Journal of Dynamics, Monitoring and Diagnostics, 2022, 1, 160-168
https://doi.org/10.37965/jdmd.2022.30

ISTP
RESEARCH ARTICLE

Bearings Intelligent Fault Diagnosis by 1-D Adder Neural Networks

Jian Tang,[1] Chao Wei,[1] Quanchang Li,[2] Yinjun Wang,[1] Xiaoxi Ding,[1] and Wenbin Huang[1]
[1]College of Mechanical Engineering, Chongqing University, Chongqing 400044, China
[2]Department of Mechanical and Materials Engineering, Queen's University, Kingston, Canada

(Received 30 July 2021; Revised 16 April 2022; Accepted 28 July 2022; Published online 08 August 2022)

Abstract: Integrated with sensors, processors, and radio frequency (RF) communication modules, intelligent bearing could achieve the autonomous perception and autonomous decision-making, guarantying the safety and reliability during their use. However, because of the resource limitations of the end device, processors in the intelligent bearing are unable to carry the computational load of deep learning models like convolutional neural network (CNN), which involves a great amount of multiplicative operations. To minimize the computation cost of the conventional CNN, based on the idea of AdderNet, a 1-D adder neural network with a wide first-layer kernel (WAddNN) suitable for bearing fault diagnosis is proposed in this paper. The proposed method uses the l_1-norm distance between filters and input features as the output response, thus making the whole network almost free of multiplicative operations. The whole model takes the original signal as the input, uses a wide kernel in the first adder layer to extract features and suppress the high frequency noise, and then uses two layers of small kernels for nonlinear mapping. Through experimental comparison with CNN models of the same structure, WAddNN is able to achieve a similar accuracy as CNN models with significantly reduced computational cost. The proposed model provides a new fault diagnosis method for intelligent bearings with limited resources.

Keywords: adder neural network; convolutional neural network; fault diagnosis; intelligent bearings; l_1-norm distance

I. INTRODUCTION

Known as the "heart" of the rotary support system, bearings are used in a wide variety of applications such as aerospace, high-speed rail and automotive wheels, large rotors and precision machine tools, etc. [1]. Machine performance is heavily dependent on the health of the bearings, and bearing failure can even carry a life-threatening risk. Therefore, it is of great significance to accurately monitor and diagnose bearings during operation [2]. In traditional fault diagnosis, the condition monitoring of bearings mainly relies on engineers' long-term experience and expertise in the fault diagnosis field. Attributed to the development of the artificial intelligence technology, bearing fault diagnosis moves toward intelligent self-diagnosis so that the health status of bearings can be automatically detected and identified [3].

Intelligent is the development trend of modern industry. Under this background, the development of high-end bearings is inevitably inseparable from the intelligent trend, and intelligent bearings emerge at the historic moment [4,5]. The intelligent bearing is based on the traditional bearing integrated with different sensors, microprocessor, and control system so that the combination into one and form a unique bearing structure unit. Figure 1 shows a sketch of the system structure. Compared with ordinary bearings, intelligent bearings have the characteristics of self-sensing, self-decision, and self-regulation. In terms of autonomous decision-making of bearings, machine learning technology is usually used for real-time state monitoring of bearings. However, these integrated bearings usually adopt battery-powered or low-power output self-powered technology due to the limitation of industrial application environment. Therefore, intelligent bearing fault diagnosis model needs to keep high precision and reduce energy consumption.

In the field of intelligent bearing fault diagnosis, a considerable number of traditional machine learning models have been effectively utilized, for example, classical support vector machines (SVMs) [6], artificial neural networks (ANNs) [7], computationally simple probabilistic graphical models (PGMs) [8], and k-nearest neighbors (kNN) [9]. In these traditional classical fault diagnosis methods, features are extracted manually by designing the corresponding algorithms and then imported into machine learning models to intelligently identify the bearing health status. These traditional machine learning models have achieved the intelligence of bearing diagnosis to a certain extent, and the bearing health status can be obtained without manual judgment. However, precise manual extraction of features is still required throughout the process to obtain a high accuracy rate. Besides, with the development of cloud technology and the advent of the era of big data, the bearing data explode, and traditional machine learning cannot achieve good generalization performance under such volume of data, which reduces the accuracy of diagnosis. [10]. Therefore, the traditional machine learning model cannot meet the needs of high-precision diagnosis performance of high-end bearings.

As machine learning enters the era of deep learning, bearing intelligent fault diagnosis also changes from traditional machine learning models to deep learning models [11]. Deep learning techniques have achieved great success in bearing fault diagnosis due to their powerful feature extraction capability and end-to-end diagnostic approach. By the deep mining of the initial signal features, the fault

Corresponding author: Wenbin Huang, (e-mails: whaung@cqu.edu.cn; dxxu@cqu.edu.cn).

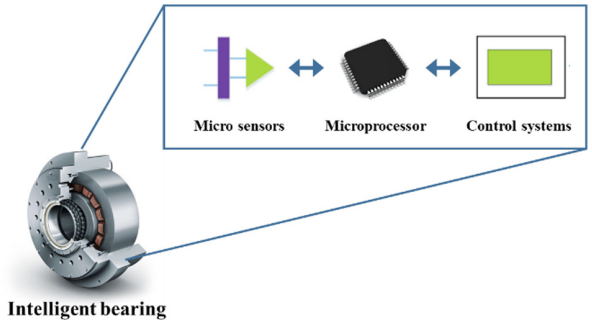

FIGURE 1. Intelligent bearing system.

features hidden in the vibration signals can be extracted accurately, and a fault recognition rate that far exceeds that of traditional machine learning can be achieved. In addition, the end-to-end diagnostic model frees human labor from feature extraction [12].

