Subject Section

Bearing Fault Diagnosis Based on Deep Belief Network

Wujie Sun [1],*

[1]School of Software Engineering, South China University of Technology, Guangzhou 510006, P.R. China.

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Abstract

It is important to judge whether the bearing is faulty and the possible fault location based on the acceleration of the bearing. However, the acceleration of the bearing is affected by noise, and it is difficult to rely on observation to find the law. This problem can be solved by using wavelet packet decomposition and deep belief network. In this paper, we use wavelet packet decomposition for noise reduction, and extract features from the noise-reduced data, and then use the deep belief network (DBN) for training. The experimental results show its effectiveness.

Keywords: Deep belief network, fault diagnosis, wavelet packet decomposition, feature extraction

Contact: wjsunscut@163.edu.cn

1 Introduction

Mechanical bearing failures are common in daily life, and the economic and time loss they cause is very serious. Therefore, it is very important to find it early in the event of a failure. Accelerometer data collected from actual operating environment is an important indicator to determine whether the bearing is faulty. However, the acceleration of the bearing is affected by noise, and it is difficult to rely on observation to find the problem. Therefore, we need a method that can efficiently determine whether the bearing is faultly and its fault type by using the accelerometer data.

Since the accelerometer data is affected by noise, we need to perform noise reduction. Wavelet packet decomposition [1][2] is widely used for noise reduction and has achieved remarkable results.

In recent years, deep learning, as an emerging method in the field of machine learning, has achieved brilliant results in the fields of image and speech recognition with its powerful capabilities. As one of the classical algorithms of deep learning, deep belief network [3] successfully solves problems such as information retrieval, dimension reduction, fault classification and so on with its excellent feature extraction and training algorithms. The deep learning method has the following advantages compared with the traditional fault diagnosis methods: 1) Deep learning has powerful feature extraction ability, can automatically extract features from a large amount of data, and reduces the needs of the expert experience and signal processing technology. It reduces the uncertainty of feature extraction and fault diagnosis caused by human

in traditional methods; 2) By establishing a deep model, it can well represent the complex mapping relationship between signal and health status, which is very suitable for big data background. The need for diagnostic analysis of diverse, nonlinear, and high-dimensional health monitoring data. Therefore, applying deep learning to the field of fault diagnosis has certain timeliness, practicability and versatility.

In addition, deep belief network have better scalability and mapping capabilities than other machine learning algorithms such as support vector machine [4] and backpropagation neural networks [5].

For the input data of the deep confidence network, some choose to use the time domain signal as the input [6], and some choose to use the frequency domain signal as the input [7]. In this paper, we choose to extract features from the time domain and the frequency domain as the input to the deep belief network [8].

2 Wavelet Packet Decomposition and Feature Extraction

Wavelet packet decomposition uses an analysis tree to represent wavelet packets, that is, using multiple iterations of wavelet transform to analyze the details of the input signal. From the perspective of function theory, wavelet packet decomposition is to map the signal into the space formed by the wavelet packet basis function. From a signal processing point of view, it is to pass the signal through a series of filters with different center frequencies but the same bandwidth.

Since the wavelet decomposition [9] is poor in frequency resolution in the high frequency band and time resolution in the low frequency band, wavelet packet is proposed to overcome this problem. It is a more sophisticated method of signal analysis that improves the time domain resolution of the signal. Fig. 1 shows the difference between them.
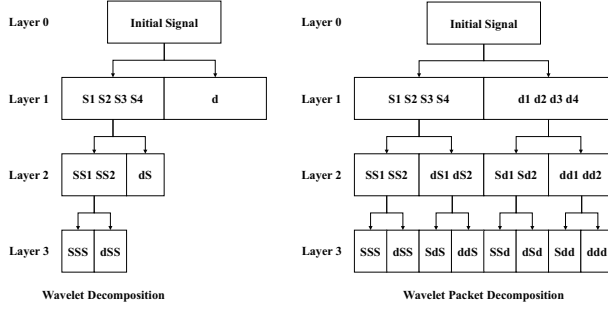

Fig. 1. Difference between wavelet decomposition and wavelet packet decomposition.

Since wavelet packet decomposition is not the main content of this article, it is not described too much. Wavelet packet decomposition can be easily implemented using functions such as wavedec, appcoef, detcoef, wbmpen, and wdencmp in MATLAB.

