Bearing Fault Diagnosis based on Interpretable Sparse Method

Shan Shi

School of Logistics Engineering, Shanghai Maritime University, Shanghai 200000, China.

836964469@qq.com, cookie_shishan@163.comm

Abstract

Rolling bearings are one of the most frequently faulty components in wind turbines. Timely bearing fault diagnosis can minimize economic losses and accident losses. For deep learning fault diagnosis methods, there are internal black box problems and traditional interpretability methods cannot reach Two-way optimization problem of interpretability and accuracy, a gate structure sparse neural network based on the LRP method is proposed. By designing an interpretable adaptive sparse gate structure, the weight of the sparse gate is adaptively adjusted to filter the input information of the DNN , So that the model achieves the purpose of sparseness. The experimental results show that the correct rate of fault diagnosis for rolling bearings using this method is 97.11%, which is 4.74% higher than that of the traditional DNN, which verifies the effectiveness of the method.

Keywords

Rolling Bearing; Deep Learning; DNN; Interpretability; LRP; Sparse Gate.

1. Introduction

Rolling bearing is one of the most important parts in mechanical equipment, and its running state directly affects the performance of the equipment, so it is of great significance to diagnose the fault of rolling bearing. In recent years, deep learning has been widely used in rolling bearing fault diagnosis. Deep learning can achieve complex function approximation by learning a deep nonlinear network structure,characterize the distributed representation of input data,and demonstrate a powerful learning data set from a small number of samples. The ability of essential characteristics has received extensive attention from various fields.Deep learning has been extensively studied especially in the aspect of machinery condition monitoring,and is superior to several other machine learning techniques [1]. In the field of fault diagnosis, according to the different network structure, researchers have in-depth research and application of structural frameworks:Convolutional neural network method, deep belief network-based method and stacked autoencoder-based method,recurrent neural network.

Convolutional neural network is a highly efficient algorithm for visual image processing and analysis. The CNN network is composed of a deep feedforward neural network. The convolutional neural network can handle high dimensionality and high nonlinearity under supervised learning. Compared with other image classification algorithms, the preprocessing process is less. Existing literature usually uses CNN network for feature extraction and feature recognition. Janssens et al [2]. proposed a one-dimensional convolutional neural network that uses the discrete Fourier transform of normalized vibration signals to diagnose bearing faults. Guo et al [3]. proposed a layered method based on convolutional neural network to diagnose bearing faults from raw vibration signals. A convolutional neural network with a new adaptive learning rate is used to diagnose the fault type, and then the separately trained convolutional neural network is used to determine the severity of the fault.

Recurrent neural network is a neural network that is good at processing time series. The topological structure guarantees the time memory ability of the neural network, and the length of time represents the depth of the network. Guo et al [4]. applied cyclic neural network to the fault detection of power transmission and transformation system of unstructured data, and compared the influence of different network parameters on the diagnosis results. Moustapha et al [5]. applied cyclic neural network to

sensor node fault detection, Using RNN to identify and fault detection of sensor nodes in the wireless sensor network, and compared with the Kalman filter method. The simulation example shows the effectiveness of the method.

The deep belief network is formed by stacking multiple layers of restrictive Boltzmann machines. Restricted Boltzmann machine is an energy probability generation model. It consists of a visible layer and a hidden layer. The visible layer and the hidden layer are connected by weights. Tao et al. use deep belief networks to diagnose rolling bearing faults. Compared with shallow learning methods, DBN can improve the accuracy of rolling bearing fault diagnosis and accurately identify rolling bearing faults. Deng et al [6]. extracted wavelet decomposition energy spectrum characteristics, time and frequency characteristics of the original signal, as the input of the DBM fault diagnosis model, the experimental results showed that the latter two can effectively characterize the fault information, which proves that the DBM is used for fault diagnosis. Choosing low-level features can get satisfactory results. Meng G et al [7]. introduced a deep trust network to classify fault types, and proposed a new hierarchical diagnosis network, which uses a two-level diagnosis network with wavelet packet energy characteristics to collect hierarchical deep belief networks for hierarchical identification of mechanical systems. Experiments The results show that HDN has high reliability for multi-stage accurate diagnosis and can overcome the overlap problem caused by noise and other interference.

