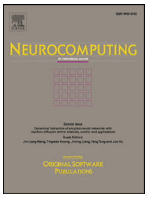
Neurocomputing 363 (2019) 88–98

Contents lists available at ScienceDirect

Neurocomputing

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



ReLTanh: An activation function with vanishing gradient resistance for SAE-based DNNs and its application to rotating machinery fault diagnosis

Xin Wang[a], Yi Qin[a,*], Yi Wang[a], Sheng Xiang[a], Haizhou Chen[b]

[a] *State Key Laboratory of Mechanical Transmission, Chongqing University, Chongqing 400044, People's Republic of China*
[b] *College of Electromechanical Engineering, Qingdao University of Science and Technology, Laoshan District, Qingdao 266061, People's Republic of China*

ARTICLE INFO

Article history:
Received 3 December 2018
Revised 7 May 2019
Accepted 5 July 2019
Available online 19 July 2019

Keywords:
Rectified Linear Tanh
Stacked autoencoder-based deep neural network
Vanishing gradient problem
Planetary gearbox faults
Rolling bearing faults

ABSTRACT

Tanh is a sigmoidal activation function that suffers from vanishing gradient problem, so researchers have proposed some alternative functions including *rectified linear unit* (*ReLU*), however those vanishing-proof functions bring some other problem such as bias shift problem and noise-sensitiveness as well. Mainly for overcoming vanishing gradient problem as well as avoiding to introduce other problems, we propose a new activation function named *Rectified Linear Tanh* (*ReLTanh*) by improving traditional *Tanh. ReLTanh* is constructed by replacing *Tanh*'s saturated waveforms in positive and negative inactive regions with two straight lines, and the slopes of the lines are calculated by the *Tanh*'s derivatives at two learnable thresholds. The middle *Tanh* waveform provides *ReLTanh* with the ability of nonlinear fitting, and the linear parts contribute to the relief of vanishing gradient problem. Besides, thresholds of *ReLTanh* that determines the slopes of line parts are learnable, so it can tolerate the variation of inputs and help to minimize the cost function and maximize the data fitting performance. Theoretical proofs by mathematical derivations demonstrate that *ReLTanh* is available to diminish vanishing gradient problem and feasible to train thresholds. For verifying the practical feasibility and effectiveness of *ReLTanh*, fault diagnosis experiments for planetary gearboxes and rolling bearings are conducted by stacked autoencoder-based deep neural network (SAE-based DNNs). *ReLTanh* alleviates successfully vanishing gradient problem and the it learns faster, more steadily and precisely than *Tanh*, which is consistent with the theoretical analysis. Additionally, *ReLTanh* surpasses other popular activation functions such as *ReLU* family, *Hexpo* and *Swish*, which shows that *ReLTanh* has certain applying potential and researching value.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

1.1. Background of diagnosis

With the increasing complexity, rotating machinery fault diagnosis play a more and more important role for the reliability and safety of modern industrial systems [1, 2]. Rotating components such as planetary gears and rolling bearings always have high fault probabilities, because of their complex structures and harsh operating conditions [3, 4]. To make matters worse, if the early minor failures are not detected and maintained timely, they will deteriorate rapidly and lead to a serious halt of the whole power transmission chain, and even catastrophic economic losses and casualties [5, 6]. Therefore, in order to prevent the faults from deteriorating, it is necessary to diagnose them as early as possible.

The researches on fault detection of rotating components have attached a lot of attention, but these tasks are still challenging [7, 8]. Traditional diagnosis methods are based on vibration signal analysis techniques such as short time Fourier transform, Wigner-Ville distribution, wavelet transform [9], but they have tedious procedures and relative unsatisfactory performance. Therefore, various intelligent and automatic methods based on deep learning models including deep neural networks (DNNs) have been proposed to simplify processes and improve accuracies [10]. For example, Jia et al. [11] compared the performance of DNNs and artificial neural networks (ANNs) on fault diagnosis for planetary gearbox. Xu et al. [12] applied a sparse autoencoder-based deep neural network on open-circuit fault diagnosis. Lu et al. [13] proposed a feature extraction method based on DNN for rolling bearing fault diagnosis.

[*] Corresponding author.
E-mail address: qy_808@aliyun.com (Y. Qin).

https://doi.org/10.1016/j.neucom.2019.07.017
0925-2312/© 2019 Elsevier B.V. All rights reserved.

1.2. Vanishing gradient problem and existing solutions

Tanh is a typical sigmoidal activation function, and it is popular in shallow networks such as ANNs [14], because it outputs zero-centered non-linear activations. However, *Tanh* is abandoned when it comes to deep models due to the vanishing gradient problem, thus we propose an improving activation function *ReLTanh* based on *Tanh* to retard this problem.

Tanh can squash large-scale inputs into an interval of [−1, 1] and provide non-linear and noise-robust representation. But meanwhile, the saturation characteristic also vanishes gradients. Once the inputs fall into the saturation regions, they get relative small gradients close to zero and slow down the updating of weights and biases [15]. Even worse, the gradients decrease exponentially as the depth of the network increases, because gradient computation in back-propagation (BP) process is based on the chain rule and all layers are interconnected and interlocked [16]. Sometimes, the neurons in the lower layers of a multi-layer network can hardly be updated and even die, which blocks DNNs from deepening further [17]. Due to vanishing gradient problem, it generally takes more computational power to train, and DNNs are more likely to converge to a local minimum.

In order to alleviate vanishing gradient problem, a lot of methods have been developed in recent years, such as layer-wise pre-training algorithm [18], *rectified linear unit* (*ReLU*) family [19]. Unsupervised pre-training is beneficial to diminish the vanishing gradient problem by providing a better weight initialization for deep models. For example, a stacked autoencoder-based DNN (SAE-based DNN) can pre-train autoencoders to preview the data, so it will be applied for diagnosis in this study [20]. *ReLU* family can overcome vanishing gradient problem by their straight lines with the fixed slope of 1 in the positive interval [21]. However, straight lines are a double-edged sword, it provides noise-sensitive representation for all layers and affect the convergence of learning [22]. Besides, it brings a certain degree of bias shift problem as well. For example, *ReLU* is identical for positive inputs and zero otherwise, so it has a non-negative mean activation output [25]. According to Refs. [23, 28], units with a non-zero mean activation output can cause bias shift problem for the next layer, and bias shift leads to oscillations and impede learning. Even worse, bias shift may aggravate with the depth of models increase, just like the vanishing gradient problem.

1.3. The proposed ReLTanh

In order to diminish vanishing gradient problem that perplexes *Tanh* and reduce bias shift and noise-sensitiveness that torments *ReLU* family, we propose a new activation function: *ReLTanh*. *ReLTanh* is created by replacing the saturated waveforms with two straight lines, so *ReLTanh* consists of the nonlinear *Tanh* in the center and two linear parts on both ends. The positive line is steeper than the negative one, and both lines start at two learnable thresholds, and their slopes are *Tanh*'s derivative values at these thresholds. These thresholds can be trained and updated along the gradient descent direction of cost function. So *ReLTanh* can improve vanishing gradient problem just like *ReLU* family, and its mean outputs are closer to zero so that it is affected less bias shift than *ReLU* family. Additionally, complete mathematical derivations are performed to verify theoretically that *ReLTanh* DNNs are effective for weakening vanishing gradient problem and feasible to training the slopes.

For further validation of the practical feasibility for *ReLTanh*, the *ReLTanh* SAE-based DNNs are employed in fault diagnosis experiments for planetary gearboxes and rolling bearings. At first, for planetary gearboxes, vibration signals are collected by ourselves from a professional test rig. Then according to Fig. 1, statistical feature extraction, vector sample construction and fault recognition are performed in sequence. The results demonstrate that *ReLTanh* relaxes vanishing gradient problem successfully and surpasses *Tanh* on multi-aspects. Even more exciting, *ReLTanh* can provide higher accuracies than popular activation functions such as *ReLU* family, *Hexpo* and *Swish*. Next, Case Western Reserve University's motor rolling bearing dataset that is internationally recognized is employed to train and test *ReLTanh* DNNs again, and similar results are obtained. These two experiments provide results and conclusions that are in line with the theoretical analysis, and they prove in theory and practice the potential of application and development.

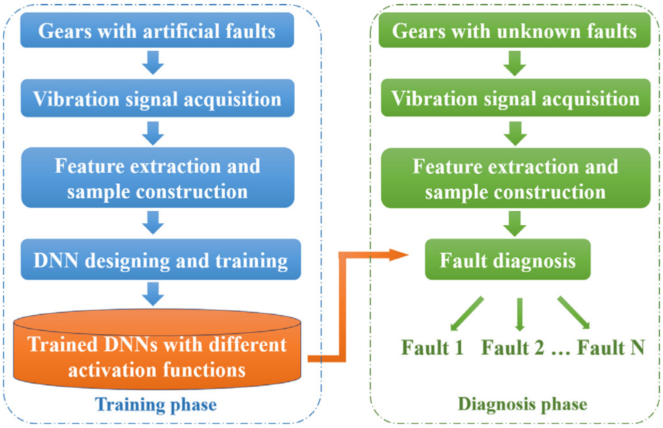

Gears with artificial faults
Vibration signal acquisition
Feature extraction and sample construction
DNN designing and training
Trained DNNs with different activation functions
Training phase
Gears with unknown faults
Vibration signal acquisition
Feature extraction and sample construction
Fault diagnosis
Fault 1
Fault 2 … Fault N
Diagnosis phase

Fig. 1. Flow diagram of the intelligent diagnosis approach based on SAE-based DNNs.

