Deep Residual Networks with Adaptively Parametric Rectifier Linear Units for Fault Diagnosis

Minghang Zhao, Shisheng Zhong, Xuyun Fu, Baoping Tang, Shaojiang Dong, and Michael Pecht, *Fellow, IEEE*

***Abstract*—Vibration signals under the same health state often have large differences due to changes in operating conditions. Likewise, the differences among vibration signals under different health states can be small under some operating conditions. Traditional deep learning methods apply fixed nonlinear transformations to all the input signals, which has a negative impact on the discriminative feature learning ability, i.e., projecting the intra-class signals into the same region and the inter-class signals into distant regions. Aiming at this issue, this paper develops a new activation function, i.e., adaptively parametric rectifier linear units, and inserts the activation function into deep residual networks to improve the feature learning ability, so that each input signal is trained to have its own set of nonlinear transformations. To be specific, a sub-network is inserted as an embedded module to learn slopes to be used in the nonlinear transformation. The slopes are dependent on the input signal, and thereby the developed method has more flexible nonlinear transformations than the traditional deep learning methods. Finally, the improved performance of the developed method in learning discriminative features has been validated through fault diagnosis applications.**

***Index Terms*—Deep learning, deep residual networks, fault diagnosis, rectifier linear units, vibration signal.**

I. Introduction

Mechanical and electrical devices play an integral role in many industries, such as metallurgy, transportation, and mining. Due to the long-time operation under harsh environments, it is often inevitable for them to encounter faults that may result in severe accidents and losses. To avoid these troubles, fault diagnosis has become a research focus in the past decades [1]-[2]. Specifically, vibration signals contain plenty of impulses and fluctuations caused by mechanical faults [3], and are frequently used in fault diagnosis. Typically, engineers can diagnose the faults through localizing fault frequencies [4]-[6]. However, this is often difficult for large machines that consist of many components, which motivates the exploration of the other approaches for mechanical fault diagnosis.

Data-driven fault diagnosis approaches are rapidly evolving in the past decades, which can avoid the trouble of localizing fault-related frequencies [7]-[8]. In general, a set of statistical features, such as kurtosis, energy, and peak-to-peak value, are configured in the first step, with the purpose of representing the fault-related information. Then, these statistical features are fed into a shallow classifier for recognizing the fault. However, it is mostly unknown whether these statistical features can fully represent the fault-related information. Furthermore, an optimal statistical feature set, which works well for a specific machine, may not work well for the other machines or the same machine running under the other operating conditions. As a consequence, configuring a satisfactory feature set, which can fully represent the fault-related information, has become a long-standing issue in the field of data-driven fault diagnosis.

As an alternative, deep learning methods can automatically learn a set of discriminative features from raw signals and yield higher accuracy than the traditional machine learning methods, which avoids the trouble of artificially configuring a feature set [9]. A variety of deep learning methods, including deep belief networks [10]-[12], auto-encoders [13], convolutional neural networks (ConvNets) [14]-[26], and some others [27]-[31], have been investigated in fault diagnosis. For example, Liu et al. [11] proposed a Gaussian–Bernoulli deep belief network, which is an impressive and remarkable breakthrough in extracting high-order semantic features for fault diagnosis of electronics-rich analog systems; Jiang et al. [16] developed a multi-scale ConvNet, which learns a group of highly comprehensive features, to diagnose wind turbine gearboxes; Ding at al. [19] developed a deep ConvNet to learn a group of energy-fluctuated features for reliable fault diagnosis of spindle bearings; Shihavuddin et al. [20] developed a deep-learning-based automated damage detection system for

Manuscript received May 3, 2019; revised September 13, 2019 and December 5, 2019; accepted January 24, 2020. This work was supported in part by the Key National Natural Science Foundation of China under Grant U1733201, in part by the National Natural Science Foundation of China under Grant 51775072, in part by the Natural Science Foundation of Shandong Province under Grant ZR2017MEE062 and ZR2019MEE096, and in part by the Natural Science Foundation Project of CQ under Grant cstc2017jcyjAX0279. *(Corresponding author: Minghang Zhao)*

M. Zhao, S. Zhong, and X. Fu are with the School of Naval Architecture and Ocean Engineering, Harbin Institute of Technology at Weihai, Weihai 264209, China (e-mail: zhaomh@hit.edu.cn; zhongss@hit.edu.cn; fuxuyun@hit.edu.cn).

B. Tang is with the State Key Laboratory of Mechanical Transmission, Chongqing University, Chongqing 400044, China (e-mail: bptang@cqu.edu.cn).

S. Dong is with the School of Mechatronics and Automotive Engineering, Chongqing Jiaotong University, Chongqing 400074, China (e-mail: dongshaojiang100@163.com).

M. Pecht is with the Center for Advanced Life Cycle Engineering, University of Maryland, College Park, MD 20742, USA (e-mail: pecht@umd.edu).

wind turbine blades and yielded almost human-level precision. However, traditional deep learning methods are generally much harder to train than shallow neural networks. Once the training failed, the deep learning methods would not be able to detect the faults.

In recent years, deep residual networks (ResNets) emerge as one of the state-of-the-art deep learning algorithms for pattern recognition tasks, which are in fact a special kind of ConvNets [32],[33]. Compared to the conventional ConvNets, the use of identity shortcuts in ResNets significantly reduces the difficulty of training of deep architectures with tens or hundreds of layers. Moreover, ResNets are gradually becoming popular in machine fault diagnosis [34]-[40]. For instance, Zhang et al. [35] applied ResNets to learn discriminative features from vibration signals for bearing fault diagnosis; Wen at al. [37] developed a ResNet with 51 layers to diagnose self-priming centrifugal pumps; Ma et al. [39] integrated ResNets with demodulated time-frequency features for diagnosing planet gearboxes under varying rotation speeds. The superiority of ResNets compared to the classical ConvNets has been validated in these studies, and therefore this paper takes ResNets as a benchmark to be further improved.

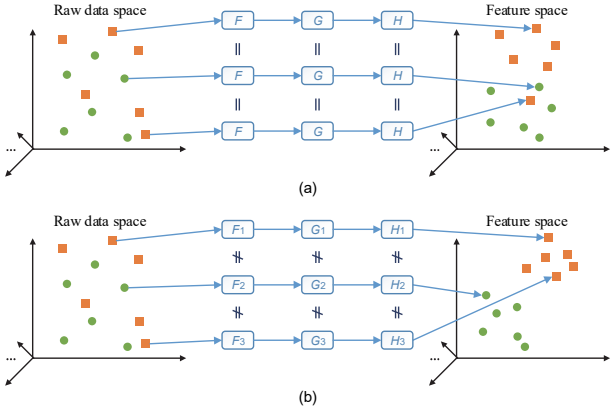


Fig. 1. (a) A sketch of nonlinear transformations in classical deep neural networks, where *F*, *G*, and *H* represent nonlinear transformations, and "=" means that the nonlinear transformations are the same. (b) A sketch of nonlinear transformations in the developed method, where "≠" means that the nonlinear transformations can be different.

However, as indicated in Fig. 1(a), a problem in classical neural networks, including ResNets with typical activation functions (e.g., sigmoid, tanh, rectifier linear units (ReLUs) [41], leaky ReLUs (LReLUs) [42], and parametric ReLUs (PReLUs) [43]), is that an identical group of nonlinear transformations is applied to each signal. This methodology has a negative impact on the feature learning ability. A problem is that two vibration signals under the same health state may have very different characteristics (e.g., different impulsive and fluctuation waveforms) due to the changing operating conditions (e.g., rotation speeds and loads). The fault-related frequencies are proportional to the variable rotating speeds; the amplitudes may be magnified or may shrink with increasing loads for mechanical devices with different structures; and the waveforms of vibration signals under the same health state are often obviously different from each other. As a result, it is difficult to project the two vibration signals into the same class by applying identical nonlinear transformations to them. Likewise, another problem is that the vibration signals under different health states may have very similar characteristics on some occasions, e.g., their fault-related frequencies may be similar to each other. The vibration signals are often projected into a close region after an identical set of nonlinear transformations in classical deep networks, which leads to misclassifications [44]-[46]. Therefore, it is meaningful to develop new deep learning methods that can automatically learn and apply different nonlinear transformations to the input signals.