Through the wavelet packet decomposition, we obtained the noise reduction data. Next, we should extract features from the data. We need to set up a sliding window to read the data by its length. For example, if there are 50,000 data and the sliding window length is 1000 and the number of statistical features is 100, then we can get 50 data with each containing 100 parameters. Table 1 shows how to calculate the statistical features. N is the length of sliding window and x_i is the ith acceleration data in the sliding window. More discriminating parameters can help to improve the accuracy of the classification. Limited to the length, we only provide 18 parameters related to the time domain in the paper.

Table 1. Statistical Features

Name	Formula
Peak-to-Peak	$\text{MAX}|x_i| - \text{MIN}|x_i|$
Peak	$\text{MAX}|x_i|$
Mean	$\frac{1}{N}\sum_i^N x_i$
Mean Square	$\frac{1}{N}\sum_i^N x_i^2$
Root Mean Square	$\sqrt{\frac{1}{N}\sum_i^N x_i^2}$
Mean Amplitude	$\frac{1}{N}\sum_i^N |x_i|$
Square Mean Root	$\left(\frac{1}{N}\sum_i^N |x_i|^{1/2}\right)^2$
Variance	$\frac{1}{N}\sum_i^N (x_i - \overline{x})^2$
Standard Deviation	$\sqrt{\frac{1}{N}\sum_i^N (x_i - \overline{x})^2}$
Skewness	$\frac{1}{N}\sum_i^N x_i^3$
Kurtosis	$\frac{1}{N}\sum_i^N |x_i|^4$
Skewness Factor	$\frac{1}{N}\sum_i^N |x_i|^3 / \left(\sqrt{\frac{1}{N}\sum_i^N x_i^2}\right)^3$
Kurtosis Factor	$\frac{1}{N}\sum_i^N |x_i|^4 / \left(\sqrt{\frac{1}{N}\sum_i^N x_i^2}\right)^4$
Peak Factor	$\text{MAX}|x_i| / \sqrt{\frac{1}{N}\sum_i^N x_i^2}$
Impulsive Factor	$\text{MAX}|x_i| / \frac{1}{N}\sum_i^N |x_i|$
Clearance Factor	$\text{MAX}|x_i| / \left(\frac{1}{N}\sum_i^N |x_i|^{1/2}\right)^2$
Waveform Factor	$\sqrt{\frac{1}{N}\sum_i^N x_i^2 / \frac{1}{N}\sum_i^N |x_i|}$
Waveform Entropy [10]	$\frac{1}{N}\sum_i^N x_i \log x_i$

3 Deep Belief Network

A common DBN model [11] proposed by Hinton consists of a classifier and multiple restricted Boltzmann machines (RBMs) as shown in Fig. 2. If we want to figure out the principles of the DBN, first we need to master the knowledge related to RBM.
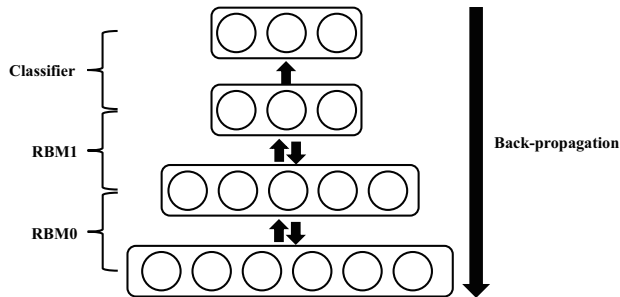

Fig. 2. A simple DBN model with 2 RBMs.

3.1 Restricted Boltzmann Machine

RBM [12] has a visible layer, a hidden layer, but there is no connection within the layer, and the layer is fully connected between the layers, and the node variable takes a value of 0 or 1. It removes the intra-layer connection of the Boltzmann machine (BM), which greatly reduces the amount of calculation. The structure of the RBM is shown in Fig. 3.
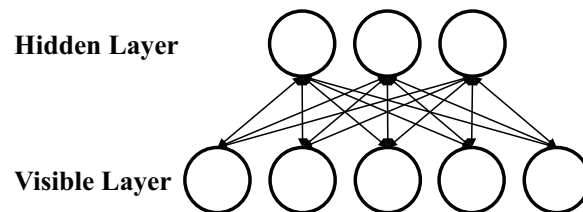

Fig. 3. A simple RBM model.

