Deep Learning-Based Motor Fault Diagnosis

Authors: He Yiyi, He Yiyi

Date: 2024-01-07T00:00:00+00:00

Abstract

Traditional motor fault diagnosis techniques are typically based on single-type state parameters, such as vibration parameters or electrical parameters. However, the monitoring scope of single-type motor state parameters is often very limited, making it difficult to meet the requirements for comprehensive and integrated fault diagnosis of motors. This study aims to propose a comprehensive motor fault diagnosis method by fusing two types of parameters—vibration data and current data—to improve diagnostic reliability and accuracy. Furthermore, considering that in actual industrial and production environments, the cost of obtaining large-scale labeled samples is often high or even infeasible, this paper further studies and improves neural networks, proposing a few-shot fault diagnosis network based on RNN and attention mechanisms. This paper utilizes motor fault feature extraction methods to investigate vibration and current signal characteristics of motors under different fault conditions. The employed fault feature extraction methods include: Fast Fourier Transform (FFT) and Hilbert-Huang Transform. According to the actual data fusion requirements of this paper, an overall data fusion implementation scheme is designed, sequentially utilizing Fast Fourier Transform (FFT), Hilbert-Huang Transform (HHT), and Convolutional Neural Network (CNN) cascaded with Multi-Layer Perceptron (MLP) to extract fault features, fusing motor vibration and current parameters for integrated fault identification and diagnosis of motors. Results demonstrate that motor fault diagnosis technology employing data fusion methods can improve the accuracy of diagnostic results, reduce uncertainty caused by single parameters, thereby enhancing the accuracy of motor fault diagnosis. The designed few-shot fault diagnosis network is used to identify equipment health conditions under small sample conditions, where the attention mechanism captures spatial and channel relationships of signals. Using single experimental samples, the network employed in this paper is verified to possess advantages in diagnostic efficiency and accuracy under different few-shot working conditions.

Full Text

Chapter 1 Introduction

Traditional motor fault diagnosis techniques typically rely on a single type of state parameter, such as vibration or electrical parameters alone. However, the monitoring scope of any single parameter type is often very limited, making it difficult to meet the comprehensive fault diagnosis requirements for electric motors. This research proposes an integrated motor fault diagnosis method by fusing two types of parameters—vibration data and current data—to improve diagnostic reliability and accuracy. Building upon this data fusion approach, we further consider that in real industrial and production environments, the cost of obtaining large-scale labeled samples is often prohibitively high or even infeasible. Therefore, we propose a few-shot fault diagnosis network based on Recurrent Neural Networks (RNN) and attention mechanisms through further learning and improvement of neural networks.

This study investigates motor vibration and current signal characteristics under different fault conditions using motor fault feature extraction methods. The employed feature extraction methods include Fast Fourier Transform (FFT) and Hilbert-Huang Transform (HHT). According to the actual data fusion requirements of this paper, we design an overall data fusion implementation scheme that sequentially utilizes FFT, HHT, and a Convolutional Neural Network (CNN) cascaded with a Multi-Layer Perceptron (MLP) to extract fault features. By fusing motor vibration and current parameters, we achieve comprehensive fault identification and diagnosis for electric motors. The results demonstrate that motor fault diagnosis technology employing data fusion methods can enhance diagnostic accuracy and reduce uncertainty caused by single-parameter analysis, thereby improving overall motor fault diagnosis performance. The designed few-shot fault diagnosis network is used to identify equipment health status under limited sample conditions, where the attention mechanism captures spatial and channel relationships in signals. Using single experimental samples, we verify that our network exhibits advantages in diagnostic efficiency and accuracy across different few-shot working conditions.

Chapter 2 Convolutional Neural Networks

2.1 Convolutional Layer

The convolutional layer is a key component of Convolutional Neural Networks (CNN) used for feature extraction. The convolution kernel is the most critical element in a convolutional layer, responsible for performing convolution operations on input signals to extract and generate corresponding features. The size and shape of convolution kernels are predetermined, and their weight parameters are automatically learned during training to suit specific tasks, allowing kernel design to be adjusted according to task requirements. In CNNs, each convolutional layer typically contains multiple kernels, with each capable of extracting different features. By optimizing kernel weight parameters during training, the network can automatically learn the most informative feature representations for input data. The specific convolution process can be expressed by the following formulas:

$$Z_n = W_n * X_{n-1} + b_n$$
$$X_n = \sigma(Z_n) = \sigma(W_n * X_{n-1} + b_n)$$

where n denotes the n-th convolutional layer, $*$ represents convolution operation, b denotes bias parameters, W represents convolution kernels, and σ denotes the activation function.