Aiming at the aforementioned issue, as indicated in Fig. 1(b), this paper develops a new deep learning method, the so-called ResNets with adaptively PReLUs (ResNet-APReLU), to assign different nonlinear transformations to the input signals, with the ultimate goal of improving diagnostic accuracy. Specifically, the developed ResNet-APReLU method allows the slopes in the activation functions (i.e., APReLU) to be adjustable according to the input signals. Here, the slopes are a kind of multiplication coefficients to be used when performing nonlinear transformations. This is achieved by inserting a specially designed sub-network into the activation function. It is notable that the developed method is inspired by the squeeze-and-excitation networks [47], in which a group of multiplication coefficients are learned by a sub-network to adjust the values of features at different channels. In this way, the developed ResNet-APReLU can automatically design a group of nonlinear transformations for each input signal.

The remainder of this paper is organized as follows. Section II reviews the basics of ResNets, and elaborates the developed ResNet-APReLU. Experimental results are given and discussed in Section III, and conclusions are summarized in Section IV.

II. Theory of the Developed ResNet-APReLU

This section introduces the design principle and architecture of the developed ResNet-APReLU in detail, after an overview of the necessary fundamentals of classical ResNets and popular activation functions.

A. Fundamentals of classical ResNets

ResNets [32],[33] are a special kind of ConvNets, which are an ensemble of various components, including a convolutional layer, a number of residual building blocks (ResBlocks), a batch normalization (BN), a ReLU activation function, a global average pooling (GAP), and a fully connected output layer (FC). The involved components are introduced as follows.

First, the convolutional layer is composed of a number of trainable filters, in which the parameters are initialized as random floats and optimized in the training process. A convolution, which is followed by adding a bias, is expressed by

$$y_j = \sum_i x_i * k_{ij} + b_j \tag{1}$$

where x is the input feature map and y is the output feature map. Since this paper takes 1-dimensional (1D) vibration signals as input, the feature maps are 2-dimensional (2D) matrices in the format of length × channels. k is the convolutional kernel, b is the bias, and i and j are indicators of the channels of the feature map.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIE.2020.2972458, IEEE Transactions on Industrial Electronics

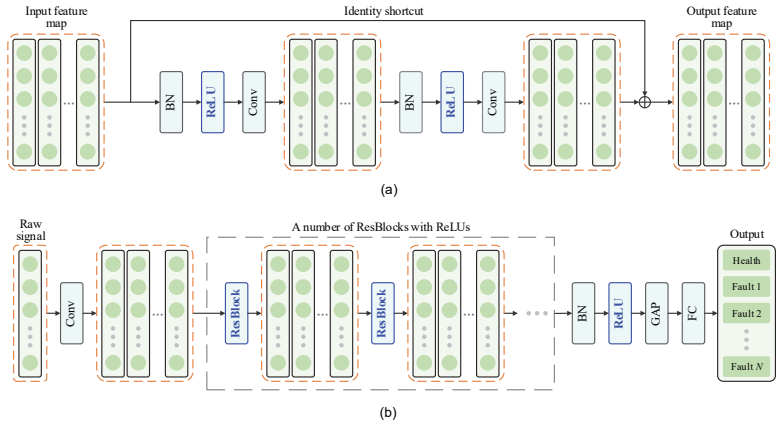


Fig. 2. (a) A ResBlock with two ReLUs, two BNs, two convolutional layers, and an identity shortcut, and (b) the architecture of a classical ResNet.

ReLUs are one of the most popular activation functions for deep learning methods [9]. Compared to the sigmoid and tanh functions, ReLUs are more effective in preventing the gradient vanishing problem. A ReLU is expressed by

$$y = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{2}$$

where x and y are the input and output features, respectively.

BN is a normalizing technique that is applied to the features within deep architectures [48]. It aims to solve the internal covariate shifting issue, i.e., the distributions of features often continuously change among the training iterations, which slows down the training process. A BN is expressed by

$$\mu = \frac{1}{N_{\text{batch}}} \sum_{s=1}^{N_{\text{batch}}} x_s \tag{3}$$

$$\sigma^2 = \frac{1}{N_{\text{batch}}} \sum_{s=1}^{N_{\text{batch}}} (x_s - \mu)^2 \tag{4}$$

$$\hat{x}_s = \frac{x_s - \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{5}$$

$$y_s = \gamma \hat{x}_s + \beta \tag{6}$$

where x_s and y_s are the input and output features of the sth observation in a pre-defined mini-batch, respectively; ϵ is a near-zero constant; and γ and β are trainable parameters for scaling and shifting the distributions.

As shown in Fig. 2(a), a ResBlock is composed of two BNs, two ReLUs, two convolutional layers, and one identity shortcut. The identity shortcut is the key component that gives ResNets an advantage over classical ConvNets. The gradients can be propagated to the early layers (i.e., the layers that are close to the input layers) directly through the identity shortcuts, which facilitates the gradient back-propagation, so that ResNets are much easier to train than the classical ConvNets.

GAP is a special pooling layer that is often used before the final FC output layer [49]. Mathematically, an average value is calculated from each channel of the feature map. GAP can reduce the amount of weights used in the final FC layer and further decreases the risk of overfitting.

The cross-entropy error [50] is used as the cost function to be minimized in ResNets. In the first step, a softmax function is applied to convert the features to the range of (0, 1), which is expressed by

$$y_j = \frac{e^{x_j}}{\sum_{i=1}^{N_{\text{class}}} e^{x_i}} \tag{7}$$

where x and y are the input and output features, respectively, and N_{class} is the number of classes. The cross-entropy error is then mathematically expressed by

$$E = -\sum_{j=1}^{N_{\text{class}}} t_j \log(y_j) \tag{8}$$

where t is the label in the one-hot format. After the calculation of the error, gradient back-propagation can be applied to optimize the parameters. The process can be repeated for a certain number of iterations to fully optimize the ResNet model, which has been depicted in Fig. 2(b).

B. Improved versions of ReLUs

As introduced in Section II.A, ReLUs are one of the most popular activation functions for deep learning methods. Research has been conducted to develop variants of ReLUs, such as LReLUs and PReLUs, to improve the performance.

LReLUs are slightly different than the traditional ReLUs in that LReLUs apply a small, non-zero multiplication coefficient (e.g., 0.1) to the negative features, rather than enforcing them to be zeros [42], which is expressed by

$$y = \max(x, 0) + 0.1 \cdot \min(x, 0) \tag{9}$$

where x and y are the input and output features, respectively.

The PReLU is a variant of LReLUs [43]. As mentioned above, the coefficient in LReLUs is a pre-defined constant. Instead, PReLUs allow the coefficients to be trainable using gradient backpropagation. A PReLU is expressed by

$$y = \max(x, 0) + \alpha \cdot \min(x, 0) \tag{10}$$

where α is the trainable multiplication coefficient (i.e., a slope). Specifically, each channel of the feature map has its own α, so that the nonlinear transformations can become highly flexible. In addition, it is notable that α in the PReLU is trainable in the training process, but becomes a constant number in the testing process, which is not adjustable according to each specific test signal.

C. Design of the developed ResNet-APReLU

The developed ResNet-APReLU is in fact a ResNet with a special kind of activation function, i.e., APReLUs, to perform adaptive nonlinear transformations. Thus, in this subsection, the motivations for applying adaptive nonlinear transformations to vibration signals are introduced, and the architectures of the APReLU and ResNet-APReLU are then explained in detail.