In RBM, the activation conditions of each hidden layer units are independent when the visible layer units' state is given. Similarly, the activation conditions of the visible layer units are independent when the hidden layer units' state is given.

The essence of RBM is a useful tool for unsupervised learning, which can be used for dimensionality reduction, feature extraction, autoencoder, deep belief network, and so on.

Next we will formulate the formula according to [13]. Assume the RBM has n visible layer units and m hidden layer units. We use the vectors $\mathbf{v}$ and $\mathbf{h}$ to represent the states of its visible layer and hidden layer where v_i is the ith visible layer unit and h_j is the jth hidden layer unit. Therefore, for a given state $(\mathbf{v}, \mathbf{h})$, the energy of the RBM can be calculated as

$$E(\mathbf{v}, \mathbf{h}) = -\sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij}v_ih_j - \sum_{i=1}^{m} a_iv_i - \sum_{j=1}^{n} b_jh_j \quad (1)$$

where w_{ij} is the weight, a_i is the visible layer unit bias, b_j is the hidden layer unit bias, and $\boldsymbol{\theta} = \{w_{ij}, a_i, b_j\}$.

After that, the joint probability distribution can be calculated as

$$P(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta}) = \frac{\mathbf{e}^{-E(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta})}}{Z(\boldsymbol{\theta})}, \quad Z(\boldsymbol{\theta}) = \sum_{\mathbf{v}, \mathbf{h}} \mathrm{e}^{-E(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta})} \quad (2)$$

and the likehood function can be calculated as

$$P(\mathbf{v}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \sum_{\mathbf{h}} \mathrm{e}^{-E(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta})} \quad (3)$$

By formula derivation, we can get the following activation probability formula

$$P(h_j = 1|\mathbf{v}, \boldsymbol{\theta}) = \sigma\left(b_j + \sum_i v_iw_{ij}\right) \quad (4)$$

$$P(v_i = 1|\mathbf{h}, \boldsymbol{\theta}) = \sigma\left(a_i + \sum_j w_{ij}h_j\right) \quad (5)$$

where $\sigma(x) = 1/(1 + e^{-x})$.

3.2 Contrastive Divergence Algorithm

The purpose of learning RBM is to find the value of $\boldsymbol{\theta}$ to fit the given training data. However, obtaining results based on methods such as Gibbs sampling is cumbersome and time consuming [13]. Hinton proposed a algorithm called Contrastive Divergence (CD) [14] which is easy to calculate. Algorithm is shown as Alg. 1.

3.3 Run DBN

Now we construct DBN as shown in Fig. 2. First, we input the data into RBM0 and run CD algorithm. Then we should consider the output of RBM0 as the input of RBM1 and run CD algorithm. After that, we can see the network as back propagation neural network and train it. The $\boldsymbol{\theta}$ trained by RBMs should be the initial value of the back propagation neural network.

This is the most basic DBN model and we can optimize it for better classification. In [8], the authors improved the

---

Algorithm 1 Contrastive Divergence

1: Training sample x_0, learning rate ϵ, max epoch T;
2: $\mathbf{v_1} = x_0$, $\mathbf{w}$, $\mathbf{a}$, and $\mathbf{b}$ are random small values;
3: **for** $epoch = 1, 2, ..., T$ **do**
4:     **for** $j = 1, 2, ..., m$ **do**
5:         Calc $P(h_{1j} = 1|\mathbf{v_1}) = \sigma(b_j + \sum_i v_{1i}w_{ij})$;
6:         Randomly select $h_{1j} \in \{0, 1\}$;
7:     **end for**
8:     **for** $i = 1, 2, ..., n$ **do**
9:         Calc $P(v_{2i} = 1|\mathbf{h_1}) = \sigma\left(a_i + \sum_j h_{1j}w_{ij}\right)$;
10:        Randomly select $v_{2i} \in \{0, 1\}$;
11:    **end for**
12:    **for** $j = 1, 2, ..., m$ **do**
13:       Calc $P(h_{2j} = 1|\mathbf{v_2}) = \sigma(b_j + \sum_i v_{2i}w_{ij})$;
14:    **end for**
15:    $\triangle\mathbf{w} = P(h_{1.} = 1|\mathbf{v_1})\mathbf{v_1^T} - P(h_{2.} = 1|\mathbf{v_2})\mathbf{v_2^T}$;
16:    $\triangle\mathbf{a} = \mathbf{v_1} - \mathbf{v_2}$;
17:    $\triangle\mathbf{b} = P(h_{1.} = 1|\mathbf{v_1}) - P(h_{2.} = 1|\mathbf{v_2})$;
18:    $\mathbf{w} \leftarrow \mathbf{w} + \epsilon\triangle\mathbf{w}$;
19:    $\mathbf{a} \leftarrow \mathbf{a} + \epsilon\triangle\mathbf{a}$;
20:    $\mathbf{b} \leftarrow \mathbf{b} + \epsilon\triangle\mathbf{b}$;
21: **end for**

