Research on Optimization Algorithm of auto-encoding neural network applied to rolling bearing fault diagnosis

Shi Shi and Qingbin Tong*

School of electric engineering, Beijing JiaoTong University, Beijing, 100044, China
* Qingbin Tong’s e-mail: qbtong@bjtu.edu.cn

Abstract. Real time, fast and batch processing of vibration signals has become a future development trend in the field of fault diagnosis, but data dimensionality disasters may arise. In view of the long running time of deep learning in the case of large samples, the gradient descent method and its variant algorithms are introduced for the loss function optimization problem, and the approximate optimal solution is solved in an iterative manner. The gradient descent method is used to minimize the loss function, and the research is carried out on the basis of MATLAB program implementation. The gradient descent method and its variant algorithm are applied to the rolling bearing fault diagnosis model for analysis. By comparing the algorithm's convergence speed, loss value and accuracy of the rolling bearing fault diagnosis model, a relatively good optimization algorithm suitable for the rolling bearing fault diagnosis model is determined.

1. Introduction

With the increasing scale and automation of modern production, the application of rolling bearings in mechanical equipment can be seen everywhere. Due to the complex working environment, rolling bearings are one of the most vulnerable mechanical parts. Therefore, it is extremely important to the state monitoring and fault diagnosis of rolling bearings. The intelligent fault diagnosis technology of rolling bearing includes two aspects: fault feature extraction and fault pattern recognition. Among them, the feature extraction step generally resorts to signal processing methods, which has a strong dependence on professional knowledge and related technical personnel. In the era of electromechanical big data, data reflecting the status of equipment often has the characteristics of large total volume, multiple forms, and low value density. Relying on manual acquisition of information can no longer meet the needs of automatically mining information from electromechanical big data [1]. The rise of artificial intelligence technology has opened up new horizons for rolling bearing fault diagnosis technology. Deep learning is an emerging force in the field of artificial intelligence, which can automatically learn from the inherent characteristics of the data in order to obtain a good characteristic expression, which is a rolling bearing fault feature extraction and diagnosis provides a new way of thinking.

As a star in the era of big data, deep learning has attracted enough attention in academia and industry, and has achieved remarkable results in many traditional recognition tasks. Relying on multiple hidden layers, deep learning can well realize the representation of complex and high-dimensional functions, so it has strong characterization capabilities, and has many advantages that other algorithms can't match in automatic feature extraction [2]. Stacked auto-encoding network (SAE) is one of the most important models in deep learning, which was improved by the auto-encoder proposed by Rumelhart by Hinton in 2006 [3]. Since then, it has been widely used in various fields, among which many tasks of fault identification and classification have been completed in the field of mechanical fault diagnosis.
In the past, building a machine learning model required four stages: cleaning data, finding features, building models, and training and applying models. In this process, the most time-cost link is to find features. Deep learning has become popular because it can reduce the labour time cost of model building by automatically finding features. A good model training algorithm can usually reduce the computational complexity of model training, and can also improve the accuracy of the model, so that the model achieves the optimal result. The algorithm used in this process is usually called an optimization algorithm. As one of the most popular optimization algorithms, gradient descent algorithm is currently the most commonly used optimization algorithm in neural network problems. After a long period of development, gradient descent algorithms have proposed a variety of new algorithms in gradient calculations, step calculations, etc. These methods have no absolute advantages and disadvantages, and are often used as black boxes in practical applications. But different gradient descent algorithms will show different performance in different model training, and it is not the most advanced gradient descent algorithm that performs best in all models [4][5]. Therefore, the purpose of this paper is to determine a better optimization algorithm for rolling bearing fault diagnosis model by comparing the performance of several gradient descent algorithms in rolling bearing fault diagnosis model, including convergence speed, loss value and accuracy of fault diagnosis. The data used is the rolling bearing data set of Western Reserve University.

2. Basic model of fault diagnosis

2.1. Deep stack auto-encoding network theory

In 1986, Rumelhart proposed the concept of automatic encoder and applied it to high-dimensional complex data processing, which promoted the development of neural network [3].

