Improved DAE and Application in Fault Diagnosis

Funa Zhou[1], Shuai Yang[1*], Chenglin Wen[2]

1. School of Computer and Information Engineering, Henan University, Kaifeng, China
E-mail: zhoufn2002@163.com, yangshuai2711@163.com (* Corresponding Author)

2. School of Automation, Hangzhou Dianzi University, Hangzhou, China
E-mail: wencl@hdu.edu.cn

Abstract: Deep neural network (DNN) with powerful feature learning ability is widely used in fault diagnosis. Stacked de-noising auto-encoder(DAE) is a common way to achieve a deep neural network. In this paper, the learning rate of DAE has been improved. Different from the traditional fixed learning rate, the idea of time-varying learning rate is adopted in the paper which effectively reduces the oscillation of the cost function of DAE during the iterative process based on the guaranteed iteration rate, so that the extracted features are more favorable for fault classification. Experimental results show that the proposed method can improve fault classification accuracy and robustness of the DNN based fault diagnosis model.

Key Words: DNN; Fault diagnosis; Time-varying learning rate; DAE; IDAE

1. INTRODUCTION

With the development of science and technology, mechanical equipment is becoming more and more complex structure. When these mechanical devices are affected by some uncertain factors in operation, it will lead to certain degrees of damage and result in some failure or more serious consequences. Mechanical failure in industrial production not only causes economic loss, but also causes casualties in serious cases[1-3]. Motor bearings and other mechanical equipment are prone to failure. Therefore, fault diagnosis has a very important meaning for machinery equipment.

The methods for fault diagnosis can be divided into qualitative model based method, quantitative model based method, and data-driven based method[4,5]. The method of qualitative model and quantitative model is more demanding for model accuracy, which limits the application of these methods in fault classification to a certain extent. In recent years, methods based on data-driven statistical features and machine learning have been extensively applied to machine failure diagnosis[6-9]. However, the method based on data-driven statistical feature extraction can only detect fault but can not classify machine faults well. Some scholars use machine learning methods such as Support Vector Machine (SVM) and Artificial Neural Network (ANN) to diagnose faults. But SVM and ANN are both shallow learning method[10].

Hinton proposed a deep learning method in 2006[11]. Deep learning has strong ability to extract potential nonlinear features involved in the observation data [12,13]. Due to the powerful feature extraction capabilities, deep learning was quickly used in fault diagnosis[14-17]. A structure of DNN can be achieved by stacking multilayer Auto-Encoder. The Auto-Encoder can not extract the features of data very well when dealing with noisy data, which results in a low robustness of DNN with stacked Auto-Encoder. In order to improve the robustness of the model, some scholars use DAE to construct DNN. The traditional training process of DNN use a fixed learning rate. However, when the learning rate is not reasonable, it not only affects the convergence speed of the algorithm, but also causes the cost function to oscillate in the vicinity of the optimal value, thus affecting the robustness of the model.

In order to solve this problem, the idea of time-varying learning rate is adopted to improve the DAE method in this paper. The improved DAE can weaken the phenomenon of oscillation in the iterative process, moreover it can make the cost function smoothly trend to the minimum. So that it is more conducive to get the optimal model parameters. Bearing data testing result shows that the method proposed in this paper has better convergence speed and the robustness. By comparing with DNN in fixed learning rate, we find that this method can also improve fault classification accuracy of the model.

The remaining part of this paper is organized as follows: Section 2 introduces theory of deep learning; Section 3 develops an improved DAE and its application in fault diagnosis; Section 4 presents the experimental results and analysis; Sections 5 is the conclusion.

2. THEORY OF DEEP LEARNING

In this section, we will give a brief introduction to DNN with stacked AE, and its variation structure-DAE.

2.1 Auto-Encoder

Auto-Encoder is a common training model in deep learning. It adopts a kind of unsupervised training method. The features of the input data are obtained by coding, and then use the decoding process to reconstruct the data[18,19] . The model of the auto-encoder is shown in Fig.1. It consists of the input layer, the hidden layer, and the output layer. AE

This research was supported in part by the Natural Science Fund of China (Grant No. U1604158).

only has one hidden layer, and its output layer has the same number of neurons as the input layer.

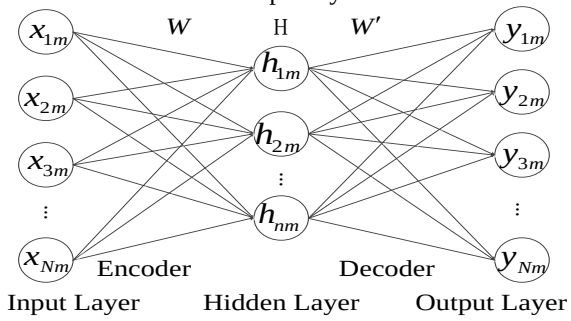

Fig.1 The model of Auto-encoder

For input data sets X, the encoding and decoding process of auto-encoder can be expressed as:

