Application of BP Neural Network in Network Fault Diagnosis

DONGYANG LI¹ and XIAORONG CHENG¹

¹School of Control and Computer Engineering, North China Electric Power University, Baoding 071000, China

Abstract. There are various deficiencies in network fault diagnosis methods but the need for fault diagnosis is further increasing. In order to adapt to the current development needs, the paper applies the artificial neural network concept to network fault diagnosis. Aiming at the slow convergence speed of traditional neural networks and the tendency to fall into the local optimal solution, first we attempt to add the momentum factor and then adopt rough set preprocessing data. The simulation results show that the improved algorithm has a certain advantages for the original BP algorithm, and it has a certain value for future fault diagnosis research.

1. Introduction

With the rapid development of network technology and the Network expansion, the degree of social dependence on the network has reached an unprecedented level, which also puts higher requirements on network management. Under the 5G wave, network management will become an important issue. As the number of users increases sharply, the probability of network failure increases, but the traditional method of manually determining faults is not applicable to the current situation. A fast, efficient and accurate fault diagnosis method has become a current need. Artificial neural networks have been widely used in the field of fault diagnosis in recent years because of the strong independent learning ability and fault tolerance. And BP neural network is one of the outstanding representatives. It is also the most successful artificial neural network learning algorithm to date. It can adapt well to the current development and gets a lot of attention in fault diagnosis, opening up a new path for intelligent diagnosis technology.

The neural network implements the mapping function from input to output. It is proved by experiments that the BP neural network can realize the ability of any complex nonlinear mapping, but the BP algorithm also has some limitations. The rationality of the parameters is difficult to control, the learning rate is slow, and it is easy to fall into the local optimum situation. For these problems, this paper proposes a method that can properly optimize these problems. The improved algorithm can have a better learning rate, improve the efficiency of network management personnel, and has certain reference significance for future network fault diagnosis.

2. Improvement of BP neural network

In practical applications, about 80% of neural network models take the form of BP neural network or BP neural network. The BP neural network is divided into three layers, which are an input layer, an implicit layer, and an output layer, wherein the hidden layer may be several layers. Studies have shown that a three-layer network structure can be applied to any complex mapping situation.

The basic idea of the BP algorithm can be summarized into two phases. The forward flow of the first stage signal propagates to the output layer, and the second stage results in the back propagation of the
error. When the signal is forward propagating, the input value is input by the input layer, reaches the hidden layer through the algorithm, and is processed by the hidden layer to flow to the output layer. When the output layer is reached, the algorithm calculates the difference between the actual output value and the preset desired output, called the error. If the actual error is greater than the expected error, the backpropagation phase is performed, otherwise the training is completed. The back propagation of the error is the process of changing the weight. The output layer is outputted in reverse to the hidden layer and then to the output layer, and is transmitted back to the layer. When it reaches each layer, the error is distributed to all the neural units in each layer. In this way, the weight of each unit is modified. This process is cyclical until the end of the pre-set condition is reached, which is also known as the training process of the neural network.

2.1 Rough set decision table theory

The state information extracted from the device must be rich and disorganized, since the redundancy and uncertainty of information if the data is directly input into the system for training, It may result in slow convergence and low accuracy of fault diagnosis. Therefore, we try to use the rough set decision table idea to preprocess the extracted data, eliminate useless data and redundant data, and improve the usability of the data. The decision table is a two-dimensional table, with each column representing a state attribute, with each row representing each piece of data extracted. Objects in the domain are divided into different decision classes according to different condition attributes. For decision attributes, not every decision attribute is necessary. We try to find the simplest attribute set that does not affect the classification, which is the rough set preprocessing idea. U is the domain, a, b, and c are conditional attributes, and D is a decision attribute. As shown in Table 1.

| U | a | b | c | D |
|---|---|---|---|---|
| 1 | 1 | 0 | 2 | 1 |
| 2 | 2 | 1 | 0 | 2 |
| 3 | 2 | 1 | 2 | 3 |
| 4 | 1 | 2 | 2 | 1 |
| 5 | 1 | 2 | 0 | 3 |

There may be several kinds of reductions in a decision table. The intersection of these reductions constitutes the core of the decision table. The attributes of the core are important attributes that affect the classification. As shown in Table 2 and Table 3, after different reductions, they have the same classification ability as the original Table 1, Obviously [c] is the core. When core is found, other attributes are added outward to form the simplest decision table. Of course, in the specific problem, we do not need to find all the reductions. heuristic search can be used to find the reduction of the conditional attributes.