The remainder of paper is organized as follows. In Section 2, the detailed introduce of vanishing gradient problem and popular activation functions, and the architectures and learning rules of SAE-based DNNs are briefly presented. The definition of *ReLTanh* and relevant mathematical derivations are described in Section 3. Section 4 introduces the application of the *ReLTanh* DNNs on fault diagnosis for planetary gearboxes according to Fig. 1. Section 5 applies the *ReLTanh* on rolling bearing fault diagnosis. Finally, some conclusions are addressed in Section 6.

2. Related work

2.1. Vanishing gradient problem

The main purpose of this study for *ReLTanh* is to weaken the vanishing gradient problem. For supervised learning models such as deep neural networks (DNNs), the ultimate training goal during BP process is to fit the labeled data and find the global minimum of the cost function by gradient descent methods. The partial gradients of the cost function with respect to the weights are a key multiplier in the updating formula. But by chain rule, they cause serious decay for updating values when several gradients less than 1 are cumproded with each other. It is known as vanishing gradient problem, which is a key drawback that limits the depth of networks.

The BP process of a DNN with N hidden layers is taken as an example to illustrate vanishing gradient problem in detail, which is shown vividly in Fig. 2.

In this study, the top classifier f_2 is *Softmax*, and f_1 is the activation function of hidden layers, and the cost function C is cross-entropy, which is given by [24]:

$$\mathrm{C} = -\sum_{j}^{m} t_j \ln\left(a_{N+1,j}\right) \tag{1}$$

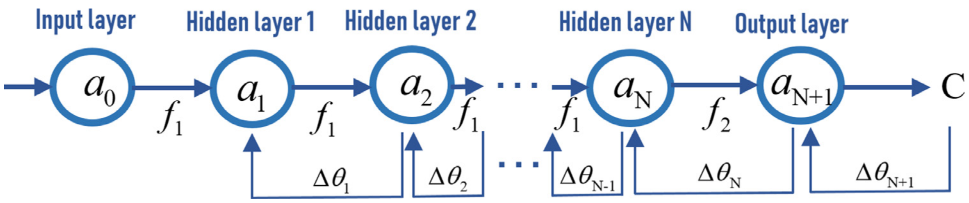


Fig. 2. The DNN BP process (where the $\mathbf{a}_i$ is the output of the layer i ($i = 0, 1, \ldots, N, N+1$), and parameters $\Delta\boldsymbol{\theta}(\Delta\mathbf{w}, \Delta\mathbf{b})$ are the updating values of weights $\mathbf{w}$ and biases $\mathbf{b}$).

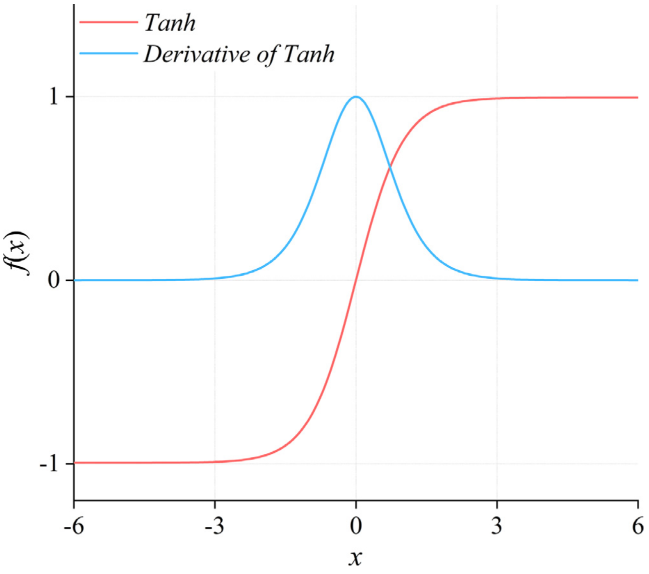


Fig. 3. *Tanh* and its derivative.

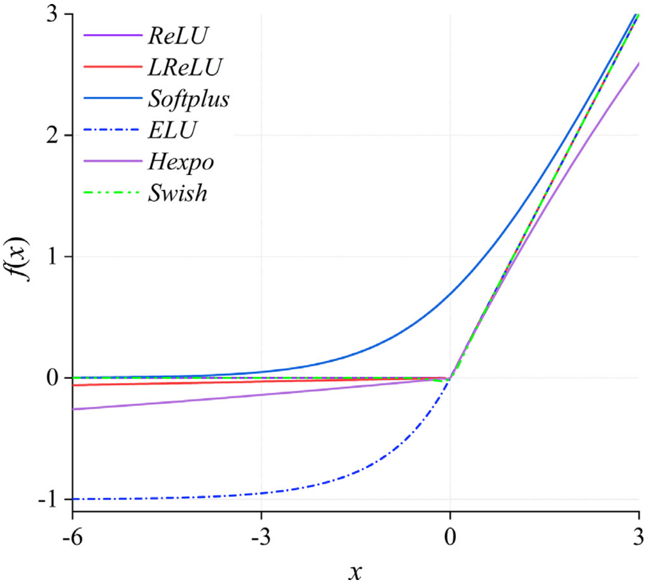


Fig. 4. Popular activation functions.

where t is the target output, m is the number of neuros in the output layer.

The updating value $\Delta\mathbf{w}_{N+1}$ of the weight $\mathbf{w}_{N+1}$ in the output layer can be calculated by chain rule.

$$\Delta\mathbf{w}_{N+1} = -\frac{\partial C}{\partial \mathbf{w}_{N+1}} = (\mathbf{a}_{N+1} - \mathbf{t})\mathbf{a}_N \tag{2}$$

Then based on back-propagation algorithm, the weights of the hidden layers can be updated by:

$$\Delta\mathbf{w}_N = -\eta\frac{\partial C}{\partial \mathbf{w}_N} = \eta(\mathbf{a}_{N+1} - \mathbf{t})\mathbf{w}_{N+1} f_1'(\mathbf{z}_N)\mathbf{a}_{N-1} \tag{3}$$

$$\Delta\mathbf{w}_1 = -\eta\frac{\partial C}{\partial \mathbf{w}_1} = \eta(\mathbf{a}_{N+1} - \mathbf{t})\mathbf{w}_{N+1} f_1'(\mathbf{z}_N)\ldots\mathbf{w}_2 f_1'(\mathbf{z}_1)\mathbf{a}_0 \tag{4}$$

where $\mathbf{z}_i = \mathbf{w}_i\mathbf{a}_i + \mathbf{b}_i$ is the input of the ith layer.

It follows from Eqs. (3) and (4) that the cumprod of $f_1'(\mathbf{z}_i)$ using chain rule will lead to exponential decrease for weights with the increase of the number of layers when the activation function has small derivative.

Tanh is a typical function that suffer from vanishing gradient problem because of the saturation regimes in positive and negative intervals. *Tanh* and its derivative are calculated by Eqs. (5) and (6), and their waveforms are shown in Fig. 3.

$$\mathrm{Tanh}(x) = \left(e^x - e^{-x}\right)\Big/\left(e^x + e^{-x}\right) \tag{5}$$

$$\mathrm{Tanh}'(x) = 4\Big/\left(e^x + e^{-x}\right)^2 \tag{6}$$

As can be seen in Fig. 3, the derivative of *Tanh* reaches the maximum of 1 when $x = 0$, but it is almost close to zero beyond the saturated interval of $[-3.5, 3.5]$. This inherent drawback will cause vanishing gradient problem. The update value $\Delta\mathbf{w}_i$ becomes smaller and smaller as the depth of the network increases, and back-propagation learning slows down exponentially. More seriously, if the network is very deep, the neurons in the lower layers may stop updating, which limits the classification performance. Similarly, the bias $\mathbf{b}$ is also impacted by vanishing gradient problem seriously.

Thus an activation function *ReLTanh* that has stronger gradient-vanishing-proof capacity is proposed in this paper by remolding the traditional *Tanh*.

2.2. Classical activation functions

To alleviate vanishing gradient problem, a number of activation functions have been proposed. The most common activation functions including *ReLU* family, *Hexpo* and *Swish* are analyzed and compared theoretically as follows, and their waveforms are illustrated in Fig. 4.

(1) *ReLU* [25]: $\mathrm{ReLU}(x) = \mathrm{Max}(0, x)$

ReLU provides a straight line with fixed slope of 1 in positive interval to avoid vanishing gradient problem. However, *ReLU* forbids the learning in negative interval, which limits data fitting performance. Besides, *ReLU* suffers from bias shift, i.e. *ReLU* has a non-zero output acting as bias for the next layer, and it will lead to oscillations and impede learning [28].

(2) *Leaky ReLU* (*LReLU*) [26]: $\mathrm{LReLU}(x) = \mathrm{Max}(\alpha x, x)$, where $\alpha = 0.01$

LReLU replaces the negative part of the *ReLU* with a linear function that has a fixed slope of 0.01, which enables a small amount of information to flow when $x < 0$. Additionally, compared to the *ReLU*, the average output of the *LReLU* is more close to zero, thus the *LReLU* is affected less bias shift problem.

(3) *Softplus* [27]: $\text{Softplus}(x) = \log(1 + \exp(x))$

Softplus can be regard as a smooth version of *ReLU*, so it also suffers bias shift.

(4) Exponential Linear Unit (ELU) [28]:

$$\text{ELU}(x) = \begin{cases} x & x \geq 0 \\ \alpha(exp(x) - 1) & x < 0 \end{cases} \tag{7}$$

The *negative* part of *ELU* is saturated, therefore it can tolerate the negative abnormal inputs and is more noise-robust, however it causes vanishing gradient in negative interval as well. The hyper parameter α can control the activation for negative net inputs, and according to Ref. [28], $\alpha = 1$ is used throughout this paper.