$$H = f(X) = \sigma_1(WX + b) \quad (1)$$

$$Y = F(H) = \sigma_2(W'H + d) \quad (2)$$

Where σ_1 is the activation function for the encoding network, W is the weight of the input layer and the hidden layer, b is the bias vector of the encoding network, H is the activation value of the hidden layer, which is the features of input data X; σ_2 is the activation function of decoding network, W' is connected to the hidden layer and output layer weights, d is the bias vector for the decoding network, Y is the network output, which is the reconstruction value of input X. The principle of auto-encoder is to minimize the error $L(X,Y)$ between X and Y, so that the output Y is as close as possible to the input X. The error between X and Y is defined as follows:

$$L(X,Y) = \frac{1}{M}\|Y - X\|^2 \quad (3)$$

There are many ways to solve the parameters of network optimization such as the gradient descent method, Gauss - Newton method and L-BFGS method, etc[20].

2.2 DAE

In order to improve the robustness of AE, the input data is polluted with additional noise, and the automatic encoder model is trained to be able to reconstruct the complete, noiseless data. This improved Auto-Encoder is DAE. The principle of DAE is shown in Fig.2.

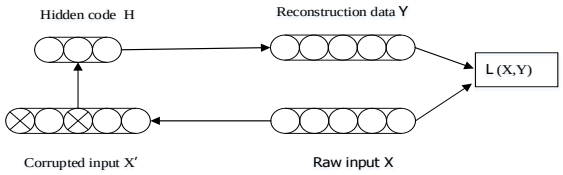

Fig.2 The principle of DAE

In Fig. 2, X is the noiseless data, and X' is the data polluted by additional noise. H is a feature extracted from data X'. Y is the output of DAE. The encoding and decoding process of DAE can be expressed as:

$$H = f(X') = \sigma_1(WX' + b) \quad (4)$$

$$Y = F(H) = \sigma_2(\bar{W}H + d) \quad (5)$$

Unlike AE, the training data of DAE is the additional noise polluted X'. The same formula (3) is used when calculating the cost function, and noiseless data X is used. Similarly, the optimal parameters of the DAE are obtained by minimizing formula (3).

3. DNN BASED ON THE IMPROVED DAE

In the actual application, due to the instability of the sensor or other uncertain factors, the measured data will generate some uncertainty. AE can hardly extract the intrinsic features of these uncertain data, which leads to poor classification accuracy, thus reduces the robustness of DNN. In this paper, DAE is used to feature extraction of noise data. DAE can extract the potential features of noise data, which is beneficial to the subsequent fault diagnosis and improve the robustness of DNN.

3.1 DAE Training at Time-Varying Learning Rate

For DAE, we assume that $X = \{x_{ij}\}, (i = 1,2,...,N; j = 1,2,...,M)$ is the unlabeled data set, which contains M samples, each sample contains N observation variables. Data set X introduces a certain proportion of noise to get data X'. The encoding process and the decoding process of the DAE are shown in (4) and (5). And the encoding network parameters and decoding network parameters were denoted as $\theta_1 = \{W, b\}$ and $\theta_1^T = \{\bar{W}, d\}$. The hyperbolic tangent function is utilized for the activation function σ_1 and σ_2. The function can be described as follows:

$$f(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (6)$$

To make the output Y as close as possible with the original noiseless data X, we need to optimize the parameters θ_1 and θ_2. Formula (3) is used to describe the degree of approximation between X and Y, by minimizing formula (3) to implement parameter optimization. In training process, gradient descent method is used for DAE parameter optimization, the concrete update form of network parameters can be formulated as formula (7) - (8).

$$W_l = W_l - \alpha \frac{\partial}{\partial W_l} L(X,Y), l = 1,2 \quad (7)$$

$$b_l = b_l - \alpha \frac{\partial}{\partial b_l} L(X,Y), l = 1,2 \quad (8)$$

Where $\frac{\partial}{\partial W_l} L(X,Y)$ and $\frac{\partial}{\partial b_l} L(X,Y)$ can be calculated by back propagation algorithm, α is the learning rate, which determines the step size of each iteration. The learning rate of the traditional DAE is a fixed value. There is no strict theory for the selection of learning rate. In practical application, it is usually determined by experience. If α is selected a too large value, it may appear oscillation phenomenon, and even the algorithm can

not converge. If the learning rate is too small, the algorithm will converge slowly and spend too much time. In this paper, we use formula (9) to express the time varying learning rate.

$$\alpha = \alpha(i) \quad (9)$$

Where i is the number of iterations. Formula (9) is substituted into formula (7) and (8), and the updated parameters can be obtained via equation (10) and equation (11).

$$W_l = W_l - \alpha(i)\frac{\partial}{\partial W_l}L(X,Y), l = 1,2 \quad (10)$$

$$b_l = b_l - \alpha(i)\frac{\partial}{\partial b_l}L(X,Y), l = 1,2 \quad (11)$$

Where $\alpha(i)$ is a function of the number of iterations, which can be reduced as the number of iterations increases. It can make sure that the learning rate is in a relatively large range in the early iteration so as to maintain a certain convergence rate. The learning rate becomes smaller at the end of the iteration. It can be seen from formula (10) and (11) that if $\alpha(i)$ is a smaller value, the parameters is updated at a relatively slow rate. So it is more conducive to reach the optimal value, and also can avoid the oscillation phenomenon near the optimal value. The specific choice of $\alpha(i)$ will be given in the subsequent chapters.