The convolution operation process is shown in Figure 2 [Figure 2: see original paper]-1. The use of convolution kernels enables CNNs to automatically extract discriminative features from raw input data that are significant for task execution. By stacking multiple convolutional layers, the network can gradually extract higher-level features, achieving understanding and representation of complex patterns and abstract concepts.

2.2 Pooling Layer

The pooling layer is an important component of CNNs, often used alternately with convolutional layers for feature extraction. It compresses and summarizes information from local regions of input feature maps through aggregation operations, serving three main purposes: (1) reducing spatial dimensions by decreasing feature map size and computational load; (2) extracting key features by preserving the most salient characteristics from local regions; and (3) providing translation invariance, as pooling operations are unaffected by input translations, thereby enhancing network robustness. Notably, pooling operations have no trainable parameters and represent fixed operations that do not introduce additional model parameters. Subsequent layers can add more convolutional and pooling layers to extract higher-level features.

The two primary pooling operations are Max Pooling and Average Pooling, illustrated in Figure 2-2. The operation principle is as follows: (1) the pooling layer divides the input feature map into non-overlapping rectangular regions (pooling windows); (2) for each pooling window, the maximum value (Max Pooling) or average value (Average Pooling) within the window is selected as the output feature.

2.3 Activation Function

In one-dimensional CNNs, activation functions are typically applied to convolutional layer outputs to enhance network nonlinearity. Common activation functions include ReLU, Sigmoid, and Tanh. Activation functions perform nonlinear transformations on inputs, enabling networks to learn more complex patterns. ReLU (Rectified Linear Unit) is a widely used activation function with the formula:

$$\mathrm{ReLU}(x) = \max(0, x)$$

where $\max(0, x)$ selects the larger value between x and 0, converting all negative values to 0 while preserving non-negative values. The ReLU activation function offers advantages such as ease of use, high computational efficiency, and fast convergence, making it widely adopted in deep learning.

2.4 Fully Connected Layer

In CNNs, the Fully Connected Layer is typically positioned between convolutional/pooling layers and the output layer, integrating and mapping extracted features to produce final classification results or regression predictions. It constitutes an important component of CNNs, providing global understanding and holistic modeling capabilities for input data. As shown in Figure 2-3, its primary role is performing nonlinear mapping and combination of features to better fit the data, capturing complex relationships between input features and extracting higher-level features through learned weights and biases. Fully connected layers typically use activation functions to introduce nonlinearity and increase network representation capacity. During CNN training, the weights and biases of fully connected layers are trainable parameters updated through backpropagation algorithms and optimizers to minimize loss functions and improve network fitting.

2.5 Multi-Layer Perceptron Algorithm Principles

The Multi-Layer Perceptron (MLP) is a fundamental feedforward neural network model widely applied in machine learning and deep learning tasks. It consists of an input layer, output layer, and one or more hidden layers, with each layer comprising multiple neurons (or nodes) that transmit information through connection weights. Its basic structure is shown in Figure 2-4.

Input Layer: Receives input data, with each neuron corresponding to one feature in the input vector. Assuming the input layer has n neurons (corresponding to n-dimensional features), the input vector is $\mathbf{x} = (x_1, x_2, \dots, x_n)$.

Hidden Layer: Situated between input and output layers, hidden layers perform nonlinear transformations and feature extraction on input data, mapping it to a higher-dimensional representation space. Each hidden neuron receives outputs from the previous layer, performs weighted summation through connection weights, and applies an activation function for nonlinear transformation to produce its output. This process can be expressed as:

$$z_j = f(v_{j1}x_1 + v_{j2}x_2 + \cdots + v_{jn}x_n + b_j)$$

where $v_{j1}, v_{j2}, \dots, v_{jn}$ are weights connecting the input and hidden layers, b_j is the bias term for the j-th neuron, and $f(\cdot)$ is the activation function.