Autoencoder is an unsupervised learning algorithm, composed of three layers: input layer, hidden layer and output layer. Stacked autoencoders are formed by stacking multiple autoencoders. The deep neural network (DNN) based on SAE can achieve more intuitive deep learning. DNN can effectively extract the fault characteristics of mechanical equipment and avoid the problem of falling into local optimality. In bearing fault diagnosis, the extracted vibration signals are mostly one-dimensional sequence data. The autoencoder has a simple structure, and the deep neural network based on stacked autoencoders has certain advantages in processing time series signals. It can directly process one-dimensional time series signals and eliminate redundant information in the data. The structure is simple and easy to implement. Muhammad et al [8]. apply batch normalization to each layer of the autoencoder before the activation function, which reduces the difficulty of training. The bearing data verifies that the batch normalized stacked sparse autoencoder has good fault diagnosis performance. Chen et al [9]. used the greedy training method to train the stacked noise reduction autoencoder, which has higher fault diagnosis accuracy compared with traditional PCA and stacked autoencoder. Lu et al [10]. directly use deep neural network (DNN) for signal feature extraction, and the obtained features can accurately and identifiably describe the bearing signal data, which is of great significance to the completion of fault diagnosis. The experiment proves that DNN is a kind of signal data A powerful tool for feature extraction.

Deep neural networks based on stacked autoencoders have been extensively studied in fault diagnosis. Fault diagnosis methods based on neural networks have a major limitation, that is, the "black box" nature of their diagnostic decision-making and learning processes. In order to solve this problem, an explanation method based on back propagation is proposed. The core idea is to use the back propagation mechanism to propagate the decision importance signal of the model from the output layer neurons of the model to the input of the model to derive the input samples. Feature importance. Methods such as Grad [11], GuidedBP [12] and Integrated [13] use this core idea to explain. However, the saliency maps obtained by these three methods through backpropagation usually contain a lot of visually visible noise. For this reason, Smilkov D et al [14]. proposed a smooth gradient backpropagation interpretation method, which passes through the input sample The introduction of noise solves the visual noise problem in Grad and other methods. Although the above method based on gradient backpropagation can locate the decision features in the input sample, it cannot quantify the contribution of each feature to the model decision result. Bach S et al [15]. proposed a layer-wise relevance propagation method to calculate the contribution of a single pixel to the prediction result of the image classifier. The general form of the LRP method assumes that the classifier can be decomposed into multiple computational layers. Each layer can be modeled as a multi-dimensional

vector and each dimension of the multi-dimensional vector corresponds to a correlation score. The core of LRP is Backpropagation is used to recursively propagate the correlation score of the high-level to the low-level until it reaches the input layer. Interpretation methods based on backpropagation are usually simple to implement, computationally efficient, and make full use of the structural characteristics of the model. John et al [16] proposed an interpretable deep convolutional neural network for fault diagnosis of gearboxes. Use vibration signal as time series data, classify by wavelet transform and discrete cosine neural network, use LRP method to decompose the contribution of the local area in the spectral image to the classification result, and determine the time-frequency point in the spectral image to the fault type And the degree of contribution to severity identification. Grezmak et al [17]. proposed an interpretable convolutional neural network for machine fault diagnosis through hierarchical correlation propagation. Using LRP as an indicator, they studied the performance of training CNN on time-frequency spectrum images of vibration signals measured on induction motors. It can be seen that the LRP index can be used to quantify the relationship between the output of each layer of neurons through its back-propagation calculation method, so as to achieve the purpose of interpretability.

Deep neural network With the continuous increase of network depth and network neurons, the network structure is becoming more and more redundant. In order to solve the above problems, trying to take appropriate sparse methods for the network will effectively improve the diagnosis effect of the network. This chapter proposes a sparse method based on network interpretable parameters. The LRP method can be used to quantify the correlation between the input and the output result, and the quantization value is used as a criterion to combine with the sparse gate structure to design an interpretable adaptive sparse gate structure, referred to as IAS gate. This soft threshold gate structure is applied to a deep neural network, and the weight of the sparse gate is adaptively adjusted to filter the input information, so that the network achieves the purpose of sparseness, and IAS-DNN is used to achieve interpretability and two-way optimization of the neural network. For the first time, it is proposed to use network interpretability index to adjust network parameters adaptively.

2. Related technical principles

2.1 Basic knowledge

2.1.1 AE feature extraction process

Autoencoder is considered to be a very useful basic model in the field of deep learning.It is a special type of feedforward neural network composed of an encoder and a decoder.The encoder is composed of an input layer and a hidden layer, and the output It is composed of a hidden layer and an output layer, and has a symmetrical structure.Autoencoder is a kind of neural network,it is an unsupervised learning method,the goal is to make the output equal to the input. Its structure is shown in Figure 1. In this model, the encoder uses nonlinear mapping to compress the input data to obtain the characteristic representation of the data;the decoder reconstructs the network and maps it to the output space to obtain the reconstructed representation of the input.The training process of AE is to optimize the network parameters by minimizing the reconstruction error,so that the output is as close to the input as possible.