Through the above process, the network parameters θ_1 and θ_1^T of the first layer of DAE can be obtained. The input data is obtained through the following formula when training the second layer DAE.

$$H_1 = f_\theta(X) = \sigma_1(WX + b) \quad (12)$$

Where $W = W_1, b = b_1$. The noiseless data X is used to compute the feature H_1 extracted by the first hidden layer. The input $\bar{H}_1$ of the second layer DAE is obtained by adding the same proportion of noise to H. The same training method is used to train the second DAE.

In order to achieve the intention of fault classification, we need to add a classifier. In this paper, we use the Softmax classifier as the output layer of DNN. We use feature H_j of the last hidden layer (j is the number of hidden layers) and the labeled data $y_m \in \{1,2,...,k\}$ to train the Softmax classifier. For a given input x_i, the probability $p(x_i = y_m \mid x_m)$ which characterize the degree of membership of each category can be calculated via the following hypothesis function:

$$h_\theta(x_i) = \begin{bmatrix} p(y_i = 1 \mid x_i;\theta) \\ p(y_i = 2 \mid x_i;\theta) \\ \vdots \\ p(y_i = k \mid x_i;\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x_i}} \begin{bmatrix} e^{\theta_1^T x_i} \\ e^{\theta_2^T x_i} \\ \vdots \\ e^{\theta_k^T x_i} \end{bmatrix} \quad (13)$$

$$label = \arg\max_{j=1,2\cdots,k}\{p(y_i = j \mid x_i;\theta),\} \quad (14)$$

We can use formula (14) to determine the classification label of x_i. Where, θ is the model parameter of Softmax. The model parameters can also be optimized by the same means to minimize the cost function. The cost function of Softmax classifier can be defined as follows:

$$J(\theta) = -\frac{1}{M}[\sum_{i=1}^{M}\sum_{j=1}^{k} 1\{y_i = j\}\log\frac{e^{\theta_j^T x_i}}{\sum_{m=1}^{k} e^{\theta_m^T x_i}}] \quad (15)$$

Where $1\{\bullet\}$ is indicate function. Finally, we fine-tune the parameters [21]. Softmax cost function is used as the cost function of the deep neural network and use the back propagation algorithm to update the parameters. The fine-tuning process uses the labeled data to improve the performance of DNN.

3.2 Improved DAE Based Fault Diagnosis

In this section, the improved DAE (IDAE) proposed in section 3.1 is used in fault diagnosis. The main step of IDAE based fault diagnosis method follows 7 steps:

Step1 The data sample X' polluted with a certain proportion of noise is used as the input of the first layer DAE to obtain the parameters W_1 and b_1;

Step2 Use the original input X and parameters W_1 and b_1 of the first layer to calculate the output Y_1 of the first hidden layer according to the following formula;

$$Y_1 = f(X) = \sigma(W_1 X + b_1) \quad (16)$$

Step3 Add the same proportion of noise to Y_1 as X', and obtain the input data y_1 of the second layer of DAE. The same method is used to train the second layer DAE to obtain the network parameters W_2 and b_2;

Step4 The output Y_2 of the second hidden layer is calculated via equation (17) using Y_1 and the network parameters W_2 and b_2 of the second hidden layer;

$$Y_2 = f(Y_1) = \sigma(W_2 Y_1 + b_2) \quad (17)$$

Step5 Train the Softmax classifier with the output Y_2 of the second hidden layer and the labeled data to obtain the network parameters W_3 and b_3 of the Softmax layer;

Step6 Use the network parameters obtained in the previous steps as initialization parameters of the deep network. Use the original labeled data X as the input of DNN to fine-tune parameters of the network by appropriate optimization algorithms;

Step7 Once test data is collected, it is used as the input of the well trained DNN classifier to get the fault classification result.

4. EXPERIMENTAL RESULTS AND ANALYSIS

In this section the convergence and the feasibility of the proposed method will be proved by an experiment of bearing fault diagnosis.

