling algorithm is used at this time, the feature value of $S = x/4$ is resulted. This leads to weakening of the key feature information significantly, which is not beneficial for the feature extraction and classification. However, in this case, the max pooling algorithm can help better extract the feature information x of the pooling domain and achieve the reduction of the feature dimension without losing the feature information. In Figure 6(b), x_1, x_2, and x_3 $(x_1 < x_2 < x_3)$ represent different input values. If the max pooling algorithm is used in this case, the output feature value is $S = x_3$, which does not consider the feature information contained in x_1 and x_2, ignores the possible connections among x_1, x_2, and x_3, including the time continuity in the one-dimensional time-series signals, and results in the loss of large quantities of useful feature information. However, in this case, the mean pooling algorithm can reduce the loss of useful information.

It should be indicated that, during the feature extraction of one-dimensional time-series signals, conventional pooling algorithms are not applicable to all cases. The max pooling algorithm ignores the correlation between input feature values in the pooling domain, while the mean pooling algorithm weakens key feature information in the case that the feature information is concentrated. The stochastic pooling algorithm considers the correlation between the input feature values. However, during the pooling process, the feature extraction does not contain all the information and the results of multiple experiments may be different. Therefore, the repeatability of the experiments is poor.

3.2. The Cubic Spline Interpolation Pooling. In the actual fault diagnosis, the machine, which should be tested, may have periodic fault signals or short-term excitation signals due to a transient shock. Therefore, the key information in the vibration signals may be periodic or instantaneous. However, the conventional pooling algorithms need to be selected according to the features of the input signals, which can obviously reduce the efficiency and versatility of the 1DCNN for fault diagnosis. Therefore, considering the limitations of the abovementioned conventional pooling algorithms, an improved pooling algorithm is proposed in the present study named the cubic spline interpolation pooling, which makes the feature information fully extracted and mined while considering the time continuity of input signals.

Cubic spline interpolation is a mathematical process of solving three-moment equations to obtain the curve function group by using a smooth curve through a series of form points [28, 29]. However, the Newton interpolation algorithm has a large amount of calculation. Moreover, when the number of interpolation polynomials is higher, the error rate increases. It should be indicated that the piecewise linear interpolation has uniform convergence, while the smoothness is poor. In comparison with the higher order spline, the cubic interpolation spline requires less computation and storage and is more stable. Moreover, compared with the quadratic interpolation spline, the cubic interpolation spline is more flexible when simulating arbitrary shapes. The cubic interpolation spline curve provides a reasonable compromise between flexibility and computational speed. Therefore, the cubic spline interpolation method is used instead of applying the original pooling method. The mathematical definition is as follows.

Let a function $y = f(x)$ have $n + 1$ equally spaced sampling points on the interval [a, b], i.e., $a = x_0 < x_1 < \cdots < x_n = b$. The value of the function at the sampling point is $f(x_i) = y_i$ $(i = 0,1, \ldots, n)$. If there is a segmentation function $z(x)$ that meets the following conditions, then $z(x)$ is called the cubic spline interpolation function:

$z(x)$ is no more than a cubic polynomial on each subinterval $[x_i, x_{i+1}]$

$z(x)$ has a continuous second derivative over the entire interval $[a, b]$

$z(x_i) = y_i$ $(i = 0,1, \ldots, n)$

During the pooling process for one-dimensional time-series signals, the improved pooling algorithm replaces the conventional pooling methods with the cubic spline interpolation method and performs quadratic feature extraction and dimension reduction on the feature signal output after the convolutional layer feature extraction.

The network pooling process shown in Figure 1 is taken as an example, and the cubic spline interpolation pooling method proposed in the present study is described as follows:

Step 1: the eigenvector of size $1 \times N$ output from the convolutional layer is divided into subvectors whose segments do not overlap each other. It should be indicated that each segment vector is an independent pooling domain and the size of each subvector is $1 \times n$.

Step 2: the function is fitted by using each point in the subvector as a series of known form points. The function fitted by the abovementioned method, which is referred to as the eigenfunction hereinafter, retains the feature information to the greatest extent and ensures the time continuity of input features.

Step 3: the value of each interpolation point is calculated according to the specified step size based on the eigenfunction obtained by the fitting process. Then, the value is used as the output eigenvalue after the cubic spline interpolation pooling. The eigenvalues obtained in each segment are connected, and the obtained vector is the eigenvector output after pooling.

The pooling method enables the pooling layer to perform feature secondary extraction on the input feature signals. It retains useful information to the greatest extent, realizes data dimension reduction, and improves the network computing efficiency. Figure 7 shows the process of the pooling algorithm.

4. Performance Evaluation

In this section, several pooling methods are utilized to verify the validity of the cubic spline interpolation method and consider the definition conditions of the cubic spline interpolation. Moreover, the corresponding 1DCNN is constructed to diagnose different fault signals and compare the diagnostic results. The network consists of one input layer, two convolutional layers, two pooling layers, one fully connected layer, and one output layer. It should be indicated that the experiment is performed in the Matlab2014a environment. The operating system is Windows 10, and the CPU is Intel i7-6700HQ with the memory of 8GB.