1) Motivations for applying adaptive nonlinear transformations to vibration signals

A significant task in fault diagnosis powered by deep neural networks is to project the intra-class signals into nearby regions

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIE.2020.2972458, IEEE Transactions on Industrial Electronics

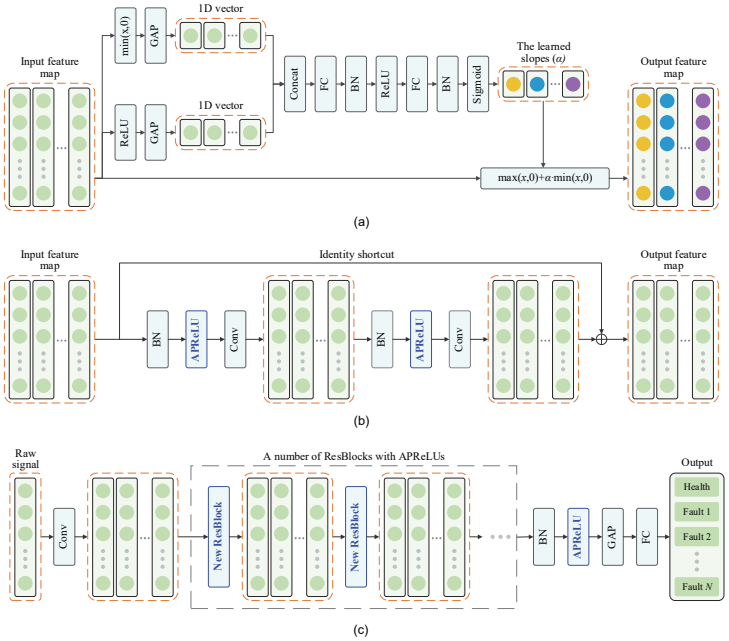


Fig. 3. (a) The developed APReLU for adaptive nonlinear transformation, (b) a new ResBlock with two convolutional layers, two BNs, and two APReLUs, and an identity shortcut, and (c) the overall architecture of the developed ResNet-APReLU method.

and the inter-class signals into distant regions. However, a long-standing problem is that vibration signals under the same fault state often have different characteristics due to the changing operating conditions. For example, the amplitudes may increase or decrease due to the changing loads, and the time intervals of fault-related impulses may change due to the changing rotation speeds. At the same time, the vibration signals under some different fault states may have similar characteristics in some circumstances. For example, although the fault-related frequencies of different faults may be quite distinct under the same rotation speed, these frequencies may become similar to each other if the vibration signals are collected under different rotating speeds.

In the deep learning methods that use traditional activation functions, the same set of nonlinear transformations are applied to all the vibration signals. As a result, it can be challenging to project the vibration signals under the same health state into a nearby region through the same set of nonlinear transformations; if some vibration signals under different health states share similar properties, it can be difficult to project them to distant regions as well. Therefore, to improve the discrimination of the learned features, new activation functions need to be developed that allow different nonlinear transformations to be assigned to different vibration signals.

2) Design of the fundamental architecture for APReLU

The APReLU integrates a specially designed subnetwork as an embedded module for adaptively estimating the multiplication coefficients to be used in the nonlinear transformations, which is the major innovation of this study.

As depicted in Fig. 3(a), in the first step, the input feature map is propagated into a ReLU and a GAP to calculate a 1D vector to represent the global information of the positive features. At the same time, the input feature map is also propagated into a min(x, 0) function and a GAP to calculate another 1D vector to represent the global information of the negative features. The motivations are as follows. First, although the learned features in deep neural networks do not have clear physical meaning, a min(x,0) function is applied along with the ReLU activation function because the negative features may contain some useful fault-related vibration signal information that the positive features do not contain. Second, GAPs can be used to address the shift variation problem when applying deep learning methods to vibration signals. The locations of fault-related impulses and fluctuations can be different in different vibration signals, which often change the final features and decrease the stability of deep neural networks; the problem can be addressed by calculating averages from the feature maps using GAPs. Moreover, the information of a feature map can be squeezed into two 1D vectors, which have much fewer data points than the original feature map and can reduce the calculation amount of the subsequent network. Therefore, the multiplication coefficients to be used in the subsequent nonlinear transformation can be determined with reference to the global information of positive features and negative features at the same time, rather than only considering the global averages directly generated from a GAP as in [47], so that the multiplication coefficients can be estimated based on a comprehensive information source.

After that, the two 1D vectors are concatenated and propagated into a calculating path (i.e., FC→BN→ReLU→FC →BN→sigmoid), in which the number of neurons in each FC layer equals the number of channels of the input feature map of the APReLU. Likewise, the motivations for the design of the calculating path are explained as follows. First, with the use of a ReLU and a sigmoid activation function, the calculating path can provide two levels of nonlinearity when determining the multiplication coefficients. Second, the gradient of the ReLU is either one or zero, which keeps the values of gradients within a reasonable range in most circumstances. Third, the sigmoid activation function converts the multiplication coefficients to floats in the range of (0, 1), which prevents the risk of assigning too large values to the multiplication coefficients. Fourth, BN

can address the internal covariate shifting problem and accelerate the optimization process; by applying BN to the features among the layers, the training process can be accelerated. In the end, the same function with the PReLU (see Eq. 10) is applied to perform nonlinear transformation and get the output feature map.

3) Architecture of the developed ResNet-APReLU for vibration-based gearbox fault diagnosis

In this section, the architecture and optimization method of the developed ResNet-APReLU are introduced, and its superiority in vibration-based gearbox fault diagnosis is clarified.

As depicted in Fig. 3(b), a new ResBlock was constructed, which is composed of two convolutional layers, two BNs, two APReLUs, and an identity shortcut. The new ResBlock has almost the same architecture as the classical ResBlock in Fig. 2(a). The only difference is that the developed APReLUs are used instead of the traditional ReLUs for adaptive nonlinear transformations, and therefore each new ResBlock has two levels of adaptive nonlinear transformations. In other words, the output feature map of the APReLU has the same shape and format as the input feature map, so that the developed APReLU can be easily inserted into the ResBlock and any locations of deep neural networks without making any other modifications.

Figure 3(c) shows the overall architecture of the developed ResNet-APReLU, which is composed of a convolutional layer, a number of new ResBlocks, a BN, an APReLU, a GAP, and an FC output layer. Because a number of new ResBlocks are stacked in the architecture, the developed ResNet-APReLU is able to apply the adaptive nonlinear transformations multiple times. Accordingly, the working principle, which is illustrated in Fig. 1(b), can be achieved, in which each input signal can have its own set of nonlinear transformations that can be different from the other signals.

ResNet-APReLU is optimized using the gradient descent algorithm [9], which is expressed by

$$w \leftarrow w - \eta \frac{\partial E}{\partial w} \tag{11}$$

where w is an indicator to represent any of the trainable parameters in the architecture, E is the cross-entropy error, and η is the learning rate. As indicated in Fig. 3(a), the developed APReLU is in fact a combination of various basic operations, including a min(x,0) function, two ReLUs, two GAPs, two FC layers, two BNs, a sigmoid function, and a $\max(x,0) + \alpha\min(x,0)$ function (i.e., Eq. 10). The gradients of the output feature with respect to the input feature of some basic operations are illustrated in Table I, where N_{feature} is the number of features in a channel of a feature map. The gradients of the FC layer and the BN are available in [9] and [48], respectively.

TABLE I
GRADIENTS OF THE OUTPUT FEATURE WITH RESPECT TO THE INPUT FEATURE OF SOME BASIC OPERATIONS

Operation	Gradient
ReLU	$\frac{\partial y}{\partial x} = \begin{cases} 1, x > 0 \\ 0, x \le 0 \end{cases}$
min(x, 0)	$\frac{\partial y}{\partial x} = \begin{cases} 0, x \ge 0 \\ 1, x < 0 \end{cases}$
GAP	$\frac{\partial y}{\partial x} = \frac{1}{N_{\text{feature}}}$
Concatenation	$\frac{\partial y}{\partial x} = 1$
Sigmoid	$\frac{\partial y}{\partial x} = \frac{e^{-x}}{(e^{-x}+1)^2}$
max(x,0) + αmin(x,0)	$\frac{\partial y}{\partial x} = \begin{cases} 1, x > 0 \\ \alpha, x \le 0 \end{cases}$

The gradients of the final cross-entropy error with respect to the trainable parameter w can then be calculated according to the chain rule [9], which is expressed by

$$\frac{\partial E}{\partial w} = \sum_k \frac{\partial E}{\partial \text{Path}_k} \cdot \frac{\partial \text{Path}_k}{\partial w} \tag{12}$$

where Path is a collection of differentiable paths that connect the trainable parameter w with the cross-entropy error at the output layer. Here, the so-called differentiable paths are composed of the basic components listed above. After the optimization, the new ResBlocks will be able to convert the input data to be highly discriminative features. In the end, machine fault diagnosis is conducted using the FC output layer, in which the number of neurons equals the total number of considered health states.

In summary, the developed ResNet-APReLU is a new deep learning method that automatically learns a group of nonlinear transformations for each specific input signal, which is suitable for performing fault diagnosis of gearboxes using vibration signals collected under various operating conditions. The multiplication coefficients, which are used to define the adaptive nonlinear transformations, are determined according to the characteristics of each specific vibration signal. Compared to the traditional way of applying identical nonlinear transformations to all the vibration signals, the developed ResNet-APReLU is more effective in extracting discriminative features and yielding higher diagnostic accuracy.

