Gear Fault Diagnosis Based on GPU-CNN

Zhipeng Dong¹, Shuo Meng¹, Kuo Chi¹ and Jianshe Kang¹
¹ Army Engineering University of PLA, Shijiazhuang, China

E-mail:398198011@qq.com

Abstract. Gear is widely used in mechanical transmission system, but it is prone to failure, which seriously affects the performance of equipment. In order to realize the diagnosis and classification of gear faults, we used Convolutional Neural Networks (CNN) to extract time-frequency image features of its vibration signal. CNN can extract the characteristics of time-frequency signals from vibration signals and identify gear faults accurately. However, due to the large training data set, CNN training costs too much time. According to the characteristics of Graphics Processing Unit (GPU), Compute Unified Device Architecture (CUDA) can improve the speed of CNN algorithm and reduce the time consumption. Therefore, this paper proposes a method based on GPU-CNN for gear fault diagnosis. The experimental results show that the method can effectively shorten the training time and significantly improve the operation efficiency.

1. Introduction

The first paragraph after a heading is not indented (Bodytext style) Vibration signal analysis technology is the main direction of gear fault detection research. According to the theory of gear meshing vibration, acceleration sensor is used to collect gear vibration signal, and signal characteristics are extracted and analyzed to diagnose and classify gear faults.

In this paper, we input time-frequency images of gear vibration signal into the deep neural networks to extract the time-frequency image features to realize the diagnosis and classification of gear faults. The application of artificial intelligence for fault diagnosis, reliable fault characteristics is the premise. Artificial intelligence methods usually need multiple features or feature vectors as input. Therefore, the more effective fault features extracted, the better. Most of the fault features are extracted by signal processing method, and then used as inputs to train the artificial intelligence method. When a new fault sample is obtained, the trained model can be used to identify the fault state of the new sample. At present, artificial intelligence methods widely used in fault diagnosis are: artificial neural network (ANN) [1], hidden Markov model (HMM) [2-5], hidden semi Markov model (HSMM) [6-9], support vector machine (SVM) [10-16], Relevance vector machine (RVM) [17-21], Bayesian networks (BN), dynamic Bayesian networks (DBN), k nearest neighbor (KNN) [22], etc.

Convolutional neural network (CNN) can effectively train and classify the gear vibration signal state characteristic information, and the parallel computing application of GPU in CUDA architecture can accelerate the implementation of the algorithm.

2. Fault characteristic signals of gear

Please follow these instructions as carefully as possible so all articles within a conference have the same title page. This paragraph follows a section title so it should not be indented.
2.1. Gear vibration signal composition and fault characteristics

In the process of gear meshing, the load and rotation speed fluctuation and other factors cause faults, resulting in amplitude and frequency changes of vibration signal and phenomenon of amplitude modulation (AM), frequency modulation (FM) or AM and FM modulation. As shown in the following Eq. (1):

\[ X_c(t) = \sum_{m=0}^{M} A_m [1 + a_m(t)] \cos[2\pi f_z t + \phi_m + b_m(t)] \]  

(1)

Where \( a_m(t) \) represents the amplitude modulated signal and \( b_m(t) \) represents the frequency modulated signal.

When the fault occurs, the vibration signal of the gear will be modulated to different degrees (Table 1). The energy distribution of the vibration signal will change compared with that of the normal state. In the signal spectrum diagram, the characteristics of the gear in fault state can be seen from the sideband. Usually, when gear failure occurs, sideband will increase. At the same time, the spacing of side frequency can reflect the source of fault information. Therefore, the sideband modulation is the carrier of fault information, and the analysis and study of the characteristics of side frequency in various fault states are helpful for more efficient and accurate diagnosis and identification of gear faults, and have important practical significance for transmission fault diagnosis.

Table 1. List of the signal characteristics of three faulty gears in the spectrum diagram

| State type       | Modulation signal                          | Carrier signal                  | Side band         | Frequency spacing | Frequency amplitude | Vibration energy |
|------------------|--------------------------------------------|---------------------------------|-------------------|-------------------|---------------------|------------------|
| Broken teeth     | frequency conversion & frequency doubling | meshing frequency/natural frequency of gears | wide and high     | larger            | larger              | extent           |
| Tooth surface wear | frequency conversion & frequency doubling | meshing frequency & its harmonics | less and little   | smaller           | smaller             | extent           |
| Lack of teeth    | frequency conversion & frequency doubling | meshing frequency & its harmonics | narrow and short  | smaller           | smaller             | to a certain extent |

2.2. Vibration signal analysis of gear faults

At present, fault diagnosis methods based on vibration signal analysis mainly include time domain analysis, frequency domain analysis and time-frequency domain analysis. The first two are based on the processing of stationary signals, and the non-stationary signals cannot be analyzed and processed effectively. The vibration signals of the fault gears are all non-stationary signals, and some fault characteristics exist in these non-stationary signals. Time-frequency analysis can directly reflect the relationship between the development and change of the main components of each frequency of the signal over time. The processing methods include short-time Fourier transform, continuous wavelet transform and S-transform.