### Table 1. decision table

| U | a | b | c | D |
|---|---|---|---|---|
| 1 | 1 | 0 | 2 | 1 |
| 2 | 2 | 1 | 0 | 2 |
| 3 | 2 | 1 | 2 | 3 |
| 4 | 1 | 2 | 2 | 1 |
| 5 | 1 | 2 | 0 | 3 |

### Table 2. Reduction[a,c]

| U | a | b | c | D |
|---|---|---|---|---|
| 1 | 1 | 2 | 1 | 1 |
| 2 | 1 | 0 | 2 | 2 |
| 3 | 2 | 2 | 3 | 3 |
| 5 | 1 | 0 | 3 | 3 |

### Table 3. Reduction[b,c]

| U | b | c | D |
|---|---|---|---|
| 1 | 0 | * | 1 |
| 2 | 1 | 0 | 2 |
| 3 | 1 | 2 | 3 |
| 4 | 2 | 2 | 1 |
| 5 | 2 | 0 | 3 |
2.2 Momentum node and inertia factor

The essence of the BP algorithm is to find the minimum value of the error function. This algorithm uses the steepest descent method in nonlinear programming to modify the weight coefficient according to the negative gradient direction of the error function. Studies have shown that the above BP algorithm has some drawbacks. 1) The learning rate is slow. Since each time the weight is adjusted, it needs to be based on a network error. If the near-minimum value is relatively flat and the partial derivative value is relatively small, the weight adjustment is small, so that it needs to "stay" for a long time and converge. Slower. In the "steep" place, the number of partial derivatives is large, and the weight adjustment range is large, there will be an "overshoot" phenomenon, the weight adjustment path is jagged, and the minimum value cannot be determined, resulting in a slow convergence rate. 2) When the training data differs little from each other or there is a certain law, it will also cause BP to converge slowly or even not, and the diagnostic accuracy is not high. 3) For a function with multiple minimum values, the BP algorithm may fall into a local optimal situation, the search time consumed increases, and the global minimum cannot be determined, resulting in a slow convergence rate.

In response to this situation, we try to improve the algorithm in two aspects. Firstly, we try to add a momentum node to guide the BP network to move quickly in the direction of minimum error and to open the distance between fault and non-fault. It shown in Figure 1.

The \( X_{n+1} \) input node is the "momentum" node, and \( X_{n+1} = X_1 + X_2 + \ldots + X_n \).

In the weight update phase of the standard BP algorithm, we attempt add the inertia factor. The purpose is to make the weight correction have a certain inertia, reduce the sensitivity of the algorithm to the local minimum, and reduce the possibility of falling into the local minimum.

The standard BP neural network weight adjustment formula is:

\[
W_{ij}^{(k-1)}(t+1) = W_{ij}^{(k-1)}(t) + \eta \sum_{h=1}^{l} \delta_{hj}^{(k)} y_{hj}^{(k-1)}
\]

The weight adjustment formula after adding the inertia factor is:

\[
W_{ij}^{(k-1)}(t+1) = W_{ij}^{(k-1)}(t) + \eta \sum_{h=1}^{l} \delta_{hj}^{(k)} y_{hj}^{(k-1)} + \alpha \Delta W_{ij}^{(k-1)}(t)
\]

\( \alpha \) is the inertia factor, \( 0 < \alpha < 1 \).

3. Simulation experiment verification

We extracted the 2026 interface group fault for testing then remove obviously useless and redundant data, establish the initial decision table, A representative from the MIB data form condition attribute, D said the fault decision attribute. Because the BP neural network accept discrete data, We discretize the data before use the data used in network training. The decision table is shown in table 4.
The rough set decision table is used to preprocess the algorithm, and MN is momentum node. It is shown in Table 5.

Table 4. Initial decision table

| number | Condition attribute (fault symptom) | Decision attribute (fault type) |
|--------|-----------------------------------|-----------------------------|
| 1      | A1 0 0 0 0 0 0 0.5 1 1 0 0 1 1 0 0 0 |
| 2      | 1 0 0 0 0 0 1 0.5 0.5 0 0 0 1 1 0 0 0 |
| 3      | 1 0 1 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 |
| 4      | 1 0 1 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 |
| 5      | 1 1 1 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 |
| 6      | 0 0 1 1 1 0 0 0 1 1 1 1 1 1 0 0 0 1 |
| 7      | 0 1 1 0 1 0 0 0 0 1 0 1 1 1 0 0 0 1 |
| 8      | 1 0 1 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 |
| 9      | 1 1 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 |
| 10     | 1 0 1 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 |
| 11     | 0 1 0 0 0 0 0 0.5 0 0 0 1 0 1 0 0 0 0 |
| 12     | 1 1 0 0 1 1 0 1 0 0 0 1 1 1 0 0 0 0 0 |
| 13     | 1 0 1 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 |
| 14     | 0 1 1 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 |
| 15     | 1 1 1 0 0 0 0 0 0 0 0 1 1 0 0 1 0 |

