BP Neural Network-Based Web Service Selection Algorithm in the Smart Distribution Grid

Lanlan Rui , Yinglin Xiong, Ke Xiao，Xuesong Qiu
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, China
E-mail: {llrui,ylxiong,xiaoke,xsqiu}@bupt.edu.cn

Abstract—A good web selection algorithm can provide the most suitable service for users. However, known for its slow convergence rate and proneness of oscillation in its learning process, the traditional error back propagation neural network algorithm cannot be applied in the service selection scenarios of actual smart distribution grid. In order to meet the requirements of telecommunication technology for smart distribution grid and improve the quality of telecommunication service, this paper proposes an improved error back propagation algorithm, in which the learning factor can be self-adjusted with every iteration. The simulation results show an optimization of the training speed and an oscillation reduction in the learning process with the new algorithm, thus obvious optimizing the web services selection in smart distribution grid.

Key words—error back propagation algorithm; web services selection; gentle factor; UPS; smart distribution grid

I. INTRODUCTION

In the web service environment, the Smart Distribution Grid (SDG) is supposed to integrate all the information for providing the best web service. Therefore, how to select the most appropriate one from similar service providers has become an urgent key problem of SDG in the web services environment.

Different types of service selection algorithm have been proposed, such as Genetic Algorithm (GA) [1], and Simulated Annealing Algorithm (SAA) [2]. Genetic Algorithm (GA), with a fast convergence rate and intrinsic parallelism, is free of the constraints of the restrictive assumption of search space. However, practice shows that GA is still not satisfactory because of its premature convergence, low post-search efficiency and poor local optimization ability.

Though good at local optimization, Simulated Annealing Algorithm (SAA) commands little of the whole search space, and so it is hard for it to get to the most promising searching area, resulting in low operation efficiency.

Considering the adaptability of the algorithm for the changing web services evaluation index, this paper uses error Back Propagation (BP) Algorithm of artificial neural network as a web service selection algorithm [3]. In the practical application, the traditional BP neural network usually suffers from low learning convergence rate and oscillation in the training process. In order to improve the performance, this paper proposes a new algorithm with a new factor added to the traditional BP neural network. Added to the original learning factor and momentum factor of the traditional BP network is an innovation factor, which can have a stable effect in the training, and thus named the smooth factor. A comparative evaluation of their performance after applying both algorithms in simulators reveals a better convergence rate, training time and frequency stability of selected UPS power of the improved BP algorithm than those of the traditional one. So it is more suitable for SDG.

The rest of this paper is organized as follows. In section II, a web service optional model is set up. And a descriptive approach of Service is defined. A method of calculation to QoS is proposed. Section III describes an innovative BP algorithm and defines an adaptive learning factor. The steps of BP arithmetic operations are described in details in this section. Section IV shows the experiment and the results analysis. Conclusion and summary of future researches is concluded in Section V.

II. WEB SERVICE SELECTION MODEL

A. Research Background

SDG is an important part of the smart grid. As SDG is supposed to provide uninterrupted service to uses based on their different requirements on power supply, security and reliability, high reliability and automation of relative important infrastructure equipment like power supply and data center are a must.

The power equipment for the controlling center is obvious important for all types of facilities working safely and continually in large data centers of the SDG. The increasing of management points [4] makes it hard for the administrators to monitor and control every device all the time. Therefore, when a power failure occurs, the administrator might not be able to shut down the computer(s) and the UPS before the battery runs out. So computers and their peripherals should be endowed with the ability to independently cope with certain foreseeable problems, automatically making adjustment according to the real situation.

This paper is to propose a method for an automated UPS power supply selection for the network equipment of the computer rooms in SDG, in which suitable UPS services meeting the requirements of the target scene are singled out to be assigned to each device according to the frequency stability of UPS power supplies.

B. Algorithm targets and service selection process

This work was partially supported by National Natural Science Foundation of China (61302078, 61372108), Funds for Creative Research Groups of China (61121061), Beijing Higher Education Young Elite Teacher Project (YETP0476).