In the field of deep learning, convolutional neural networks (CNNs) have been widely used in the field of bearing fault diagnosis [13,14,15,16,17]. Compared with other deep learning models such as stacked autoencoder (AE) [18] and deep belief network (DBN) [19], CNN models can accurately capture useful features from the original vibration signal directly. More importantly, the CNN-based models share weights, thus drastically reducing the number of parameters and facilitating the training and storage of the models. These advantages make CNN models known as the first choice for bearing diagnosis [20]. However, the convolutional operations and multilayer network architecture cause extremely high computational cost for CNNs, resulting in much higher requirement for the processor in the energy and computational power. Therefore, the implementation of CNNs usually requires high-performance graphics processing unit (GPU) and application-specific integrated circuit (ASIC), and they are hard to be applied in end devices of the Internet of Things (IoT) system with a low-performance microprocessor. To overcome these problems, it is worth exploring how to effectively reduce the computational cost of deep learning fault diagnosis models for bearings.

It is well known that the four basic operations in mathematics are addition, subtraction, multiplication, and division. Among them, the multiplication operation is more expensive to perform in the microprocessor than the addition operation [21]. However, most of the operations in deep learning models are multiplications between floating point numbers. Therefore, there are a large number of papers investigating how to replace the multiplication with addition and thus reduce the computational cost of the model. Courbariaux et al [22] presented BinaryConnect, which enables addition to replace multiplication operations in the network by adjusting the weight parameters throughout the network to binary. Hubala et al [23] proposed a binarized neural network (BNN) network that binarized the weights and activation functions. In addition, Rastegari et al [24] added a scale factor to implement binary convolution operations. Cai et al [25] proposed a half-wave Gaussian quantizer for forward approximation, thus improving the accuracy of the model. Although the binary network reduces the computational cost of the model, the accuracy of the model is not satisfactory. Therefore, Chen et al. [26] proposed a network called AdderNets for converting the multiplication in deep neural networks, especially in CNNs, into the simpler additive operations. The researchers used the l_1-norm distance between the filter and the input features as a feedback for the output. To achieve better performance, the researchers constructed a special backpropagation method and found that this neural network, which uses almost exclusively additive methods, can converge efficiently with excellent speed and accuracy.

With the development of intelligent bearing technology, autonomous diagnosis of bearing ends has become a development trend. Based on the idea of AdderNet, this paper proposes a 1-D wide first-layer kernel adder neural network model for bearing fault diagnosis to reduce the computational cost. We use l_1-norm distance instead of mutual correlation in the whole network to measure the relevance between features and filters, thus making the whole network almost free of multiplication operations. The following contributions of this paper are summarized from the above discussion:

(1) Aiming at the problem that the mainstream CNN diagnosis model cannot be applied to the end processor of the bearing due to the excessive computation, based on the idea of AdderNet, this paper proposes the bearing fault diagnosis model WAddNN to effectively reduce the computation of the model while ensuring the diagnosis accuracy.

(2) WAddNN uses l_1 norm to extract features, which makes the feature extraction layer in the model contain only additive operations, thus significantly reducing the computational overhead of the network.

(3) The effectiveness of WAddNN is verified through experiments. It is shown that the WAddNN proposed in this paper can effectively extract features from vibration signals, showing its great potential to be applied to end processors of bearings.

The rest of the paper is divided into the following sections. In Section II, the theoretical background of AdderNet is discussed. Section III introduces the principle and steps of the WAddNN method. In Section IV, the effectiveness of the proposed WAddNN is verified by the experimental data. Section V draws the final conclusions.

II. A BRIEF INTRODUCTION TO AdderNet

Addition is faster than multiplication in a microprocessor, and the computational cost of addition is much lower than that of multiplication. However, multiplication between floating point numbers takes up almost the entire deep neural network, imposing a huge computational cost. For this reason, an additive network (AdderNet) has been proposed in [26] to exchange these large-scale multiplications in deep neural networks, especially CNNs, in order to obtain easier additions to reduce the computational cost. We will present additive neural networks for 2-D image recognition in this section.

A. 2-D ADDER NETWORK LAYER

The convolution operation in CNN is to compute the mutual correlation between the features and the convolution kernel [27]. As the mutual correlation is a distance metric, the convolution operation can be seen as a way to measure the distance between the features and the convolution kernel.

Given an intermediate convolution kernel $F \in \mathbb{R}^{d\times d\times v_{\text{in}}\times v_{\text{out}}}$ and the input features $X \in \mathbb{R}^{H\times W\times v_{in}}$, where d represents the volume size of the convolution kernel, v_{in} represents the input channel, v_{out} is the output channel, and H and W are the length and width of the features, respectively, the output features of the convolution operation can be calculated as:

$$Y(m,n,t) = \sum_{i=0}^{d}\sum_{j=0}^{d}\sum_{k=0}^{c_{\text{in}}} S(X(m+i,n+j,k),F(i,j,k,t)) \quad (1)$$

where $S(\cdot\,,\cdot)$ represents the distance metric function, and here it is the mutual correlation metric $S(x,y) = x \times y$.

In fact, there are many other metric functions besides the mutual correlation which can be used to extract the relationship between the filter and the input features; however, most of them contain multiplication, which brings a large computational cost. Therefore, using l_1-norm distance, which contains only additive operations, instead of convolutional operations, can reduce the computational cost of the CNN network effectively. By using l_1-norm distance, the output features is calculated as:

$$Y(m,n,t) = -\sum_{i=0}^{d}\sum_{j=0}^{d}\sum_{k=0}^{v_{in}} |X(m+i,n+j,k) - F(i,j,k,t)| \quad (2)$$

The l_1-norm distance (2) and the cross-correlation (1), which are also used to measure the distance between the filter and the input features, still have some differences in their outputs. The most obvious difference is that the output of the convolution filter represents a weighted sum, which has both positive and negative results. Whereas the results of the additive filter are only negative as can be seen in (2). So using the same calculation process leads to a continuous increase in the output of the additive layer. For this reason, the additive network uses batch normalization (BN) techniques in the calculation process to keep the output constant to a range that is conducive to network training optimization. By changing the metric to the l_1-norm distance, it is possible to use only adders in the network to extract features and build adder neural networks on top of this.