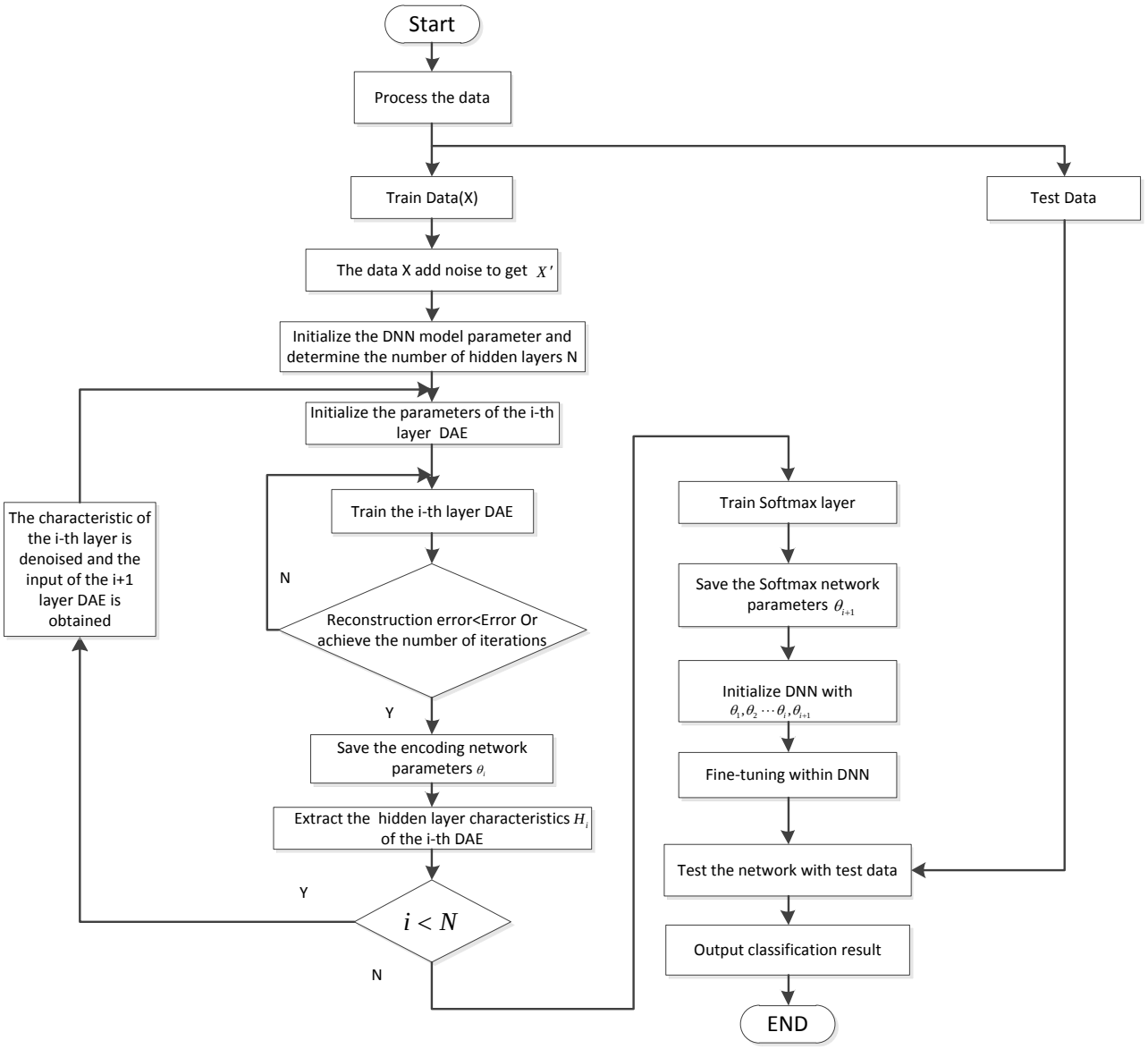

Fig.3 Flow chart of algorithm based on DAE

4.1 Data Description

The experimental data set used in this paper is collected by Case Western Reserve University Bearing Data Center [21], the experimental platform shown in Fig.4. Acceleration sensor is used in the experiment to collect the vibration signal of motor drive end as experimental data of bearing fault diagnosis. In this experiment, the acceleration sensor is used to collect the motor-driven vibration signal of 0hp, the sampling frequency is 48kHz. There are four types of bearing health: (1) inner ring fault; (2) outer ring fault; (3) roller fault; (4) normal condition. Fourier transform is used to preprocess the vibrations signal to get 8000 training samples and 8000 test samples.

4.2 Convergence Analysis of Time-Varying Learning Rate Algorithm

In this section convergence of the proposed DNN algorithm is analyzed. In the experiments, we first use the traditional method and choose a fixed value as the learning rate of DAE. Fig.5 draws a graph of the cost function that varies with the number of iterations in different learning rates. In order to eliminate the influence of randomness, each learning rate has been conducted 20 experiments. The cost function of each experiment is recorded to get the average cost function value of 20 experiments.