4.1. Simulation Signal Verification. In this section, two typical simulation signals in signal processing are developed, including the sinusoidal signals and periodic pulse signals, to verify the effects of the proposed method. The expression of the sinusoidal signals is as follows:

$$Y = \sin(500\,\pi t + 10). \tag{5}$$

The two typical simulation signals are pooled by mean pooling, max pooling, and cubic spline interpolation pooling methods of different pooling steps. Figure 8 shows the time-domain waveform of the periodic pulse signals when the pooling step is taken as 2 and 8, respectively. Moreover, it illustrates the results after pooling.

It is observed, that for the pooling step size of 2, the pooling domain is small. Therefore, three types of methods completely preserve the pulse components contained in the original signals. However, as the pooling step size increases and reaches 8, the data dimension is reduced, while the feature sparseness constantly increases. Meanwhile, the feature information in the signals has a certain loss. It should be indicated that most feature information is lost by mean pooling, while the most feature information is retained by the cubic spline interpolation pooling. Moreover, due to the

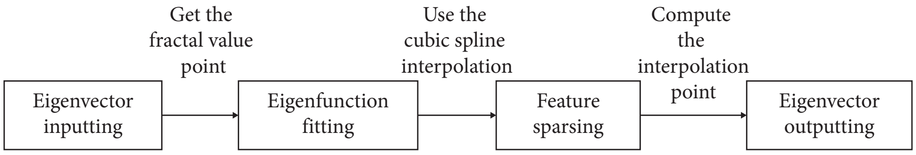
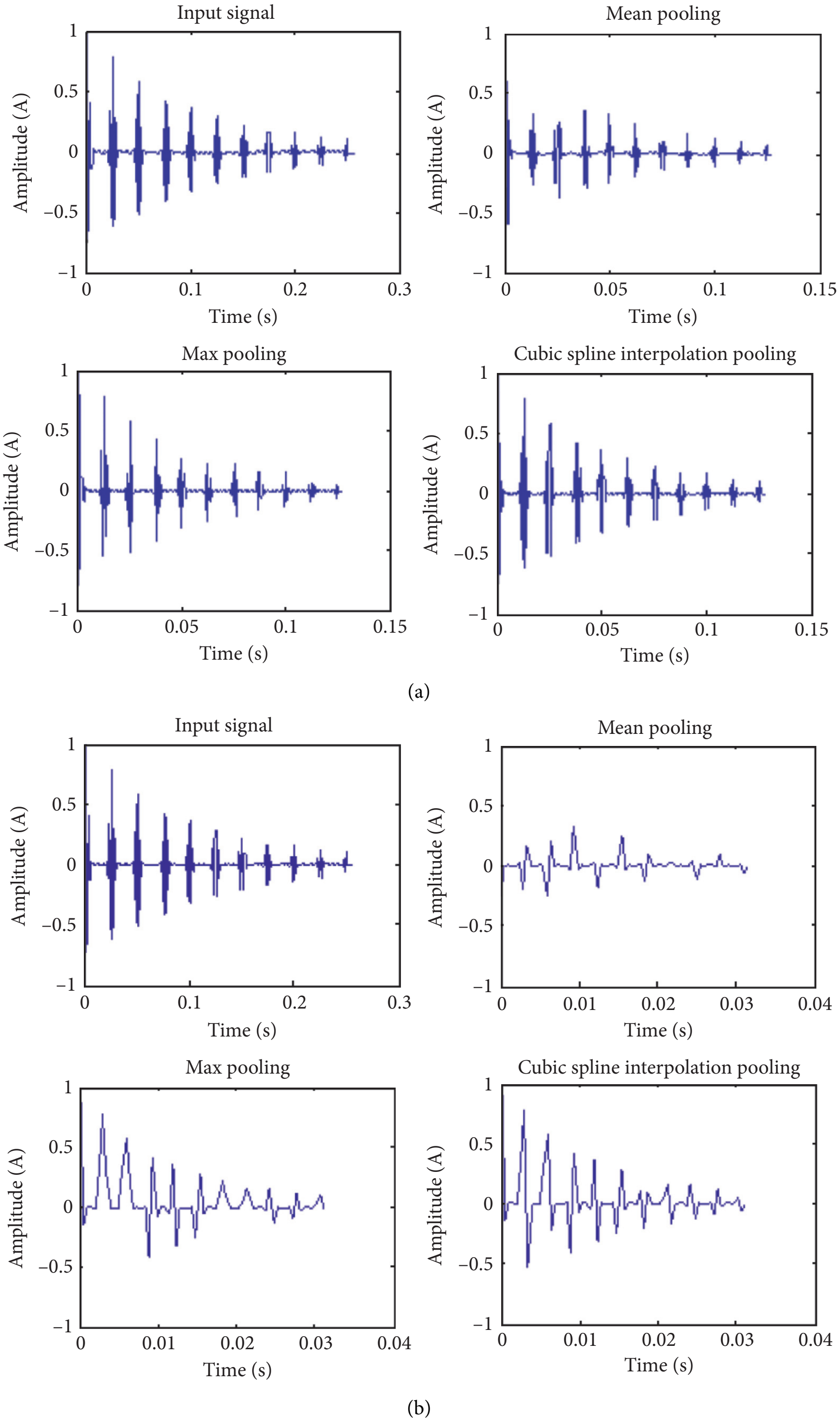

Eigenvector inputting
Get the fractal value point
Eigenfunction fitting
Use the cubic spline interpolation
Feature sparsing
Compute the interpolation point
Eigenvector outputting

Figure 7: Process of the cubic spline interpolation pooling.