B. OPTIMIZATION METHOD

In the training process of neural networks, the gradients of the parameters are obtained by backpropagation and then optimized by the gradient descent method. Similarly in additive neural networks, gradient descent is also used. Therefore, the partial derivatives of the output features Y with respect to the filter F are calculated as:

$$\frac{\partial Y(m,n,t)}{\partial F(i,j,k,t)} = \text{sgn}(X(m+i,n+j,k) - F(i,j,k,t)) \quad (3)$$

where $\text{sgn}(\cdot)$ represents the sign function.

As can be seen from (3), the output of the partial derivative has only three values, i.e., $+1$, 0, or -1. Thus, (3) is optimized by a method called signSGD. However, the signSGD does not descend along the optimal path during optimization, and more seriously, the optimal path deviates more from the model after the dimensionality increases, which is detrimental to the optimization of the model. For this reason, Chen [26] suggested using (4) to update the gradient:

$$\frac{\partial Y(m,n,t)}{\partial X(m+i,n+j,k)} = X(m+i,n+j,k) - F(i,j,k,t) \quad (4)$$

After processing the gradient of the filter F, the same needs to be done for the gradient of the input feature X to facilitate the optimal update of the parameters. However, using (4) to calculate the gradient of X, the obtained results will exceed $[-1, +1]$, which is unbeneficial to the optimization of the network. If F_n and X_n are used to denote the filter and input features of the n_{th} layer, respectively, then the gradient calculation of $\partial Y/\partial F_n$ will only have an effect on the gradient of F_n itself. While $\partial Y/\partial X_n$ will have an effect on the gradients of all previous network layers in the chain rule. In particular, using (4) to calculate the gradient of input Y can easily lead to a serious consequence of gradient explosion when the number of layers increases. Therefore, the gradient of input X needs to be limited within $[-1, +1]$. The calculation formula is calculated as:

$$\frac{\partial Y(m,n,t)}{\partial X(m+i,n+j,k)} = \text{H}(F(i,j,k,t) - X(m+i,n+j,k)) \quad (5)$$

where $\text{H}(\cdot)$ represents the function of HardTanh:

$$\text{H}(x) = \begin{cases} 1 & x > 1 \\ x & -1 < x < 1, \\ -1 & x < -1. \end{cases} \quad (6)$$

C. ADAPTIVE LEARNING RATE SCALING

The gradient in AdderNets is too small compared to the gradient of the filter in CNNs to provide the required parameter update rate for the whole network due to the difference in calculation methods. To address this problem, Chen [26] proposed an adaptive algorithm that takes a different learning rate for each layer in the adder network. The learning rate of the nth layer in the network is calculated as:

$$\Delta F_n = \varepsilon \times \phi_n \times \Delta L(F_n) \quad (7)$$

where ε represents the global learning rate, including the additive and BN layers, $\Delta L(F_n)$ represents the gradient of the filter at the n_{th} layer, and ϕ_{n} represents the local learning rate at the n_{th} layer. The purpose of subtracting the additive filter and the input features is to obtain useful feature information from the input features, so it is necessary to keep the size of the filter and the input features within the same range as much as possible. The input features have been normalized after BN layers, so only the normalization operation needs to be performed for each layer of the filter. The local learning rate is calculated as:

$$\phi_n = \frac{\phi\sqrt{d}}{\|\Delta L(F_n)\|} \quad (8)$$

where d represents the number of elements in the n_{th} layer filter and ϕ is a hyperparameter to control the magnitude of the learning rate. Adjusting the learning rate by the adaptive algorithm enables the whole additive network to be trained more efficiently.

III. PROPOSED WAddNN INTELLIGENT DIAGNOSIS METHOD

In comparison with traditional fault diagnosis methods, deep learning methods represented by CNN models have

powerful feature mining capabilities and the ability to build end-to-end diagnosis systems. Therefore, CNNs have a wide range of applications in the field of bearing fault diagnosis. However, with the development of intelligent bearings, the end processors embedded in bearings cannot directly apply deep learning models to complete fault autonomous diagnosis because of the limitation of computational resources and energy. To solve the above problems, this paper proposes the WAddNN model to reduce the consumption of computational resources. Using the adder layer instead of the convolutional layer in WAddNN, the whole network contains only additive operations compared to the CNN model without losing diagnostic accuracy. In microprocessors, the operation cycle of multiplication is much larger than that of addition, so WAddNN uses addition instead of multiplication to effectively save the working time of the processor and reduce the consumption of computational resources.

A. WAddNN

The model structure of WAddNN is shown in Fig. 2. In order to reduce the calculating cost of the model, we use all the additive layers in the three-layer network model, thus avoiding the multiplication operation of the whole model. Meanwhile, 64×1 wide kernels are used in the first layer of the model to extract the effective information in the low-frequency band. The use of wide kernels can effectively reduce the depth of the model and reduce the complexity of the network. Then, two 3×1 small kernel adder layers were used to obtain a better feature representation. Finally, the extracted features were used for fault classification. BN was implemented after each adder layer to speed up the training process. We name the model WAddNN, where W stands for the wide kernel in the first layer and AddNN represents the adder neural network with almost no multiplication operations. The model parameters are shown in Table I.