The whole goal of web service selection process is to choose a similar service set, and choose a most appropriate service to the user from the similar services set. To select a service that can meet the needs of computer equipments, we need to achieve the following goals:

Goal 1: Make the response time as short as possible, which means the algorithm convergence speed should be as fast as possible.

Goal 2: Select UPS power that has the more reliable QoS.

Our web selection process can be described as: Compute they stability value of UPS power supplies of each service in similar services set, and put them into the service selection controller. The service selection controller choose the most suitable stability value of UPS power supplies to meet the simulation environment's need, and then assign the corresponding UPS service of the value to the network equipment.

Suppose that the number of web services providing the same function is m, each service has several QoS attributes corresponding to the attributes of UPS power. The QoS attributes matrix Q is presented as figure 1.

$$Q = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1n} \\ q_{21} & q_{22} & \cdots & q_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ q_{m1} & q_{m2} & \cdots & q_{mn} \end{bmatrix}$$

Figure 1 QoS attributes matrix

Basically, we suppose a weight vector ω to measure and assess every web service in Q matrix. The assessment formula is presented as (2).

$$score_i = \sum_{j=1}^{n} \omega_j q_{ij} \ , \ \sum_{j=1}^{n} \omega_j = 1 \qquad (2)$$

The weight vector illustrates the importance of each QoS attribute in the final evaluation results and formula (3) is the output value of QoS. Function f is neuron activation function in the BP neural network. Using function f can compute the QoS evaluation value of all the candidate web services, in which the web service with the most appropriate evaluation value is considered to be the global optimal service.

$$y = f\left(\sum_{i=1}^{m} \omega_i score_i\right) \qquad (3)$$

From (2) we can see that the evaluation value is not only influenced by the QoS attributes value, but also affected by the weights assigned to them. When web services are changed, some QoS attributes value will be changed and the service registry would update the changed QoS attributes. However ω may also be changed due to some mechanism problems. Therefore, then Q matrix and ω must be considered at the same time to establish an automatic evaluation mechanism of the web service based on (2).

III. ALGORITHM IMPROVEMENTS

A. Basic BP Algorithm

Typical BP neural network is a kind of forward network that has a structure of three or more than three layers, without feedback and no interconnection fabric within a layer. The training method of the BP neural network is to add a sample to the input layer according to the forward propagation rules:

$$x_i^k = f\left(u_i^k\right) \qquad (4)$$

u_i^k is the total input of the i-th neuron in the k-th layer, and x_i^k is the output, excitation function of neurons are sigmoid function:

$$f(x) = \frac{1}{1+e^{-x}} \qquad (5)$$

Finally we can get the output x_i^m through the calculation of step by step, which means the output layer is the m-th layer, and i-th node is chosen. There is only one node in this paper. Putting x_i^m and the desired output Y_i into type (6) can calculate the error E.

$$E = \frac{1}{2}\sum_i \left(x_i^m - Y_i\right)^2 \qquad (6)$$

Then we use (7) to calculate the gradient of error function, and pass the gradient from the output back to the hidden layers and then back to the input layer, adjust the connection weights reversely.

$$\Delta\omega_{ij} = -\alpha \frac{\partial E}{\partial \omega_{ij}} \qquad (7)$$

(7) is based on the steepest descent method in nonlinear programming, it can modify the connection weights between nodes according to the error function of negative gradient direction, in which α is the learning rate.

The standard gradient descent algorithm is a greedy algorithm, although it can solve the optimization problem effectively, but may be trapped in local minima. In order to avoid this kind of defect, D.E.Rumelhart [5] proposes a method, and adds a momentum item in the rule changes considering the previous iteration of the gradient contribution. The formula of the iterative weights is as follows:

$$\Delta\omega_{ij}(n) = -\alpha \frac{\partial E}{\partial \omega_{ij}} + \beta \Delta\omega_{ij}(n-1) \qquad (8)$$

The n is the number of iterations, β is called momentum coefficient, $0<\beta<1$.