2.2.1 Short-Time Fourier Transform

By windowing the continuous time domain signal \( x(t) \), the Fourier Transform is applied to each signal...
as a transient and stationary small signal. The definition of the Short-Time Fourier Transform (STFT) is in Eq.(2):

$$STFT(t, f) = \int_{-\infty}^{\infty} x(t)\omega(t-\tau)e^{-j2\pi ft}\,d\tau$$

(2)

Where $x(t)$ represents the time domain signal, $\omega(t-\tau)$ represents the window function, $\tau$ represents the center of window function.

When the window function is selected, the resolution is fixed as a result of a fixed window length of the window function. Since the time resolution is inversely proportional to the frequency resolution, the frequency resolution is higher when the window length is longer, while the time resolution is higher when the window length is shorter[23]. However, in the process of actual signal analysis, good frequency resolution is required at low frequency and good time resolution at high frequency[25], which makes the application of STFT have certain limitations.

2.2.2 Continuous wavelet transform

Suppose the function $\varphi(t) \in \mathcal{L}$, if its Fourier Transform $\varphi(f)$ satisfies the following conditions Eq.(3)

$$C_{\varphi} = \int_{-\infty}^{\infty} \left|\frac{\varphi(\omega)}{\omega}\right|^2 d\omega < \infty$$

(3)

$\varphi(t)$ is called a mother wavelet or a base wavelet. After scaling and shifting the parent wavelet, a cluster of wavelet functions can be generated, whose expression is in Eq 4:

$$\varphi_{a,b}(t) = \frac{1}{\sqrt{a}}\varphi\left(\frac{t-b}{a}\right)$$

(4)

Where $a$ represents the scale factor, and $b$ represents the translation factor. The scale factor $a$ is used to scale the wavelet base, and the translation factor $b$ is used to change the position of the window on the time axis.

The continuous wavelet transform of signal $x(t)$ is defined as in Eq.(5):

$$CWT(a, b) = \frac{1}{\sqrt{a}}\int_{-\infty}^{\infty} x(t)\varphi\left(\frac{t-b}{a}\right)dt$$

(5)

In contrast to the STFT, the wavelet basis of the wavelet transform is not an infinitely long trigonometric function, but a finite length decaying wavelet basis function. The wavelet base can be scaled to solve the problem that time resolution and frequency resolution cannot be achieved simultaneously[24]. Morlet wavelet has the characteristic of shock attenuation waveform, so Morlet wavelet is often chosen as the wavelet basis.

2.2.3 S-transformation

Signal $x(t)$, whose one-dimensional continuous S-transformation definition is in Eq.(6):

$$S(\tau, f) = \int_{-\infty}^{\infty} x(t) \frac{|f|}{\sqrt{2\pi}} e^{-\frac{(t-\tau)^2}{2}} e^{-j2\pi ft} dt$$

(6)

$$\frac{|f|}{\sqrt{2\pi}} e^{-\frac{(t-\tau)^2}{2}}$$

is gaussian window function.

The corresponding inverse transformation of $S$ is defined as Eq.(7):
\[ x(t) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} S(\tau, f) e^{j2\pi f \tau} d\tau df \]  

(7)

It can be seen from Eq.(6) and Eq.(7) that S-transformation is a special Fourier Transform with tuned Gaussian window. Gaussian window function is a function of both time and frequency. The window width is controlled by frequency. The window width is large at low frequency and small at high frequency. This enables the window function to provide a better frequency resolution at low frequencies and a better time resolution at high frequencies, and overcomes the shortcoming of STFT with a fixed resolution.

3. Deep Neural Networks

Deep neural networks (DNN) can be understood as neural networks with many hidden layers, also known as deep feedforward networks (DFN) and multi-layer perceptron (MLP). They include deep belief network (DBN), sparse auto-encoder (SAE) and convolutional neural network (CNN)[26-27].

