Deep residual network for enhanced fault diagnosis of rotating machinery

Shoucong Xiong¹2, Tielin Shi¹
¹School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan, China
²E-mail: xiongsc@hust.edu.cn

Abstract. Deep residual network (DRN) is a recently-developed powerful algorithm in the deep learning filed. This paper introduces the superiority of DRN into the fault diagnosis of rotating machinery for simplifying traditional diagnosing process as well as enhancing predicting performance. DRN can not only extract features automatically from raw or processed signals but also benefit from its especially deep architecture to continually improve representation capacity without worrying gradient divergence issues. The functions of DRN result from the unique structure design called residual building block, which will be described clearly with the network overall architecture in this paper. Additionally, the comparison of the DRN with other machine learning-based and neural network-based fault diagnosis methods are presented.

1. Introduction
Due to the structure complexity of modern rotating machinery, recent fault diagnosis tasks mainly depend on analyzing sampling signals that are collected from the monitoring sensors. Vibration signals are the most frequently used signals in rotating machines since it can straightly reflect the fault pattern in defective components [1]. After acquiring the signal data, traditional fault diagnosis methods usually consist of two processing steps: one is the feature extraction and the other is the fault classification. However, manual feature extraction requires skilled expertise and heavy labor, meanwhile, is limited to certain diagnosis situations. Besides that, the quality of classification model decides the upper bound of fault diagnosis performance, and common machine learning-based diagnosis methods such as support vector machine are hard to perfectly handle the issues if there are too many data volume and fault classes to distinguish. Deep learning algorithms do well at large-scale datasets and some of their variants like convolutional neural network (CNN) are capable of extracting features automatically, which covers the shortages of traditional fault diagnosis methods. Currently, deep learning-based fault diagnosis methods become more and more popular [2].

Deep residual network (DRN) is an excellent variant of CNN, which has been widely implemented in the field of image recognition [3]. It is a common sense that deeper network structure theoretically gives arise to stronger representativeness. However, researchers found that deeper network is hard to train successfully, that is, network accuracy tends to be saturated with network depth keeps increasing. This limits the development of CNN in fault diagnosis field since deeper networks are also necessary for diagnosis process to extract well enough sets of features from very complicated signals. DRN solves the saturation problem with special structure design. By adding parameter-free identity shortcut connections across several stacked convolutional layers, network instead fits a residual mapping.
between input and output, making training much easier compared to the original unreferenced mapping schedule. To sum up, DRN is the ideal solution for fault diagnosis since it can not only extract features automatically but also continually improve the model performance by deepening network structure.

This paper is going to discuss the basic details of DRN, including its special structure design: residual building block, overall network architecture and an application example. Comparison between DRN and other methods is also illustrated in the following part. At last, a brief conclusion is given.

2. Deep residual network
DRN is mainly originated from plain CNN, so this part introduces the basic concept of CNN firstly, then illustrates the details of residual building block. Overall architecture of a DRN is given at the rest of this section.

2.1. Convolutional neural network

2.1.1. Artificial neural network. Artificial neural network (ANN) is a mathematical algorithm model that mimics the behaviour characteristics of biological neural network in animal brains, depending on adjusting the interconnections between a large number of internal nodes to achieve the information processing purpose. Unlike biological neural network that each neuron is interconnected with other adjacent neurons, ANN allocates the neurons into different neural layers and only the neurons in different and adjacent layer have connections. Common neural layers include input layer, hidden layer and output layer. To exhibit the structure characters of ANN, neural layers in ANN are also called fully-connected layers. Neuron is the basic component of neural layer, and contains three kinds of elements: weight, bias and activation function. Weight is the multiplier of the input information that the former neuron sends to, and bias is the extra term added to the summation of the whole input information that the current neuron receives. Activation function is an operation that the neuron handles the input information. In input layer and output layer, activation function is consistently the identity function, and in hidden layer, sigmoid function, hyperbolic tangent function and rectified linear function are usually chosen to be activation function. In this paper, rectified linear function is our activation function in hidden layer, abbreviated as ReLU.

2.1.2. Convolutional layer. CNN is a variant of ANN, aiming to process the input information that concerns the dimensionality. The merits of CNN are three points: local connectivity, shared weights, down-sampling [4]. Convolutional layer is the core of CNN, which uses convolutional kernels to partially connect sub region of input neurons to each output neuron once at a time. Figure 1 illustrates an exemplar convolution operation in convolutional layer.