Output Layer: Produces the network's final results, with each neuron corresponding to a class or value. The number of output neurons depends on the problem type. This process can be expressed as:

$$y_k = g(w_{k1}z_1 + w_{k2}z_2 + \cdots + w_{km}z_m + c_k)$$

where $w_{k1}, w_{k2}, \dots, w_{km}$ are weights connecting hidden and output layers, c_k is the bias term for the k-th neuron, and $g(\cdot)$ is the activation function.

MLPs are trained through backpropagation, which adjusts weights and biases by minimizing loss functions. Common loss functions include mean squared error and cross-entropy. Backpropagation computes gradients of the loss function with respect to weights and biases, then updates parameters accordingly to minimize the loss.

In summary, MLP is a feedforward neural network that achieves nonlinear mapping through multiple hidden layers and activation functions, ultimately generating predictions through a softmax function in the output layer. Through backpropagation, network parameters can be trained to minimize the loss function, enabling effective modeling and classification of input data.

Chapter 3 Fusion Method Design and Experimental Data Analysis

3.1 Experimental Data Acquisition

Fault data for three-phase asynchronous motors was collected from an experimental test bench. The fault simulation test bench overview and sensor layout are shown in Figure 3 [Figure 3: see original paper]-1. Measurement point details are provided in the table below:

Table 3 -1. Motor Measurement Point Description - Motor free-end bearing horizontal measurement point (vibration) - Motor free-end bearing vertical measurement point (vibration) - Motor drive-end bearing horizontal measurement point (vibration) - Motor drive-end bearing vertical measurement point (vibration) - Motor housing vertical measurement point (vibration) - Motor Phase A current measurement point - Motor Phase B current measurement point - Motor Phase C current measurement point - Motor speed measurement point

Motor specifications are shown in the following table:

Table 3-2. Motor Parameter Table - Rated power, Rated current, Rated speed, Rotor bar count, Stator slot count, Pole count: 6.31A, 2860rpm, 10N • m

The collected data includes: normal operation (norm), rotor broken bar fault (Broken_{RotorBar}), dynamic eccentricity (Dynamic_{eccentricity}), static eccentricity (Static_{eccentricity}), bearing cage fault (Bearing_{cage}), bearing inner ring fault (Bearing_{innerring}), and bearing outer ring fault (Bearing_{outerring}). The sampling frequency is 40960Hz (except for Static_{eccentricity} at 12800Hz). All data were collected under 50Hz power supply frequency at 100% load condition (full load), with actual speed around 2870rpm.

3.2 Overall Data Fusion and Fault Diagnosis Scheme Design

Vibration monitoring and electrical monitoring are commonly used methods in motor fault diagnosis that can cover most mechanical and electrical faults. Therefore, this paper primarily employs vibration data and current data fusion for motor fault identification and diagnosis. The fusion approach uses simultaneous data from different spatial locations in a parallel, feature-level fusion manner. The overall technical flow is shown in the figure and mainly includes: (1) research and analysis of fault mechanisms and characteristics in the dataset; (2) feature extraction using Hilbert-Huang Transform (HHT) to extract vibration data envelope spectra and Fast Fourier Transform (FFT) to extract current data frequency spectra, followed by training and testing set division; (3) neural network fusion and recognition by training the network with training data and evaluating it with test data to output classification results and accuracy statistics; and (4) comparative experiments using vibration data alone and current data alone to evaluate the effectiveness of the fusion network.

3.3.1 Fault Feature Extraction Scheme Design

Current data typically contains periodic and instantaneous information, while vibration data may contain frequency and amplitude variation information. Theoretically, FFT can transform current signals to the frequency domain to extract spectral information for analyzing frequency components and detecting anomalies or resonance phenomena. For vibration data, HHT can perform time-frequency analysis to capture signal variations across time and frequency, extracting instantaneous frequency, energy distribution, and time-frequency characteristics to identify abnormal vibration patterns and fault features.

To select and validate the feature extraction approach, three schemes were designed: (1) using FFT to obtain frequency spectra for both vibration and current data as fault features; (2) using HHT to obtain envelope spectra for both vibration and current data as fault features; and (3) using HHT envelope spectrum for vibration data and FFT frequency spectrum for current data as fault features. The third scheme will be verified as the optimal choice after designing the specific data fusion method.