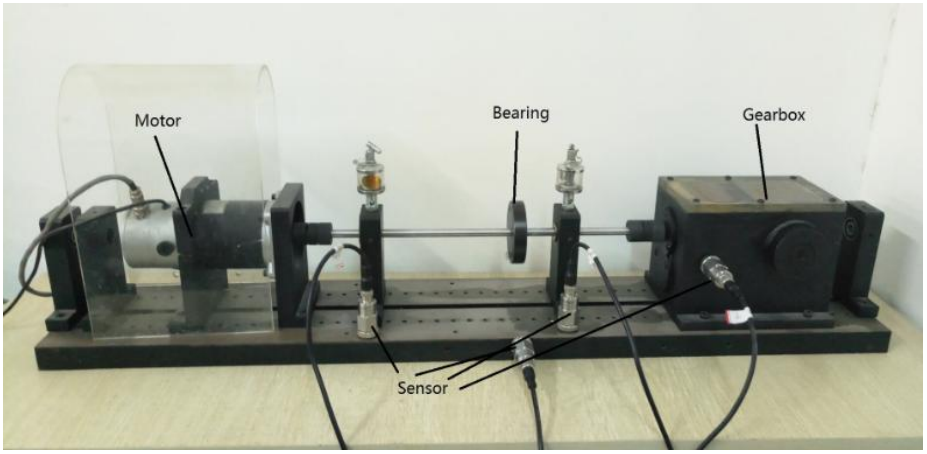

Fig.4 Experiment platform of rolling bearing

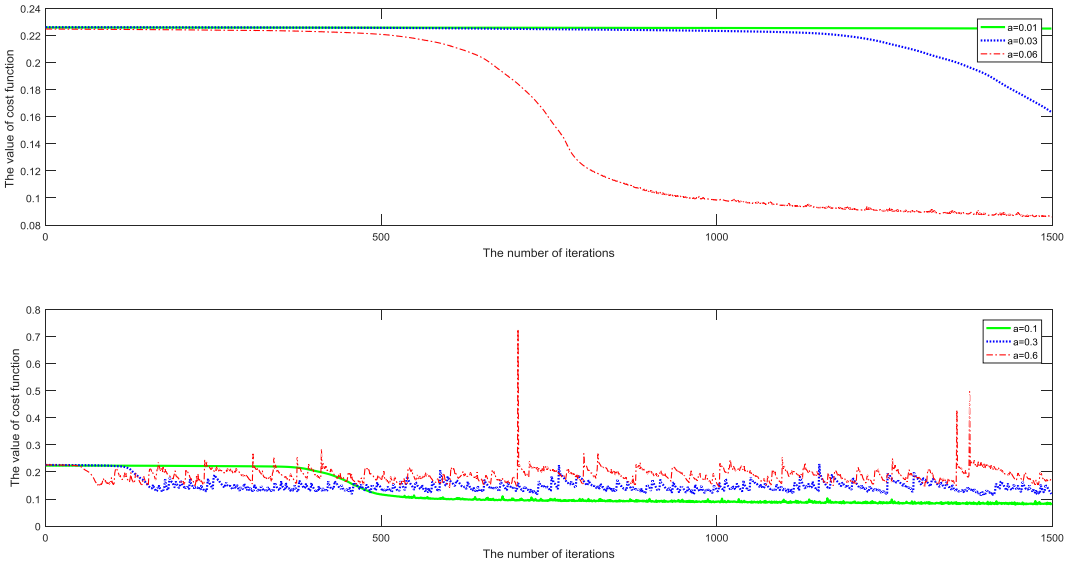

Fig.5 Diagram of the cost function in the case of fixed learning

The curves in Fig.5 show the learning rates of 0.01, 0.03, 0.06, 0.1, 0.3 and 0.6, respectively. Fig.5 shows that as the learning rate increases, the cost function converges faster. However, with the increase of learning rate, severe concussion occurs near the minimum value. When oscillation occurs, the cost function will fluctuate around the minimum value. When the preset number of iterations is reached, there is a great probability that the cost function does not reach the minimum and the obtained parameters are not optimal.

Usually, we will set a small threshold δ, when the error e meet the condition $e<\delta$ quit the iteration loop. However, this condition is often not satisfied in practice. We try to find a learning rate that can maintain a certain rate of iteration while avoiding concussion. Through a lot of experiments, the following formulas is screened out.

$$\alpha(i)=\frac{1}{\sqrt{i}} \tag{18}$$

$$\alpha(i)=\frac{1}{i} \tag{19}$$

$$\alpha(i)=\frac{2}{\sqrt{i}}-\frac{1}{i} \tag{20}$$

Where $\alpha(i)$ is the time varying learning rate, i is the number of iterations. Equations (18) - (20) are relatively ideal learning rate functions in the sense that they both can guarantee that the learning rate is in the range of (0,1).

Cost function varying with the number of iterations is plotted in Fig.6, each line corresponds to a different time varying learning rate. The green line, blue line and red line indicate the learning rates $\alpha(i)=\frac{1}{i}$, $\alpha(i)=\frac{1}{\sqrt{i}}$ and $\alpha(i)=\frac{2}{\sqrt{i}}-\frac{1}{i}$, respectively. The curves are also drawn by averaging of 20 experiments.