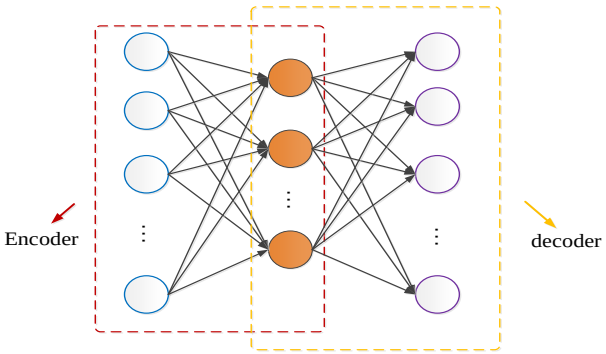

Fig. 1 AE basic model structure diagram

Given an unlabeled sample set $\{x_i^k\}, (i = 1,2,\dots,I\,; k = 1,2,\dots,K)$, in which i epresents sample dimensions, k Indicates the number of samples, the encoding process of the encoder is shown in the formula (1):

$$h_i = f_\theta(x_i) = \sigma(Wx_i + b) \tag{1}$$

Among them, f_θ is the coding function, σ is the activation function, usually the Sigmoid function, W is the network weight matrix between the autoencoders, that is, the weight matrix between the input layer and the hidden layer, b is the bias vector in the coding network, $\theta = \{W, b\}$ is Combine weights and biases together as a connection parameter between the input layer and the hidden layer. The general form of the sigmoid function is shown in formula (2):

$$\sigma(x) = log(\,1 + e^{-x}) \tag{2}$$

Similarly, for the decoding network, the encoding vector h obtained by the encoding network is reconstructed through the decoding network to obtain y_i which is equal to the input, that is, y_i is equal to the input x_i. The decoding process is shown in equation (3):

$$y_i = g_{\theta^T}(h_i) = \sigma(W^T h_i + d) \tag{3}$$

Where g_{θ^T} is the decoding function, σ is the activation function of the encoding process, W^T is the network weight matrix from the hidden layer to the output layer, and d is the bias vector generated during the encoding process.

The essence of the process of training AE is the training and optimization of the network parameter θ and θ^T In order to make the output y_i as close to the input x_i as possible, the closeness between the input and the output is characterized by minimizing the reconstruction error $J_{AE}(x, y\,; \theta\,, \theta^T)$, as shown in equation (4):

$$J_{AE}(x, y\,; \theta\,, \theta^T) = \frac{1}{m}||y - x||^2 \tag{4}$$

In each training process, the gradient descent method is used to update the AE network training parameters and the entire parameter update process is as follows

$$W = W - \alpha\frac{\partial}{\partial W}J_{AE}(x, y\,; \theta\,, \theta^T) \tag{5}$$

$$W^T = W^T - \alpha\frac{\partial}{\partial W^T}J_{AE}(x, y\,; \theta\,, \theta^T) \tag{6}$$

$$b = b - \alpha\frac{\partial}{\partial b}J_{AE}(x, y\,; \theta\,, \theta^T) \tag{7}$$

$$b^T = b^T - \alpha\frac{\partial}{\partial b^T}J_{AE}(x, y\,; \theta\,, \theta^T) \tag{8}$$

Among them, α is the learning rate, $\frac{\partial}{\partial W_i}J_{AE}(x, y\,; \theta\,, \theta^T)$ and $\frac{\partial}{\partial b_i}J_{AE}(x, y\,; \theta\,, \theta^T)$ are calculated using the back propagation algorithm, and the gradient direction.

2.1.2 Deep neural network training

The deep neural network constructed in this paper is a multi-hidden neural network stacked by multiple autoencoders. In the unsupervised learning stage, bottom-up layer-by-layer feature extraction is used, and the output of the previous autoencoder is used as the latter. Input from the encoder. Then, the result of unsupervised layer-by-layer feature representation is used as the initial value of the backpropagation optimization algorithm, supervised parameter fine-tuning, and more abstract feature representations are extracted from the original input information.