(a)

(b)

Figure 8: Time-domain waveform of the periodic pulse signals when the step size is (a) 2 and (b) 8.

algorithm characteristics of mean pooling, it compresses the signal amplitude to some extent after the input signal is pooled, which also leads to a certain degree of loss of feature information. Therefore, the mean pooling method is not applicable to signals with large amplitude changes or more transient impact components in the input.

Figure 9 shows the time-domain waveform of the sinusoidal signals when the pooling step is taken as 2 and 8, respectively. Moreover, it shows the results after pooling.

Figure 9 shows that when the pooling step size is 2, the three types of methods preserve the periodic features in the signals more reasonably, while mean pooling can also result in a certain compression of the amplitude. It should be indicated that when the pooling step gradually increases to be equal to one periodic component of the sinusoidal signal or an integer multiple of the period, the eigenvalues extracted by each pooling domain of mean pooling and max pooling are the same, and the output eigenvector is a straight line at last. Figure 9(b) shows that when the pooling step size is 8, the size of the pooling domain is exactly equal to one periodic component of the sinusoidal signals. Taking the max pooling as an example, the eigenvalues obtained in each pooling domain are only the largest amplitude in a sinusoidal cycle and the result after the pooling is a straight line. This kind of method results in the fact that the pooling results only contain the partial amplitude of the original signals and lose the periodic features in the original signals. In the actual diagnosis process, this is obviously not conducive to further extracting features since it reduces the diagnostic accuracy. For the cubic spline interpolation pooling, due to the characteristics of its algorithm, curve fitting is performed in each pooling domain and the corresponding interpolation points are calculated in the fitted curve, which are used as pooling eigenvalue outputs so as not to result in a lot of useful information to be lost.

4.2. Experiment and Comparison of Results. The experimental data in this section are derived from the Case West Bearing University Rolling Bearing Data Center. The CWRU dataset is the world's recognized dataset for bearing fault diagnostic criteria. The object of this test is the drive end bearing. The bearing type is deep groove ball bearing SKF6205. Moreover, the bearing is single-point damaged by EDM, and the sampling frequency is 12 kHz. The rolling bearing has four states, including the normal state, rolling element damage, inner ring damage, and outer ring damage. It should be indicated that the damage diameter is 0.007 inch. In the test, every 1024 data points are used as one input signal, and no overlap sampling is used. Then, the input dataset contains a total of 500 signal samples, of which 400 are training samples and the remaining 100 are test samples. Figure 10 shows the time-domain waveforms of the four types of bearing signals.

The aforementioned datasets are trained by the 1DCNN with different pooling methods. Prior to the training, the network-related parameters need to be set. The convolution kernel size of the two-layer convolutional layers is 1 × 301, and the number of convolution kernels is 4. The learning rate α is 0.01, and the batch size is 10. In other words, the network updates the weight when training every 10 samples. Moreover, the maximum number of training is 1000 times.

Table 1 presents the classification accuracy of the CWRU bearing dataset of each network model after a number of experiments. It is observed that, for a bearing in the normal operation, application of three pooling methods results in the recognition rate higher than 98%. In the case of the bearing rolling element fault state, the recognition rates of the network with both mean pooling and max pooling decrease, and the recognition rate of mean pooling is the lowest. When identifying and diagnosing the faults of the inner and outer rings of the bearing, both the max pooling method and the cubic spline pooling method have reasonable recognition rates. However, the recognition rate of the mean pooling method is low, which is difficult to meet the requirements of the actual diagnostic accuracy.

For the feature extraction and classification of one-dimensional vibration signals, the cubic spline interpolation method preserves the feature information in the input signals. While realizing data dimension reduction, it ensures that useful information will not be lost. From the perspective of the overall recognition accuracy of the bearing dataset, the network using the cubic spline interpolation method has 96% recognition accuracy, which is higher than that of the other two methods. This indicates that this pooling method is superior to max pooling and mean pooling methods commonly used in the CNN for feature extraction and classification of one-dimensional vibration signals. Moreover, it is more suitable for the processing of the one-dimensional input.

4.3. Analysis of Test Results. In this section, the eigenvector visualization of the first layer of pooling layers is extracted to further study the working principles of pooling methods and compare the advantages and disadvantages of several types of pooling methods and perform qualitative analysis. Then, the correlation coefficients between the eigenvector and the original signals are calculated for quantitative analysis to determine which pooling method can retain the feature information in the original input signals to the greatest extent.

4.3.1. Qualitative Analysis. In this section, the eigenvectors of the first layer of the pooling layers in the CNN constructed by the three types of pooling methods are extracted. Figure 11 shows the visualization results.

Figure 11 shows that the mean pooling method needs to be averaged in the local area in the pooling process of the four types of bearing state signals. Therefore, the amplitude range of the original input signals is compressed to some extent. In the processing of the one-dimensional signals, the essence is to obtain the trend line of the signal by taking the mean value. Although this pooling method can extract the global and periodic characteristics of the input signal, some of the pulsed local features that exist in the vibration signals are ignored. Therefore, the signal recognition with large amplitude changes is not effective.