2.1.1 Auto Encoder.

Automatic Encoder (AE) is a typical unsupervised learning symmetric neural network, which is essentially a three-layer neural network structure. The encoding process transforms the high-dimensional features of the input data into the low dimensional features of the hidden layer through the activation function, and the decoding process reconstructs the feature representation of the hidden layer through the activation function as the output target.

Assume that the input of the network is $X^d=\{x_1^d, x_2^d, \ldots, x_n^d\}$, $d=1,2,\ldots,m$. Where $m$ is the total number of samples and $n$ is the dimension of each sample. The process of mapping the original data $X^d$ to the hidden layer is as follows:

$$h^d = f(\omega_1 x^d + b_1)$$

(1)

where $\omega_1$ represents the weight matrix of coding layer, and $b_1$ is the threshold of hidden layer. The decoding process is the reconstruction process of the original data. The process of $h^d$ decoding the reconstructed data is as follows:

$$z^d = f(\omega_2 h^d + b_2)$$

(2)
where $\omega_2$ represents the weight matrix of decoding layer, $b_2$ is the threshold of hidden layer. $f$ is sigmoid function, the expression is $f(x) = 1/(1+\exp(-x))$. The activation function is used to add nonlinear factors, so that it has the ability of nonlinear mapping learning, and can learn more complex features. Figure 1 shows the structure of the automatic encoder.

### 2.1.2 The training process of AE.

The purpose of the AE is to train the feature parameters unsupervised so that the input value can be compressed with the least loss.

It is assumed that the samples in each dataset are independent of each other and that the training set and the test set are identically distributed. KL divergence, also known as relative entropy, is used to describe the difference between two probability distributions. Then the loss function is constructed as follows:

$$J(\omega_1, \omega_2, b_1, b_2) = KL[Z \parallel X] = -[Z \log X + (1-Z) \log(1-X)]$$

where $J$ the loss function, $X$ is the input sample and $Y$ is the target output.

The formula of gradient descent method is as follows:

$$\nabla \omega_i = \frac{\partial J}{\partial \omega_i}, \nabla b_i = \frac{\partial J}{\partial b_i}(i = 1, 2)$$

After the gradient of each coefficient is obtained, the parameters are updated along the opposite direction of the gradient:

$$\omega_i = \omega_i - \alpha \nabla \omega_i, b_i = b_i - \alpha \nabla b_i(i = 1, 2)$$

where $\alpha$ is the learning rate. After updating the parameters, the forward propagation process is continued until the error between the actual output value and the target output value is acceptable.

### 2.2 Deep stack auto-encoder neural network

![Figure 2. Fault diagnosis model block diagram of SAE neural network](image-url)
After the weights of the deep neural network are initialized by layer by layer unsupervised learning, a small number of label samples are used to fine tune the network. The greedy training method can effectively avoid the network falling into local optimum, so as to improve the network performance and make the training of the deep neural network possible [9][10].

Softmax regression model is an extension of logistic regression model, which can be used to solve the problem of multi classification. It is a supervised learning algorithm. Softmax regression is used to construct a classifier to classify the features learned by SAE.

The block diagram of SAE neural network fault diagnosis model is as shown in Figure 2.

In the pre-training stage, because of the huge training data, the traditional gradient descent optimization algorithm will make the training time become very long and consume a lot of memory, so it is necessary to explore a more suitable optimization algorithm to further improve the fault diagnosis performance of the model.

3.Optimization algorithm based on gradient descent

The gradient descent algorithm continuously updates the model parameters along the opposite direction of the gradient (first derivative) of the objective function \( J(\theta) \) (parameter \( \theta \in \mathbb{R} \)) to reach the minimum point (convergence) of the objective function [6]. The update step is \( \eta \).

There are three kinds of gradient descent algorithm frameworks. The difference between them is the number of samples used in each learning (updating model parameters).