(5) *Swish* [29]: $\text{Swish}(x) = x\sigma(\beta x)$, where $\sigma(z) = (1 + exp(-x))^{-1}$, and $\beta = 10$.

Swish is a *smooth* function with properties similar to *Softplus*, but *Swish* is not strictly positive and monotonic.

(6) *Hexpo* [30]:

$$\text{Hexpo}(x) = \begin{cases} -a\left(\exp(-x/b) - 1\right)x \geq 0 \\ c\left(\exp(x/d) - 1\right)x \leq 0 \end{cases} \tag{8}$$

Hexpo has four presetting parameter including *a, b, c, d*, hence it can produce scalable limits on both positive and negative intervals, while *ReLU* family including *ReLU, LReLU, Softplus* and *ELU* provide identity mapping for positive inputs. According to Ref. [30], *Hexpo* can perform best with parameter combination: $a = 10$, $b = 10$, $c = 1$, $d = 20$, and these parameters will be applied in this paper.

2.3. SAE-based DNNs

In this study, we will conduct all diagnosis experiments using SAE-based DNNs.

A SAE-based DNN is constructed by several autoencoders (AEs) stacked with each other and a *Softmax* classifier on the output layer. Comparing to traditional DNNs, SAE-based DNNs need to pre-train the AEs by *Greedy Layer-Wise* Training algorithm [31] before the BP fine-tuning process, and the unsupervised pre-training is beneficial to alleviate the vanishing gradient problem by providing a better weight initialization for BP process. The pre-training and fine-tuning process are introduced as follows.

An AE is the basic component of SAE-based DNN, in this study we consider the simplest form that owns three layers: an input, a hidden and an output layer [13]. As depicted in Fig. 5, the structure of the AE is generally symmetrical, i.e., the output layer has the same number of nodes as the input layer for the main purpose of reconstructing its own inputs. The AE consists of two parts: encoder and decoder. During encoding process, the encoder network trains the recognition parameters (including weights and biases) to transform the high-dimension inputs into feature codes in the low-dimension hidden layer, and then during encoding process, the decoder network trains the generative parameters to reconstruct approximately the inputs from the corresponding feature codes [32]. After the training of AEs, the pre-trained weights and biases of hidden layer are applied to construct the SAE-based DNNs.

The encoder network can be explicitly defined as Eq. (9), given a unlabeled vector sample $\mathbf{x}$ [33, 34]:

$$\mathbf{h} = f(\mathbf{w}_1\mathbf{x} + \mathbf{b}_1) \tag{9}$$

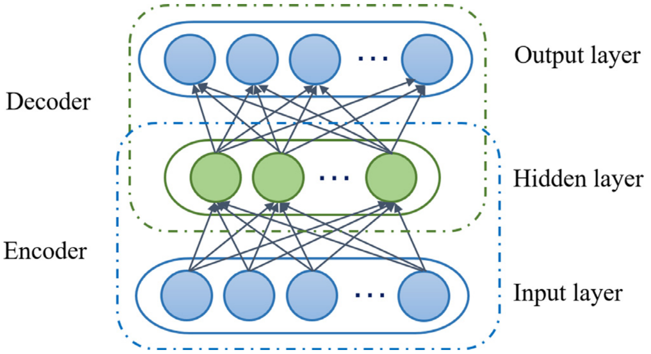


Fig. 5. The structure of an autoencoder neural network with one hidden layer.

in which $\mathbf{h}$ represent the feature codes in the hidden layer, and $\boldsymbol{\theta}_1 = \{\mathbf{w}_1, \mathbf{b}_1\}$ are the recognition parameters, and f is the encoder activation function.

Similarly, the decoder network can be defined as Eq. (10).

$$\hat{\mathbf{x}} = g(\mathbf{w}_2\mathbf{x} + \mathbf{b}_2) \tag{10}$$

where $\hat{\mathbf{x}}$ are the approximate reconstruction of the inputs, and $\boldsymbol{\theta}_2 = \{\mathbf{w}_2, \mathbf{b}_2\}$ are the reconstructing parameters, and g is the activation function of the decoder.

According to Ref. [35], the reconstruction error $Loss(\mathbf{x}, \hat{\mathbf{x}})$ between the inputs $\mathbf{x}$ and outputs $\hat{\mathbf{x}}$ are defined by Eq. (11), and the parameter set $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$ are trained simultaneously to incur the lowest reconstruction error.

$$Loss(\mathbf{x}, \hat{\mathbf{x}}) = \left[\frac{1}{m}\sum_{i}^{m}\left(\frac{1}{2}\parallel x_i - \hat{x}_i\|^2\right)\right] + \lambda T \tag{11}$$

where the first part is the mean square variance used to measure the average discrepancy and m is the number of neuros in the output layer, and second part λT is the regularization term used to prevent overfitting [35].

An AE is trained by gradient descent algorithm, and for example, the weight $\mathbf{w}_i$ of the ith layer can be updated by [13]:

$$\mathbf{w}_i = \mathbf{w}_i - \eta\frac{\partial Loss(\mathbf{x}, \hat{\mathbf{x}})}{\partial \mathbf{w}_i} \tag{12}$$

where η is the learning rate, and the partial derivatives terms $\frac{\partial Loss(\mathbf{x}, \hat{\mathbf{x}})}{\partial \mathbf{w}_i}$ can be calculated by chain rule. Similarly, biases $\mathbf{b}_i$can be updated by the above back-propagation algorithm.

During pre-training process, each AE is trained one after the other. After pre-training, it can be assumed that most information of the input samples is included in the code vectors of the hidden layer. Therefore, the code vectors can be used as inputs to train the following AE that is stacked on the trained one [34].

After pre-training, N AEs could be stacked together to construct a N-hidden-layer SAE-Based DNN [11]. For example, the structure of the SAE-based DNN that has two hidden layers is shown in Fig. 6.

Then based on the pre-trained parameters, the fine-tuning process can be performed by BP algorithm. Unfortunately, vanishing gradient problem introduced in Section 2.1 occurs during this process, thus *ReLTanh* is proposed and applied to overcome this problem.

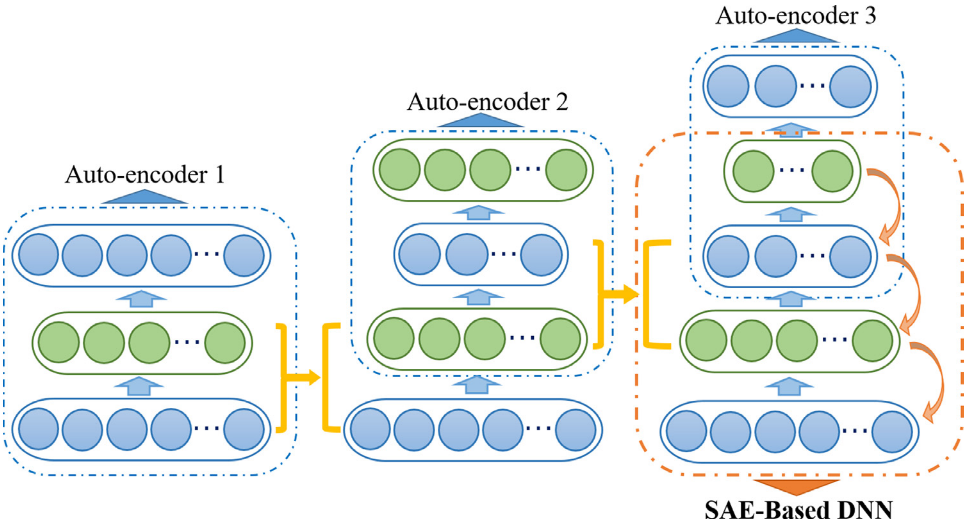


Fig. 6. The structure of a four-layer SAE-Based DNN.

3. ReLTanh

3.1. *Definition of* ReLTanh

The activation function *ReLTanh* is defined as follows.

$$\mathrm{ReLTanh}(x)=\begin{cases}\mathrm{Tanh}'(\lambda^+)(x-\lambda^+)+\mathrm{Tanh}(\lambda^+) & x\geq\lambda^+\\ \mathrm{Tanh}(x) & \lambda^-<x<\lambda^+\\ \mathrm{Tanh}'(\lambda^-)(x-\lambda^-)+\mathrm{Tanh}(\lambda^-) & x\leq\lambda^-\end{cases}$$
$$\text{where } \lambda^+_{lower}\leq\lambda^+\leq\lambda^+_{upper},\ \lambda^-_{lower}\leq\lambda^-\leq\lambda^-_{upper} \tag{13}$$

It is obvious that *ReLTanh* consists of a piece of nonlinear *Tanh* waveform in the center and two linear parts on both ends, and λ^+ and λ^- are respectively the positive and negative thresholds that determine the start positions and slopes of the straight lines. Besides, both λ^+ and λ^- can be trained by BP algorithm, and it is beneficial to decrease the cost function and search the global minimum.

It is worth noting there are extra limiting conditions for both λ^+ and λ^-: $\lambda^+_{lower}\leq\lambda^+\leq\lambda^+_{upper}$ and $\lambda^-_{lower}\leq\lambda^-\leq\lambda^-_{upper}$, and they are mainly used to constrain the learnable range of slopes to avoid unreasonable waveform and guarantee the gradient-vanishing-proof capacity. Empirically, $0\leq\lambda^+\leq0.5$ and $\lambda^-\leq-1.5$ are used throughout in this paper.