3.1 Deep Belief Network (DBN)

DBN is a probabilistic generation model. It establishes a joint probability distribution between observation data and labels and evaluates both \( P(\text{Observation} | \text{Label}) \) and \( P(\text{Label} | \text{Observation}) \). Structurally, it is made up of multiple limited Boltzmann machine (RBM). In the training, the layered training method is adopted to solve the problem that the traditional neural network training method is not applicable to the multi-layer network training. The network structure is shown in Figure 1.

![Figure 1. Structure of deep belief network](image_url)

3.2 Sparse Auto-Encoder (SEA)

SAE is a hidden layer neural network with identical input and output. The input dimension must be larger than the output dimension, which belongs to unsupervised learning. A basic SAE can be thought of as a three-layer neural network: an input layer, a hidden layer, and an output layer, where the output layer is of the same size as the input layer. SAE adds a sparse constraint condition on the basis of SAE. In other words, the characteristics obtained at a time mean code is sparse as far as possible (most nodes in each layer are 0, and a few are not), making the expression more effective. The network structure of SAE is shown in Figure 2.
3.3 Convolutional Neural Network (CNN)
CNN is used at local awareness in the research of cerebral cortex in the cat and the direction choice of neurons when found the unique network structure. CNN, a kind of completely link neural network structure, can efficiently reduce the complexity of feedback neural network and widely used in deep learning. CNN has the characteristics of the rotation, translation invariance, which achieved fairly good results in the field of image recognition[28]. The network structure of CNN is shown in Figure 3.

4. CNN implementation of CUDA
A large number of samples are needed to train the CNN. Therefore, for the vibration signals collected
in each column, the signals are divided into several small segments of a certain length, and then time-
frequency transformation is performed on these small segments to obtain multiple time-frequency
graphs. The pixels of the time-frequency graph are adjusted to the appropriate size for input into the
CNN. Through multi-layer feature extraction of input data, each layer extracts more abstract features
on the basis of the previous layer. The input image data is pixel points, and becomes edge after further
extraction. Extracting the features of this layer again will eventually provide you some features of the
whole object. The higher the features are obtained, the more applications of classification recognition
are available. However, with the increase of input data, the calculation time will definitely increase
sharply, and the CPU training model time becomes more and more unacceptable. So, parallel is an
inevitable choice. Due to its multicore structure, GPU is destined to be the inevitable choice of
optimization algorithm.

4.1 CUDA
CUDA is a computing platform developed by NVIDIA that can process large, complex, parallel
applications using GPU. A large number of algorithms are used in the process of CNN training and
recognition of time-frequency image input of gear faults, and each complex algorithm inevitably
brings a large amount of computation. Traditional serial programming can no longer meet the need of
recognition, but CUDA technology can.

GPU runs on the device in the form of kernel function, but only these parts of the program perform
parallel computation on the GPU. The compilation process of CUDA program is shown in Figure 4.
CUDA programs include the host code running on CPU and the device code running on GPU (parallel
code kernel). The NVIDIA CUDA compiler divides the code into these two parts, and then the device
code is put on GPU by the specific compiler.

![Figure 4. The compilation process of CUDA program.](image)

The data in CNN includes images, convolution kernel and convolution results, and parallelization
is realized in the process of forward propagation and error back propagation. The parallelization
process includes weight update, gradient calculation and sensitivity deconvolution. Next, we first
introduce several data storage methods, and then illustrate the process of parallelization with the
parallelization of gradient computation.

4.2 Data storage mode
The convolutional layer mainly uses convolution kernels to realize feature extraction. The storage of all convolution kernels in a convolutional layer is as follows: the information in each line of the convolution kernel is read successively, and then the information in different channels is spliced. The storage mode of image information in convolutional layer is similar to that of convolution kernel.

The modular segmentation of images facilitates the calculation of pixel positioning. The size of modules is usually the size of the convolution kernel, and the total number of modules is the number of convolution processes carried out by the convolution kernel. The result of convolution is a cube, and then the information inside the cube is stored in a one-dimensional array. Only after the completion of each convolution process can a pixel value on the corresponding feature map plane be obtained. Each image goes through a convolution kernel and the convolution produces a new mapping plane. The pixel storage order of the feature map is stored as a column vector in the sequence of module Numbers. The storage order of the whole cube is to store a vertical plane along the X-axis, and then along the Y-axis according to the order of the plane, until the cube is all stored in the one-dimensional array.

4.3 Specific process of parallelization
The layout of threads, thread blocks and grids during parallelization is shown in Figure 5. When the convolution kernel is updated, each module updates the ownership value of the convolution kernel. If there are multiple images, for a thread, the update values of the corresponding module on each image are calculated. To parallelize weight updates, we first need to define three parameters: the length of the bx thread block in the X direction, the length of the by thread block in the Y direction, and the number of pixelsPerThread processed by pixelsPerThread.