3.3.2 Raw Fault Data and Feature Extraction Display

Vibration and current waveforms, FFT spectra, and HHT envelope plots for normal state, dynamic eccentricity fault, rotor broken bar fault, bearing cage fault, bearing inner ring fault, and bearing outer ring fault are shown in Figures 3-2 and 3-3.

Figure 3-2 shows raw data and features for normal state, dynamic eccentricity fault, and rotor broken bar fault. **Figure 3-3** shows raw data and features for bearing cage fault, bearing inner ring fault, and bearing outer ring fault.

3.4.1 Neural Network Structure Design

Given the experimental data, a two-network cascade approach is proposed, establishing a deep CNN (Net1) cascaded with an MLP (Net2). Net1 defines a dual-channel CNN model that can simultaneously input 2D vibration and current data for further feature extraction and fusion. The specific structure is detailed in the table below:

Table 3-3. Net1 Convolutional Neural Network Model Structure - **conv1**: Input channels 2, kernel size 3, padding 1 (captures local features at different scales), Batch normalization (32 channels, normalizes per-channel data to accelerate training, improve convergence, and enhance model stability), Max pooling (downsamples to reduce feature map dimensions and computational load) - **conv2**: Input channels 32, kernel size 3, padding 1 (captures local features), Batch normalization (64 channels), Max pooling (downsamples) - **conv3**: Input channels 64, kernel size 3, padding 1 (captures local features), Batch normalization (256 channels), Max pooling (downsamples) - **fc1**: Input features 256*91, Dropout layer (drop probability 0.5, randomly drops nodes to prevent overfitting and improve generalization)

Net2 is an MLP for fault type classification and recognition, with the following structure:

Table 3-4. Net2 Multi-Layer Perceptron Network Structure - **fc1**: Fully connected layer maps input feature vectors to specified length with linear transformation, using ReLU activation to enhance nonlinear fitting capability - **fc2**: Fully connected layer maps hidden layer outputs to class-specific space, followed by Softmax function that normalizes each category's output to probability values for classification

The final fault diagnosis architecture is shown in Figure 3-4.

3.4.2 Experimental Results Analysis

3.4.2.1 Neural Network Fusion Classification Effect Verification Using HHT-extracted vibration envelope spectrum features and FFT-extracted current frequency spectrum features, we divided the data into training and test sets at a 7:3 ratio and input them into the constructed network. The training and validation results are as follows:

1. **Vibration and current data fusion**: Accuracy and loss function curves are shown in the figure, achieving 76.9% accuracy on the training set.
2. **Vibration data only**: Accuracy and loss function curves are shown in the figure, achieving 68.3% accuracy on the training set.
3. **Current data only**: Accuracy and loss function curves are shown in the figure, achieving 58.7% accuracy on the training set.

Due to potential experimental randomness, five additional experiments were conducted and averaged. The results are shown in the table:

Table 3-5. Neural Network Fusion Classification Effect Verification

Method	Classification Accuracy	Average Accuracy
Vibration & Current Fusion	80.3%, 76.6%, 76.9%, 77.2%, 78.7%	77.94%
Vibration Only	61.3%, 66.4%, 70.9%, 69.1%, 67.2%	67.2%
Current Only	58.7%, 64.1%, 59.4%, 61.5%, 61.1%	60.96%

The statistics demonstrate that the average classification accuracy using fused vibration-current data is 77.94%, compared to 67.2% using vibration data alone and 60.96% using current data alone, proving that the neural network has excellent data fusion and fault classification capabilities.

Feature extraction and classification visualization before and after fusion are shown in Figure 3-8, demonstrating effective feature extraction and fusion with good fault separation.

3.4.2.2 Feature Extraction Method Selection and Verification To validate the three proposed feature extraction schemes, three tasks were constructed. Due to potential experimental randomness, five data fusion and classification experiments were conducted for each scheme, with accuracy statistics shown in Table 3-6:

Table 3-6. Feature Extraction Method Verification Experiments

Scheme	Classification Accuracy	Average Accuracy
Vibration: HHT, Current: FFT	80.3%, 76.6%, 76.9%, 77.2%, 78.7%	77.94%
Both Vibration & Current: HHT	73.3%, 70.6%, 70.9%, 68.5%, 66.3%	69.92%
Both Vibration & Current: FFT	68.7%, 64.1%, 60.4%, 61.5%, 69.1%	64.76%

The results completely align with theoretical analysis: using HHT for vibration envelope spectrum features and FFT for current frequency spectrum features yields the best fault identification and classification performance.