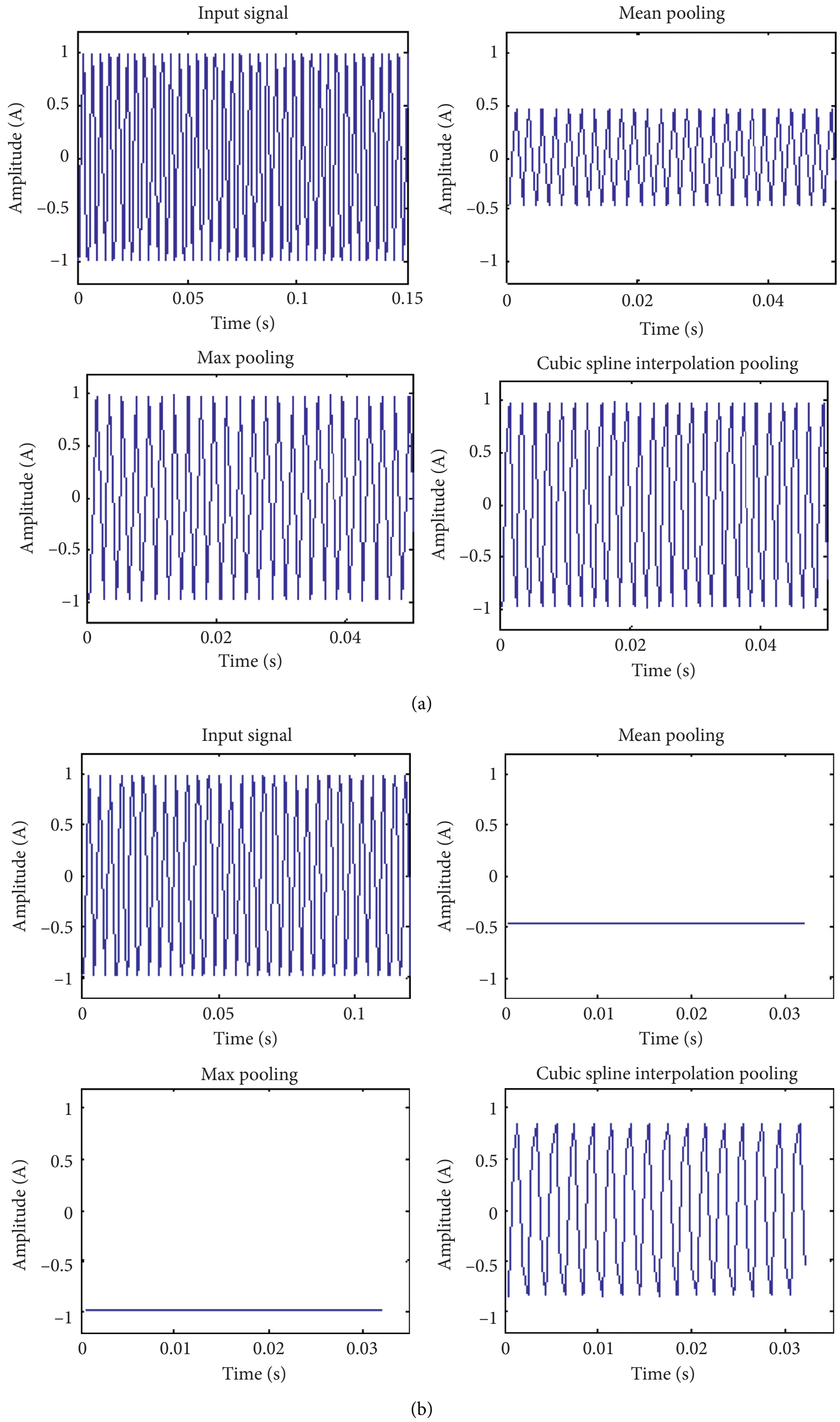


Figure 9: Time-domain waveform of the sinusoidal signals when the step size is (a) 2 and (b) 8.

In contrast, the max pooling method overcomes the abovementioned issue more reasonably. The principle of the max pooling method is to retain the local maximum as the pooling feature output. Moreover, this method is equivalent to taking the envelope of the original signals in a one-dimensional network. The max pooling can extract