The derivative of *ReLTanh* with respect to the input x is mathematically given by Eq. (14), and the waveforms are illustrated in Fig. 7. It is obvious that no matter where the *ReLTanh* is in the red variable range, it can provide larger gradients than *Tanh*, so *ReLTanh* can retard vanishing gradient problem.

$$\mathrm{ReLTan}h'(x)=\begin{cases}\mathrm{Tanh}''(\lambda^+) & x\geq\lambda^+\\ \mathrm{Tanh}'(x) & \lambda^-<x<\lambda^+\\ \mathrm{Tanh}''(\lambda^-) & x\leq\lambda^-\end{cases} \tag{14}$$

3.2. ReLTanh *for vanishing gradient problem*

At first, the availability for weakening vanishing gradient problem is proven as follows.

The second derivative of *Tanh* can be calculated by:

$$\mathrm{Tanh}''(x)=8\left(e^{-2x}-e^{2x}\right)/\left(e^x+e^{-x}\right)^4 \tag{15}$$

It follows from Eq. (15) that $\mathrm{Tanh}''(x)>0$ when $x<0$, and $\mathrm{Tanh}''(x)<0$ when $x>0$. It is obvious that the first derivative $\mathrm{Tan}h'(x)$ monotonically increases in the interval $(-\infty,0)$, while it monotonically decreases in the interval $(0,+\infty)$. Consequently, given a positive real number κ^+ and a negative real number κ^-, $\mathrm{Tan}h'(x)$ has the flowing properties:

$$\mathrm{Tan}h'(\kappa^+)>\mathrm{Tan}h'(x)\qquad x>\kappa^+ \tag{16}$$

$$\mathrm{Tan}h'(\kappa^-)>\mathrm{Tan}h'(x)\qquad x<\kappa^- \tag{17}$$

According the above properties, when $\lambda^+=\kappa^+$ and $\lambda^-=\kappa^-$, and the derivative of *ReLTanh* satisfies:

$$\mathrm{ReLTan}h'(x)=\mathrm{Tan}h'(\lambda^+)>\mathrm{Tan}h'(x)\qquad x>\lambda^+ \tag{18}$$

$$\mathrm{ReLTan}h'(x)=\mathrm{Tan}h'(\lambda^-)>\mathrm{Tan}h'(x)\qquad x<\lambda^- \tag{19}$$

Then it follows from Eqs. (2)–(4) that the updating value calculated by *ReLTanh* is larger than that by *Tanh*, which is also compared intuitively in Fig. 7.

$$\Delta\mathbf{w}_{\mathrm{ReLTanh}}>\Delta\mathbf{w}_{\mathrm{Tanh}}\qquad x>\lambda^+\mathrm{or}x<\lambda^- \tag{20}$$

Thus, *ReLTanh* can provide DNNs a higher convergence speed, and the lower layers can be trained more effectively.

3.3. *Learnability of* ReLTanh

Then, the feasibility for training the thresholds of *ReLTanh* is proved as follows.

Without loss of generality, we consider that each neuro in the same layer has different thresholds. Taking the positive thresholds of *i*th layer: $\boldsymbol{\lambda}^+_i$ as an example, $\boldsymbol{\lambda}^+_i$ can be updated by:

$$\boldsymbol{\lambda}^+_i=\boldsymbol{\lambda}^+_i+\gamma\Delta\boldsymbol{\lambda}^+_i \tag{21}$$

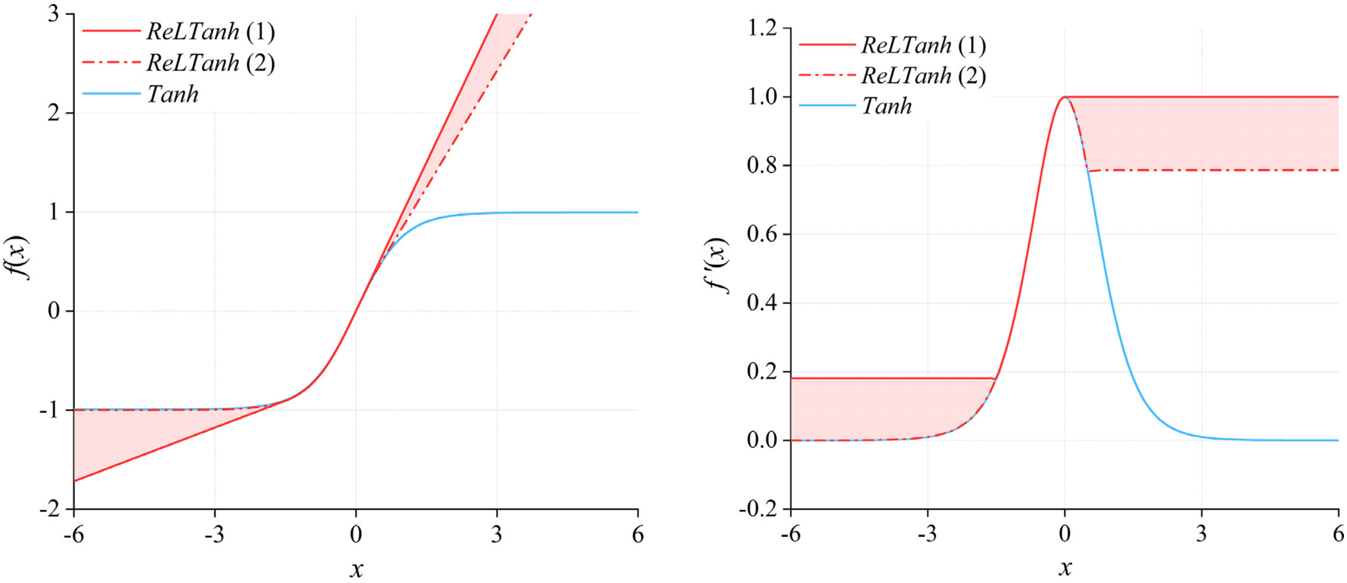
in which γ is the specific learning rate for training thresholds.

The updating value $\Delta\boldsymbol{\lambda}^+_i$ for $\mathbf{x}>\boldsymbol{\lambda}^+_i$ can be calculated by chain rule:

$$\Delta\boldsymbol{\lambda}^+_i=-\frac{\partial C}{\partial\mathbf{a}_i}\frac{\partial\mathbf{a}_i}{\partial\boldsymbol{\lambda}^+_i}=(\mathbf{a}_i-\mathbf{t})\frac{\partial\mathbf{a}_i}{\partial\boldsymbol{\lambda}^+_i} \tag{22}$$

$$\frac{\partial\mathbf{a}_i}{\partial\boldsymbol{\lambda}^+_i}=\frac{\partial\mathrm{ReLTanh}_i(\mathbf{x})}{\partial\boldsymbol{\lambda}^+_i}=\mathrm{Tan}h''\left(\boldsymbol{\lambda}^+_i\right)\left(\mathbf{x}-\boldsymbol{\lambda}^+_i\right) \tag{23}$$

$$\Delta\boldsymbol{\lambda}^+_i=\frac{\partial C}{\partial\mathbf{a}_i}\frac{\partial\mathbf{a}_i}{\partial\boldsymbol{\lambda}^+_i}=\mathrm{Tan}h''\left(\boldsymbol{\lambda}^+_i\right)(\mathbf{a}_i-\mathbf{t})\left(\mathbf{x}-\boldsymbol{\lambda}^+_i\right) \tag{24}$$



(a) *Tanh* and *ReLTanh*

(b) Derivatives of *Tanh* and *ReLTanh*

Fig. 7. *Tanh* and *ReLTanh* and their derivatives, in which *ReLTanh* (1) ($\lambda^+ = 0$ and $\lambda^- = 1.5$) and *ReLTanh* (2) ($\lambda^+ = 0.5$ and $\lambda^- = -\infty$) present the learnable range of thresholds. (a) *Tanh* and *ReLTanh*. (b) Derivatives of *Tanh* and *ReLTanh*.

However, if each neuro updates its thresholds independently, massive calculation is needed. In order to decrease the computational burden, the thresholds are recommended to be layer-wise shared. According to this recommendation, Eq. (23) can be simplified as:

$$\frac{\partial \bar{a}_i}{\partial \lambda_i^+} = \frac{\partial \mathrm{ReLTanh}_i(\bar{x}_i)}{\partial \lambda_i^+} = \mathrm{Tanh}''\left(\lambda_i^+\right)\left(\bar{x}_i - \lambda_i^+\right) \qquad \mathbf{x}_i \geq \lambda_i^+ \tag{25}$$

in which $\bar{x}_i$ is the average of the inputs $\mathbf{x}_i$ that is larger than λ_i^+, and $\bar{a}_i$ is the average of the corresponding outputs $\mathbf{a}_i$.

In summary, the updating values of thresholds in *i*th layer can be updated by:

$$\Delta\lambda_i = \begin{cases} \mathrm{Tanh}''\left(\lambda_i^+\right)\left(\bar{a}_i - \bar{t}_i\right)\left(\bar{x}_i - \lambda_i^+\right) & \mathbf{x}_i \geq \lambda_i^+ \\ 0 & \lambda_i^- < \mathbf{x}_i < \lambda_i^+ \\ \mathrm{Tanh}''\left(\lambda_i^-\right)\left(\bar{a}_i - \bar{t}\right)\left(\bar{x}_i - \lambda_i^-\right) & \mathbf{x}_i \leq \lambda_i^- \end{cases} \tag{26}$$

The initialization of thresholds can also influence the learning effect. In this study, the initial thresholds are set as $\lambda^+ = 0$ and $\lambda^- = -1.5$ for all layers, so that larger gradients can be provided to speed up the training process in the early stage, and then thresholds can be adjusted according to the demands of data fitting.