4.4 Thread layout within a thread block
There are a total number of bx*by size threads in a block. Which convolution kernel is updated by a thread in the Block is determined by Thread.x. Thread.y determines which part of the pixel value of a convolution kernel is updated to calculate the weight of pixelsPerThread. In summary, a thread block is updated by*pixelsPerThread for different pixels of the number of bx-sized convolution kernels based on a module.

In the case of gradient parallel computation, the thread layout within a thread block is shown in Figure 6. The pixelsPerThread-size weight update of a convolution kernel for each thread mainly consists of two processes: transferring the data to the shared memory (this step involves a complex positioning process) and completing the weight update. _syncthreads() is a built-in function of CUDA, which is used for communication of threads within a block to realize synchronization of threads within a block.
4.5 Overall layout of thread blocks within the grid

Figure 7 shows the overall layout of the block during the parallel calculation of gradient. Each module goes through all the convolution kernels for the convolution process. All convolution kernels appear in the X-axis direction of the block in the sequence of modules, and each module divides the same number of modules according to the number of convolution kernels. The Y-axis direction is divided into several groups of size by*pixelsPerThread according to the pixel size of a convolution kernel.

5. Comparative test and analysis
5.1 Test environment
Hardware environment:
I.Comprehensive mechanical fault simulation test bench: gear fault simulation experiment platform MCDS - II. The test bed is mainly composed of a motor, a sensor, a tachometer, a magnetic particle regenerator and a two-stage parallel gear box, and a corresponding supporting motor control software is used to adjust the motor speed. The structure of the test bed is shown as bellow.
Figure 8. Test bed

The relevant parameters of the gearbox rotating shafts in the test are shown in the table below.

Table 2. Bearing Box parameters

| Parameters of gearbox rotating shafts          | tooth number |
|-----------------------------------------------|-------------|
| Input shaft drive wheel G1                    | 41          |
| Intermediate shaft slave wheel G2             | 79          |
| Intermediate shaft drive wheel G3             | 36          |
| Output shaft drive wheel G4                   | 90          |

II. CPU: AMD Phenom(TM) II X4 B97 Processor @3.2GHz
III. The GPU: NVIDIA GeForce 405
Operating system:
Windows 7 64 bit flagship edition

5.2 Test steps

In the test, the sampling frequency was set as 40960 Hz and the motor speed was set as 1500 r/min to collect the test data in any interval under different fault conditions. The preinstalled fault gear is the intermediate shaft drive wheel G3, and the test respectively set normal, missing tooth, breaking tooth and wearing tooth, four different cases. Because deep learning models need a large amount of data for model learning, this experiment collected 100 sets of data for each fault condition mentioned, and the total data are 400 sets. Considering that the time-frequency analysis effect of S-transform is better, this paper adopts the time-frequency analysis method of S-transform to analyze the data

I. Time-frequency processing was performed on the test data collected above, and 400 time-frequency images were obtained. There are 100 images for each case, stored in four folders with fault names. Four images were randomly selected for each case, as shown in Figure 9.
II. 70% (280 pieces) of 400 time-frequency images were randomly selected as the training data of DBN, SAE and CNN3 deep learning models, and the remaining 30% (120 pieces) were used as the test data of the model.

III. Test the classification accuracy of three deep learning models running on CUDA platform and the acceleration ratio of CPU and GPU execution time.

In summary, the structural flow and related steps of this study can be summarized as shown in the following Figure 10.

Figure 9. Time-frequency image

Figure 10. The structural flow and related steps of this study
5.3 Analysis of test data

By permutation and combination of DBN, SAE, CNN and STFT, CWT, ST, nine combination modes are formed to obtain the accuracy of graph classification and the acceleration ratio of parallel and serial execution of the program[7]. The test results are shown in the following Table 3.

| Time-frequency method | Deep learning model | Accuracy | CPU time-consuming | GPU time-consuming | Speedup ratio |
|-----------------------|---------------------|----------|--------------------|--------------------|---------------|
|                       | DBN                 | 93.58%   | 3129.56s           | 2333.71s           | 1.34          |
| STFT                  | GPU-CNN             | 96.85%   | 185.69s            | 133.08s            | 1.40          |
|                       | SAE                 | 93.96%   | 2463.89s           | 1706.98s           | 1.44          |
|                       | DBN                 | 94.85%   | 298.36s            | 228.07s            | 1.31          |
| CWT                   | GPU-CNN             | 97.58%   | 192.65s            | 141.60s            | 1.36          |
|                       | SAE                 | 95.58%   | 2536.98s           | 1979.10s           | 1.28          |
|                       | DBN                 | 95.30%   | 3545.68s           | 2813.50s           | 1.26          |
| ST                    | GPU-CNN             | 99.05%   | 163.56s            | 123.78s            | 1.32          |
|                       | SAE                 | 97.68%   | 2456.39s           | 1969.53s           | 1.25          |