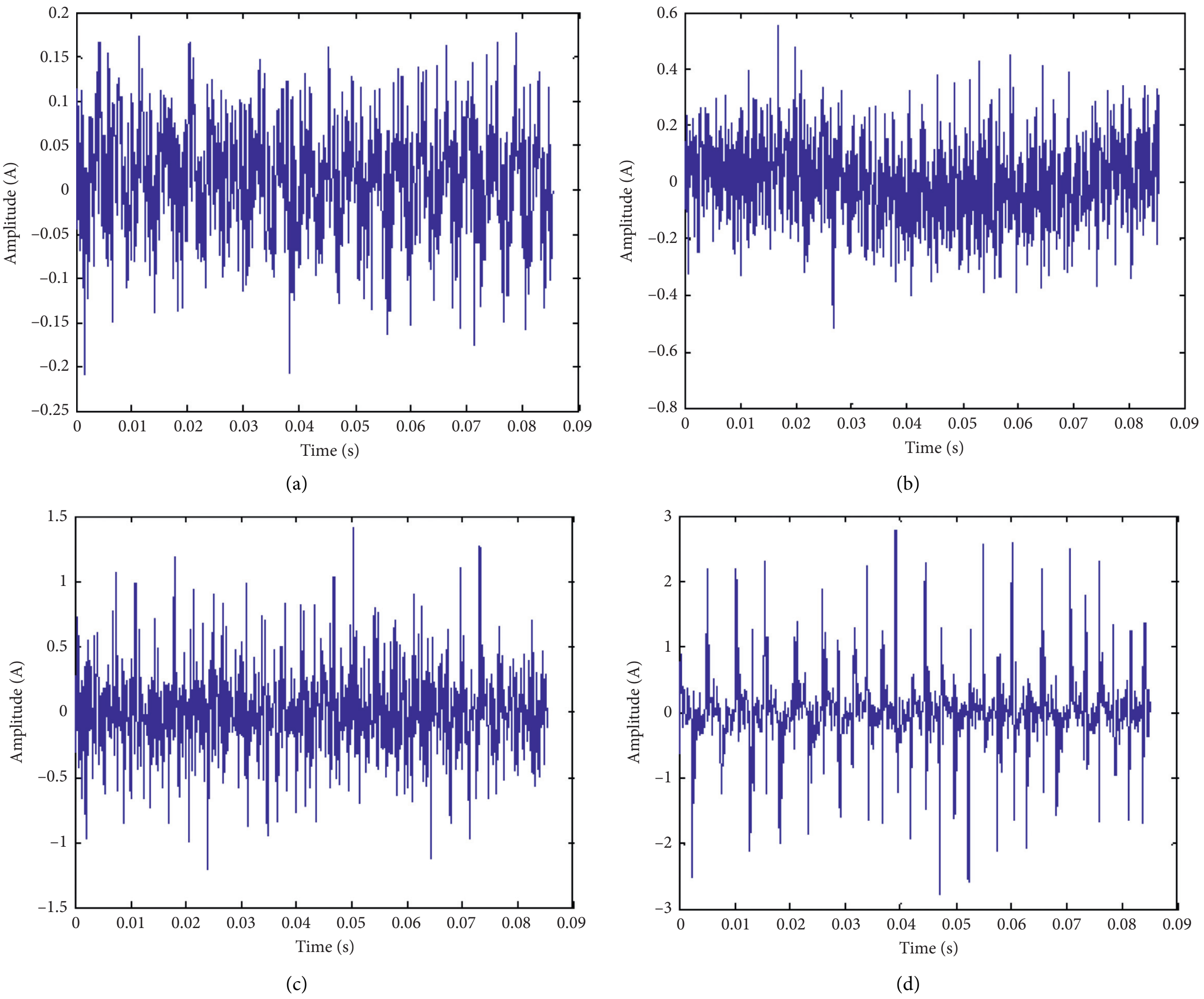


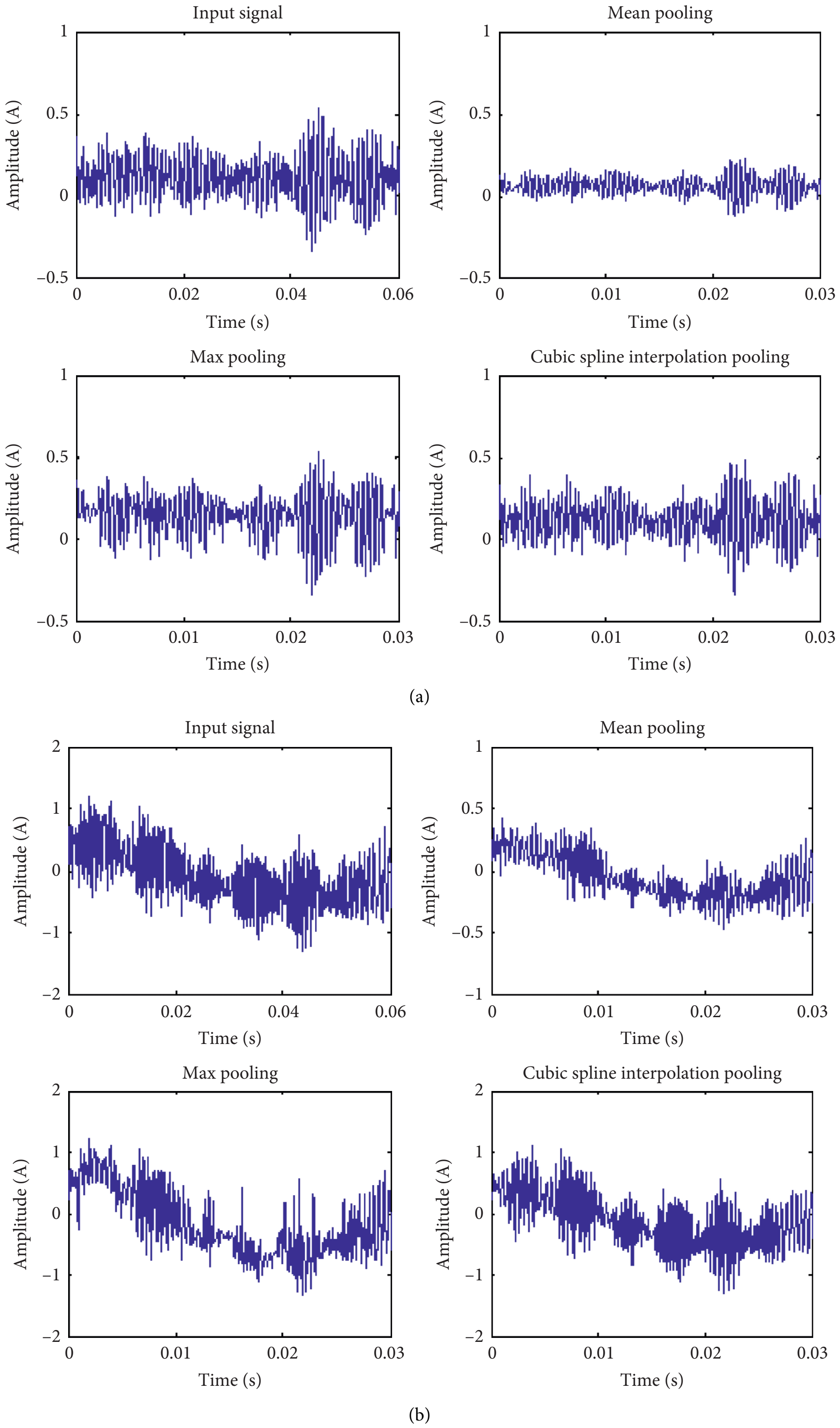
Figure 10: Time-domain waveforms of the four types of bearing signals. (a) Normal state. (b) Rolling element damage. (c) Inner ring damage. (d) Outer ring damage.