3.4. *Superiority of* ReLTanh

To sum up, the advantages of *ReLTanh* comparing against other common activation functions are listed as follows.

(1) *ReLTanh* has better derivative performance compared to *Tanh*, and it can diminish the vanishing gradient problem as *ReLU* family do.

(2) For mean activation, the outputs of *ReLTanh* are more close to zero, thus it affected less by bias shift than *ReLU* family. Compared to *ReLTanh*, *ReLU* and *Softplus* are absolutely non-negative, and *LReLU* and *Swish* have negligible negative outputs compared to their positive ones. With lighter bias shift influence, *ReLTanh* can speed up and smooth training process.

(3) The advantage of learnable thresholds helps *ReLTanh* to approach more closely to the global minimum. With training goes on, *ReLTanh* can adjust automatically the learnable thresholds in the descending direction of loss. It is worth noting that *ReLTanh* have a similar waveform to *ELU*, but *ReLTanh* can outperform *ELU*, not only because *ReLTanh* can updating thresholds to help to search the minimize of cost function, but also because *ELU* still suffers from vanishing gradient problem in the negative interval.

(4) To some extent, *ReLTanh* is more robust to noise and abnormal inputs. As learning goes on, the negative thresholds become close to $-\infty$ gradually, and the slope of the negative line become smaller. This property can make *ReLTanh* more noise-robustness and gentle.

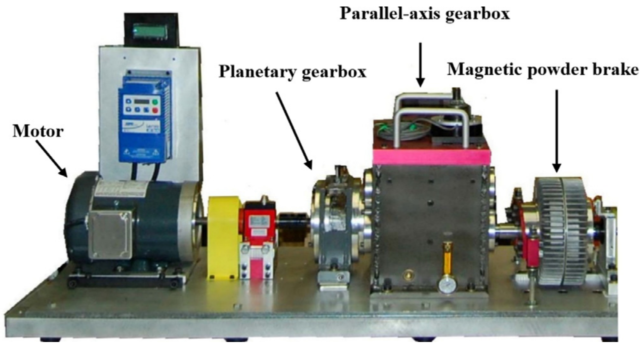


Fig. 8. The drivetrain diagnostics simulator.

All those advantages will be proved further in the experiment section.

4. Fault diagnosis for planetary gearbox

4.1. Experimental setup

As shown in Fig. 8, the test rig is a drivetrain diagnostics simulator (DDS) designed by SpectraQuest Inc (the company website can be visited with "http://www.pinxuntech.com/"), and it mainly consists of a driving motor, a two-stage planetary gearbox, a two-stage parallel-axis gearbox, a programmable magnetic brake. In this

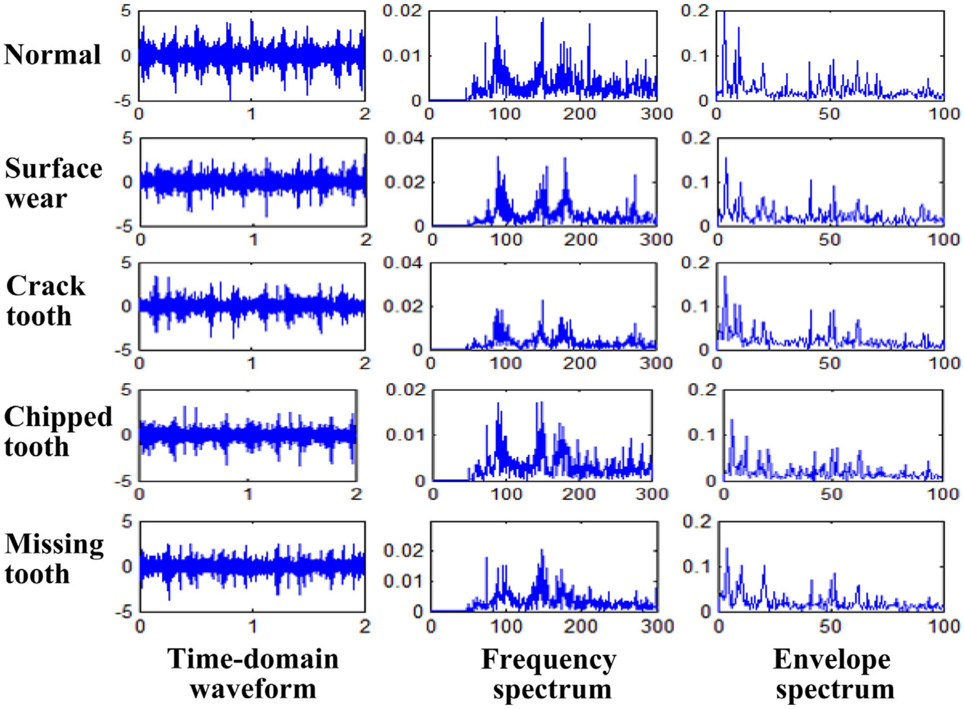


Fig. 9. The time-domain waveforms, frequency spectra and envelope spectra of original signals under the load of 1.4 N.m.

study, we focus on the secondary sun gear of the planetary gearbox because of its higher failure rate than other components in the gearbox [36]. Four most typical gear faults including surface wear, crack tooth, chipped tooth and missing tooth are discussed. Meanwhile, a normal gear is taken for comparison.

Via controlling the magnetic brake, vibration signals are collected under four different load conditions (0 Nm, 1.4 Nm, 2.8 Nm and 25.2 Nm respectively), so that the obtained signals could be more diverse [37]. In order to enrich information further, impulsive signals are extracted from original ones by Morlet wavelet transform proposed in Ref. [38], because vibration transients excited at specific frequency are more sensitive to early failures.

Taking the original and impulsive signals under the load of 1.4 Nm as an example, their time-domain waveforms, frequency spectra and envelope spectra are shown in Figs. 9 and 10 respectively.

4.2. Feature extraction

It can be seen from Figs. 9 and 10 that the waveforms of five kinds of gears are different, and the difference can be learned by DNNs to achieve diagnosis. However, Training DNNs using directly signals are computationally expensive, so statistical features such as Mean and Peak, etc. that can reflect the fault information quantitatively are applied [39].

In this study, as shown in Table 1, 25 features are calculated. Features t_1-t_{17} are extracted form original time-domain signals, and f_1-f_4 are frequency-domain features that can be extracted from frequency spectra and envelope spectra respectively. Similarly, 25 features are also extracted from impulsive signals.

Totally, 50-dimension samples are constructed in vector form by those 50 features.

4.3. Fault diagnosis and result analysis

In this subsection, *ReLTanh* can present practically a superior diagnosis performance for planetary gearboxes compared to the common activation functions, which can verify the theoretical analysis. For analysis on vanishing gradient problem, a 13-layer SAE-based DNN is built. For better reliability and generality of this study results, 10 parallel experiments are performed by using different initialization for weights and bias, and each experiment contains 22 testing subsets that are relatively independent. In this case, total 220 comparison results among those activation functions can be obtained, so the results are compelling.

The mean curves and the variation ranges of training accuracy of *ReLTanh* and *Tanh* on the 10 experiments are shown in Fig. 11, and *ReLTanh* thresholds before and after training of a certain experiment are illustrated in Fig. 12. Mean diagnostic accuracies and mean outperforming rates are applied as key measurable indicators for comparisons among those activation functions. These two indicators are defined below, and the comparative results are shown in Figs. 13 and 14.

Mean accuracies of activation functions are calculated by:

$$A = \frac{1}{N}\sum_{i}^{N}\left(\frac{1}{M}\sum_{j}^{M}A_{ij}\right) \tag{27}$$

where $M = 22$ and $N = 10$ are respectively the numbers of parallel experiments and testing subsets, A_{ij} is the accuracy of a certain accuracy of jth subset on ith experiment.

Outperforming rates represent the percentages of testing subsets on which *ReLTanh* outperforms other functions. Similarly, mean outperforming rates are defined as follows.

$$R = \frac{1}{N}\sum_{i}^{N}\left(\frac{m_i}{M}\right) \tag{28}$$

where m_i is the number of testing subsets on which *ReLTanh* performs better than the baseline function.

It is obvious from Fig. 11 that the learning processes of *ReLTanh* are steady and fast, and all the training accuracies on 10 experiments converge to 100% after epoch of 50. Fig. 12 illustrates that the positive and negative thresholds have been successfully trained and changed according to the cost function, and every hidden layer

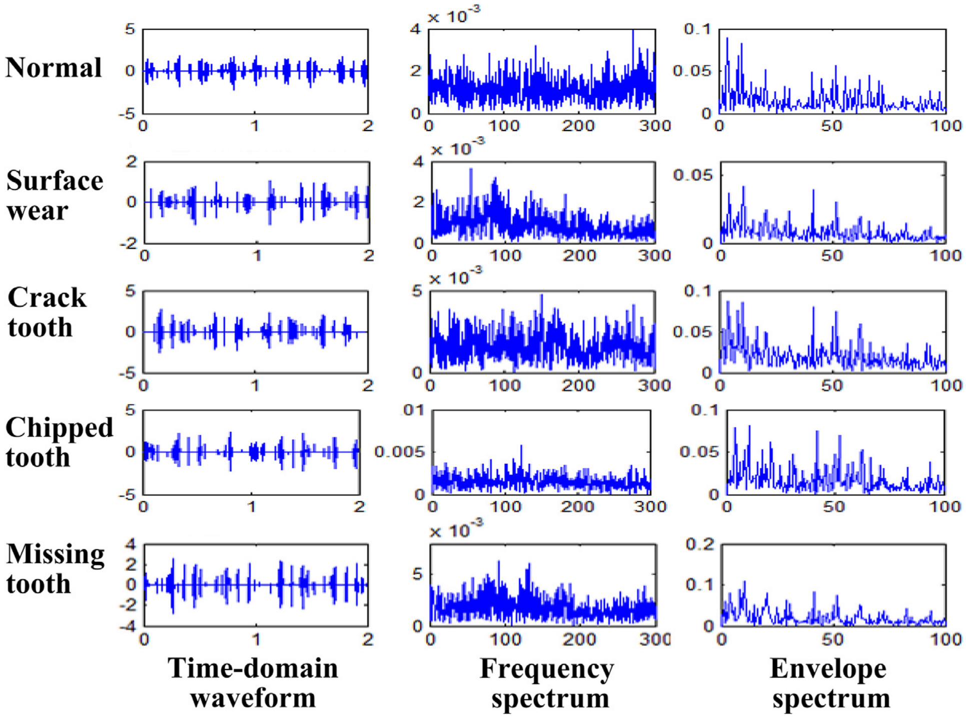


Fig. 10. The time-domain waveforms, frequency spectra and envelope spectra of impulsive signals under the load of 1.4 N.m.