![Figure 1. An exemplar convolution operation in convolutional layer.](image-url)
The outputs of a convolution layer are called feature maps, which are the input samples of the next layer. Convolution operation may shrink the output’s size comparing to the input, limiting the depth and capacity of model. To handle this, zero-padding is used by adding some circles of zero around the outermost edge of input to offset the shrunken size.

2.1.3. Pooling layer. Pooling layer is another kind of special layer uniquely existed in CNN. The function of pooling layer is to reduce redundant information as well as enlarge the receptive field of model. A pooling layer should be alternatively inserted in the network after several convolutional layer having been stacked to ensure that the model extracts feature gradually more abstractly. Usual types of pooling layer include max pooling layer and average pooling layer. In this paper, a special type of average pooling layer called global average pooling layer [5] is also adopted, in which the size of pooling kernel is equal to the input size. Figure 2 shows how a pooling layer works in the neural network.

![Figure 2. Max pooling and average pooling.](image)

2.2. Residual building block

Before discussing the residual building block, there is a deep learning trick called batch normalization (BN) [6] needed explaining in advance. BN resolves the problem that the distribution of each layer’s inputs keeps shifting in training due to the update of parameters of previous layers, by normalizing the input statistics into a stable range thus accelerating the network convergence. For a layer with d-dimensional input $x = (x^1, ..., x^d)$ over a mini-batch $B = \{x_1, ..., x_m\}$, BN normalizes the inputs as follow:

$$
\begin{align*}
\mu_B &= \frac{1}{m} \sum_{i=1}^{m} x_i, \\
\sigma^2_B &= \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu_B)^2, \\
\hat{x}_i &= \frac{x_i - \mu_B}{\sqrt{\sigma^2_B + \epsilon}}, \\
y_i &= \gamma \hat{x}_i + \beta.
\end{align*}
$$

where $x_i$ is the $i$th input vector from mini-batch $B$, $\mu_B$ and $\sigma^2_B$ are the expectation and variance of inputs over $B$, $\epsilon$ is a constant added for numerical stability, $\gamma$ and $\beta$ scale and shift the normalized inputs elementwise to restore the representation capability of the neural layer, and $y_i$ is the output of BN layer.
When network becomes deep, it is found that model performance will get saturated with stacking plain convolutional layers, which is called accuracy degradation problem. Residual learning framework helps to ease the training of network and break up the accuracy upper limit. When each few stacked layers of a plain network try to fit an unreferenced mapping $H(x)$, a residual network instead asks layers to fit a residual mapping of $F(x) = H(x) - x$, and $H(x)$ becomes $F(x) + x$. Both forms are able to obtain the desired mapping, but the ease of learning is different. It should be much easier to find those perturbations with respect to the input than to learn a brand-new function.

Residual learning framework is built on residual building blocks. The detailed structure of a residual building block is illustrated in Figure 3. The element-wise additive operation $F(x) + x$ is performed on two feature maps, channel by channel for convolutional layers. When channel numbers of two layers are not matched, zero-padding feature maps are appended for increasing dimensions, and the shortcut $x$ is parameter-free.

![Figure 3. Detailed structure of a residual building block.](image)

### 2.3. Architecture of DRN

The network structure in this paper is piled up by the residual building blocks described in section 2.2, and a global average pooling layer, a full connected layer and a softmax layer are added after the stacked convolutional layers, in which the convolution operation with a step of 2 is used instead of pooling operation. Details are shown in Figure 4.

![Figure 4. Overall architecture of the DRN, where $m$ denotes the kernel number.](image)
All convolution kernels in the DRN have the same size of 3 by 3, the reason is that two stacked layers of 3 by 3 convolution can obtain an effective 5 by 5 receptive field, and three stacked layers of 3 by 3 convolution can get a 7 by 7 receptive field [7], so there is no need to deliberately design the sizes of the convolution kernels. The global average pooling layer averagely synthesizes the spatial information of feature map, making the network more robust to spatial translation of the input, and able to accept input images of any size.