B. 1-D ADDER NETWORK LAYER

In this paper, the proposed 1-D adder network uses mainly 1-D arrays as kernel and feature maps compared to 2-D adder networks [28]. The traditional convolutional layer uses inner product to calculate the similarity of the feature map to the convolutional kernel, which is given by:

TABLE I Details of WAddNN model

No.	Layer type	Kernel size/ stride	Kernel number	Output size (width × depth)
1	Adder layer1	$64 \times 1/16 \times 1$	16	60×16
2	BN1	/	/	60×16
3	Pooling1	$2 \times 1/2 \times 1$	16	30×16
4	Adder layer2	$3 \times 1/1 \times 1$	32	28×32
5	BN2	/	/	28×32
6	Pooling2	$2 \times 1/2 \times 1$	32	14×32
7	Adder layer3	$3 \times 1/1 \times 1$	64	12×64
8	BN3	/	/	12×64
9	Pooling3	$2 \times 1/2 \times 1$	64	6×64
10	Flatten	/	/	384×1
11	Softmax	/	/	10×1

$$Y^{l}(i,t) = \sum_{m=0}^{H} \sum_{k=0}^{v_{in}} \left(X^{l-1}(i+m,k) \times F^{l-1}(i,k,t) \right) \tag{9}$$

In contrast, in the one-dimensional adder layer, the similarity between the feature map and the convolution kernel is computed using the l_1 norm, thus changing all the multiplication operations for computing the inner product to additive operations for computing the l_1 norm. Therefore, compared with the convolutional layer, the adder layer can effectively reduce the consumption of computational resources. The equation of the adder layer is calculated as:

$$Y^{l}(i,t) = -\sum_{m=0}^{H} \sum_{k=0}^{v_{in}} \left| X^{l-1}(i+m,k) - F^{l-1}(i,k,t) \right| \tag{10}$$

Two fully connected layers in the classification stage are used for the final classification of the extracted features. Since the faults are divided into 10 categories, the softmax function is transformed using 10 neurons to obtain the probability distribution. The softmax function is expressed as:

$$q(z_j) = \frac{e^{z_j}}{\sum_{k}^{10} e^{z_k}} \tag{11}$$

where z_j denotes the logarithm of the j_{th} output neuron.

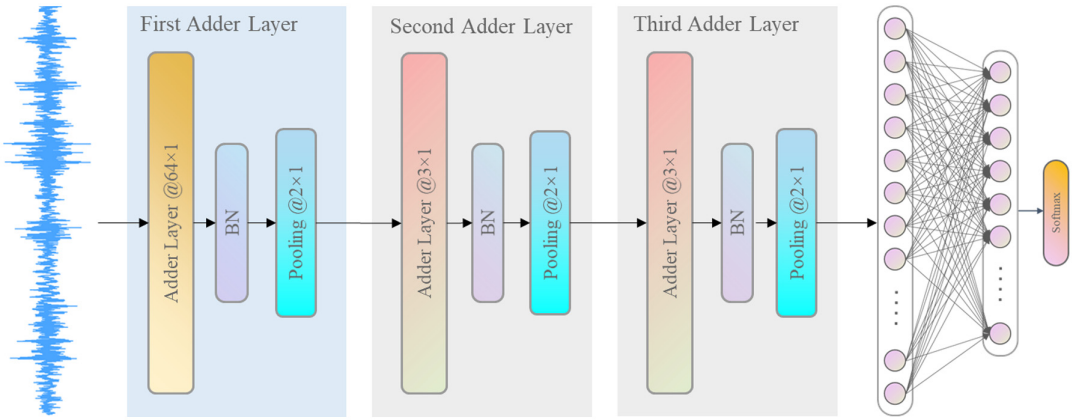

FIGURE 2. Architecture of the proposed WAddNN model.

C. TRAINING OF THE WAddNN

The structural design of WAddNN is mainly based on the one-dimensional characteristics of the vibration signal, which leads to the establishment of a one-dimensional additive network model. Meanwhile, the backpropagation algorithm is used in the training process of the model. In this subsection, the training process of the model is described.

As the loss function of the whole network, cross-entropy is an effective function used for the classification task. Let $p(x)$ denote the target distribution, $q(x)$ denote the estimated distribution, and the cross-entropy between $p(x)$ and $q(x)$ is

$$Loss = \mathrm{H}(P,Q) = -\sum_{x} P(x)\log Q(x) \tag{12}$$

The neural network uses BP to calculate the gradient of the filter and stochastic gradient descent to update the parameters. In adder neural networks, in order to improve the training speed, the following equation is used to perform the gradient operation on the filter F:

$$\begin{aligned}\Delta \mathrm{L}(F^{l-1}) &= \frac{\partial X^{l}(i,,t)}{\partial F^{l-1}(m,k,t)} \\ &= X^{l-1}(i+m,k) - F^{l-1}(m,k,t)\end{aligned} \tag{13}$$

At the meantime, the gradient of the input feature X^{l-1} is also important for the parameter update. To avoid gradient explosion, the input features are truncated when performing the derivation on them as follows:

$$\begin{aligned}\Delta L(X^{l-1}) &= \frac{\partial X^{l}(i,t)}{\partial X^{l-1}(i+m,k)} \\ &= \mathrm{H}(X^{l-1}(i+m,k) - F^{l-1}(m,k,t))\end{aligned} \tag{14}$$

The iteration of gradient descent updates the parameters as follows:

$$X_{l-1} = X_{l-1} - \theta \times \Delta L(X_{l-1}) \tag{15}$$

$$F_{l-1} = F_{l-1} - \theta \times \Delta \mathrm{L}(F_{l-1}) \tag{16}$$

where θ is the global learning rate of the whole network. To improve the training speed of the model, we likewise use an adaptive learning rate of (8) for the filter F. So the iterative formula for F becomes

$$F_{l-1} = F_{l-1} - \theta \times \alpha_{l-1} \times \Delta L(F_{l-1}) \tag{17}$$

The whole network can be better optimized after scaling the learning rate using the adaptive algorithm.