Table 1: Results of the experiment.

	Number of bearing status signals		Number of misidentified bearing states		
Bearing condition	Train sample	Test sample	Max pooling	Mean pooling	Cubic spline interpolation pooling
Normal state	400	100	8	4	5
Rolling element damage	400	100	12	16	3
Inner ring damage	400	100	3	20	2
Outer ring damage	400	100	1	28	6
Accuracy of the testing sample (%)			94.0	83.0	96.0

the typical features contained in the signals, which is helpful for fault diagnosis of the bearing status. However, Figure 11(b) shows that, during the process of diagnosing the rolling element fault state signals, the extracted feature signals are affected in continuity and smoothness, which may result in a decrease of the recognition rate. This is because this method only extracts the maximum value and does not consider the correlation before and after the signal time.

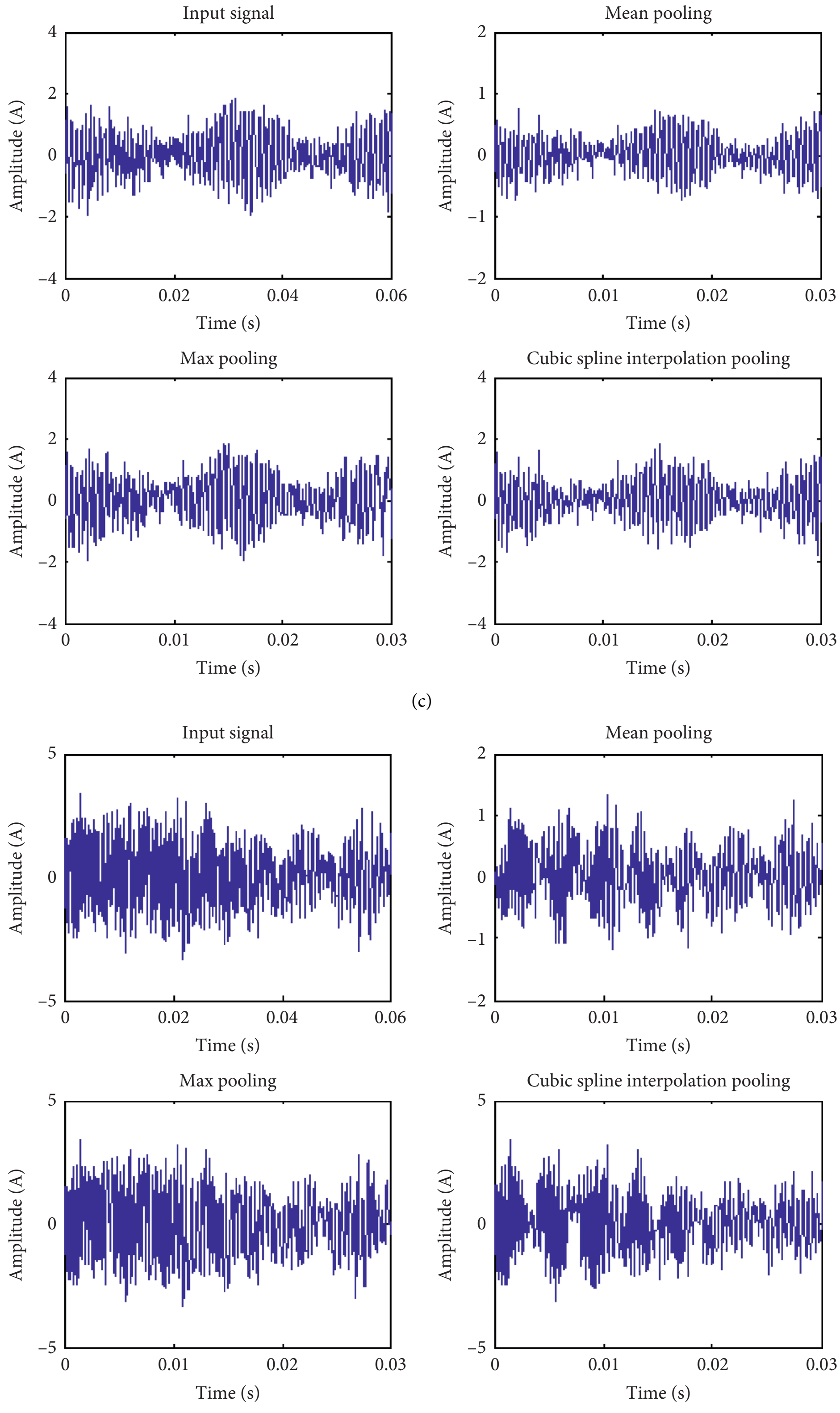
The cubic spline interpolation pooling method proposed in the present study is excellent to avoid the abovementioned problems. The cubic spline interpolation function has the characteristics of simple calculation, good stability, and good smoothness, while the fitting function fully extracts the original input signal feature information. Therefore, it can ensure the continuity of the whole fitting curve, which is more conducive to the following deep feature extraction of the convolutional layers and the classification of the fully connected layer.