Table 5. Simplified sample data

| number | Condition attribute | Decision attribute |
|--------|---------------------|--------------------|
| 1      | A1 0 0 1 2 0 0 0 0 0 |
| 3      | 1 0 1 0 2 0 1 0 0 0 |
| 4      | 1 0 1 1 3 1 0 0 0 0 |
| 5      | 1 1 1 1 4 0 0 1 0 0 |
| 6      | 0 0 1 2 0 0 0 0 1 0 |
| 7      | 0 1 1 2 0 0 0 0 1 0 |
| 10     | 1 0 1 0 2 0 1 0 0 0 |

Table 6. main failure table

| main failure table (1 Means there is a fault, 0 means there is no fault) |
|---------------------------------------------------------------|
| D1 Routing card fault                                         |
| D2 Wired discontinuous                                        |
| D3 Administrative closure                                     |
| D4 Link protocol error                                        |

Table 7. MIB variable scale

| MIB variable scale |
|--------------------|
| A1 Protocol type (1 for PPP type, 0 for HDLC type) |
| A2 Interface management status (1 for off, 0 for on) |
| A3 Current state of the interface (1 for down, 0 for up) |
| A12 Serial port detection (1 for up, 0 for down) |

Table 5 shows that sample data are greatly reduced after preprocessing, but the original classification effect is maintained and A1, A2, A3, A12 are the key attributes that determine attribute classification. The simulation experiment was carried out through the MATLAB neural network toolbox. The experimental results are shown in Table 8 and Figure 2.

Table 8. performance comparison

| type               | time(s) | Learning step |
|--------------------|---------|---------------|
| BP algorithm       | 2.8     | 29            |
| Momentum BP algorithm | 1.3   | 8             |
The experimental data shows that the decision table momentum BP algorithm has better learning efficiency than the standard BP algorithm, and the convergence is obviously accelerated to reach the set error.

Next, we use the five data of Initial decision table to test the experimental accuracy. The experimental results are shown in Table 9. The experimental results show that the accuracy of decision table momentum BP algorithm is better than the standard BP algorithm. The fault Diagnosis ability is better than the standard algorithm.

![Figure 2](image-url)  
Figure 2. Comparison diagram of neural network

| num | Actual fault data | Momentum BP algorithm output of decision table | Standard BP algorithm output |
|-----|-------------------|-----------------------------------------------|------------------------------|
|     | D1    | D2    | D3    | D4    | D1    | D2    | D3    | D4    | D1    | D2    | D3    | D4    |
| 11  | 1     | 0     | 0     | 0     | 0.94298 | 1.75E-18 | 5.19E-15 | 0.990243 | T     | 0.94938 | 2.29E-05 | 3.12E-04 | 2.46E-07 | T     |
| 12  | 0     | 0     | 0     | 0     | 0.00031298 | 9.19E-14 | 0.990396 | 5.38E-14 | T     | 0.988E-05 | 2.66E-03 | 0.43758 | 2.38E-03 | T     |
| 13  | 0     | 1     | 0     | 0     | 0.00069953 | 1     | 0.0002753 | 1.63E-13 | T     | 0.99073598 | 9.99E-04 | 2.61E-13 | 7.82E-09 | T     |
| 14  | 0     | 0     | 0     | 1     | 0.0372 | 8.71E-12 | 1.99E-08 | 0.99921 | T     | 0.99043699 | 3.50E-06 | 4.40E-05 | 0.9968 | T     |
| 15  | 0     | 0     | 1     | 0     | 0.0020878 | 1.58E-04 | 0.99949 | 3.53E-12 | T     | 2.96E-06 | 2.62E-01 | 0.60243 | 4.13E-07 | F     |

4. Conclusion

This paper analyzes the performance of basic BP neural network algorithm in fault diagnosis. In view of the slow learning rate and the inability to find the global optimal value in the learning process, we try to add the momentum factor and Rough set preprocessing. The simulation experiment verifies the effectiveness of the improved algorithm. The results show that the training rate of the neural Network and the accuracy of classification are improved. The improved algorithm appropriately reduces the diagnostic failure time in network fault diagnosis, improves the diagnostic accuracy rate, and provides a reference method for intelligent fault diagnosis.

Reference

[1] Zhixin Shang, Yu Zhou, Qingwei Ye, Xiaodong Wang. (2012.) Fault Diagnosis of Computer Network Based on Rough Sets and BP Neural Network[P].

[2] Yuhao Wang, Wang Haibo. (2010) Application of improved BP neural network in local area network fault diagnosis [J]. Computer simulation, 27(04):96-98+205.

[3] Tang Hongfa, (2005) Network fault diagnosis system based on BP neural network [D]. Nanjing university of science and technology.