IV. EXPERIMENTAL VERIFICATION

With the development of intelligent bearing technology, edge computing and remote autonomous diagnosis become the development trend. In this paper, we propose WAddNN with adder layers as the main structure, aiming to reduce the computational load of the fault diagnosis model, thus reducing the energy consumption and the area used by the bearing end processor. In this section, we build a CNN model with the same structure as WAddNN for comparison and validate the effectiveness of the proposed approach in fault diagnosis on two bearing datasets.

A. CASE I: CWRU DATASET

1) DATA DESCRIPTION. The data for this experiment were obtained from the Case Western Reserve University (CWRU) Bearing Data Center [29]. The signals were collected from a motor-driven mechanical system with an accelerometer sampling frequency of 12 kHz as shown in Fig. 3. In total, there are four types of bearing states: normal state, ball failure, inner ring failure, and outer ring failure. In each category of bearing failure, there are divided into different degrees of failure of 0.007 inch, 0.014 inch, and 0.021 inch, respectively. So there are a total of 10 categories of bearing status. In this experiment, each sample contains 1024 data points and has 7000 training samples and 1000 test samples. The details of all data are shown in Table II.

2) COMPARISON MODEL. WAddNN uses additive operations instead of multiplicative operations, thus reducing the computational cost of the processor to accommodate the energy and area constraints of terminal autonomous diagnosis. The adder layer in WAddNN uses the l_1-norm distance metric to measure the similarity between the input and the filter to extract fault features of the input data. The CNN model, on the other hand, uses mutual correlation to extract fault features and has been used with great success in previous fault diagnosis studies. In order to verify the diagnostic capability of WAddNN for bearing faults, a CNN model with the same structure as WAddNN is built in this paper for comparison experiments. The specific parameters of the CNN model are shown in Table III.

3) COMPARISON EXPERIMENTS. For an objective comparison, we trained and tested both models under the same conditions. Both models use the dataset described above, 7000 samples for training the models and 1000 samples for testing the accuracy of the models. In this process, the same computer device was used to train both models. The training process was optimized using the stochastic gradient descent with a learning rate of 0.005, a momentum of 0.9, and a weight_decay of 0.0005. After training the same epoch, the final models were obtained and the model accuracy was tested on the test set, and Table IV shows the fault classification results for both models. The number of multiplication and addition operations required for WAddNN and CNN is related to the structure of the network itself. So for a fair comparison, both models are the same network structure. The number of multiplication and addition operations required for one of the CNNs is related to the input of each layer, the size of the convolutional kernel, and the step size.

The whole CNN network consists of three convolutional layers and a fully connected layer. Since the comparison of CNN model used in this paper is a one-dimensional convolutional network, the number of multiplications in one of the convolutional layers is calculated by the formula:

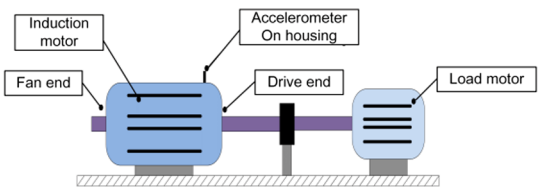

FIGURE 3. Motor driving mechanical system used by CWRU.

TABLE II Description of rolling element bearing datasets

Fault location	None	Ball			Inner race			Outer race			Load
Category labels	0	1	2	3	4	5	6	7	8	9	/
Fault diameter (inch)	0	0.007	0.014	0.021	0.007	0.014	0.021	0.007	0.014	0.021	0
Number of train data	700	700	700	700	700	700	700	700	700	700	/
Number of test data	100	100	100	100	100	100	100	100	100	100	/

TABLE III Details of CNN model used in experiments

No.	Layer type	Kernel size/stride	Kernel number	Output size (width × depth)
1	Convolution1	64 × 1/16 × 1	16	60 × 16
2	BN1	/	/	60 × 16
3	Pooling1	2 × 1/2 × 1	16	30 × 16
4	Convolution2	3 × 1/1 × 1	32	28 × 32
5	BN2	/	/	28 × 32
6	Pooling2	2 × 1/2 × 1	32	14 × 32
7	Convolution3	3 × 1/1 × 1	64	12 × 64
8	BN3	/	/	12 × 64
9	Pooling3	2 × 1/2 × 1	64	6 × 64
10	Flatten	/	/	384 × 1
11	Softmax	/	/	10 × 1

TABLE IV The fault classification results of CNN and WAddNN

Model	#Mul.	#Add.	Accuracy
CNN	194 k	194 k	99.90%
WAddNN	4 k	384 k	99.50%

$$C_{in} \times C_{out} \times \left(\frac{(H-K)}{S} + 1 \right) \times K \tag{18}$$

where C_{in} and C_{out} are the input and output channels, respectively, S is the step size, H is the input feature length, and K is the convolution kernel size. The number of multiplications in the fully connected layer is calculated by multiplying the number of input neurons I with the number of output neurons O: $I \times O$.

Therefore, the multiplication computation of the first convolutional layer of the CNN in this paper is 61440, the second convolutional layer is 46080, the third layer is 86016, and the multiplication computation of the final fully connected layer is 3840.The total multiplication computation of the CNN is 194 k, while the corresponding number of addition operations is equal to the multiplication operations. After the first three convolutional layers are replaced with adder layers, the whole WAddNN contains only the multiplication operations of the last fully connected layer, and the corresponding multiplication operations are converted to additive. Finally, the number of operations in all layers is summed to get the result in Table IV.

Table IV shows that the number of multiplications and additions required for CNN and WAddNN with the same structure are different. 194 k multiplications and 194 k additions are required for CNN to complete a diagnosis, while only 4 k multiplications are required for WAddNN, and the rest of the multiplications are converted to additions (194 k + 190 k). Taking the Intel Pentium CPU as an example, the 16-bit multiplication operation requires 10 times more instruction cycles than the addition operation. Therefore, converting multiplication operations in the network to addition operations can effectively reduce the processor's running time. Hence, WAddNN can reduce the consumption of processor resources, and the reduction of instruction cycles enables the model to run more quickly and complete the diagnosis, thus reducing the time consumption during the diagnosis and improving the real-time performance.