Given an unlabeled input data x as the input of the encoder, the hidden layer feature h_1 of the first autoencoder AE_1 and the network parameter $\theta_1 = \{W_1, b_1\}$ of the first layer are obtained through unsupervised training, and then h_1 will be used as the second autoencoder The input of the AE_2, through unsupervised training, the hidden layer feature h_2 of AE_2 and the network parameter $\theta_2 = \{W_2, b_2\}$ of the second layer are obtained. Repeat the above process to obtain the last hidden layer feature h_N of the autoencoder AE_N and the network parameter $\theta_N = \{W_N, b_N\}$ of this layer.

Then the extracted feature h_N is used as the input of the Softmax classifier, and the labeled dataset $\{1,2,\cdots,K\}$ is used as the output to train the Softmax classifier. For an observation sample $x(m) = [x_1(m), x_2(m), \cdots, x_k(m)]$ at time m, first use it as the input of the deep neural network to obtain the hidden layer feature $h_N(m)$, and then input $h_N(m)$ into the trained Softmax classifier to obtain the classification result of the observation sample $x(m)$.

$$label(m) = \underset{j=1,2,\cdots,K}{argmax}\{p(label(m) = k|x(m);\theta)\} \tag{9}$$

$$h_\theta(x(m)) = \begin{bmatrix} p(label(m)=1 \mid x(m);\theta) \\ p(label(m)=2 \mid x(m);\theta) \\ \vdots \\ p(label(m)=k \mid x(m);\theta) \end{bmatrix} = \frac{1}{\sum_{k=1}^{K} e^{\theta_k^T x(m)}} \begin{bmatrix} e^{\theta_1^T x(m)} \\ e^{\theta_2^T x(m)} \\ \vdots \\ e^{\theta_k^T x(m)} \end{bmatrix} \tag{10}$$

Among them, $\theta = [\theta_1, \theta_2, \cdots, \theta_k]$ is the model parameter of the Softmax classifier, which is similar to the autoencoder model. In order to ensure the accuracy of classification, the cost function J_θ is minimized to optimize the network parameters of the model. The cost function of the training process of the Softmax classifier is shown in equation (10), and its network parameters can be obtained by minimizing $J_\theta(x(m))$.

The last is the reverse fine-tuning optimization process of SAE, which uses a supervised back-propagation algorithm.Use labeled data to optimize and fine-tune the parameters of the entire deep neural network, and complete the fine-tuning process by minimizing the reconstruction error $E(\theta)$. The optimization update process of the parameters is as follows:

$$\theta = \theta - \alpha \frac{\partial E(\theta)}{\partial \theta} \tag{11}$$

$$E(\theta) = \frac{1}{M}\sum J_\theta(Y', T_k; \theta) \tag{12}$$

In which, T_k is the known label data set, Y' is the actual output of the deep neural network, α is the learning rate, θ and is the model parameter of the Softmax classifier. The model parameters θ can be optimized by formulas (11) and (12).

2.1.3 Layer-wise Relevance Propagation

In the LRP method, attribute scores, called relevance scores, are used to calculate the structure of the classifier in a top-down manner. For neural network classifiers, the relevance score is propagated from the output. The output layer relevance score is usually regarded as the output layer pre-activation value corresponding to the class of the associated score, and the value corresponding to the class that requires the relevance score is obtained by forwarding the input. The correlation score is propagated to the previous layers, so that the sum of the correlation scores for each layer is constant.

$$f(x) = \cdots = \sum_{d\in l+1} R_d^{(l+1)} = \sum_{d\in l} R_d^{(l)} = \cdots = \sum_d R_d^l \tag{13}$$

Where $f(x)$ is the real-valued prediction output, $R_d^{(l)}$ is the neuron d of the l th layer. For the fully connected layer, there are several methods to propagate the correlation with the previous layer, and at the same time satisfy the above formula, use the z -rule, and its correlation The propagation formula of is:

$$R_i^{(l)} = \sum_j \frac{z_{ij}}{\sum_{i'} z_{i'j}} R_j^{(l+1)} \tag{14}$$

Where z_{ij} is the contribution of activation at neuron i to the total pre-activation of neuron j, which is defined as $z_{ij} = a_i^{(l)}\omega_{ij}^{(l,l+1)}$, where $a_i^{(l)}$ is the activation of neuron i in layer l, and $\omega_{ij}^{(l,l+1)}$ is the weight that connects neuron i and j. The z -rule of relevance propagation, the relevance score can bear both positive and negative values. Since the probability of an input belonging to a certain class is ultimately measured by the value of its corresponding output layer neuron, the correlation score

can represent evidence of classification decision. Through the global analysis of the input correlation score, the region in the input with a large correlation score can be used as an indication of the mode that contributes the most to the classification decision. Based on the correlation score, the improved deep neural network can be explored from the interpretability level.