(a)

(b)

Figure 11: Continued.



(c)

(d)

Figure 11: Pooling results of bearing status signals. (a) Normal state. (b) Rolling element damage. (c) Inner ring damage. (d) Outer ring damage.

Table 2: Correlation coefficients between the proposed method and the input signals.

Bearing states	Max pooling	Mean pooling	Cubic spline interpolation pooling
Normal state	0.3574	0.2824	0.3617
Rolling element damage	0.2513	0.2671	0.3336
Inner ring damage	0.3744	0.2463	0.3789
Outer ring damage	0.3912	0.1746	0.3558

4.3.2. Quantitative Analysis. In order to compare the pooling effects of the conventional pooling methods with that of the proposed cubic spline interpolation pooling method, this section utilizes the eigenvectors extracted in the previous section and analyzes the correlation between the original input signals. Table 2 shows the correlation coefficients between the proposed method and the input signals of different bearing states.

Table 2 shows that both the max pooling method and the cubic spline interpolation pooling method have a higher correlation with the original input signals, while the correlation between the eigenvectors obtained by the mean pooling method and the original input signals is lower. The obtained results correspond to the diagnostic recognition accuracy of the respective networks. After mean pooling, although this method roughly expresses the trend of the input signals of this segment and retains the periodic features, the details and local features are weakened or even lost. Therefore, the degree of similarity between feature vectors obtained after pooling and the original signals is not high, which results in the loss of useful information. This affects subsequent feature extraction and final classification effects. In contrast, when the max pooling method is used to process most of the one-dimensional vibration signals, the pooled feature vector is highly correlated with the original signals, and more useful information can be retained. However, due to the pooling method characteristics, which are regardless of the temporal relationship between the vibration signals, the maximum value is taken as the eigenvalue in the respective specified pooling domain. Finally, the eigenvectors obtained by arranging these eigenvalues in series are affected by their continuity. For example, Figure 11(b) shows that the max pooling result in the rolling element fault state is not ideal and the correlation coefficient with the original signals is also low. This indicates that the max pooling method is not affected by the fluctuation degree of the signals. It should be indicated that if the fluctuation degree of a certain segment signal is large, the use of the max pooling results in poor continuity of the obtained feature signals. Moreover, the original signals are characterized properly so that the overall recognition accuracy of the network is lowered. However, the cubic spline interpolation pooling overcomes the limitations of the abovementioned methods. The calculation mode of the segmentation interpolation fits the function corresponding to the segment signal reasonably so that the feature information remains intact. Moreover, the time continuity of the signal is considered, ensuring that the obtained feature vector is smooth and continuous and improves the network recognition rate.

5. Conclusion

For the periodicity and short-time pulse characteristics of one-dimensional time-series signals in fault diagnosis and considering the time continuity of sampling points before and after the signals, a CNN pooling method based on the cubic spline interpolation is proposed in the present study. It should be indicated that the eigenvectors input from the previous layer of convolutional layers are divided into several nonoverlapping subsignal segments, initially. Then, the corresponding fitting function is calculated by the cubic spline interpolation in each signal segment. Moreover, each of the interpolated points after the fitting is sequentially arranged as an eigenvector output. The proposed method can preserve the feature information in the original signals as much as possible while ensuring the continuity of the signals. The comparison tests on the CWRU public dataset show that the proposed method in the present study has a higher recognition rate and stability for one-dimensional signals. Moreover, it better preserves the signal characteristics while achieving data dimension reduction.

Data Availability

The data used to support the findings of this study are available in [30].

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was funded by the National Natural Science Foundation of China (grant number 51705531).

References

[1] L. Deng and D. Yu, “Deep learning: methods and applications,” *Foundations and Trends in Signal Processing*, vol. 7, no. 3-4, pp. 197–387, 2014.

[2] J. Schmidhuber, “Deep learning in neural networks: an overview,” *Neural Networks*, vol. 61, pp. 85–117, 2015.

[3] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[4] Y. Keiji and K. Yoshiyuki, “Food image recognition using deep convolutional network with pre-training and fine-tuning,” in *Proceedings of the 2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pp. 1–6, IEEE, Turin, Italy, June 2015.