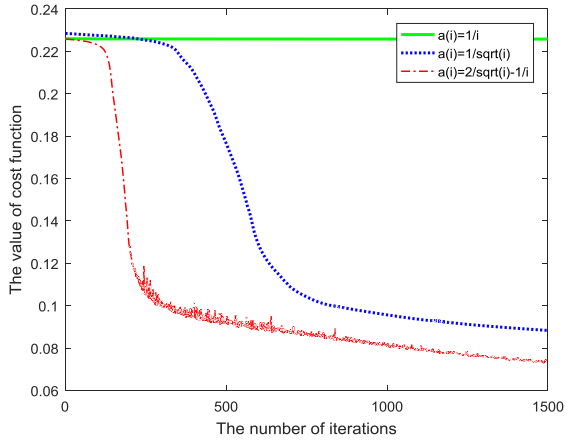

Fig.6 Diagram of the cost function in the case of time-varying learning

According to Fig.6, when the learning rate in formula (19) was used, iteration process is slow. Furthermore, when the learning rate in (20) is used, oscillations occur during the iteration. The learning rates of formula (18) can not only maintain a certain convergence speed, but also satisfy the convergence requirements of the algorithm. So, in the following DNN fault diagnosis experiment, we choose $\alpha(i)=\frac{1}{\sqrt{i}}$ as the learning rate.

4.3 The Effectiveness Analysis of DAE in Fault Diagnosis Based on Time-Varying Learning Rate

In Section 4.2, we analyze the convergence of the proposed algorithm. And $\alpha=0.06$ and $\alpha(i)=\frac{1}{\sqrt{i}}$ are chosen as the fixed learning rate and time-varying learning rate, respectively.

First, additional noise is added to the experimental data. The noise adding method is the same as that mentioned by Vincent et al. [22]. For training samples, the proportion of noise is 0%,20%, 30% and 40% , respectively. For the test data, the proportion of noise is 0%, 20%, 30%, 40%, 50%, 60% and 70% of the noise respectively.

In order to eliminate the randomness in the experiment, average classification accuracy of 20 experiments is used as the final classification accuracy. Fault classification accuracy is defined as following:

$$p=\frac{n}{m}\times 100\% \qquad (21)$$

Where n is the number of samples correctly classified, m is the total number of samples, p is the accuracy of fault diagnosis.

Table 1 compares the classification accuracy under different conditions when learning rates are $\alpha=0.06$ and $\alpha(i)=\frac{1}{\sqrt{i}}$, respectively. Fig.7 shows the fault classification result when the learning rates are $\alpha=0.06$ and $\alpha(i)=\frac{1}{\sqrt{i}}$, respectively. In Fig.7, the blue line and red line correspond to the learning rates $\alpha=0.06$ and $\alpha(i)=\frac{1}{\sqrt{i}}$, respectively. The four sub-graphs in Fig. 7 represent the ratios of adding noise to training data are 0%, 20%, 30% and 40% respectively. It can be seen from Table 1 that the performance with time-varying learning rate is better than that of fixed learning rate $\alpha=0.06$.

When $\alpha(i)=\frac{1}{\sqrt{i}}$ and the training data adds 30% of the noise, the performance of the trained model can also get better fault classification result. It can be easily seen from Fig.7 that the fault classification accuracy of the time-varying learning rate is higher than that of the fixed learning rate in the case when noise ratio of the training samples and the testing samples are same. Simultaneously, Table 1 tells us that when 70% noise is added in the test data, classification accuracy of DNN with time-varying learning rate can reach the best fault classification accuracy 87.28%, while DNN with fixed learning rate can only reach a accuracy of 85.06% .

From above all, it can be concluded that DNN model with time-varying learning rate is better than DNN with constant learning rate in the sense of model robustness.

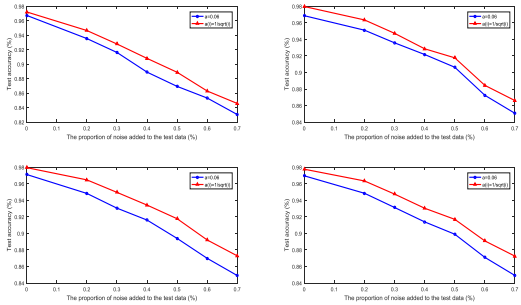

Fig.7 The classification accuracy of different noise ratio

Fig.8 and Fig.9 illustrate the classification results of different test samples when the noise ratio of the training samples is 30%. Fig.8 is the fault classification results of DNN with fixed learning rate. And Fig.9 is the fault classification results of DNN with time-varying learning rate. In Fig.8 and Fig.9, the small blue circle represents the label, and the red asterisk represents the model output category. And The six sub-graphs in Fig. 8 and Fig.9 represent the noise ratios added to test data 0%, 20%, 30%, 40%, 50% and 70%, respectively. Comparing the classification results of Fig.8 and Fig.9, the classification results in Fig.9 are better than the classification results in Fig.8, which are mutually proofed with the accuracy in Table1.In conclusion, in the field of fault diagnosis, features extracted by DAE with time-varying learning rate DAE are more superior to DAE with fixed learning rate.