3.1. Batch gradient descent method (BGD)

The batch gradient descent method is the basic gradient descent method. The basic idea is to use all the training set samples to update the model parameters each time [12]. This method can solve any extreme value problem with first derivative through iterative calculation. The direction determined by the full data set can better represent the sample population. Each update will proceed in the correct direction, and finally it can be guaranteed to converge to the extreme point (convex function converges to the global extreme point, non-convex function may converge at local extreme points). But there are also some shortcomings:

- Computational efficiency problem [7].
- Gradient calculation problem: due to many characteristics of neural network, the objective function is sometimes a nonconvex function, so it is difficult to avoid the parameter trapped near the minimum. This kind of situation often leads to the gradient close to zero, so it cannot converge to the global minimum (mainly in the process of neural network training).
- The value of learning rate \( \eta \). This is a common basic problem in the basic gradient descent algorithm. Too small learning rate leads to slow convergence of the objective function, while too large learning rate leads to violent oscillation of the objective function near its minimum. In addition, because all parameters use the same learning rate, the features with low frequency in the data often cannot achieve the optimal effect.
- Moreover, model parameters cannot be updated online.

3.2. Stochastic gradient descent (SGD)

Each time a sample is randomly selected from the training set to learn. The advantage of this method is that each learning is very fast and can be updated online. But there are also some shortcomings [8]:

- For non-convex error function, it is easy to fall into local optimum.
- The most important point of SGD is that it is difficult to choose the right learning rate. Same as BGD.
- Because SGD uses random and small batch data to calculate the gradient, which approximates the real gradient of the whole dataset to reduce the gradient noise and variance caused by calculation density.
- Since a single sample cannot represent the trend of all samples, it may converge to the local optimum. It is not easy to implement in parallel.
3.3. Small batch gradient descent method (MBGD)

Small batch gradient descent method is a compromise. It combines batch gradient descent method and random gradient descent method to achieve a balance between each update speed and update times. Each update randomly selects a fixed number of samples from the training set for learning \[9\].

Compared with the stochastic gradient descent method, the small batch gradient descent method reduces the convergence volatility, that is, it reduces the variance of parameter update, making the update more stable. Compared with the total gradient descent method, it improves the speed of each learning. And it doesn't need to worry about memory bottleneck, so it can use matrix operation for efficient calculation.

3.4. Momentum

The main idea of momentum is to accelerate the learning process by introducing a new variable \(V\) to accumulate the exponentially decaying moving average (the exponentially weighted average). Momentum is introduced to speed up the learning process, especially for high curvature, small but consistent gradient or high noise gradient. The difference between SGD and momentum is that momentum adds an update quantity (i.e. momentum term) to the parameter update term:

\[
d_i = \beta d_{i-1} + g(\theta_{i-1})
\]

\[
\theta_i = \theta_{i-1} - \alpha d_i
\]

where \(d_i\) and \(d_{i-1}\) are the update directions of this time and the last time respectively, \(g(\theta)\) representing the gradient of the objective function at \(\theta\), and the super parameter \(\beta\) is the attenuation weight of the last update direction, so it is generally between 0 and 1, and \(\alpha\) is the learning rate. Generally speaking, in an iteration, the total parameter update quantity includes two parts, the first is obtained from the last update quantity \(\alpha \beta d_{i-1}\), and the second is obtained from this gradient \(\alpha g(\theta_{i-1})\). The larger the value of momentum super parameter \(\beta\), the greater the influence of the previous gradient on the current direction.

4. Comparison of algorithms applied in Fault Diagnosis Models

The gradient descent optimization algorithm mentioned in the previous section shows good results in different angles, and can improve some defects. In order to verify their optimization effect in the SAE neural network fault diagnosis model proposed in the previous paper, the rolling bearing fault data collected by the laboratory of Case Western Reserve University (CWRU) \[11\] in the United States is used as the research object for experimental verification. The rolling bearing data set of Case Western Reserve University has high quality and obvious fault characteristics, so it is a common rolling bearing fault diagnosis standard data set in academic circles.