[5] R. M. Santos, L. N. Matos, and H. T. Macedo, “Speech recognition in noisy environments with convolutional neural networks,” in *Proceedings of the 2015 Brazilian Conference on*

Intelligent Systems, pp. 175–179, IEEE, Natal, Brazil, November 2015.

[6] T. Yoshioka, S. Karita, and T. Nakatani, “Far-field speech recognition using CNN-DNN-HMM with convolution in time,” in *Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4360–4364, IEEE, Brisbane, Australia, April 2015.

[7] A. Krizhevsky and I. Sutskever, *ImageNet Classification with Deep Convolutional Neural Networks. NIPS*, Curran Associates Inc., New York, NY, USA, 2012.

[8] L. Wen, X. Li, L. Gao, and Y. Zhang, “A new convolutional neural network-based data-driven fault diagnosis method,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5990–5998, 2018.

[9] T. Ince, S. Kiranyaz, L. Eren, M. Askar, and M. Gabbouj, “Real-time motor fault detection by 1D convolutional neural networks,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 7067–7075, 2016.

[10] D. Peng and Z. Liu, “A novel deeper one-dimensional CNN with residual learning for fault diagnosis of wheelset bearings in high-speed trains,” *IEEE Access*, vol. 7, pp. 10278–10293, 2019.

[11] J. Ponce and Y. A. Lecun, “Theoretical analysis of feature pooling in visual recognition,” in *Proceedings of the 27th International Conference on Machine Learning (ICML-2010)*, pp. 111–118, Haifa, Israel, June 2010.

[12] D. H. Hubel and T. N. Wiesel, “Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex,” *The Journal of Physiology*, vol. 160, no. 1, pp. 106–154, 1962.

[13] J. J. Koenderink and A. J. V. Doorn, “The structure of locally orderless images,” *International Journal of Computer Vision*, vol. 31, no. 2-3, pp. 159–168, 1999.

[14] D. Scherer, A. Müller, and S. Behnke, “Evaluation of pooling operations in convolutional architectures for object recognition,” in *Proceedings of the 20th International Conference on Artificial Neural Networks*, pp. 92–101, Thessaloniki, Greece, September 2010.

[15] K. Fukushima and S. Miyake, “Neocognitron: a new algorithm for pattern recognition tolerant of deformations and shifts in position,” *Pattern Recognition*, vol. 15, no. 6, pp. 455–469, 1982.

[16] Y. Lecun, B. Boser, and S. Denker, “Handwritten digit recognition with a back-propagation network,” *Advances in Neural Information Processing Systems*, pp. 396–404, ACM, San Francisco, CA, 1990.

[17] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[18] K. Jarrett, K. Kavukcuoglu, and M. Ranzato, “What is the best multi-stage architecture for object recognition?” in *Proceedings of the 2010 IEEE International Conference on Computer Vision*, Kyoto, Japan, September 2010.

[19] Z. Matthew, “Hierarchical convolutional deep learning in computer vision,” New York University, New York, NY, USA, Doctoral dissertation, 2013.

[20] Z. Matthew and F. Rob, “Stochastic pooling for regularization of deep convolutional neural networks,” 2013, https://arxiv.org/abs/1301.3557.

[21] D. Yu, H. Wang, P. Chen, and Z. Wei, “Mixed pooling for convolutional neural networks,” in *Proceedings of the 9th International Conference on Rough Sets and Knowledge Technology (RSKT 14)*, pp. 364–375, Shanghai, China, October 2014.

[22] W. Hai and G. Xiao, “Max-pooling dropout for regularization of convolutional neural networks,” *Neural Information Processing*, Springer, Berlin, Germany, pp. 46–54, 2015.

[23] H. Lee, G. Kim, H.-G. Kim, S.-H. Oh, and S.-Y. Lee, “Deep CNNs along the time Axis with intermap pooling for robustness to spectral variations,” *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1310–1314, 2016.

[24] Q. Chun and W. Jiang, *Research on Fault Diagnosis Method for Reciprocating Compressors Based on Whole Cycle Acting Signal Collection*, Naver, Seongnam-si, South Korea, 2015.

[25] X. Ling, T. Guiji, and H. Aijun, “Analysis of rotor rubbing fault signal based on Hilbert-Huang transform,” in *Proceedings of the International Conference on Measuring Technology & Mechatronics Automation*, pp. 586–589, Changsha, China, April 2009.

[26] L. Zhan, L. Jian, and Z. Pei, “Study on engine vibration signal collection method,” in *Proceedings of the International Conference on Information Engineering & Computer Science*, pp. 25-26, Wuhan, China, December 2010.

[27] S. Huang, J. Tang, and J. Dai, “Signal status recognition based on 1DCNN and its feature extraction mechanism analysis,” *Sensors*, vol. 19, no. 9, p. 2018, 2019.

[28] S. A. Dyer and J. S. Dyer, “Cubic-spline interpolation,” *IEEE Instrumentation & Measurement Magazine*, vol. 4, no. 1, pp. 44–46, 2001.

[29] B. C. Wheeler and S. R. Smith, “High-resolution alignment of action potential waveforms using cubic spline interpolation,” *Journal of Biomedical Engineering*, vol. 10, no. 1, pp. 47–53, 1988.

[30] CWRU, *Case Western Reserve University Bearing Date Center Website*, CWRU, 2006, http://csegroups.case.edu/bearingdatecenter/home, Cleveland, OH, USA.