---

activation function by combining Sigmoid with LReL to achieving the goal. Authors of [15] proposed the semi-supervised DBN to improve the structure of the DBN to achieve better performance.

In this project training, I reproduced these two papers. Since there is no suitable optimization method at present, in the DBN and experiment part, only the optimization methods of the published papers are involved.

4 Experiments

In the experiments in this paper, we used the drive end bearing accelerometer data provided by the Electrical Engineering Laboratory of Case Western Reserve University [16]. Since the content only covers the recurring part, the main goal is to compare whether the various methods speed up the convergence or improve the classification accuracy rate when the parameters are basically consistent, instead of finding the highest classification accuracy rate.

In the classification of faults, we refer to [6] to classify faults into 10 categories. In this experiment, 500 data were used, that is, 50 data per category, and each data was extracted from 2048 points.

First we use MATLAB to denoise the data, some of which are shown in the Fig. 4.

Then we use the sliding window to segment the noise reduction data and calculate its features according to Table 1. The feature distribution can be seen in Fig. 5.

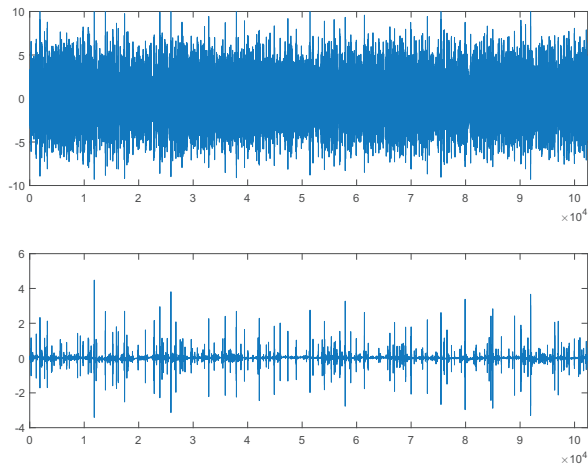## 5 Conclusions

test

    test

        test

Fig. 4. The above figure is the original data, and the figure below is the data after noise reduction. By noise reduction, a large amount of noise is filtered out so that features can be more easily resolved.
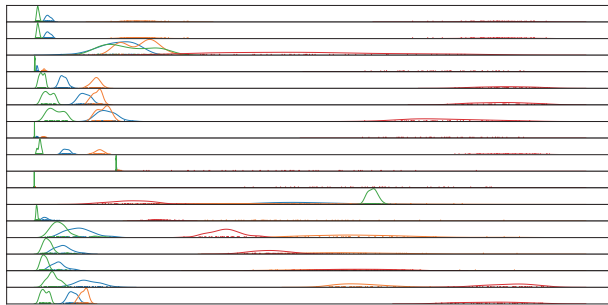