III. EXPERIMENTAL RESULTS

The developed ResNet-APReLU was applied to diagnose planetary gearboxes under changeable operating conditions by taking vibration signals as input. Experimental comparisons were conducted with classical ConvNets and ResNets that use sigmoid, tanh, ReLU, LReLU, and PReLU activation functions. The experimental results have been summarized and discussed in this section.

A. Vibration signal collection

As shown in Fig. 4, a drivetrain dynamics simulator was used for simulating the faults and collecting vibration signals, which is composed of a motor, a 2-stage planetary gearbox (with four planetary gears in the first stage and three planetary gears in the second stage), a 2-stage fixed-shaft gearbox, and a heavy-duty programmable magnetic brake. The magnetic brake can be used to adjust the torsional loads to simulate the changed operating conditions. An accelerometer was mounted at the input end of the planetary gearbox in the vertical direction, and the vibration signals were collected with a sampling frequency of 12.8 kHz.

As summarized in Table II, eight health states of the planetary gearbox were simulated, including one healthy state, three kinds of faults on the bearing that support a planet gear in the first stage of the planetary gearbox, and four kinds of faults on the sun gear in the first stage of the planetary gearbox. As summarized in Table III, vibration signals were collected under 3 torsional loads and 3 rotation speeds. To be specific, 200 vibration signals were collected under each torsional load and rotation speed, so that each health state has $200 \times 3 \times 3 = 1800$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIE.2020.2972458, IEEE Transactions on Industrial Electronics

observations. Each observation contains 4096 data points, i.e., a 0.32-second vibration signal. Further, to investigate the performance of the developed ResNet-APReLU in fault diagnosis under different amounts of noise, additive white Gaussian noise was manually inserted into each observation to enforce signal-to-noise ratios (SNRs) of 5 dB, 3 dB, and 1 dB, respectively.

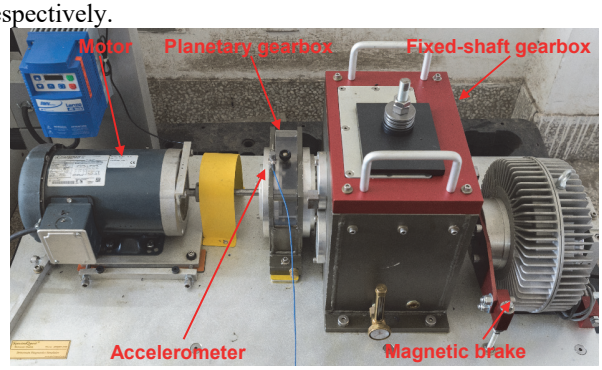


Fig. 4. Drivetrain dynamics simulator for collecting vibration signals.

TABLE II
SUMMARY OF EIGHT HEALTH CONDITIONS OF PLANETARY GEARBOX IN THE EXPERIMENTS

Class	Description	Label
1	No fault in the gearboxes	H
2	Inner race fault on a bearing	B1
3	Outer race fault on a bearing	B2
4	Rolling element fault on a bearing	B3
5	Root cracking fault on a gear	G1
6	Surface pitting fault on a gear	G2
7	Tooth damaging fault on a gear	G3
8	Tooth missing fault on a gear	G4

TABLE III
SUMMARY OF THE OBSERVATIONS UNDER EACH HEALTH CONDITION

Rotation speed (Hz)	Torsional load (lb·ft)	Number of observations under each specific rotation speed and torsional load	Total number of observations in each health condition
20	1, 6, 18	200	
30	1, 6, 18	200	1800
40	1, 6, 18	200	

B. Hyperparameter setup

In the experiment, the same hyperparameters are adopted in the involved deep learning methods with different activation functions. Likewise, although the fine-tune of hyperparameters is still an unsolved challenge that needs much research effort, the focus of this study is to develop a new deep learning method with specially designed architecture, rather than optimizing the hyperparameters. The hyperparameters are set as follows.

The learning rate is 0.1 in the first 40 epochs, 0.01 in next 40 epochs, and 0.001 in the final 20 epochs [40]. The parameters in the deep architecture can be updated with large step sizes at the beginning and slightly tuned in the final stage. Momentum is a training technique that makes use of the updates in the last iteration to accelerate the optimization process; in this study, the momentum ratio is 0.9, as suggested in [9]. The weights are initialized with the method in [43], and the biases are initialized as zeros. L2 regularization is used to reduce overfitting, which pushes the weights towards zero through adding a penalty term in the error function. The coefficient of L2 regularization is 0.0001, which follows the setup in [32]. Mini-batch is a group of randomly selected signals that are propagated into a network at the same time; the mini-batch size is 128, following a setup in [40]. Padding with four zeros is applied to the two ends of each training signal, and random cropping with a length of 4096 is used for data augmentation, which further prevents overfitting.

Likewise, the numbers of convolutional kernels are 4 in the first convolutional layer and the first third of the ResBlocks, 8 in the next one third of the ResBlocks, and 16 in the final one third of the ResBlocks, which follows [34]. The length of the convolutional kernels is 3 in all the layers. It is notable that the length of convolutional kernels can be larger when dealing with larger datasets. The setup in this study is only for verifying the superiority of the method. The number of ResBlocks M is set to 6, 9, and 12, to test the performance with different depths.

TABLE IV
EXPERIMENTAL RESULTS, WHERE M IS THE NUMBER OF RESBLOCKS OR CONVBLOCKS (I.E., RESBLOCKS WITHOUT IDENTITY SHORTCUTS) (%)