Table 1 The fault classification accuracy of DNN with different learning rate

Train / Test	Sample noise ratio:20%		Sample noise ratio:30%		Sample noise ratio:40%		Sample noise ratio:0%	
Learning rate	0.06	$1/\sqrt{i}$	0.06	$1/\sqrt{i}$	0.06	$1/\sqrt{i}$	0.06	$1/\sqrt{i}$
Noise ratio:0%	96.86%	97.97%	96.89%	97.95%	96.96%	97.76%	96.74%	97.22%
Noise ratio:20%	95.11%	96.35%	94.83%	96.47%	94.85%	96.34%	93.54%	94.65%
Noise ratio:30%	93.58%	94.73%	93.05%	94.96%	93.13%	94.75%	91.62%	92.79%
Noise ratio:40%	92.17%	92.86%	91.61%	93.42%	91.39%	93.04%	88.90%	90.80%
Noise ratio:50%	90.63%	91.69%	89.39%	91.77%	89.90%	91.70%	86.93%	88.88%
Noise ratio:60%	87.24%	88.44%	86.98%	89.20%	87.10%	89.09%	85.32%	86.29%
Noise ratio:70%	85.06%	86.60%	84.90%	87.28%	84.91%	87.25%	83.04%	84.57%

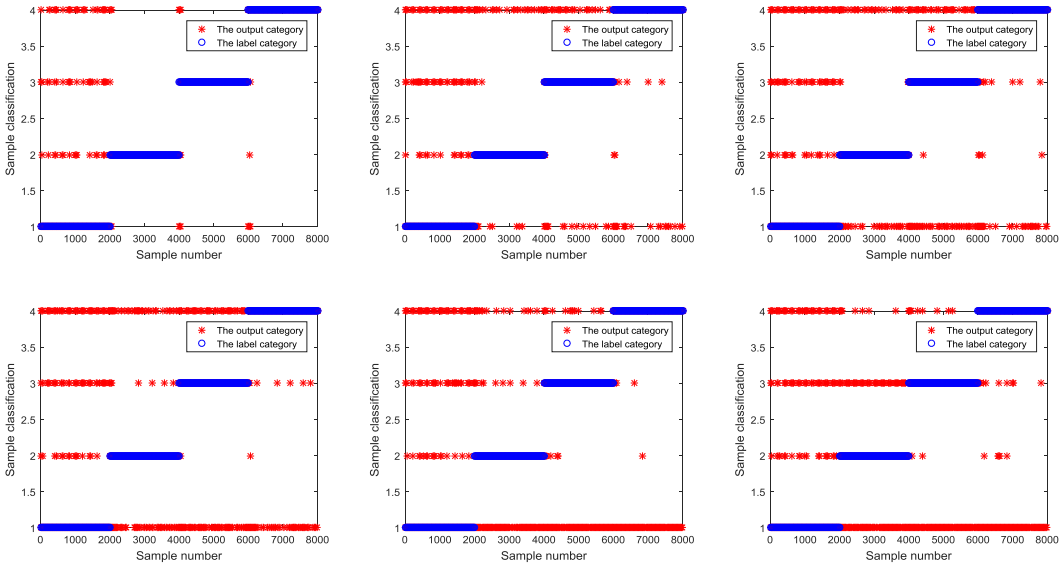

Fig.8 Fault classification results of DNN with fixed learning rate

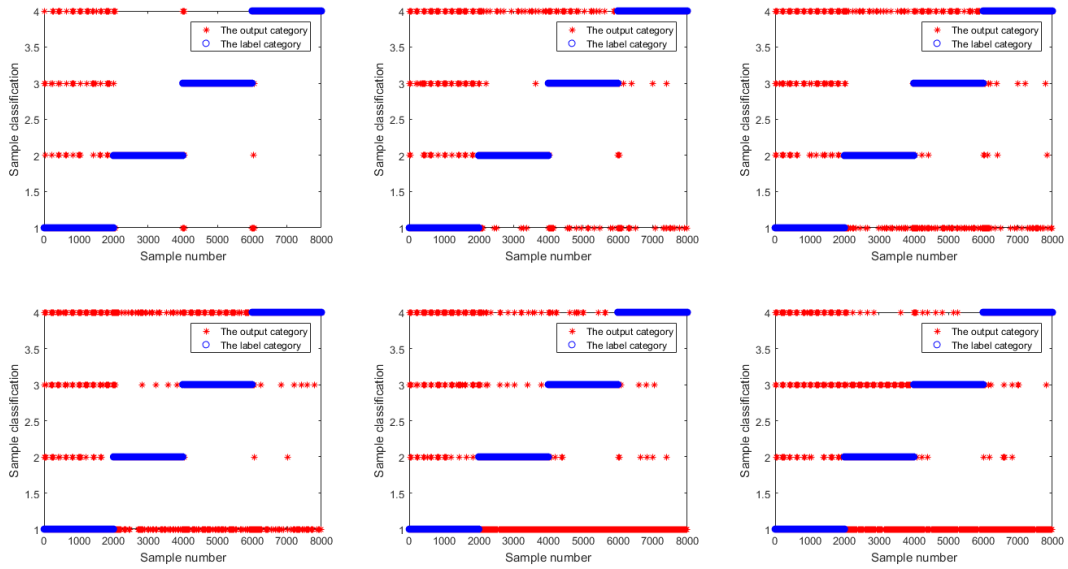

Fig.9 Fault classification results of DNN with time-varying learning rate

5. CONCLUSION

In this paper, we propose a fault diagnosis method of DNN with time-varying learning rate and the DNN is constructed by stacked DAE. The main idea is to construct a learning rate function, which decreases with the number of iterations. This mechanism not only guarantees the convergence rate of iteration, but also smoothly approximates the minimum value of cost function. Experimental verification shows the efficiency of the improved method proposed in this paper.