2.2 Sparse neural network of gate structure based on LRP method

For traditional sparse gates, the correlation of each neuron to the classification results is calculated through each training process, and the sparse gate is set using its size as the basis for discrimination, and each layer of the network is effectively sparsed according to the correlation size through an adaptive method. In essence, it is an improvement of the neural network structure, and it is not interpretable.

This section proposes an adaptive sparse gate algorithm based on network interpretability index design. Find out the interference information in the feature extraction process from the interpretable level, and suppress it in a targeted manner. The remaining relevant useful information is propagated to the next layer to achieve the network sparse process. After suppression, the training parameters are all parameters that contribute greatly to the output. So as to achieve the purpose of improving the accuracy of network training.

The design of the IAS gate is shown in Figure 2.

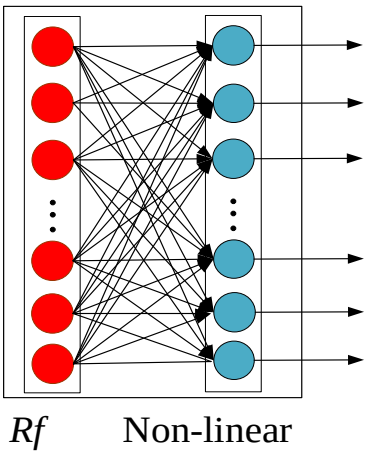

Fig. 2 IAS gate diagram

The red nodes in the figure represent the correlation parameters calculated based on the LRP algorithm for each layer of neurons, and the blue nodes represent the neurons of the sparse gate. When the correlation parameter of a neuron is processed by a layer of neurons, a set of sparse indicators is given, and its dimension is the same as the number of neurons, that is, each neuron has a sparse indicator, and it is between 0 and 1. During the optimization training process of the network, the parameters of the sparse gate are optimized and adjusted synchronously, so that the self-adaptation of the sparse value of the network is realized.

Use correlation to control the degree of opening and closing of the sparse gate. The sparse process is as follows. First, in each forward process, the correlation matrix $R^{(h_i)}$ of each layer and node is calculated, and normalized, so that the correlation score is mapped In the range of [0,1].

$$R_{norm}^{(l_i)} = \frac{R^{(l_i)} - R_{min}^{(l_i)}}{R_{max}^{(l_i)} - R_{min}^{(l_i)}} \tag{15}$$

Use the normalized correlation value as the input of the gate to control the IAS gate. IAS gateis a group of neurons whose output dimension is consistent with the number of neurons in this layer. The number of neuron layers can also be set flexibly

$$I(R_{norm}^{(l_i)}) = \sigma(W_I R_{norm}^{(l_i)} + b_I) \quad (16)$$

In the formula: W_I and b_I are the weights and biases of the sparse gate, and the σ function is the nonlinear function of the sparse gate. In order to ensure that the output of the nonlinear function is in the range of 0 to 1, the nonlinearity is shown in formula (4):

$$\sigma(x) = log(1 + e^{-x}) \quad (17)$$

Because the correlation score represents the contribution of the input data to a certain extent, and the degree of opening and closing of the gate is adjusted by the size of the correlation score, through the sparse gate, the data with large correlation can be passed quickly, and the data with small correlation can be effectively suppressed, To achieve the purpose of sparsity, which can better extract features.

In the training process of the network, according to each round of prediction output, the LRP value of each layer is calculated, and the LRP value of this layer is normalized and sent to the sparse gate neural layer of the layer as the input value adaptive Adjust the sparse ratio of each layer. The sparse neural network model with gate structure based on the LRP method is shown in Figure 3.