M	Method	Training accuracy			Test accuracy		
		5 dB	3 dB	1 dB	5 dB	3 dB	1 dB
6	ConvNet-Sigmoid	76.81 ± 6.44	72.20 ± 3.55	66.72 ± 4.33	75.95 ± 6.14	71.51 ± 3.70	66.37 ± 4.49
	ConvNet-Tanh	88.11 ± 3.24	85.13 ± 2.76	78.69 ± 4.42	87.21 ± 3.33	84.01 ± 2.80	77.87 ± 4.77
	ConvNet-ReLU	91.94 ± 1.07	89.02 ± 2.03	85.23 ± 1.58	90.63 ± 1.19	87.64 ± 2.27	83.92 ± 2.16
	ConvNet-LReLU	91.03 ± 1.02	88.88 ± 1.75	85.15 ± 1.57	90.55 ± 1.09	87.88 ± 2.06	84.20 ± 1.80
	ConvNet-PReLU	94.50 ± 0.74	91.41 ± 1.50	87.85 ± 1.75	93.77 ± 0.97	90.19 ± 1.75	86.10 ± 2.31
	ConvNet-APReLU	97.70 ± 0.65	96.81 ± 1.10	95.17 ± 0.70	97.14 ± 0.99	96.15 ± 1.20	94.15 ± 0.89
	ResNet-Sigmoid	80.13 ± 3.35	75.21 ± 2.59	68.19 ± 5.38	79.51 ± 3.36	74.63 ± 2.31	67.97 ± 4.33
	ResNet-Tanh	89.50 ± 1.70	84.30 ± 3.88	79.70 ± 1.82	88.69 ± 2.17	83.28 ± 3.94	78.85 ± 1.48
	ResNet-ReLU	94.42 ± 0.83	91.79 ± 0.47	87.73 ± 0.76	93.99 ± 0.91	90.55 ± 0.85	86.63 ± 0.73
	ResNet-LReLU	94.53 ± 0.71	91.86 ± 0.74	87.49 ± 0.85	93.47 ± 1.12	91.15 ± 1.31	86.21 ± 0.76
	ResNet-PReLU	95.24 ± 0.58	92.75 ± 0.72	89.11 ± 1.32	94.50 ± 0.82	91.86 ± 0.87	88.02 ± 2.32
	ResNet-APReLU	98.75 ± 0.19	98.07 ± 0.36	96.43 ± 0.37	98.36 ± 0.37	97.07 ± 0.49	95.57 ± 0.76
9	ConvNet-Sigmoid	82.26 ± 2.51	77.17 ± 3.90	72.91 ± 6.03	81.63 ± 2.74	76.44 ± 4.06	71.78 ± 5.88
	ConvNet-Tanh	89.65 ± 2.38	87.25 ± 1.35	82.33 ± 2.78	88.72 ± 2.62	86.36 ± 1.49	81.18 ± 3.43
	ConvNet-ReLU	90.53 ± 2.49	87.22 ± 1.69	81.91 ± 4.87	89.40 ± 2.69	85.01 ± 1.79	80.13 ± 5.02
	ConvNet-LReLU	92.14 ± 1.47	87.52 ± 2.28	82.98 ± 3.16	91.11 ± 2.27	86.54 ± 3.06	81.79 ± 3.48
	ConvNet-PReLU	95.12 ± 1.47	92.55 ± 1.87	89.66 ± 2.93	94.29 ± 1.85	91.06 ± 1.98	88.65 ± 2.73
	ConvNet-APReLU	97.06 ± 1.11	95.29 ± 1.79	92.39 ± 4.17	96.31 ± 1.15	94.17 ± 1.79	91.24 ± 4.26
	ResNet-Sigmoid	86.75 ± 3.31	80.81 ± 3.90	76.20 ± 3.30	86.25 ± 3.00	79.97 ± 3.66	75.10 ± 2.86
	ResNet-Tanh	93.96 ± 0.96	91.01 ± 1.33	86.51 ± 1.61	93.19 ± 0.88	89.60 ± 1.42	85.12 ± 1.84
	ResNet-ReLU	96.55 ± 0.46	93.97 ± 0.56	91.15 ± 0.95	95.64 ± 0.62	92.66 ± 0.85	89.50 ± 1.29
	ResNet-LReLU	96.58 ± 0.23	94.51 ± 0.69	90.85 ± 0.60	95.83 ± 0.65	93.33 ± 0.49	89.54 ± 0.62
	ResNet-PReLU	97.29 ± 0.64	95.75 ± 0.54	92.42 ± 1.01	96.55 ± 0.97	94.44 ± 0.64	90.73 ± 0.86
	ResNet-APReLU	99.36 ± 0.18	98.50 ± 0.25	97.36 ± 0.46	98.80 ± 0.30	97.78 ± 0.31	96.08 ± 0.49
12	ConvNet-Sigmoid	84.48 ± 3.56	80.01 ± 3.49	73.52 ± 4.81	83.64 ± 3.81	79.28 ± 3.21	72.49 ± 4.49
	ConvNet-Tanh	88.03 ± 4.34	87.14 ± 2.76	83.52 ± 0.95	87.06 ± 4.54	85.69 ± 2.56	81.94 ± 0.93
	ConvNet-ReLU	89.66 ± 2.48	85.73 ± 3.78	81.00 ± 4.82	88.44 ± 2.98	84.24 ± 3.87	79.08 ± 5.26
	ConvNet-LReLU	90.26 ± 2.81	87.44 ± 2.16	81.53 ± 1.83	88.89 ± 3.08	86.30 ± 2.53	79.17 ± 1.98
	ConvNet-PReLU	94.18 ± 1.59	89.83 ± 1.13	88.76 ± 3.14	92.47 ± 2.14	87.75 ± 1.32	86.58 ± 3.95
	ConvNet-APReLU	94.64 ± 2.92	91.26 ± 5.35	82.78 ± 11.36	93.50 ± 3.40	90.10 ± 6.39	81.34 ± 11.54
	ResNet-Sigmoid	88.32 ± 3.34	85.07 ± 1.71	78.27 ± 2.79	87.86 ± 2.86	84.24 ± 1.91	77.31 ± 3.07
	ResNet-Tanh	94.98 ± 1.59	91.74 ± 2.73	89.60 ± 1.37	94.03 ± 1.93	90.23 ± 2.35	87.80 ± 1.37
	ResNet-ReLU	97.20 ± 0.34	95.29 ± 0.38	93.05 ± 0.80	96.03 ± 1.01	93.90 ± 0.78	91.29 ± 1.58
	ResNet-LReLU	97.17 ± 0.36	95.18 ± 0.54	92.77 ± 0.46	96.10 ± 0.43	93.68 ± 0.73	91.14 ± 1.16
	ResNet-PReLU	98.13 ± 0.38	96.92 ± 0.18	94.66 ± 0.82	97.15 ± 0.85	95.49 ± 0.48	93.06 ± 0.97
	ResNet-APReLU	99.36 ± 0.17	99.00 ± 0.18	98.04 ± 0.50	98.86 ± 0.38	98.13 ± 0.55	96.94 ± 0.56

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIE.2020.2972458, IEEE Transactions on Industrial Electronics

C. Performance comparison

In this study, experiments were conducted under a scheme of 10-fold cross validation. The indicator "ConvNet-ReLU" refers to a ConvNet that uses ReLU activation functions, and so is the other. Detailed results have been summarized in Table IV, and the overall average of the accuracies have been given in Table V, which are discussed as follows.

TABLE V
OVERALL AVERAGE OF THE ACCURACIES IN TABLE IV

Method	Training accuracy	Test accuracy
ConvNet-Sigmoid	76.23 ± 4.29	75.45 ± 4.28
ConvNet-Tanh	85.54 ± 2.78	84.45 ± 2.94
ConvNet-ReLU	86.92 ± 2.76	85.39 ± 3.03
ConvNet-LReLU	87.44 ± 2.01	86.27 ± 2.37
ConvNet-PReLU	91.54 ± 1.79	90.10 ± 2.11
ConvNet-APReLU	93.68 ± 3.24	92.68 ± 3.51
ResNet-Sigmoid	79.88 ± 3.30	79.20 ± 3.04
ResNet-Tanh	89.03 ± 1.89	87.87 ± 1.93
ResNet-ReLU	93.46 ± 0.62	92.24 ± 0.96
ResNet-LReLU	93.44 ± 0.58	92.27 ± 0.81
ResNet-PReLU	94.70 ± 0.69	93.53 ± 0.98
ResNet-APReLU	98.32 ± 0.30	97.51 ± 0.47

1) Comparison between the APReLU and the classical activation functions

As indicated in Table V, the APReLU is superior to the traditional activation functions, including sigmoid, tanh, ReLU, LReLU, and PReLU. To be specific, the ConvNet-APReLU achieved an overall average test accuracy of 92.68%, which is 17.23%, 8.23%, 7.29%, 6.41%, and 2.58% higher than the ConvNets with sigmoid, tanh, ReLU, LReLU, and PReLU, respectively; the ResNet-APReLU achieved an overall average test accuracy of 97.51%, yielding 18.31%, 9.64%, 5.27%, 5.24%, and 3.98% improvements compared to the ResNets with sigmoid, tanh, ReLU, LReLU, and PReLU, respectively.

Then, a nonlinear dimension reduction method, t-distributed stochastic neighbor embedding (t-SNE) [51] is used to project the learned features at the final GAP layer to 2D spaces. Note that the 2D features generated from t-SNE are highly dependent on the neighboring relations among the observations, which can be different for most datasets. Although the low-dimensional features generally loss much information after dimensionality reduction, the objective of 2D visualizations is only to give an intuitive idea on the characteristics of learned features, rather than using the 2D visualizations for fault classification.

As shown in Fig. 5, the health states become more separable in the ConvNet-APReLU. For example, the observations under the fault "B2" are distinguishable from the other health states in the ConvNet-APReLU, but inseparable in the other ConvNets. As shown in Fig.6, the health states are almost totally separable in the ResNet-APReLU, and only a few misclassifications can be observed. In contrast, the health states are highly overlapped in the other ResNets; further, the observations under the same health state mostly distribute in a few different regions which are distant from each other. The reason is that the vibration signals were collected under various operating conditions and have different characteristics. It is challenging for the classical deep learning methods to project them into the same region. As a consequence, the developed APReLU is helpful to learn a set of highly discriminative features for classification. In addition, Fig. 7 depicts the error tendencies in the training process. It can be observed that the ConvNet-APReLU resulted in the lowest training and test errors among the four ConvNets; likewise, the ResNet-APReLU yielded lower training and test errors than the other three ResNets as well, which validated that the developed APReLU is helpful in optimizing the error towards zero.

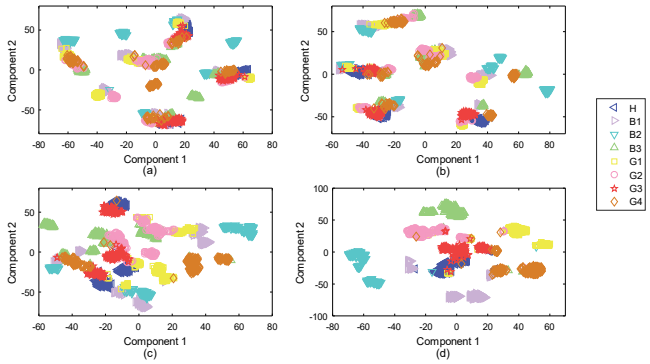


Fig. 5. Visualizations of learned features at the final GAP layer of (a) ConvNet-ReLU, (b) ConvNet-LReLU, (c) ConvNet-PReLU, and (d) ConvNet-APReLU, when the number of ConvBlocks equals 9 and SNR = 5 dB. A ConvBlock is different from the ResBlock in that the ConvBlock does not have the identity shortcut.