REFERENCES

[1] Alavi S M M, Izadi-Zamanabadi R, Hayes M J. Robust fault detection and isolation technique for single-input/single-output closed-loop control systems that exhibit actuator and sensor faults[J]. Iet Control Theory & Applications, 2008, 2(11):951-965.

[2] Touati Y, Merzouki R, Bouamama B O. Fault Detection and Isolation in Presence of Input and Output Uncertainties Using Bond Graph Approach[J]. Imaaca, 2012.

[3] Patan K, Witczak M. Towards Robustness in Neural Network Based Fault Diagnosis[M]. Versita, 2008.

[4] Li H, Xiao D Y. Survey on data driven fault diagnosis methods[J]. Control & Decision, 2011, 26(1):1-9+16.

[5] R. M. An and Y. Gao, "Spacecraft fault classification based on hierarchical neural network," Spacecraft environment engineering, vol. 30, no. 2, pp. 203-208, 2013.

[6] F. N. Zhou , C. L. Wen, Y. B. Leng and Z. G. Chen, "A data-driven fault propagation analysis method", Journal of Chemical Industry and Engineering(China), vol. 61, no. 8, pp. 1993-2001,2010.

[7] H. Q. Ji, X. He and D. H. Zhou, "On the use of reconstruction-based contribution for fault classification, "Journal of Process Control, vol. 40, pp. 24-34, 2016.

[8] D. J. Yu, M. F. Chen, J. S. Cheng and Y. Yang, "A fault classification approach for rotor systems based on empirical mode decomposition method and support vector machines, " Proceedings of the Chinese society for electrical engineering, vol. 26,no. 16, pp. 162-167, 2006.

[9] M. Gan, C. Wang and C. A. Zhu, "Construction of hierarchical classification network based on deep learning and its application in the fault pattern recognition of rolling element bearings" Mechanical Systems and Signal Processing, vol.72-73, pp. 92–104, 2016.

[10] Kahraman F, Capar A, Ayvaci A, et al. Comparison of SVM and ANN performance for handwritten character classification[C]// Signal Processing and Communications Applications Conference, 2004. Proceedings of the IEEE. IEEE, 2004:615-618.

[11] T. Kuremoto, S. Kimura, K. Kobayashi and M. Obayashi, "Time series forecasting using a deep belief network with restricted Boltzmann machines," Neurocomputing, vol. 137, pp. 47–56, 2014.

[12] Wen L, Li X, Gao L, et al. A New Convolutional Neural Network Based Data-Driven Fault Diagnosis Method[J]. IEEE Transactions on Industrial Electronics, 2017, PP(99):1-1. G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," Science, vol.313, pp. 504-507, 2006.

[13] P. L. Wang , C. J. Xia, "Fault detection and self-learning identification based on PCA-PDBNs," Chinese Journal of Scientific Instrument, vol.36, no. 5, pp.1147-1154, 2015.

[14] Jia F, Lei Y, Guo L, et al. A neural network constructed by deep learning technique and its application to intelligent fault diagnosis of machines[J]. Neurocomputing, 2017.

[15] Wen L, Li X, Gao L, et al. A New Convolutional Neural Network Based Data-Driven Fault Diagnosis Method[J]. IEEE Transactions on Industrial Electronics, 2017, PP(99):1-1.

[16] Zhao W, Lu C, Ma J, et al. A deep learning method using SDA combined with dropout for bearing fault diagnosis[J]. 2015.

[17] Zhao G, Zhang G, Ge Q, et al. Research advances in fault diagnosis and prognostic based on deep learning[C]// Prognostics and System Health Management Conference. IEEE, 2017:1-6.

[18] R. Pang, Z. B. Yu, W. Y. Xiong and H. Li , "Faults recognition of high -speed train bogie based on deep learning ," Journal of Railway Science and Engineering, vol.12, no.6, pp. 1283-1288, 2015.

[19] Tan CC，Eswaran C．Reconstruction and recognition of face and digit images using autoencoders [J]. Neural Computing and Applications ,2010,19 (7)： 1069-1079．

[20] Ngiam J, Coates A , Lahiri A, et a1．On optimization methods for deep learning [C] //28th International Conference on Machine Learning, 2011:265-272 .

[21] Bearing data Centre, Case Western Reserve University, Available: http://csegroups.case.edu/bearingdatacenter/home

[22] Vincent, Pascal, Larochelle, Hugo, Lajoie, Isabelle, et al. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion.[J]. Journal of Machine Learning Research, 2010, 11(12):3371-3408.