3. Fault diagnosis

3.1. Data description

A public rolling bearing dataset from Case Western Reverse University (CWRU) bearing data centre [8] is conducted to verify the DRN. The experiment was built on a test stand that consisted of a 2 hp motor, a torque transducer/encoder, a dynamometer and a control electronics, as shown in Figure 5. Vibration data was collected by the accelerometer attached to the housing of driven end and sampled 12000 points per second. There are three types of bearing faults and each fault type contains three damage sizes. In all, there are total ten types of bearing conditions in each load condition.

![Figure 5. The CWRU test stand.](image)

3.2. Diagnosis process

a) To find out a DRN model that satisfies the diagnosis requirement, the first step is to set kernel number \( m \), this case it is set to 16.

b) Confirm the data length, e.g. 1024, of signal segments that are fed into the model and sample a training dataset and a validation dataset from collected sensor data. Label related fault category to each signal segment.

c) Transform each signal segment with wavelet packet transform into time-frequency coefficients that can expose richer fault information and then stack up sets of coefficients into a two-dimensional coefficient matrix [4]. Now each coefficient matrix becomes the input of model.

d) Initialize the parameters in the DRN model with Gaussian distribution and train the model with training dataset. Optimizer like Adam, initial learning rate value like 0.0001 need to be defined in advance. Evaluate the model performance with validation dataset at the end of every training epoch and stop the training when validation performance is maximal. After training, the DRN model is ready for diagnosis.

e) For an unknown vibration signal segment, transform it into time-frequency coefficient matrix and feed it into the DRN, then the diagnosis result can be given by the model.

4. Method comparison

DRN is compared the performance with a convolutional neural network (CNN), an artificial neural network (ANN) and a support vector machine (SVM) under the fault diagnosis scenario described in section 3.1. A testing dataset is extra sampled for testing the model generalization performance. Each model is repeated training five times and average testing accuracy is shown in Table 1.
Table 1. Comparison results of different methods.

|     | DRN   | CNN   | ANN   | SVM   |
|-----|-------|-------|-------|-------|
| Mean| 99.16 | 97.13 | 93.78 | 94.23 |
| Max | 99.38 | 98.28 | 94.66 | 94.56 |
| Min | 99.09 | 96.20 | 92.08 | 93.98 |
| Std | 0.0049| 0.0152| 0.0176| 0.0055|

From the results in Table 1, it is found that the proposed DRN method can achieve better performance than other diagnosis methods by a wide margin, exhibiting the superiority of DRN.

5. Conclusion

This paper presents a thorough fault diagnosis process achieved by DRN algorithms. Manual feature extraction is not necessary anymore and DRN automatically finds out the relationship between category labels and vibration signals. Time-frequency transform is added only for enhancing the diagnosis quality of complex tasks. Comparison between DRN and other algorithms shows the potential of DRN in fault diagnosis of rotating machinery field.

Although, it is difficult for people to perfectly visualize the reasons of superiority of DRN as well as its automatic feature extraction process. But predicting accuracy still proves DRN’s powerfullness. Find more direct evidences to explain why DRN works in fault diagnosis field may be an interesting topic in future research, and it is also helpful to guide the design of stronger network structures.

References

[1] Xiong S, He S, Xuan J, Xia Q and Shi T 2020 Enhanced deep residual network with multilevel correlation information for fault diagnosis of rotating machinery JVC/Journal Vib. Control
[2] Zhao R, Yan R, Chen Z, Mao K, Wang P and Gao R X 2019 Deep learning and its applications to machine health monitoring Mech. Syst. Signal Process. 115 213–37
[3] He K, Zhang X, Ren S and Sun J 2016 Deep residual learning for image recognition Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 12 770–8
[4] Xiong S, Zhou H, He S, Zhang L, Xia Q, Xuan J and Shi T 2020 A novel end-to-end fault diagnosis approach for rolling bearings by integrating wavelet packet transform into convolutional neural network structures Sensors (Switzerland) 20 1–26
[5] Lin M, Chen Q and Yan S 2014 Network in network 2nd Int. Conf. Learn. Represent. ICLR 2014 - Conf. Track Proc. 1–10
[6] Ioffe S and Szegedy C 2015 Batch normalization: Accelerating deep network training by reducing internal covariate shift 32nd Int. Conf. Mach. Learn. ICML 2015 1 448–56
[7] Simonyan K and Zisserman A 2015 Very deep convolutional networks for large-scale image recognition 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc. 1–14
[8] Smith W A and Randall R B 2015 Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study Mech. Syst. Signal Process. 64–65 100–31