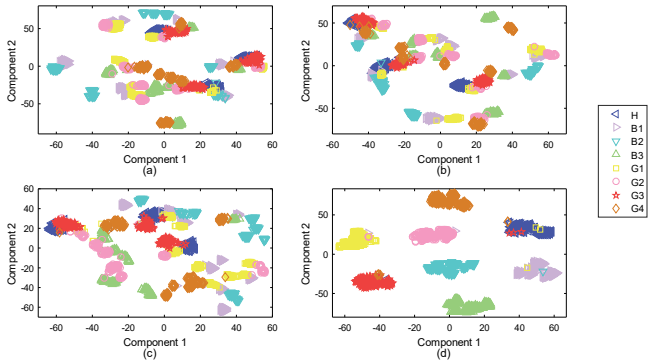


Fig. 6. Visualizations of learned features at the final GAP layer of (a) ResNet-ReLU, (b) ResNet-LReLU, (c) ResNet-PReLU, and (d) the developed ResNet-APReLU, when the number of ResBlocks equals 9 and SNR = 5 dB.

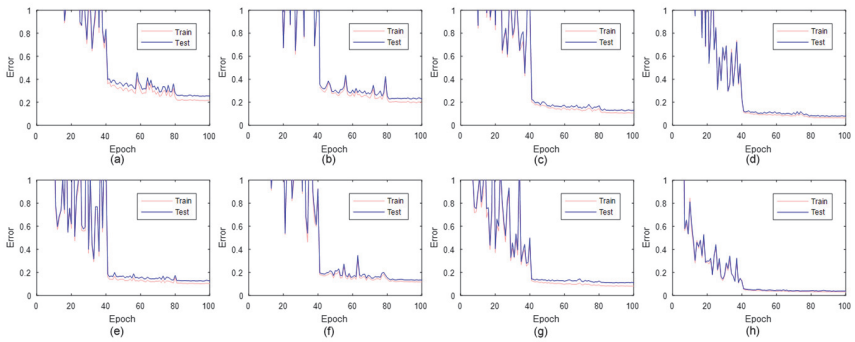


Fig. 7. The varying tendency of errors of (a) ConvNet-ReLU, (b) ConvNet-LReLU, (c) ConvNet-PReLU, (d) ConvNet-APReLU, (e) ResNet-ReLU, (f) ResNet-LReLU, (g) ResNet-PReLU, and (h) ResNet-APReLU, when the number of building blocks equals 9 and SNR = 5 dB..

2) The efficacy of the ResNet-APReLU compared to the ConvNet-APReLU

As indicated in Table V, the ResNet-APReLU is superior to the ConvNet-APReLU by yielding an improvement of 4.83% in terms of overall average test accuracy; likewise, the average standard deviation of test accuracies of the ResNet-APReLU is 0.47%, which is 3.04% lower than that of the ConvNet-APReLU, which validated that the ResNet-APReLU can achieve higher and more stable test accuracy than the ConvNet-APReLU.

Furthermore, although the ConvNet-APReLU yielded higher accuracy than the ConvNets with other activation functions in most cases, it suffered low accuracy when $M = 12$ and SNR = 1 dB. The reason is that the inclusion of APReLUs increases the complexity of the architecture, and it is difficult to optimize the parameters in the ConvNet-APReLU with many layers. As a contrary, the developed ResNet-APReLU achieved the highest training and test accuracy among the ResNets in each case. The identity shortcuts in the ResNet-APReLU can greatly facilitate the flowing of gradients and ease the difficulty of optimization, so that the diagnostic performance can be more stable.

As shown in Figs. 5(d) and 6(d), the health states are more separable in the ResNet-APReLU than the ConvNet-APReLU. For example, the fault "G3" is overlapped with the healthy state "H" in the ConvNet-APReLU; they are very distinguishable in the developed ResNet-APReLU. Besides, as shown in Figs. 7(d) and 7(h), both the training and test errors of ResNet-APReLU are lower than ConvNet-APReLU. Thus, the ResNet-APReLU is more effective in learning highly discriminative features and easier to be optimized than the ConvNet-APReLU.

3) The superiority of the structure of the developed APReLU compared to the other structures

As indicated in Fig. 3(a), the input feature map is not only propagated into a min(x, 0) function and a GAP, but also propagated into a ReLU activation function and a GAP. After that, two 1D vectors are obtained, concatenated, and propagated to the subsequent layers. In order to verify the superiority of this structure, an experimental comparison with another three structures is conducted. As shown in Table VI, "average" means that the input feature map is directly propagated to a GAP; "positive" means that the input feature map is propagated to a ReLU activation function and a GAP; "negative" means that the input feature map is propagated to a min(x, 0) function and a GAP. Then, the output feature map of the GAP is propagated into the calculation path (i.e., FC→BN →ReLU→FC→BN→Sigmoid), which is exactly the same as the developed APReLU.

The experimental configurations and hyperparameters are the same as the experiments in Table IV with $M = 6$. The training and test accuracies are summarized in Table VI. It can be observed that the developed APReLU achieved the highest training and test accuracies when it is compared with the other structures.

TABLE VI
ACCURACIES OF THE RESNET-APRELUS WITH DIFFERENT STRUCTURES

Method	Training accuracy	Test accuracy
Average	93.18 ± 0.74	92.31 ± 0.87
Positive	96.04 ± 0.44	95.04 ± 0.68
Negative	95.71 ± 0.38	94.75 ± 0.60
Developed	96.43 ± 0.59	95.38 ± 0.76

4) An analysis on the values of the slopes

An example of the values of the slopes on negative features is summarized in Table VII (see Eq. 10). Eight vibration signals, which are different from each other, are taken as the examples to be analyzed. It can be seen that the slope on the negative features is a fixed number (i.e., 0.1) in the LReLU activation function for the 8 vibration signals. The slope becomes a trainable parameter in the PReLU activation function; it is initialized as 0 and trained to be 0.495, which is the same for all the vibration signals as well. In contrast, the learned slopes in the developed APReLU are different for different vibration signals, which is proof that the developed APReLU activation function can assign different nonlinear transformations to different vibration signals.

TABLE VII
THE SLOPES ON THE NEGATIVE FEATURES IN THE ACTIVATION FUNCTIONS

Vibration signal	LReLU	PReLU	APReLU
1	0.1	0.495	0.588
2	0.1	0.495	0.497
3	0.1	0.495	0.575
4	0.1	0.495	0.580
5	0.1	0.495	0.699
6	0.1	0.495	0.714
7	0.1	0.495	0.513
8	0.1	0.495	0.590

D. Further verification on a public dataset

The effectiveness of the developed ResNet-APReLU was also validated on a public gearbox dataset, i.e., PHM 2009 challenge dataset [52]. Eight health states were considered. Each health state has 1000 observations, and each observation is a vibration signal containing 4096 data points. Moreover, white Gaussian noise was added to each vibration signal to yield a SNR of 1 dB. The same hyperparameters as the above experiment were used. The number of ResBlocks was set to 12. More details about the dataset can be found in [52].

The experimental results are summarized in Table VIII. The developed ResNet-APReLU not only yielded the highest training accuracy (i.e., 99.99%), but also achieved the highest test accuracy (i.e., 99.55%) among the considered methods. However, the ResNet-APReLU takes more optimization time than the other methods. Therefore, the architecture of the ResNet-APReLU should be optimized in the future to reduce time consumption.