Table 1
Statistical features.

Name	Formula	Name	Formula	Name	Formula
Peak-to-peak (Pk-pk)	$t_1 = \text{MAX}\lvert x(i)\rvert - \text{MIN}\lvert x(i)\rvert$	Variance (Var)	$t_8 = \frac{1}{N}\sum_i^N (x(i)-\bar{x})^2$	Impulsive factor (IF)	$t_{15} = \text{MAX}\lvert x(i)\rvert \Big/ \frac{1}{N}\sum_i^N \lvert x(i)\rvert$
Peak	$t_2 = \text{MAX}\lvert x(i)\rvert$	Standard deviation (SD)	$t_9 = \sqrt{\frac{1}{N}\sum_i^N (x(i)-\bar{x})^2}$	Clearance factor (CF)	$t_{16} = \text{MAX}\lvert x(i)\rvert \Big/ \left(\frac{1}{N}\sum_i^N \lvert x(i)\rvert^{1/2}\right)^2$
Mean	$t_3 = \frac{1}{N}\sum_i^N x(i)$	Skewness (Ske)	$t_{10} = \frac{1}{N}\sum_i^N x(i)^3$	Waveform factor (WF)	$t_{17} = \sqrt{\frac{1}{N}\sum_i^N x(i)^2} \Big/ \frac{1}{N}\sum_i^N \lvert x(i)\rvert$
Mean square (MS)	$t_4 = \frac{1}{N}\sum_i^N x(i)^2$	Kurtosis (Kur)	$t_{11} = \frac{1}{N}\sum_i^N \lvert x(i)\rvert^4$	Mean frequency (MF)	$f_1 = \frac{1}{N}\sum_j^N X(j)$
Root mean square (RMS)	$t_5 = \sqrt{\frac{1}{N}\sum_i^N x(i)^2}$	Skewness factor (SF)	$t_{12} = \frac{1}{N}\sum_i^N \lvert x(i)\rvert^3 \Big/ (\sqrt{\frac{1}{N}\sum_i^N x(i)^2})^3$	Frequency center (FC)	$f_2 = \sum_j^N (f(j) \times X(j)) \Big/ \sum_j^N X(j)$
Mean amplitude (MA)	$t_6 = \frac{1}{N}\sum_i^N \lvert x(i)\rvert$	Kurtosis factor (KF)	$t_{13} = \frac{1}{N}\sum_i^N \lvert x(i)\rvert^4 \Big/ \left(\sqrt{\frac{1}{N}\sum_i^N x(i)^2}\right)^4$	RMS frequency (RMSF)	$f_3 = \sqrt{\sum_j^N (f(j)^2 X(j)) \Big/ \sum_j^N X(j)}$
Square mean root (SMR)	$t_7 = \left(\frac{1}{N}\sum_i^N \lvert x(i)\rvert^{1/2}\right)^2$	Peak factor (PF)	$t_{14} = \text{MAX}\lvert x(i)\rvert \Big/ \sqrt{\frac{1}{N}\sum_i^N x(i)^2}$	Standard deviation frequency (SDF)	$f_4 = \sqrt{\sum_j^N ((f(j)-f_c)^2 X(j)) \Big/ \sum_j^N X(j)}$

has different trained thresholds to meet the demands of data fitting as far as possible. By comprehensive comparison, it follows from Figs. 13 and 14 that *ReLTanh* provides superior average accuracy of 95.63% to all other functions, and the highest outperforming rate reaches 100%.

Fig. 11 states that *Tanh* converges at the epoch of around 150, which is 100 iterations slower than *ReLTanh*. Figs. 13 and 14 illustrate that *ReLTanh* surpasses *Tanh* overwhelmingly with an accuracy difference of 7% and an outperforming rate of 100%. This result shows that the improvement on *ReLTanh* is effective, it reduces vanishing gradient problem that always perplexes *Tanh* to accelerate the learning and increase diagnostic accuracies.

For *ReLU* family, *ELU* produces highest mean accuracy, and *Softplus* takes the second place. Compared to *ReLTanh*, more than 97% of testing results of *ReLU* and *LReLU* are defeated by *ReLTanh*. The mean accuracy of *ReLTanh* is 0.6% higher than *ELU*, and the superiority of *ReLTanh* to *ELU* is mainly due to help of learnable thresholds on searching the global minimum, which has been described in Section 3.4.

Compared with *Hexpo* and *Swish*, *ReLTanh* beats them with large differences of about 4% and 6% respectively. *Hexpo* and *Swish* have basically non-negative average outputs and their gradients in negative interval are relatively small, so they suffer from a certain degree of bias shift problem and vanishing gradient problem.

It is found out in the experiments that ReLU family Hexpo and Swish cannot converge given larger initial weights, while ReLTanh can tolerate large initialization and provide relatively satisfactory accuracies.

To sum up, for this fault dataset, *ReLTanh* performs best and is worth further development and application.

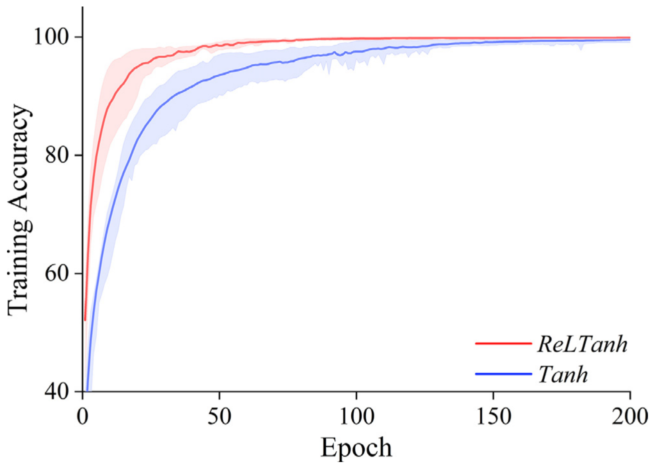


Fig. 11. Mean training accuracy curves of *ReLTanh* and *Tanh*.

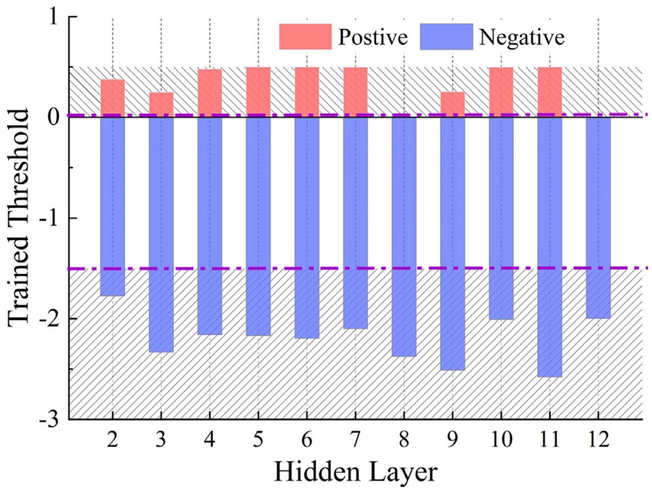


Fig. 12. *ReLTanh* thresholds on hidden layers (in which the two purple lines are the initial thresholds: $\lambda^{+} = 0$ and $\lambda^{-} = -1.5$, and the shadow areas represent the learnable range of thresholds: $0 \leq \lambda^{+} \leq 0.5$ and $\lambda^{-} \leq -1.5$).

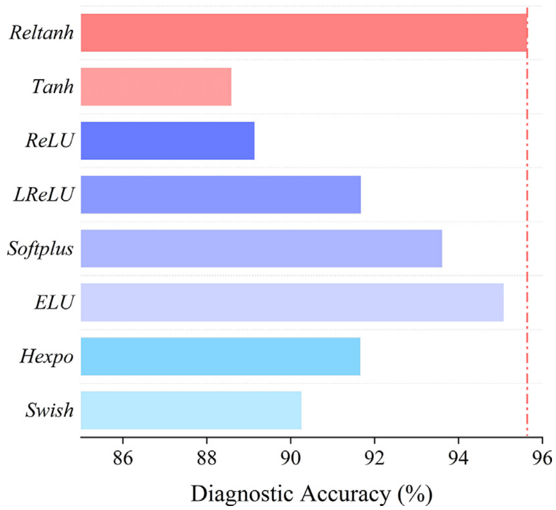


Fig. 13. Mean diagnostic accuracies.