It can be seen from Table 2 that:

I. Among the three time-frequency analysis methods of STFT, CWT and ST, the operation time of CNN in the deep learning model is the least. This is because DBN and SAE have similar structures and adopt fully connected methods.

II. In the aspect of observation accuracy of calculation results, the time-frequency analysis method of time-frequency resolution is higher, the higher the accuracy of diagnosis results, also illustrates from the side, then the energy frequency graph clustering, the better, the higher the resolution, the image characteristics is more easy to extract, therefore, the higher the classification the accuracy of the results.

III. The program processing results combined with the above nine deep learning models and time-frequency analysis methods show that the speed of parallel execution is better than that of serial execution.

6. Conclusions

In this paper, we used time-frequency image of gear vibration signal into the deep neural networks to extract the time-frequency image feature to realize the diagnosis and classification of gear fault. Compared with DBN, SAE and CNN has obvious advantages in completing gear time-frequency signal image classification. Meanwhile, parallel execution of CNN on GPU can greatly shorten training and recognition time.
References

[1] Hariharan V and Srinivasan P. 2009 J. Mechan. Engineer 40119
[2] Tobon-MejiaDA, MedjaherK, Zerhouni N and TripotG. 2012 IEEE Transact. Reliab 61 491
[3] Li ZN, He YY, Chu FL, Han J and HaoW. 2006 J. Sound Vibra 29160
[4] Kang JS and Zhang XH. 2012 J. Inform. Intel. Interdispli. 156829
[5] Boutros T and Liang M. 2011 Mechan. Syst. Sign. Process 252102
[6] BechhoeferE, BernhardAPE and HeD. 2006 Proc. Am. Helicopt. Soc. 62nd Annual Forum Phoenix AZ
[7] Peng Y, Dong M. 2011 Mechan. Syst. Sign. Process 25237
[8] JiangR, KimMJ and MakisV. 2012 Quality Reliab. Engineer. Inter. 28783
[9] DongM and HeD. 2007 Mechan. Syst. Sign. Process 212248
[10] WidodoA and YangBS 2007 Mechan. Syst. Sign. Process 212560
[11] XianGM and ZengBQ 2009 Expert Syst. Appl. 361231
[12] WidodoA, KimEY, SonJD, YangBS, TanACC, GuDS, ChoiBK and MathewJ 2009 Expert Syst. Appl. 367252
[13] ZhangXL, ChenXF and HeZJ 2010 Proc. Institu. Mechan. Engineer. Part C - J. Mechan. Engineer. Science 224217
[14] SaimuruganM, RamachandranKI, SugunanarV and SakhivelINR 2011 Expert Syst. Appl. 383819
[15] ChenFF, Tang BP and ChenRX 2013 Meas. 46220
[16] CaesarendraW, WidodoA and YangBS 2010 Mechan. Syst. Sign. Process. 241161
[17] HeC, LiuC, LiY and YuanJ 2011 Proceed. Institu. Mechan. Engineer. Part C - J. Mechan. Engineer. Science 2251718
[18] WidodoA and YangBS 2011 Expert Syst. Appl. 382592
[19] Maio FD, TsuiKL and ZioE 2012 Mechan. Syst. Sign. Process 31405
[20] TranVT, Yang BS, Gu FS and BallA 2013 Mechan. Syst. Sign. Process 38601
[21] LeiYG, ZuoMJ 2009 Mechan. Syst. Sign. Process 231535
[22] Wang P, Li T Y, Gao X J and et al 2019 J. Vib. Engineer 32 548
[23] Zhu X Y, Wang Y J, Zhang Y Q and et al 2018 J. Vib. Test. Diagno. 38 1021
[24] Huang Z L and Zhang J Z 2016 Scientia Sinica
[25] Ren H, Qu J F, Chai Y and et al 2017 Contr. Decis. 32 345
[26] Wang Y C and Duan X S 2018 Tacti. Miss. Tech. 191 31
[27] Zhang M B, Wang L W. and Feng Y Q 2019 Systems Engineer. Electr. Tech. 41 178