B. Improved BP algorithm

In the traditional BP algorithm, the smaller the learning factor α is, the smaller the weight of network highlight variation from one iteration to the next iteration is, the error surface is smooth, but it will make the neural network training speed slower. On the other hand, if increase the value of α to accelerate the learning speed, neural network weight may change unstable, which is prone to oscillation. In order to reduce vibration and improve the training speed of network, this paper proposes a novel BP neural network algorithm. The algorithm's performance is that the difference between the output value and the target value in each iteration proportionally added to the connection weights of adjustment formula.

$$\Delta\omega_{ij}(n) = -\alpha\frac{\partial E}{\partial\omega_{ij}} + \beta\Delta\omega_{ij}(n-1) + \eta\left(x_i^m - Y_i\right) \qquad (9)$$

The η in (9) is the proportion, referred as the smooth coefficient. The addition considered the difference between the actual output value and the standard output value. In order to approximate the standard output as soon as possible, it shorten the gap for η times in this direction. Compared with the traditional BP algorithm, which has only learning factor and momentum factor, the new algorithm is faster to approximate the standard output value and can overcome the defect of oscillation in the process of training.

Figure 2 simply described the working process of the adaptive learning rate algorithm.

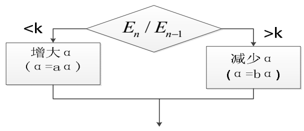

Figure 2 flow chart of adaptive learning rate algorithm

According to figure 2, the automatically adjustment of learning rate in adaptive learning rate algorithm is described as follows: observe the ratio between two consecutive iteration error function value to make sure if it has reduced the error function value. If the error function value decreased, it indicates that the learning rate value α is small. Else, α is large .We need to keep adjusting the value of α, until the neural network is converged.

Propose such a method to adjust the learning rate: if the ratio between new error and the previous old error is larger than a fixed value (defined as k in this paper, set k = 1.05[6]), learning rate cut work is needed (in this paper, learning rate of the actual value multiplied by a factor b, set b = 0.6 [6]). If the ratio between error of the new and the previous old error is smaller than a fixed value of k, the learning rate would need to be raised(in this paper, learning rate of the actual value multiplied by a factor a, set a = 1.06[6]).

Based on the experience and the actual needs of this paper, the value of variables is set as follows: a = 1.06, b = 0.6, k = 1.05. The initial value of learning rate in the neural network algorithm is usually a float number selected from (0,1).

C. *computing steps of the improved BP algorithm*

The improved BP neural network algorithm's learning step is shown in figure3:

The learning steps of algorithm in the flow chart above are described as follows:

1) Neural network should be initialized at first and set node number for each layer. The node number of input layer is generally decided by the web service attribute. And node number of output layer is usually set to 1. For the three-layer feed-forward neural network with m node number in input layer, the node number of hidden layer is set to 2 m + 1.

2) Set the sample input and the expected sample output. Randomly select a study factor α, momentum factor β, gentle factor η, a matrix with smaller connection weight and the error precision allowed by the neural network learning process. Then the neural network is ready to be trained.