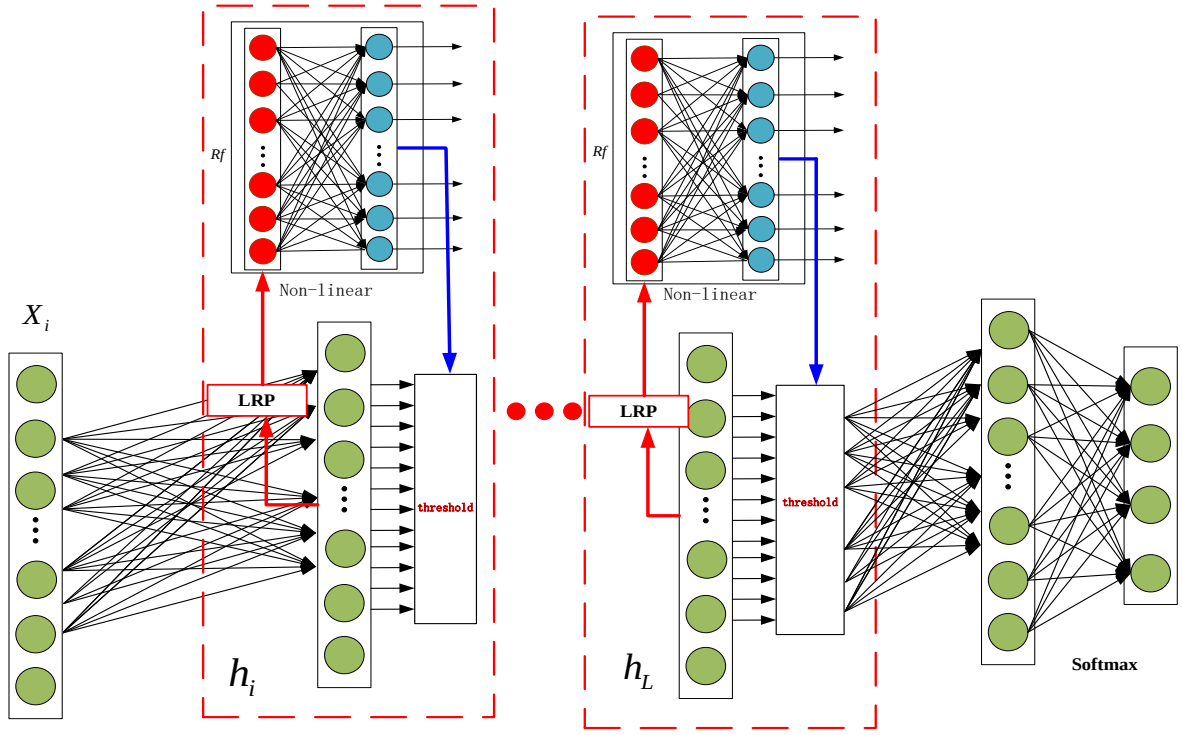

Fig. 3 DNN structure diagram with IAS gate structure

The DNN structure with the IAS gate structure uses the LRP value based on the DNN network to adaptively adjust the sparse value of the sparse gate, and then the output value of the layer of neurons passes through the sparse gate, so that the neural network achieves the goal of sparseness.

Assuming that the input feature of each layer is h, and the weight w and bias b of the neurons in this layer are sums respectively, then the output of the neurons in this layer is:

$$H(h_i) = \sigma_i(W_s h_i + b_s) \quad (18)$$

The output value of the neuron of the diagnosis network is multiplied by the threshold value of the IAS gate output to obtain the output of the neuron of this layer, as shown in formula (19)

$$h_{i+1} = I(R_{norm}^{(l_i)}) \circ H(h_i) \quad (19)$$

In the formula, 1 is the dot product symbol, which means that the corresponding elements are multiplied as the input of the next hidden layer, and so on. The process of IAS-DNN is: first, process the collected data, and then initialize the network parameters randomly. Then, the training data is

forward propagated in the network, and the prediction output is calculated in the process of forward propagation. At the same time, the correlation score $R^{(h_i)}$ between each layer and the prediction result is calculated backward, and the parameters are updated. When the network parameters meet the optimization goal, the network parameter update is stopped, the network parameters are saved, and the online diagnosis mode is entered.

3. Experimental Results

In order to verify the feasibility of the algorithm, a simulation experiment was carried out on the proposed algorithm.This paper takes rolling bearings as the fault diagnosis object to verify the effectiveness of the IAS-DNN fault diagnosis method.

The experimental data used in this chapter is the open source bearing test data set provided by the Bearing Data Center of Case Western Reserve University in the United States. There are three types of failures and one normal state for the types of failures involved in the experiment. The three types of failures are: inner ring failure; outer ring failure; ball failure.