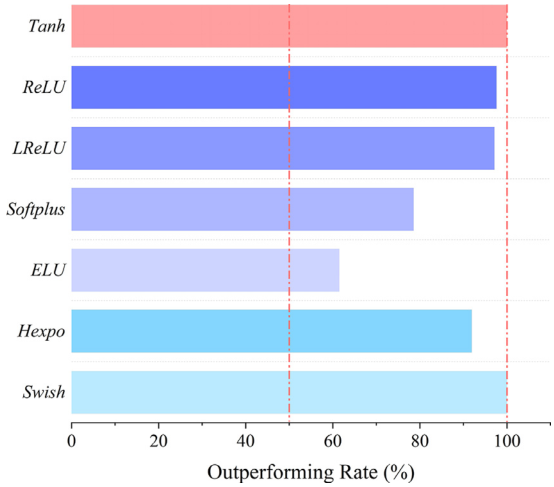


Fig. 14. Mean outperforming rates.

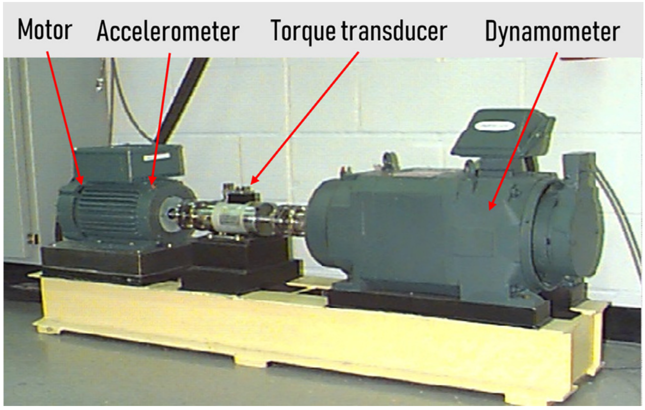


Fig. 15. The rolling bearing test rig of CWRU.

5. Fault diagnosis for rolling bearing

In order to prove further the effectiveness and superiority of *ReLTanh* on vanishing gradient problem, rolling bearing faulty datasets provided by Bearing Data Centre of Case Western Reserve University (CWRU) (which can be visited with “http://csegroups.case.edu/bearingdatacenter/home”) are applied to test *ReLTanh* again.

5.1. Experimental setup

Bearings are used to guide and support the rotating shafts of motors and rotary machinery, such as aero-engine, machine tool, etc., and any tiny fault can lead to severe losses on product quality and apparatus. To make matters worse, the rolling bearing is easily affected with many types of faults (e.g., localized defect, crack and wear on its bolls, inner ring and so on), and the failures are difficult to diagnose because of its complex structure [40]. Thereby the *ReLTanh* DNN is employed to achieve fault diagnosis for rolling bearings.

The faulty datasets of CWRU are widely used to verify various models, because of their better data integrity and faulty diversity. The rolling bearing test rig consists of a motor, a torque transducer, a dynamometer and control electronics, which is shown in Fig. 15.

We consider a normal operating conditions and three kinds of faults including inner race fault, ball fault and outer race fault, and

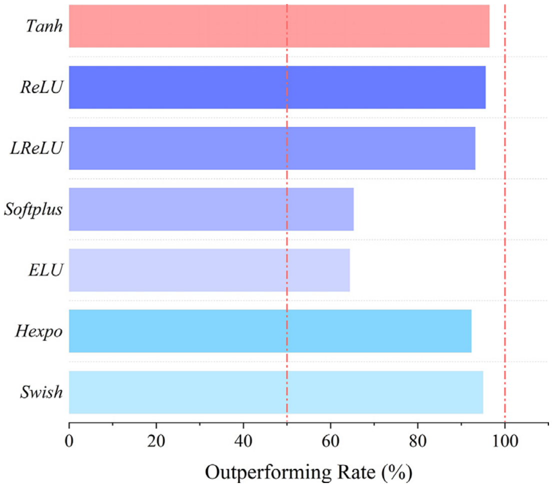


Fig. 16. Mean diagnostic accuracies.

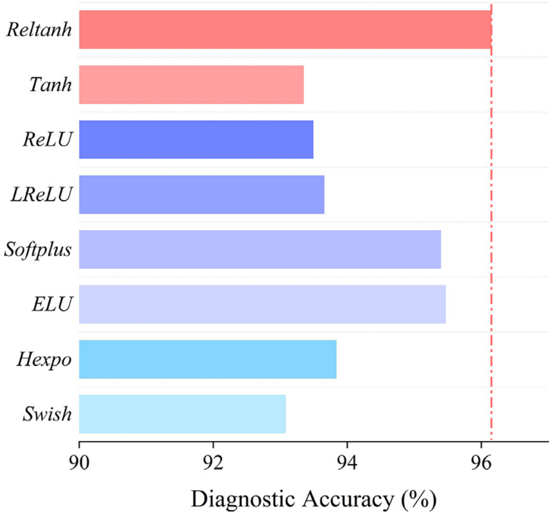


Fig. 17. Mean outperforming rates.

each kind of fault contains three different faulty diameters of 0.007, 0.014, and 0.021 inch, respectively. So signals of one normal condition and total 9 fault conditions can be obtained under 4 different load conditions: 0, 1, 2 and 3 horsepower. Similarly to the experiments for planetary gearboxes, feature extraction and sample construction are performed in the same way.

5.2. Fault diagnosis and result analysis

Like the experiment introduced in Section 4, a SAE-based DNN with 12 hidden layers are built for fault diagnosis experiment for rolling bearings. A subset containing 1000 samples are employed to train the DNNs armed with different activation functions, and 35 mutually independent subsets of 300 samples are used for test. For generality, 10 parallel experiments using different initializations are repeated.

Mean diagnostic accuracies and mean outperforming rates are compared in Figs. 16 and 17.

Similarly, *ReLTanh* produces a mean testing accuracy of 96.15% and defeats the other functions with obvious differences.

For traditional *Tanh*, *ReLTanh* transcend it with a gap of 2.8% and an outperforming rate of more than 96%, and the results are in accordance with those of the experiment in Section 4.2, which further verify the improvement of *ReLTanh* is effective for vanishing gradient problem. For *ReLU* family, *ELU* and *Softplus* perform better than *ReLU* and *LReLU*, but also lose against to *ReLTanh* with accuracy gaps of about 0.7% and outperforming rates of about 65%. Compared to *Hexpo* and *Swish*, *ReLTanh* wins again like the previous diagnosis experiments for gearboxes.

In conclusion, the results of those experiments are certainly consistent with those of planetary gearbox fault diagnosis, and it further verifies the conclusion that *ReLTanh* is effective for improving vanishing gradient problem.

6. Conclusions

Aiming to overcome the vanishing gradient problem suffered by *Tanh*, *ReLTanh* is created by improving *Tanh* for faster and more precise learning in SAE-based DNNs. *ReLTanh* is composed of three parts: a line with a relatively slight slope in negative interval, a non-linear part that reserved by *Tanh* in middle, and a line with a steeper slope in positive interval. The slopes of two lines can be trained by updating the thresholds according to the cost function, and detailed mathematical derivations demonstrate that the learning for thresholds is feasible. Besides, it is proved theoretically by using second derivatives that *ReLTanh* is effective to yield larger gradients to accelerate learning and reduce vanishing gradient problem. Two diagnosis experiments for planetary gears and rolling bearings illustrate the effectiveness and outperformance of *ReLTanh* to other common functions. The experimental results show that *ReLTanh* plays an important role in overcoming vanishing gradient problem and produces a faster and more precise learning for SAE-based DNNs than traditional *Tanh*. Compared to *ReLU* family, *Hexpo* and *Swish*, *ReLTanh* provides mean activation that is closer to zero, so it is hardly affected by bias shift problem and it presents more accurate performance. Two main contributions of this paper lie in that (1) a creative activation function *ReLTanh* is proposed by improving *Tanh*, and it is testified that *ReLTanh* has better learning performance than common activation functions; (2) an integrated diagnosis technique based on *ReLTanh* and SAE-Based DNNs is presented for rotating machinery fault diagnosis. Comprehensively, *ReLTanh* is worth further development and application.

Declaration of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The work described in this paper was supported by the National Natural Science Foundation of China (no. 51675065), Chongqing Research Program of Basic Research and Frontier Technology (no. cstc2017jcyjAX0459), National Key R&D Program of China (no. 2018YFB2001300) and the Fundamental Research Funds for the Central Universities (nos. 2018CDQYJX0011 and 2018CDJDCD0001).

References

[1] Yi Qin, Jingqiang Zou, Baoping Tang, Yi Wang, Haizhou Chen, Transient feature extraction by the improved orthogonal matching pursuit and K-SVD algorithm with adaptive transient dictionary, IEEE Trans. Ind. Info. (2019) (in press), doi:10.1109/TII.2019.2909305.
[2] Z.R. Wang, J. Wang, Y.R. Wang, An intelligent diagnosis scheme based on generative adversarial learning deep neural networks and its application to planetary gearbox fault pattern recognition, Neurocomputing 310 (2018) 213–222.
[3] L.L. Cui, N. Wu, C.Q. Ma, H.Q. Wang, Quantitative fault analysis of roller bearings based on a novel matching pursuit method with a new step-impulse dictionary, Mech. Syst. Signal Proc. 68–69 (2016) 34–43.
[4] L. Wang, Z. Zhang, H. Long, J. Xu, R. Liu, Wind turbine gearbox failure identification with deep neural networks, IEEE Trans. Ind. Info. 13 (3) (2017) 1360–1368.