Table 1. Parameter table in the model

| hidden layers | Hidden nodes 1 | Hidden layer nodes2 | Activation function | classifier | Learning rate | momentum factor | Batch size | Iterations |
|---------------|----------------|---------------------|---------------------|-----------|---------------|----------------|-----------|------------|
| 2             | 200            | 100                 | Sigmoid             | softmax   | 0.06          | 0.69           | 256       | 400        |

The object of diagnosis is deep groove ball bearing skf6205. Different faults are simulated by EDM on the outer race, inner race and rolling element. The fault diameters are 0.001, 0.014 and 0.021 inch respectively. In addition, the undamaged bearing is added. In the experiment, a total of 10 kinds of bearings with different states need to be identified. The data set is the vibration of the bearing mount collected by the acceleration sensor, with the acquisition frequency of 12 kHz and the sampling length of 1200 points. Each state has 1000 samples, a total of 10000. Each state selects 500 samples for training, and the remaining 500 samples for testing.

SGD, BGD, MBGD and momentum method are used to train the single automatic encoder in the fault diagnosis model mentioned above, and the training effect is shown in the figure 3 and figure 4.
As can be seen from figures that different optimization algorithms are used in the training of automatic encoder, and the training loss and time are completely different. In the SAE model with two hidden layers, the algorithm is compared. In the first hidden layer training, the SGD algorithm takes the shortest time to train, but there is a large oscillation, and the BGD optimization algorithm has the best stability, but the training time is the longest. MBGD is more stable and faster than SGD. And momentum optimization algorithm has better stability and faster training speed, and can get lower loss value, which is undoubtedly a more superior optimization algorithm.

5. Conclusion and Prospect
From the above analysis, a good optimization algorithm can greatly improve the training efficiency and accuracy of the model. Momentum is proved to be effective in the pre training of deep auto encoder neural network model for rolling bearing fault recognition. In the future, we can further optimize the learning rate and the mixing effect of each optimization algorithm.

Acknowledgments
The authors wish to extend their sincere thanks for the support from the Beijing Municipal Science and Technology Commission of China. This work was supported by the Natural Science Foundation of Beijing municipality, China (Grant No. 3212032).

References
[1] LEI, Y G. JIA, F. KONG, D T. et al. (2018) Opportunities and challenges of machinery intelligent fault diagnosis in big data era. Chinese Journal of Mechanical Engineering, 54(5):94-104.
[2] ZHANG, Q C. YANG, L T. CHEN, Z K. (2016) Deep computation model for unsupervised feature learning on big data. IEEE Transactions on Services Computing, 9(1): 161-171.
[3] RUMELHART, D E. HINTON, G E. WILLIAMS, R J. (1988) Learning representations by backpropagating errors. Nature, 323(6088): 399-421.

[4] ZHOU, Z H. (2016) Machine Learning. TSINGHUA UNIVERSITY PRESS, Beijing

[5] HINTON, G E. OSINDERO, S. THE, Y W. (2006) A fast learning algorithm for deep belief nets. Neural Computation, 18(7):1527-1554.

[6] LAROCHELLE, H. BENGIO, Y. LOURADOUR, J. et al. (2009) Exploring strategies for training deep neural networks. Journal of Machine Learning Research, 10(12):1-40.

[7] Sebastian, R. (2017) An overview of gradient descent optimization algorithms. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences.

[8] Ashia, C. Wilson, Rebecca, R. Mitchell, S. Nathan, S. Benjamin, R. (2017) The Marginal Value of Adaptive Gradient Methods in Machine Learning. 31st Conference on Neural Information Processing Systems (NIPS).

[9] ERHAN,D. BENGIO, Y. COURVILLE, A. et al. (2010) Why does unsupervised pre-training help deep learning?. Journal of Machine Learning Research, 11(2):625-660.

[10] RASTOGI, N K. RASHMI, K R. (1999) Optimisation of enzymatic liquefaction of mango pulp by response surface methodology. European Food Research &. Technology, 209(1): 57-62.

[11] Case Western Reserve University Bearing Data Center. (2018) Bearing data file [EB/OL]. Http://csegroups.case.edu/bearingdatacenter/home.

[12] Nesterov, Y. (2013) Gradient methods for minimizing composite functions. Mathematical Programming, 140(1):125-161.