Collect the vibration signal of the motor drive end bearing under different load conditions.The data collected by the sensor contains four health states: inner ring fault, outer ring fault, rolling ball fault and normal. This experiment uses a normal data set and three fault data sets. The three sets of failure data are that the load is 3 horsepower, the size of the failure is 0.007 inches, and the types of failures are inner ring failure, outer ring failure, and ball failure. The sampling frequency of the vibration sensor is 12kHz, and the motor speed is set to 1772. There are about 450 sampling points per lap. This experiment selects two-period timing signals as the input signal of the diagnostic network, and the input data dimension is 900 dimensions. For each type of failure, select 2000 sets of training data, a total of 8000 sets, and 2000 sets of test data. In order to compare the influence of different depth DNNs and different neuron redundancy levels on the diagnosis network, the design of the network parameters is shown in Table 1 and Table 2.

Table 1. 5-layer DNN network parameter settings

parameter	Input layer	1st hidden layer	2nd hidden layer	3rdhidden layer	Output layer
Net1	900	1000	100	50	4
Net2	900	1300	500	100	4
Net3	900	1800	1000	200	4

Table 2. 6-layer DNN network parameter settings

parameter	Input layer	1sthidden layer	2ndhidden layer	3rdhidden layer	4thhidden layer	Output layer
Net1	900	1000	500	100	50	4
Net2	900	1300	900	500	100	4
Net3	900	1800	3000	1000	200	4

Table 3. Comparison table 1of diagnostic accuracy of different network parameters

Net	Data set	1	2	3	4	5
Net1 DDN	Training	97.95	97.06	98.50	98.77	97.36
	Testing	92.72	92.73	92.98	92.94	92.10
Net1 IAS-DDN	Training	100	100	100	100	100
	Testing	96.89	96.03	95.86	96.15	96.35
Net2 DDN	Training	98.75	98.70	98.79	99.02	98.75
	Testing	93.69	92.86	93.80	93.52	93.26
Net2 IAS-DDN	Training	100	100	100	100	100
	Testing	95.65	96.52	96.18	96.42	96.56
Net3 DDN	Training	99.61	98.61	99.27	97.52	95.52
	Testing	95.12	92.79	94.73	91.75	90.78
Net3 IAS-DDN	Training	100	100	100	100	100
	Testing	97.09	96.36	96.77	96.56	96.93

In order to effectively measure the sparse ability of the proposed sparse network, a comparative experiment was carried out with traditional DNN. The method of increasing the depth of the network by increasing the number of neurons makes the number of neurons in the network have a larger number of redundant neurons. The optimization goal setting is the same, and the experimental comparison conclusion is given in the next section.

The Loss function optimization goals of the three network structures are set to 0.001 respectively to compare the diagnostic accuracy of the traditional DNN and IAS-DNN on the training data set and the test data set. Carry out 10 experiments respectively, and compare the experimental results. The experimental results are shown in Table 3 and Table 4

Table 4. Comparison table 1of diagnostic accuracy of different network parameterrs

Net	Data set	1	2	3	4	5
Net4 DDN	Training	96.63	97.68	96.81	97.79	97.16
	Testing	91.58	93.08	91.53	92.10	91.95
Net4 IAS-DDN	Training	99.45	100	100	100	100
	Testing	94.80	95.51	94.32	95.06	94.93
Net5 DDN	Training	97.18	97.32	97.91	98.53	97.18
	Testing	92.96	91.88	93.01	92.48	91.29
Net5 IAS-DDN	Training	100	100	100	100	100
	Testing	97.08	96.97	96.37	96.42	95.86
Net6 DDN	Training	97.94	98.86	97.98	97.55	94.51
	Testing	92.51	94.17	93.96	92.18	89.03
Net6 IAS-DDN	Training	100	100	100	100	100
	Testing	97.87	97.17	96.01	96.67	97.83

The above network diagnosis accuracy comparison shows that with the increase of network neurons, the traditional DNN method can achieve better diagnosis accuracy for the training set, and slightly improve the diagnosis result for the test set, but the effect is not obvious.

Table 5. Comparison of average diagnosis accuracy of 6 network structures

Net	Net 1	Net 2	Net 3	Net 4	Net 5	Net 6	
DNN	97.92	98.80	98.10	97.21	97.62	97.36	Training
IAS-DNN	100	100	100	99.89	100	100	
DNN	92.69	93.42	93.03	92.05	92.32	92.37	Testing
IAS-DNN	96.25	96.26	96.74	94.92	96.54	97.11	

Comparing the diagnosis networks of the six structures, it is obvious that the improved IAS-DNN network based on the network interpretability sparse gate is better than the traditional DNN in both the training accuracy of the test set and the diagnosis accuracy of the test set. Great improvement. It is also worth pointing out that the IAS-DNN diagnosis network will not have a significant impact on the diagnosis results as the network structure changes, that is to say, the redundant structure of the network designed based on IAS-DNN will not affect the diagnosis accuracy of the entire network.