[5] C. Li, R.V. Sanchez, G. Zurita, M. Cerrada, D. Cabrera, R.E. Vásquez, Multimodal deep support vector classification with homologous features and its application to gearbox fault diagnosis, Neurocomputing 168 (2015) 119–127.

[6] C. Yin, X.G. Huang, S. Dadras, Y.H. Cheng, J.W. Cao, H. Malek, J. Mei, Design of optimal lighting control strategy based on multi-variable fractional-order extremum seeking method, Inf. Sci. 465 (2018) 38–60.

[7] H. Geng, Y. Liang, F. Yang, L.F. Xu, Q. Pan, Model-reduced fault detection for multi-rate sensor fusion with unknown inputs, Inf. Fusion 33 (2017) 1–14.

[8] C. Yin, Y.Q. Chen, S.M. Zhong, Fractional-order sliding mode based extremum seeking control of a class of nonlinear systems, Automatica 50 (12) (2014) 3173–3181.

[9] Y. Qin, Y. Mao, B. Tang, Multicomponent decomposition by wavelet modulus maxima and synchronous detection, Mech. Syst. Signal Process. 91 (2017) 57–80.

[10] Y. Qin, X. Wang, J.Q. Zou, The optimized deep belief networks with improved logistic sigmoid units and their application in fault diagnosis for planetary gearboxes of wind turbines, IEEE Trans. Ind. Electron. 66 (5) (2019) 3814–3824.

[11] F. Jia, Y.G. Lei, J. Lin, X. Zhou, N. Lu, Deep neural networks: a promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data, Mech. Syst. Signal Proc. 72-73 (2016) 303–315.

[12] L. Xu, M.Y. Cao, B.Y. Song, J.S. Zhang, Y.R. Liu, F.E. Alsaadi, Open-circuit fault diagnosis of power rectifier using sparse autoencoder based deep neural network, Neurocomputing 311 (2018) 1–10.

[13] W.N. Lu, X.Q. Wang, C.C. Yang, T. Zhang, Ieee, a novel feature extraction method using deep neural network for rolling bearing fault diagnosis, in: 2015 27th Chinese Control and Decision Conference, New York, IEEE, 2015, pp. 2427–2431.

[14] G.F. Bin, J.J. Gao, X.J. Li, B.S. Dhillon, Early fault diagnosis of rotating machinery based on wavelet packets-Empirical mode decomposition feature extraction and neural network, Mech. Syst. Signal Proc. 27 (2012) 696–711.

[15] M.M. Lau, K.H. Lim, Investigation of activation functions in deep belief network, in: International Conference on Control and Robotics Engineering, 2017, pp. 201–206.

[16] P. Ghahremani, J. Droppo, M.L. Seltzer, Linearly augmented deep neural network, in: IEEE International Conference on Acoustics, Speech and Signal Processing, 2016, pp. 5085–5089.

[17] M.S. Ebrahimi, H.K. Abadi, Study of residual networks for image recognition, arXiv preprint arXiv:1805.00325, (2018).

[18] M. Heydarzadeh, S.H. Kia, M. Nourani, H. Henao, G.A. CapolinoIEEE, Gear fault diagnosis using discrete wavelet transform and deep neural networks, in: Proceedings of the IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society, New York, IEEE, 2016, pp. 1494–1500.

[19] G.E. Dahl, T.N. Sainath, G.E. Hinton, Improving deep neural networks for LVCSR using rectified linear units and dropout, in: IEEE International Conference on Acoustics, Speech and Signal Processing, 2013, pp. 8609–8613.

[20] G.F. Liu, H.Q. Bao, B.K. Han, A stacked autoencoder-based deep neural network for achieving gearbox fault diagnosis, Math. Probl. Eng. 10 (2018) 1–10.

[21] A. Ashiquzzaman, A.K. Tushar, A. Rahman, Applying data augmentation to handwritten arabic numeral recognition using deep learning neural networks, arXiv preprint arXiv:1708.05969, (2017).

[22] J. Li, H. Xu, J.H. Deng, X.M. Sun, Ieee, hyperbolic linear units for deep convolutional neural networks, in: 2016 International Joint Conference on Neural Networks, New York, IEEE, 2016, pp. 353–359.

[23] Y.A. LeCun, L. Bottou, G.B. Orr, K.-R. Müller, Efficient backprop, neural networks: Tricks of the trade, (Springer, 2012), pp. 9–48.

[24] B.K. Humpert, Improving back propagation with a new error function, Neural Netw. 7 (8) (1994) 1191–1192.

[25] M.D. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q.V. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, G.E. HintonIEEE, On rectified linear units for speech processing, in: 2013 Ieee International Conference on Acoustics, Speech and Signal Processing, New York, IEEE, 2013, pp. 3517–3521.

[26] A.L. Maas, A.Y. Hannun, A.Y. Ng, Rectifier nonlinearities improve neural network acoustic models, ICML Workshop on Deep Learning for Audio, Speech, and Language Processing, WDLASL, 2013 2013.

[27] V. Nair, G.E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Proceedings of the 27th International Conference on International Conference on Machine Learning, Haifa, Israel, Omnipress, 2010, pp. 807–814.

[28] D. Clevert, T. Unterthiner, S. Hochreiter, Fast and accurate deep network learning by exponential linear units (ELUs), International Conference on Learning Representations, 2016.

[29] P. Ramachandran, B. Zoph, Q.V. Le, Searching for activation functions, arXiv preprint arXiv:1710.05941, (2017).

[30] S.M. Kong, M. TakatsukaIEEE, Hexpo: a vanishing-proof activation function, in: 2017 International Joint Conference on Neural Networks, New York, IEEE, 2017, pp. 2562–2567.

[31] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, Greedy layer-wise training of deep networks, in: Proceedings of the 19th International Conference on Neural Information Processing Systems, MIT Press, Canada, 2006, pp. 153–160.

[32] G.E. Hinton, R.S. Zemel, Autoencoders, minimum description length and Helmholtz free energy, in: International Conference on Neural Information Processing Systems, 1993, pp. 3–10.

[33] Y. Bengio, A. Courville, P. Vincent, Unsupervised feature learning and deep learning: a review and new perspectives, in: IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, pp. 1–30.

[34] F. Wang, B. Dun, G. Deng, H. Li, Q. Han, A deep neural network based on kernel function and auto-encoder for bearing fault diagnosis, in: 2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), IEEE, 2018, pp. 1–6.

[35] P. Jiang, Z.X. Hu, J. Liu, S.N. Yu, F. Wu, Fault diagnosis based on chemical sensor data with an active deep neural network, Sensors 16 (10) (2016) 22.

[36] M.H. Zhao, M.S. Kang, B.P. Tang, M. Pecht, Multiple wavelet coefficients fusion in deep residual networks for fault diagnosis, IEEE Trans. Ind. Electron. 66 (6) (2019) 4696–4706.

[37] H. Geng, Y. Liang, Y.R. Liu, F.E. Alsaadi, Bias estimation for asynchronous multi--rate multi-sensor fusion with unknown inputs, Inf. Fusion 39 (2018) 139–153.

[38] Y. Qin, A new family of model-based impulsive wavelets and their sparse representation for rolling bearing fault diagnosis, IEEE Trans. Ind. Electron. 65 (3) (2018) 2716–2726.

[39] X. Wang, Y. Qin, A.B. Zhang, An intelligent fault diagnosis approach for planetary gearboxes based on deep belief networks and uniformed features, J. Intell. Fuzzy Syst. 34 (6) (2018) 3619–3634.

[40] S.L. Lu, X.X. Wang, A new methodology to estimate the rotating phase of a BLDC motor with its application in variable-speed bearing fault diagnosis, IEEE Trans. Power Electron. 33 (4) (2018) 3399–3410.

Xin Wang received the B. Eng. degree in Automotive engineering from Chongqing University, Chongqing, China, in 2017.

He is currently working toward the M.S. degree in School of Automotive Engineering of Chongqing University, Chongqing, China. His research interests mainly include signal processing, intelligent mechanical fault diagnosis and artificial intelligence.

Yi Qin received the B. Eng. and Ph.D. degrees in mechanical engineering from Chongqing University, Chongqing, China, in 2004 and 2008 respectively.

Since January 2009, he has been with the Chongqing University, Chongqing, China, where he is currently a Professor in the College of Mechanical Engineering. His current research interests include signal processing, fault prognosis, mechanical dynamics and smart structure.

Dr. Qin is a Member of IEEE.

Yi Wang received his B. Eng. degree in mechanical engineering from Southwest Jiaotong University, Chengdu, China, in 2011, Ph.D. degree in mechanical engineering from Xi'an Jiaotong University, Xi'an, China, in 2017, respectively. During 2016.8–2017.2, he was a visiting scholar in City University of Hong Kong, Hong Kong, China.

Since January 2009, he has been with the Chongqing University, Chongqing, China, where he is currently a Lecture in the College of Mechanical Engineering. His current research interests include mechanical signal processing, weak signal detection, rotating machinery fault diagnosis under speed variation conditions, manifold learning and deep learning.

Sheng Xiang received the B. Eng. degree in mechanical engineering from Yangtze University, Hubei, China, in 2017.

He is currently working toward the a Ph.D. degree in mechanical engineering of Chongqing University, Chongqing, China. His research interests mainly include signal processing, mechanical fault diagnosis and residual life prediction.

Haizhou Chen received the M.A. Eng. and Ph.D. degrees in mechanical engineering from Chongqing University, Chongqing, China, in 2010 and 2017 respectively.

Since July 2017, he has been with the Qingdao University of Science and Technology, Qingdao, China, where he is currently a Lecturer in the College of Electromechanical Engineering. His current research interests include failure mechanism analysis, fault prognosis and tribology.