Chapter 4 Few-Shot Fault Diagnosis Based on RNN and Attention Mechanism

In real industrial and production environments, obtaining large-scale labeled samples is often costly or even infeasible. Therefore, few-shot fault diagnosis better aligns with practical application scenarios, adapting to data limitations and difficulties in acquiring large samples while enhancing system robustness when certain sensors fail. Under resource constraints, few-shot methods either leverage regularization techniques and model feature extraction advantages, generate high-quality samples based on real data distributions, or apply emerging machine learning techniques. Large convolutional kernels enhance robustness, while deep small kernels effectively extract abstract features. Moreover, time-step information cannot be ignored in vibration signals, making RNNs more suitable than CNNs for this requirement.

Building upon previous fault diagnosis methods and further studying deep learning, we propose a few-shot fault diagnosis network that uses dual convolutional layers to extract high and low-frequency signal features, an attention mechanism to weight fused features and focus on primary spectra, and Bi-RNN to capture hidden information at different time sequence positions.

4.1 Recurrent Neural Network (RNN) and Derivative Networks

Standard RNNs perform data analysis based on time sequences, requiring reference to previous moment data to predict future changes. When data processing needs to connect 前后 moments or when data has strong time dependencies, such network structures often produce significant errors. Additionally, these networks require large-scale training samples; with insufficient elements, ideal models are difficult to obtain. Therefore, this paper adds a reverse-time RNN to the original model, with both RNNs sharing input and output layers at each time step. The Bi-RNN structure is shown in Figure 4 [Figure 4: see original paper]-1.

4.2 Dual-Path Convolution and Feature Fusion

The dual convolutional layer structure employs two different paths to extract high and low-frequency signal features. The first path uses two larger convolutional kernels to learn low-frequency features, which literature has proven can enhance model robustness to noise. The second path uses small convolutional kernels, deepening the network through four nonlinear activation layers to improve discriminative capability. Features are ultimately fused through element-wise multiplication, ensuring each channel contains rich information.

AdaBN is introduced to improve model adaptability across different domains. It replaces traditional Batch Normalization (BN) by adjusting statistics from source to target domains to enhance model generalization capability.

4.3 Attention Mechanism

The attention mechanism is a key technology in neural networks that enables models to flexibly focus on important parts when processing sequential or set data, thereby improving performance. The basic structure is shown in Figure 4-2. It mainly includes: (1) context weight allocation, where the mechanism assigns different weights to each element in the input sequence, allowing the model to focus on task-relevant information; and (2) adaptability, enabling the model to adjust each element's importance according to different input situations.

As a powerful technique, attention mechanisms can be given different structures to flexibly focus on important parts of input sequences, improving neural network performance and efficiency in processing sequential and set data. Its wide application provides stronger modeling capabilities for few-shot fault diagnosis.

4.4 Neural Network Structure Design

We propose a few-shot fault diagnosis method comprising data augmentation, dual-path convolution, attention mechanism, Bi-RNN, Global Average Pooling (GAP), and a diagnosis layer, as shown in Figure 4-3.

Based on the experimental data introduced in Chapter 3, using vibration data alone or current data alone achieves accuracy up to 99% with very short processing time, demonstrating diagnostic efficiency advantages, as shown in Figures 4-5 and 4-6.

This implementation realizes the few-shot fault diagnosis concept, providing a reliable solution for fault detection in actual industrial scenarios with high accuracy, fast efficiency, and good adaptability to small sample data.

All code mentioned in this paper has been uploaded to GitHub at https://github.com/itsnot11/deeplearning.git. Both experimental datasets involve confidential information and have not been uploaded. The first data fusion network includes complete code for CNN, MLP, training, and testing modules. The second few-shot fault diagnosis network includes the three most important code components: activation functions, attention mechanism, and Adamp optimizer implementation.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv — Machine translation. Verify with original.