Fig. 5. The figure includes a total of eighteen axes. In each axis, curves of different colors represent different fault type data. It can be seen that different fault types have distinct feature distributions, so we should extract features which can distinguish different fault types easily and classify them.
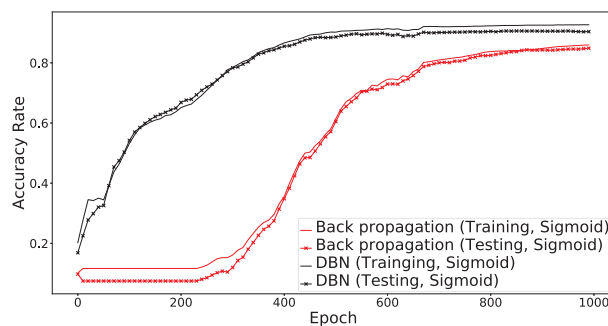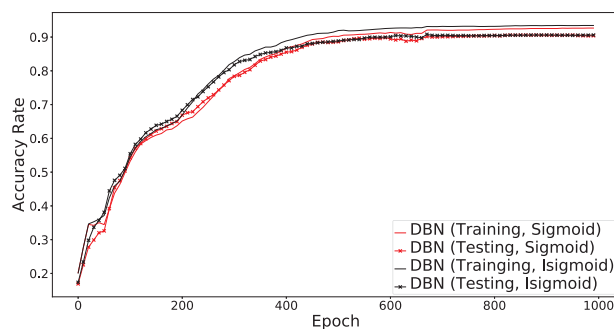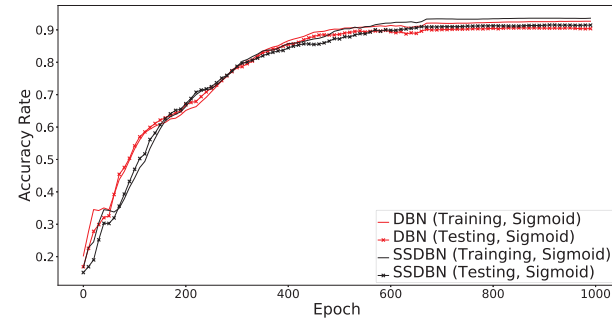

Fig. 6. Accurancy.

Fig. 7. Accurancy.

Fig. 8. Accurancy.

Acknowledgement

References

[1] Yi Wang, Guanghua Xu, Lin Liang, and Kuosheng Jiang. Detection of weak transient signals based on wavelet packet transform and manifold learning for rolling element bearing fault diagnosis. *Mechanical Systems & Signal Processing*, 54-55:259–276, 2015.

[2] L. U. Yongle, Yingjun Pan, Chunhua Ren, Yu Liu, and Hui Peng. Zero drift compensation of mems accelerometer based on wavelet packers-neural network. *Piezoelectrics & Acoustooptics*, 37(1):27–31, 2015.

[3] Geoffrey E. Hinton. A practical guide to training restricted boltzmann machines. *Momentum*, 9(1):599–619, 2012.

[4] Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

[5] M. A. Kramer and J. A. Leonard. Diagnosis using backpropagation neural networksanalysis and criticism. *Computers & Chemical Engineering*, 14(12):1323–1338, 1990.

[6] ZHAO Guangquan, GE Qiangqiang, LIU Xiaoyong, and Peng Xiyuan. Fault feature extraction and diagnosis method based on deep belief network. *Chinese Journal of Scientific Instrument*, 39(7):1946–1953, 2016.

[7] Feng Jia, Yaguo Lei, Jing Lin, Xin Zhou, and Na Lu. Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. *Mechanical Systems and Signal Processing*, 72:303–315, 2016.

[8] Qin Yi, Wang Xin, and Zou Jingqiang. The optimized deep belief networks with improved logistic sigmoid units and their application in fault diagnosis for planetary gearboxes of wind turbines. *IEEE Transactions on Industrial Electronics*, 2019.

[9] B. Walczak and D. L. Massart. Noise suppression and signal compression using the wavelet packet transform. *Chemometrics & Intelligent Laboratory Systems*, 36(2):81–94, 1997.

[10] Bin Zhang, Shaohui Zhang, and Weihua Li. Bearing performance degradation assessment using long short-term memory recurrent network. *Computers in Industry*, 106:14 – 29, 2019.

[11] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[12] Geoffrey E Hinton, Terrence J Sejnowski, et al. Learning and relearning in boltzmann machines. *Parallel distributed processing: Explorations in the microstructure of cognition*, 1(282-317):2, 1986.

[13] ZHANG Chun-xia, JI Nan-nan, and WANG Guan-wei. Restricted boltzmann machines. *CHINESE JOURNAL OF ENGINEERING MATHEMATICS*, 2015.

[14] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

[15] Gongming Wang, Junfei Qiao, Xiaoli Li, Lei Wang, and Xiaolong Qian. Improved classification with semi-supervised deep belief network. *IFAC-PapersOnLine*, 50(1):4174–4179, 2017.

[16] Electrical Engineering Laboratory of Case Western Reserve University. Bearing data. *Case Western Reserve University Bearing Data Center Website*.