TABLE VIII
EXPERIMENTAL RESULTS ON THE PUBLIC DATASET

Method	Training accuracy (%)	Test accuracy (%)	Time (second)
ConvNet-Sigmoid	94.37 ± 1.74	92.54 ± 1.43	384.72
ConvNet-Tanh	96.92 ± 1.72	95.05 ± 1.76	385.33
ConvNet-ReLU	99.13 ± 0.33	96.60 ± 0.90	385.44
ConvNet-LReLU	99.20 ± 0.30	97.39 ± 0.77	413.22
ConvNet-PReLU	99.70 ± 0.11	98.15 ± 0.84	430.89
ConvNet-APReLU	99.30 ± 1.46	98.43 ± 1.92	920.59
ResNet-Sigmoid	95.33 ± 1.12	93.99 ± 1.68	390.18
ResNet-Tanh	98.64 ± 0.33	97.14 ± 0.77	389.68
ResNet-ReLU	99.54 ± 0.15	98.03 ± 0.65	388.12
ResNet-LReLU	99.39 ± 0.13	98.31 ± 0.36	414.50
ResNet-PReLU	99.78 ± 0.27	98.53 ± 0.51	434.08
ResNet-APReLU	99.99 ± 0.01	99.55 ± 0.33	924.38

IV. CONCLUSION

This paper develops a new activation function, i.e., APReLUs, which can be easily inserted into ResBlocks and any locations of deep neural networks, without making any other modifications. The feature learning ability of deep neural networks can be improved through assigning different nonlinear transformations to the input signals, so that the ultimate goal of yielding high diagnostic accuracy can be achieved. Specifically, a sub-network is designed to automatically learn slopes to be used in the APReLU, so that

each input signal can have its own nonlinear transformations. As a consequence, the developed ResNet-APReLU has highly flexible nonlinear transformations, and a high performance in projecting intra-class vibration signals into the same region and inter-class vibration signals into different regions.

The superiority of the developed APReLU compared to the traditional activation functions has been demonstrated through experiments on fault diagnosis of planetary gearboxes. As indicated in Table V, the ResNet-APReLU yielded improvements of 5.27%, 5.24%, and 3.98% compared to the other ResNets with ReLU, LReLU, and PReLU activation functions, in terms of overall average test accuracy. These improvements were because of the use of APReLU in achieving adaptive nonlinear transformation, which enables the coefficients in nonlinear transformation to be adjustable according to the input signal. Hence, the developed ResNet-APReLU gained a high discriminative feature learning ability and improved the diagnostic accuracy.

Likewise, the developed ResNet-APReLU outperformed the ConvNet-APReLU by achieving an improvement of 4.83% in terms of overall average test accuracy (see Table V). The APReLU, in deed, is more complex and difficult to be optimized than the other considered activation functions, including ReLU, LReLU, and PReLU. If the parameters in ConvNet-APReLU was not fully optimized, it will not be able to diagnose the faults accurately. In contrast, the ResNet-APReLU eases the difficulty of training through the usage of identity shortcuts, so that the parameters in the network can be effectively optimized.

Finally, the developed APReLU can be easily inserted into deep transfer learning methods for a higher diagnostic accuracy under non-stationary operation conditions, and also applicable to the other deep learning methods, such as deep auto-encoders, capsule networks, and generative adversarial networks.

REFERENCES

[1] M. Pecht and M. Kang, *Prognostics and health management of electronics: Fundamentals, machine learning, and the Internet of things*, Wiley, New York, NY, 2018.

[2] P. Henriquez, J.B. Alonso, M.A. Ferrer, and C.M. Travieso, "Review of automatic fault diagnosis systems using audio and vibration signals," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 44, no. 5, pp. 642–652, 2014.

[3] S. Lu, Q. He, and J. Wang, "A review of stochastic resonance in rotating machine fault detection," *Mech. Syst. Signal Process.*, vol. 116, pp. 230–260, 2019.

[4] X. Jin, F. Cheng, Y. Peng, W. Qiao, and L. Qu, "Drivetrain gearbox fault diagnosis: Vibration- and current-based approaches," *IEEE Ind. Electron. Mag.*, vol. 24, no. 6, pp. 56–66, 2018.

[5] O.E. Hassan, M. Amer, A.K. Abdelsalam, et al., "Induction motor broken rotor bar fault detection techniques based on fault signature analysis – a review," *IET Electr. Power App.*, vol. 12, no. 7, pp. 895–907, 2018.

[6] B. Hou, Y. Wang, B. Tang, Y. Qin, Y. Chen, and Y. Chen, "A tacholess order tracking method for wind turbine planetary gearbox fault detection," *Measurement*, vol. 138, pp. 266–277, 2019.

[7] R. Liu, B. Yang, E. Zio, and X. Chen, "Artificial intelligence for fault diagnosis of rotating machinery: A review," *Mech. Syst. Signal Process.*, vol. 108, pp. 33–47, 2018.

[8] F. Chen, B. Tang, and R. Chen, "A novel fault diagnosis model for gearbox based on wavelet support vector machine with immune genetic algorithm," *Measurement*, vol. 46, no. 1, pp. 220–232, 2013.

[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[10] H. Oh, J. Jung, B. Jeon, and B. Youn, "Scalable and unsupervised feature engineering using vibration-imaging and deep learning for rotor system diagnosis," *IEEE Trans. Ind. Electron.*, vol. 65, no. 4, pp. 3539–3549, 2018.

[11] Z. Liu, Z. Jia, C. Vong, S. Bu, J. Han, and X. Tang, "Capturing high-discriminative fault features for electronics-rich analog system via deep learning," *IEEE Trans. Ind. Inform.*, vol. 13, no. 3, pp. 1213–1226, 2017.

[12] Y. Qin, X. Wang, and J. 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.*, vol. 66, no. 5, pp. 3814–3824, 2019.

[13] H. Shao, H. Jiang, K. Zhao, et al., "A novel tracking deep wavelet auto-encoder method for intelligent fault diagnosis of electric locomotive bearings," *Mech. Syst. Signal Process.*, vol. 110, pp. 193–209, 2018.

[14] T. Ince, S. Kiranyaz, L. Eren, M. Askar, and M. Gabbouj, "Real-time motor fault detection by 1-D convolutional neural networks," *IEEE Trans. Ind. Electron.*, vol. 63, no. 11, pp. 7067–7075, 2016.

[15] R. Chen, X. Huang, L. Yang, X. Xu, X. Zhang, and Y. Zhang, "Intelligent fault diagnosis method of planetary gearboxes based on convolution neural network and discrete wavelet transform," *Comput. Ind.*, vol. 106, pp. 48–59, 2019.

[16] G. Jiang, H. He, J. Yan, and P. Xie, "Multiscale convolutional neural networks for fault diagnosis of wind turbine gearbox," *IEEE Trans. Ind. Electron.*, vol. 66, no. 4, pp. 3196–3207, 2019.

[17] M. Xia, T. Li, L. Xu, et al. "Fault diagnosis for rotating machinery using multiple sensors and convolutional neural networks," *IEEE/ASME Trans. Mechatron.*, vol. 23, no. 1, pp. 101–110, 2018.

[18] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-driven fault diagnosis method," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5990–5998, 2018.

[19] X. Ding and Q. He, "Energy-fluctuated multiscale feature learning with deep ConvNet for intelligent spindle bearing fault diagnosis," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 8, pp. 1926–1935, 2017.

[20] A.S.M. Shihavuddin, X. Chen, V. Fedorov, et al., "Wind turbine surface damage detection by deep learning aided drone inspection analysis," *Energies*, vol. 12, no. 4, article no. 676, 2019.

[21] F. Jia, Y. Lei, N. Lu, and S. Xing, "Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization," *Mech. Syst. Signal Process.*, vol. 110, pp. 349–367, 2018.

[22] J. Pan, Y. Zi, J. Chen, Z. Zhou, and B. Wang, "LiftingNet: A novel deep learning network with layerwise feature learning from noisy mechanical data for fault classification," *IEEE Trans. Ind. Electron.*, vol. 65, no. 6, pp. 4973–4982, 2018.

[23] Y. Han, B. Tang, and L. Deng, "An enhanced convolutional neural network with enlarged receptive fields for fault diagnosis of planetary gearboxes," *Comput. Ind.*, vol. 107, pp. 50–58, 2019.

[24] L. Guo, Y. Lei, S. Xing, T. Yan, and N. Li, "Deep convolutional transfer learning network: A new method for intelligent fault diagnosis of machines with unlabeled data," *IEEE Trans. Ind. Electron.*, In Press.

[25] J. Jiao, M. Zhao, J. Lin, and J. Zhao, "A multivariate encoder information based convolutional neural network for intelligent fault diagnosis of planetary gearboxes," *Knowl.-Based Syst.*, vol. 160, pp. 237–250, 2018.

[26] W. Sun, R. Zhao, R. Yan, S. Shao, and X. Chen, "Convolutional discriminative feature learning for induction motor fault diagnosis," *IEEE Trans. Ind. Inform.*, vol. 13, no. 3, pp. 1350–1359, 2017.