4. Conclusion

Based on the network interpretability index LRP, this chapter proposes an improved soft threshold sparse gate structure to sparse the traditional DNN. First, the role and significance of sparse methods for deep neural networks are introduced, and then the defects of hard sparse methods and the interpretability limitations of existing soft threshold sparse gates are summarized. Based on the above analysis, a soft threshold sparse gate structure based on the improvement of interpretability parameters is proposed. The specific algorithm flow of the fault diagnosis method of DNN with improved sparse gate structure is given. Finally, a detailed comparison of the improvement degree of the method proposed in this paper compared with the traditional DNN method is carried out through

experimental simulation. The experimental results show that the method proposed in this paper can effectively avoid the over-fitting of the diagnostic network and effectively improve the accuracy of the network.

Acknowledgements

Thanks to my alma mater, Shanghai Maritime University School of Logistics Engineering for training and education.

References

[1] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, “Deep learning and its applications to machine health monitoring,” Mech. Syst.Signal Process., vol. 115, pp. 213–237, Jan. 2019.

[2] O. Janssens et al., “Convolutional neural network based fault detection for rotating machinery,” J. Sound Vib., vol. 377, pp. 331–345, Sep. 2016.

[3] X. Guo, L. Chen, and C. Shen, “Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis,” Measurement, vol. 93, pp. 490–502, Nov. 2016.

[4] Guo Lili, Ding Shifei. Research progress of deep learning[J]. Computer Science, 2015, 042(005): 28-33.

[5] Moustapha A I, Selmic R R. Wireless Sensor Network Modeling Using Modified Recurrent Neural Networks: Application to Fault Detection[J]. IEEE Transactions on Instrumentation & Measurement, 2008, 57(5):981-988.

[6] Tao Jie, Liu Yilun, Yang Dalian, et al. Rolling bearing fault diagnosis based on bacterial foraging algorithm and deep belief network[J]. Vibration and Shock, 2017, 036(023): 68-74.

[7] Meng G, Cong W, Zhu C. Construction of hierarchical diagnosis network based on deep learning and its application in the fault pattern recognition of rolling element bearings[J]. Mechanical Systems & Signal Processing, 2016, 72-73(May):92-104.

[8] Muhammad Sohaib and Jong-Myon Kim.Reliable Fault Diagnosis of Rotary Machine Bearings Using a Stacked Sparse Autoencoder-Based Deep Neural Network[J].Shock and Vibration,2018.

[9] Chen Z, Deng S, Chen X, ct al. Deep ncural nctworks-bascd rolling bearing fault diagnosis[J]. Microclcctronics Rcliability, 2017, 75:327-333.

[10] Lu W, Wang X, Yang C, et al. A novel feature extraction method using deep neural network for rolling bearing fault diagnosis [C]// The 27th Chinese Control and Decision Conference (2015 CCDC). IEEE, 2015.

[11] Simonyan K, Vedaldi A, Zisserman A. Deep inside concolutional networks: Visualising image classification models and saliency maps [J]. arXiv preprint arXiv :1312.6032,2013

[12] Springenberg J T, Dosocitekiy A, Brox T, el al. Striving for simplicity: The all concolutional net[J]. arXiv preprint arXiv:1412.6806,2014

[13] Zelier M D,Fergus R.Visualizing and understanding convolutional networks [C]// Proc of the 13th European Conf on Computer Vision. Berlin:Springer,2014:818-833

[14] Smilkov D,Thorat N,Kim B,et al.Smoothgrad:Removing noise by adding noise[J].arXiv preprint arXiv:1706.03825,2017

[15] Bach S, Binder A, Montavon G, et al. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation[J]. ploS One, 2015,10(7):e0130140

[16] John Grezmak, Peng Wang, Chuang Sun, et al. Explainable Convolutional Neural Network for Gearbox Fault Diagnosis. 2019, 80:476-481.

[17] J. Grezmak, J. Zhang, P. Wang, K. A. Loparo and R. X. Gao, "Interpretable Convolutional Neural Network Through Layer-wise Relevance Propagation for Machine Fault Diagnosis," in IEEE Sensors Journal, vol. 20, no. 6, pp. 3172-3181, 15 March15, 2020, doi: 10.1109/ JSEN. 2019. 2958787.