In summary, the computational cost of addition is much lower than that of multiplication. Most of the multiplication operations in WAddNN are replaced by addition operations, so the computational cost of WAddNN is lower than that of the CNN model. Meanwhile, the fault recognition rate of WAddNN reaches more than 99%, and the

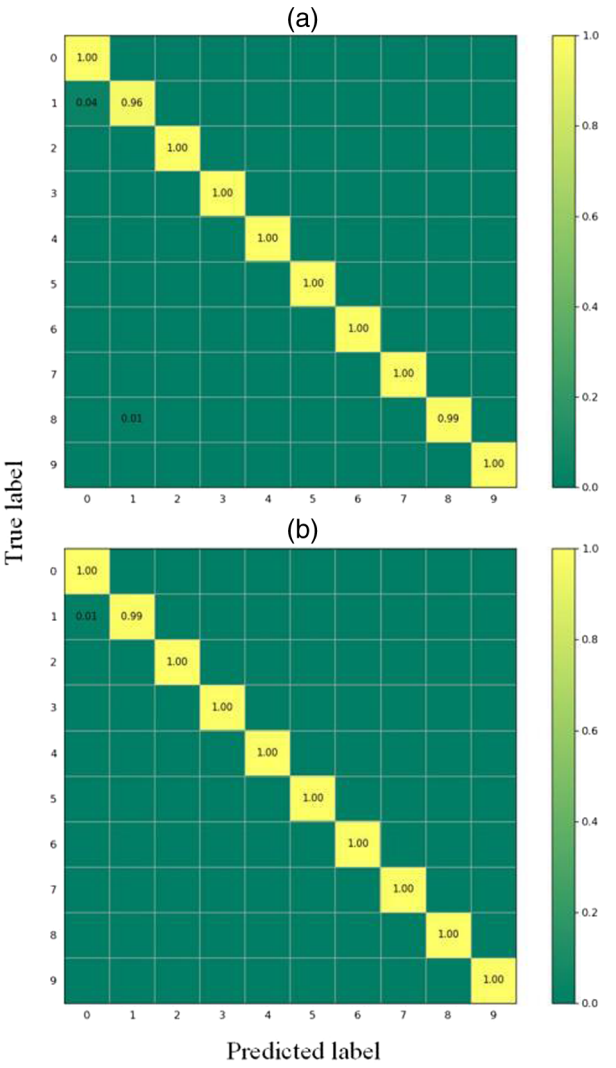

FIGURE 4. The confusion matrix: (a) WAddNN and (b) CNN.

accuracy rate is similar to the CNN model. In conclusion, WAddNN can achieve similar diagnostic accuracy as CNN with less computational cost, and it also verifies the feasibility of using additive layer instead of convolutional layer in bearing fault diagnosis.

4) VISUALIZATION OF RESULTS. The WAddNN model uses l_1-norm distance instead of mutual correlation, which in turn reduces the model computation by using additive operations instead of multiplicative operations. However, the sensitivity of WAddNN to fault features needs to be verified. Therefore, we performed the visual experimental analysis of the two models to further explore the ability of WAddNN to extract features in fault identification.

Figure 4 shows the classification accuracy of both models for each category of labels. The CNN model has a 99% accuracy on label 1, misclassifying 1% of label 1 as label 0. It has 100% accuracy on the other labels. The WAddNN model also misclassifies 4% of label 1 as label 0, while mislabeling 1% of label 8 as label 1, and 100% accuracy on the rest of the categories. It can be concluded that both models have high classification accuracy. Meanwhile, the categories of misclassification are all related to label 1, indicating that the two models have similarity in classification mechanism. Therefore, WAddNN has the same fault classification capability as the CNN model.

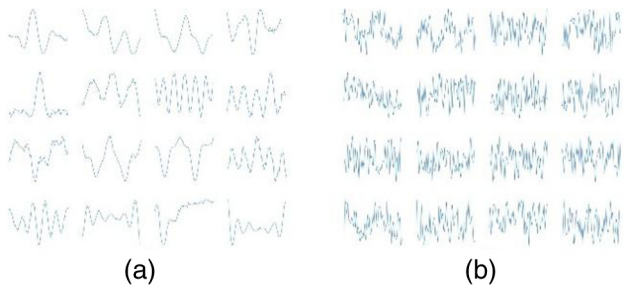

FIGURE 5. Visualization of filters in the first layer: (a) WAddNN and (b) CNN.

We visualized the first-layer filters of both network models in Fig. 5. Since the first layer used 16 64×1 wide kernels, the visualization resulted in 16 filters with different waveforms. Although WAddNN and CNN utilized different distance metrics, the filters of the proposed WAddNN network (see Fig. 5(a)) still shared similar patterns with the convolutional filters (see Fig. 5(b)). Both models extracted features at different frequencies of the signal and captured the fault information effectively. Figure 5 shows that the CNN filter is higher in frequency than WAddNN, while the WAddNN model focuses more on low-frequency features, which helps reduce the influence of local high-frequency features. The visualization experiments further demonstrated that the WAddNN filter could effectively extract useful information from the input signal.

The adder layer in WAddNN used l_1-norm distance instead of mutual correlation to measure the similarity between the input and the filter for extracting fault features of the input data. Therefore, it is necessary to further explore the feature space of WAddNN and CNN. t-distributed stochastic neighbor embedding (t-SNE) was used to investigate the feature distribution learned in each layer of the two models in Fig. 6. From the first layer of the WAddNN and CNN networks (see Fig. 6(a), (d), respectively), it can be seen that both models separate label class 3 and label class 7, and the other types are in a mixed state. After the second layer of the network, all 10 classes of labels in the WAddNN network are basically separable (see Fig. 6(b)), with labels 0, 4, and label 1 partially mixed. After the third layer of the network, the 10 class labels of both networks are already divisible (see Fig. 6(c), (f), respectively). From the feature visualization experiments of each layer in the network, it can be seen that the adder layer in the WAddNN model has the same fault classification capability as the convolutional layer in the CNN.