[27] 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, 2019.

[28] R. Razavi-Far, E. Hallaji, M. Farajzadeh-Zanjani, M. Saif, S. H. Kia, H. Henao, and G. Capolino, "Information fusion and semi-supervised deep learning scheme for diagnosing gear faults in induction machine systems," *IEEE Trans. Ind. Electron.*, vol. 66, no. 8, pp. 6331–6342, 2019.

[29] C. Sun, M. Ma, Z. Zhao, and X. Chen, "Sparse deep stacking network for fault diagnosis of motor," *IEEE Trans. Ind. Inform.*, vol. 14, no. 7, pp. 3261–3270, 2018.

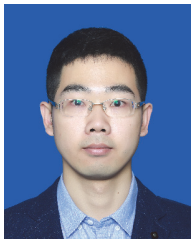
[30] X. Li, W. Zhang, and Q. Ding, "Cross-domain fault diagnosis of rolling element bearings using deep generative neural networks," *IEEE Trans. Ind. Electron.*, vol. 66, no. 7, pp. 5525–5534, 2019.

[31] K. Zhang, B. Tang, Y. Qin, et al., "Fault diagnosis of planetary gearbox using a novel semi-supervised method of multiple association layers networks," *Mech. Syst. Signal Process.*, vol. 131, pp. 243–260, 2019.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Seattle, WA, USA, Jun. 27–30, 2016, pp. 770–778.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual

networks," in *Computer Vision—ECCV 2016* (Lecture Notes in Computer Science 9908), B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Cham, Switzerland: Springer, 2016, pp. 630–645.
[34] M. Zhao, M. Kang, B. Tang, and M. Pecht, "Multiple wavelet coefficients fusion in deep residual networks for fault diagnosis," *IEEE Trans. Ind. Electron.*, vol. 66, no. 6, pp. 4696–4706, 2019.
[35] W. Zhang, X. Li, and Q. Ding, "Deep residual learning-based fault diagnosis method for rotating machinery," *ISA Trans.*, In Press.
[36] M. Zhao, S. Zhong, X. Fu, B. Tang, and M. Pecht, "Deep residual shrinkage networks for fault diagnosis," *IEEE Trans. Ind. Inform.*, 2019, DOI: 10.1109/TII.2019.2943898.
[37] L. Wen, X. Li, and L. Gao, "A transfer convolutional neural network for fault diagnosis based on ResNet-50," *Neural Comput. Appl.*, In Press.
[38] M. Zhao, B. Tang, L. Deng, and M. Pecht, "Multiple wavelet regularized deep residual networks for fault diagnosis," *Measurement*, In Press.
[39] S. Ma, F. Chu, and Q. Han, "Deep residual learning with demodulated time-frequency features for fault diagnosis of planetary gearbox under nonstationary running conditions," *Mech. Syst. Signal Process.*, vol. 127, pp. 190–201, 2019.
[40] M. Zhao, M. Kang, B. Tang, and M. Pecht, "Deep residual networks with dynamically weighted wavelet coefficients for fault diagnosis of planetary gearboxes," *IEEE Trans. Ind. Electron.*, vol. 65, no. 5, pp. 4290–4300, 2018.
[41] V. Nair and G.E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. Int. Conf. Mach. Learn.*, Haifa, Israel, Jun. 21–24, 2010, pp. 807–814.
[42] A.L. Maas, A.Y. Hannun, and A.Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learn.*, Atlanta, Georgia, USA, Jun. 16–21, 2013.
[43] K He, X Zhang, S Ren, and J Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vision*, Santiago, Chile, Dec. 7–13, 2015, pp. 1026–1034.
[44] W. Huang, G. Gao, N. Li, X. Jiang, and Z. Zhu, "Time-frequency squeezing and generalized demodulation combined for variable speed bearing fault diagnosis," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 8, pp. 2819–2829, 2019.
[45] R. Ding, J. Shi, X. Jiang, et al., "Multiple instantaneous frequency ridge based integration strategy for bearing fault diagnosis under variable speed operations," *Meas. Sci. Technol.*, vol. 29, article no. 115002, 2018.
[46] Z. An, S. Li, J. Wang, et al., "Generalization of deep neural network for bearing fault diagnosis under different working conditions using multiple kernel method," *Neurocomputing*, vol. 352, pp. 42–53, 2019.
[47] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 18–23, 2018, pp. 7132–7141.
[48] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, Lille, France, Jul. 7–9, 2015, pp. 448–456.
[49] M. Lin, Q. Chen, and S. Yan, "Network in network," in *Proc. Int. Conf. Learn. Represen.*, Banff, Canada, Apr. 14–16, 2014.
[50] P. Zhou and J. Austin, "Learning criteria for training neural network classifiers," *Neural Comput. Appl.*, vol. 7, no. 4, pp. 334–342, 1998.
[51] L. J. P. van der Maaten and G. E. Hinton, "Visualizing high-dimensional data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.
[52] PHM 2009 challenge dataset. Available: https://c3.nasa.gov/dashlink/resources/997/. Accessed: Nov. 20, 2019.



Minghang Zhao was born in Shandong, China, in June 1991. He received the B.E. and Ph.D. degrees in mechanical engineering from Chongqing University, Chongqing, China, in 2013 and 2018, respectively. He is currently a lecturer of mechanical engineering with the School of Naval Architecture and Ocean Engineering, Harbin Institute of Technology at Weihai, Weihai, China.

He was previously a visiting research scholar with the Center for Advanced Life Cycle Engineering, University of Maryland, College Park, MD, USA, from 2016 to 2017. His research interests include machine learning (especially deep neural networks) powered fault diagnosis, prognostics, and health management of mechanical and electrical systems.

Shisheng Zhong received the M.E. degree in mechanical engineering from Harbin Institute of Technology, Harbin, China, and the Ph.D. degree in mechanical engineering from Huazhong University of Science and Technology, Wuhan, China, in 1992 and 1995, respectively.

He is currently a Professor and Ph.D. Supervisor of mechanical engineering with the School of Naval Architecture and Ocean Engineering, Harbin Institute of Technology at Weihai, Weihai, China. His main research interests include intelligent manufacturing, prognostics and health management, and maintenance, repair and overhaul.

Xuyun Fu received the B.E., M.E., and Ph.D. degrees in mechanical engineering from Harbin Institute of Technology, Harbin, China, in 2003, 2007, and 2010, respectively.

He is currently an Associate Professor of mechanical engineering with the School of Naval Architecture and Ocean Engineering, Harbin Institute of Technology at Weihai, Weihai, China. His main research interests include intelligent operation and maintenance, prognostics and health management, aeroengine health management and maintenance decision.

Baoping Tang received the M.Sc. and Ph.D. degrees in mechanical engineering from Chongqing University, Chongqing, China, in 1996 and 2003, respectively.

He is currently a Professor and Ph.D. Supervisor of mechanical engineering with the College of Mechanical Engineering, Chongqing University, Chongqing, China. More than 150 papers have been published in his research career. His main research interests include wireless sensor networks, mechanical and electrical equipment security service and life prediction, and measurement technology and instruments.

Shaojiang Dong received the B.E. degree in mechanical engineering from Hohai University, Nanjing, China, the M.E. degree in mechanical engineering from Chongqing University of Technology, Chongqing, China, and the Ph.D. degree in mechanical engineering from Chongqing University, Chongqing, China, in 2006, 2009, and 2012, respectively.

He is currently a Professor and Ph.D. Supervisor of mechanical engineering with the School of Mechatronics and Automotive Engineering, Chongqing Jiaotong University, Chongqing, China. His research interest includes equipment condition monitoring, fault diagnosis, and signal processing.

Michael Pecht (S'78–M'83–SM'90–F'92) received the B.S. degree in acoustics, M.S. degrees in electrical engineering and engineering mechanics, and Ph.D. degree in engineering mechanics from the University of Wisconsin at Madison, Madison, WI, USA, in 1976, 1978, 1979, and 1982, respectively.

He is the Founder of the Center for Advanced Life Cycle Engineering, University of Maryland, College Park, MD, USA, where he is also a Chair Professor. He is a Professional Engineer and a Fellow of the American Society of Mechanical Engineers. He was the recipient of the IEEE Undergraduate Teaching Award and the International Microelectronics Assembly and Packaging Society William D. Ashman Memorial Achievement Award for his contributions in electronics reliability analysis.