B. CASE II: XJTU-SY DATASET

1) DATA DESCRIPTION. The entire dataset XJTU-SY [30] was obtained from Xi'an Jiaotong University and

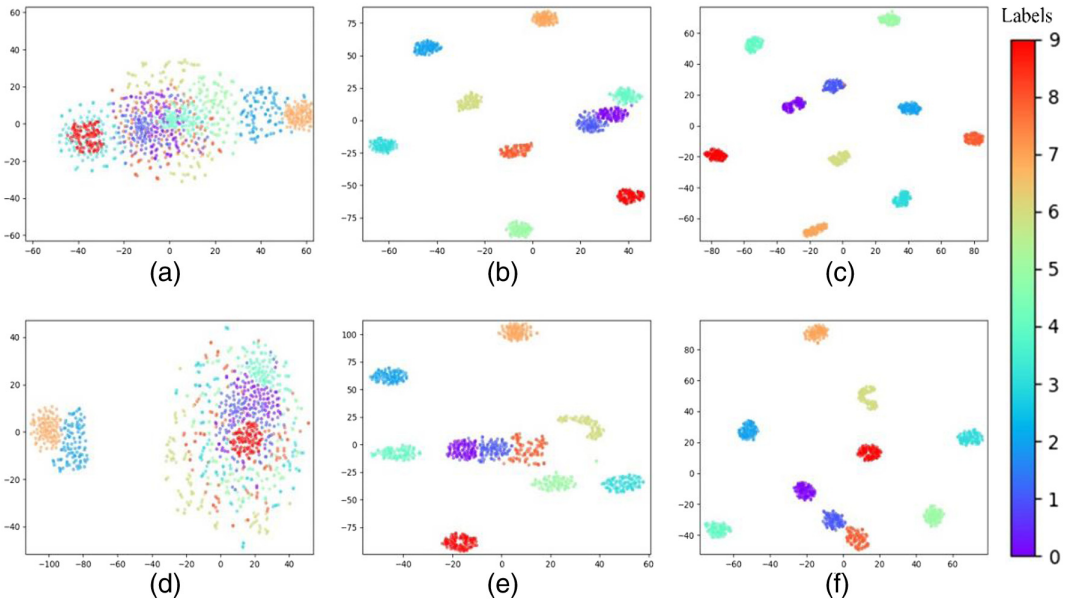

FIGURE 6. Feature visualization via t-SNE: (a) first layer of WAddNN, (b) second layer of WAddNN, (c) last layer of WAddNN, (d) first layer of CNN, (e) second layer of CNN, and (f) last layer of CNN.

TABLE V Classification results of CNN and WAddNN under different intensity noise

Model	Without noise (%)	0 dB (%)	2 dB (%)	4 dB (%)	6 dB (%)	8 dB (%)	10 dB (%)
CNN	98.84	85.72	91.90	93.08	95.91	97.88	98.30
WAddNN	97.40	82.58	90.08	91.11	94.87	96.22	96.89

Changxing Sumyoung Technology Co. The entire dataset contains 15 bearings operating under three different operating environments (1. Speed: 35 Hz, Load: 12 kN; 2. Speed: 37.5, Hz Load: 11 kN; 3. Speed: 40 Hz, Load: 10 kN) until failure. The sampling frequency during the acquisition was 2.56 kHz. The acquisition time was 1 min. Therefore, the diagnostic task for this dataset is a 15 classification task. The principle of dividing the training and test sets remains the same as in the previous experiments.

2) COMPARISON EXPERIMENT. In this subsection, we continue to use the same structure of CNN and WAddNN networks. To further validate the effectiveness of the proposed method, Gaussian white noise of different intensities was added to the dataset to test the robustness of the model to noise. Gaussian white noise is added to the original signal to synthesize signals with different signal-to-noise ratios (SNRs). The definition of SNR is shown below:

$$\mathrm{SNR}_{dB} = 10\log_{10}\left(\frac{P_{signal}}{P_{noise}}\right) \tag{19}$$

where P_{signal} and P_{noise} are the power of the signal and noise, respectively. If the SNR of the composite noise signal is 0 dB, the power of the noise is equal to the power of the original signal.

The accuracy of the two models under different noises is shown in Table V. The accuracy of the CNN model decreases as the noise intensity increases. The accuracy of WAddNN is lower than that of CNN by about 2% in different cases, and it is also robust to noise. On the XJTU-SY dataset, WAddNN significantly reduces the computational consumption compared to CNN with a certain loss of accuracy. The lost accuracy is acceptable in edge fault diagnosis. The experiments on both datasets validate the effectiveness of WAddNN, which can significantly reduce the reliance of the model on computational resources.

V. CONCLUSION

In this paper, a new WAddNN bearing fault diagnosis model is proposed to reduce the computational cost of the deep learning model, which accommodates the current development of intelligent bearing technology.

WAddNN extracts features using the l_1-norm distance, thus making the adder layer contain only additive operations. The use of the adder layer substantially reduces the computational cost in the feature extraction layer, demonstrating the great potential of the proposed method for autonomous diagnosis of bearings. Through comparison experiments with the same structured CNN model, WAddNN can obtain comparable diagnostic accuracy, which proves the powerful feature extraction capability of the adder layer while reducing the computational effort. The results show that the adder layer can replace the convolutional layer in bearing faults. This shows that the adder layer can be a basic building block for edge-end diagnostic models with limited computational resources.

However, the current WAddNN is still a bit lower in accuracy than the traditional CNN model. In the later work, we will study to further improve the accuracy of WAddNN and apply the additive network to the actual hardware fault diagnosis.

Acknowledgments

The authors are grateful for the financial support provided by the China National Key Research and Development Program of China under Grant 2019YFB2004300 and the National Natural Science Foundation of China under Grant 51975065 and 51805051.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

References

[1] J. Tang et al., "Rolling bearing remaining useful life prediction via weight tracking relevance vector machine," *Meas. Sci. Technol.*, vol. 32, no. 2, p. 024006 (12pp), 2021.

[2] Z. H. Liu et al., "A stacked auto-encoder based partial adversarial domain adaptation model for intelligent fault diagnosis of rotating machines," *IEEE Trans. Ind. Inf.*, vol. PP.99, p. 1, 2020.

[3] H. Pan, "Research on intelligent fault diagnosis of rolling bearing based on improved deep residual network," *Appl. Sci.*, vol. 11, no. 22, p. 10889, 2021.

[4] Y. Gong et al., "Self-powered wireless sensor node for smart railway Axle box bearing via a variable reluctance energy harvesting system," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.

[5] R. X. Gao, B. T. Holm-Hansen, and C. Wang, "Design of a mechatronic bearing through sensor integration," *Sens. Controls Intell. Mach. Agile Manuf. Mechatron.*, vol. 3518, International Society for Optics and Photonics, pp. 244–250, 1998.

[6] Y. Ke et al., "Intelligent fault diagnosis method of common rail injector based on composite hierarchical dispersion entropy and improved least squares support vector machine," *Digit. Signal Process.*, vol. 114, p. 103054, 2021.

[7] M. Unal et al., "Fault diagnosis of rolling bearings using a genetic algorithm optimized neural network," *Measurement*, vol. 58, pp. 187–196, 2014.

[8] J. Lu et al., "Enhanced K-nearest neighbor for intelligent fault diagnosis of rotating machinery," *Appl. Sci.*, vol. 11, no. 3, p. 919, 2021.

[9] H. Yuan et al., "An improved initialization method of D-KSVD algorithm for bearing fault diagnosis," *J. Mech. Sci. Technol.*, vol. 31, no. 11, pp. 5161–5172, 2017.

[10] F. Jia et al., "Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating

machinery with massive data," *Mech. Syst. Sig. Process.*, vol. 72, pp. 303–315, 2016.

[11] X. Xu, Y. Lei, and Z. Li, "An incorrect data detection method for big data cleaning of machinery condition monitoring," *IEEE Trans. Ind. Electron.*, vol. 67, no. 3, pp. 2326–2336, 2019.

[12] X. Guo, C. Shen, and L. Chen, "Deep fault recognizer: An integrated model to denoise and extract features for fault diagnosis in rotating machinery," *Appl. Sci.*, vol. 7, no. 1, p. 41, 2017.

[13] F. Xue, et al., "A novel intelligent fault diagnosis method of rolling bearing based on two-stream feature fusion convolutional neural network," *Measurement*, vol. 176, p. 109226, 2021.

[14] Y. Chen et al., "ACDIN: Bridging the gap between artificial and real bearing damages for bearing fault diagnosis," *Neurocomputing*, vol. 294, pp. 61–71, 2018.

[15] J. Zhao et al., "A new bearing fault diagnosis method based on signal-to-image mapping and convolutional neural network," *Measurement*, vol. 176, no. 1, p. 109088, 2021.

[16] F. Jia et al., "Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization," *Mech. Syst. Sig. Process.*, vol. 110, pp. 349–367, 2018.

[17] D. K. Appana, A. Prosvirin, and J.-M. Kim, "Reliable fault diagnosis of bearings with varying rotational speeds using envelope spectrum and convolution neural networks," *Soft Comput.*, vol. 22, no. 20, pp. 6719–6729, 2018.

[18] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[19] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[20] W. Fuan et al., "An adaptive deep convolutional neural network for rolling bearing fault diagnosis," *Meas. Sci. Technol.*, vol. 28, no. 9, p. 095005, 2017.

[21] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," in *2014 IEEE Int. Solid-State Circuits Conf. Digest Techn. Papers (ISSCC)*, San Francisco, CA, USA, IEEE, pp. 10–14, 2014.

[22] M. Courbariaux, Y. Bengio, and J.-P. David, "Binaryconnect: Training deep neural networks with binary weights during propagations," *arXiv preprint arXiv:1511.00363*, 2015.

[23] I. Hubara et al., "Binarized neural networks," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, Advances in neural information processing systems, pp. 4107–4115, 2016.

[24] M. Rastegari et al., "Xnor-net: Imagenet classification using binary convolutional neural networks," in *European Conference on Computer Vision*. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds) Computer Vision, Cham: Springer, pp. 525–542, 2016.

[25] Z. Cai et al., "Deep learning with low precision by half-wave Gaussian quantization," in *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, IEEE, pp. 5406–5414, 2017.

[26] H. Chen et al., "AdderNet: Do we really need multiplications in deep learning?," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recogn.*, IEEE, pp. 1465–1474, 2020.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Adv. N.A. Inf. Process. Syst.*, vol. 25, pp. 1097–1105, 2012.

[28] T. Ince et al., "Real-time motor fault detection by 1-D convolutional neural networks," *IEEE Trans. Ind. Electron.*, vol. 63, no. 11, pp. 7067–7075, 2016.

[29] W. Zhang et al., "A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals," *Sensors*, vol. 17, no. 2, p. 425, 2017.

[30] Y. Wang, N. Lei, N. Li, and N. Li, "A hybrid prognostics approach for estimating remaining useful life of rolling element bearings," *IEEE Trans. Reliab.*, vol. 69, no. 1, pp. 1–12, 2018.