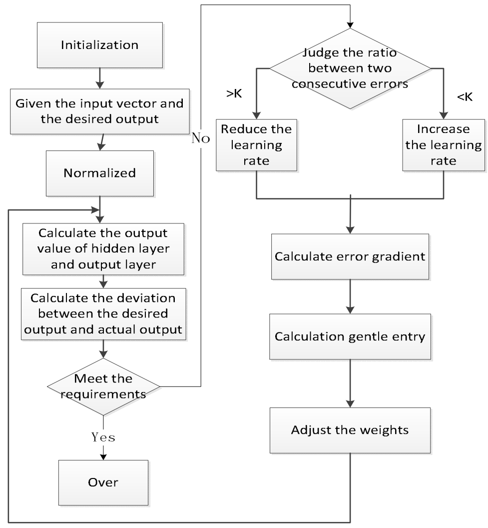

Figure 3 new BP neural network algorithm's flow chart

3) The input sample data need to be normalized pretreatment. Because the unit of the input data is different, the range of some data may be particularly big. This causes the neural network convergence is slow and long training time. The larger the range data has, the more important the role plays in pattern classification. As the value range of the output layer activation function is limited, target data of neural network training should be mapped to the value range of activation function .In this paper, the neural network uses sigmoid as the activation function. And sigmoid function has a domain limit in (0,1) region. Because the sigmoid activation function outside the (0,1) region, the output of the training data should be normalized to [0,1] interval. Simple and rapid normalization algorithm is linear transformation algorithm. The general form of linear transformation algorithm[7] is as follows:

$$x_i' = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \qquad (10)$$

x_{max}, x_{min} are the maximum and the minimum values of each group of input attribute. x_i, x_i' are respectively the data not normalized and normalized.

4) Respectively calculate the output of each unit in hidden layer and output layer. The transfer function from the input layer and hidden layer, the transfer function from hidden layer to output layer are both sigmoid functions. Calculate the output of each node of each layer according to the forward propagation rules. Finally, the output will be a practical level at the only output node.

5) Calculated the error from target value to the actual output according to formula 6), record it, and then transfer to the next step.

6) Compare the value of step 5) with the error precision set in step 2). If the error value is less than the

error precision, the purpose of the training has been achieved, and then we can turn to step 7). If not, turn to step 8).

7) When the BP neural network has been trained successfully, the error reaching the limit of the accuracy range, stop training.

8) Get the error value of the adjacent two iterations divided, if the ratio is less than k, and make adjustments according to the type 10). If the ratio is more than k, make adjustments according to the type 11). After the new learning factor values obtained, proceed to step 9)

9) Turn into the error back propagation stage from the step and we can regulate the connection weights in the process of back propagation. Get the error gradient values according to the error function, and then multiplied by the value of the new learning factors, turning into step 10).

10) The result of subtracting the theory standard output value from actual output value in the current iteration should be multiplied by smooth factor. Then turns to the step 11).

11) Two values from step 9) and 10) are need to add up, and then plus the result with momentum item. The final result is used to adjust the connection weight between output layer and hidden layer, hidden layer and input. Turn to step 4) after complete the adjustment.

IV. EXPERIMENT AND RESULTS ANALYSIS

Compare the traditional BP neural network algorithm and the improved BP neural network algorithm by the training time and efficiency. Table 1 shows the performance value of the two algorithms after training. It record the number of iterations and iteration time of the two BP algorithms by the statistics of the training data in ten times to each algorithm.

Table 1 Performance value of the two algorithms

	Least	Most	Average	Total iteration time
Traditional BP algorithm	1062	1353	1151	673ms
Improved BP algorithm	478	786	536	345ms

We can see from the table that the novel algorithm has the less number of training which determine the whole iteration time. The improved algorithm takes only half the time of the traditional algorithm, which also shows that its learning performance is far more superior than the traditional algorithm.

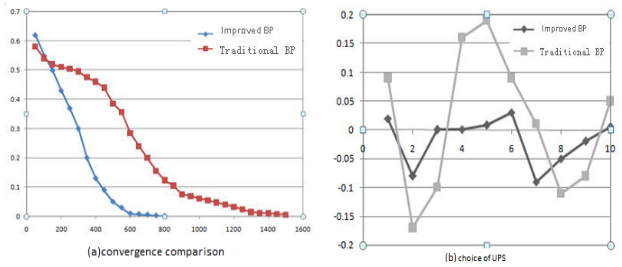

Figure 5 The algorithm comparison chart

The convergence rate of the two algorithms is shown in figure 5(a). And the horizontal axis is the error values in each iteration. With the increase of the numbers of training, the error of the traditional BP algorithm falling slowly, and the error of the improved BP algorithm has been maintaining rapid decline rate and convergence speed are optimized obviously.

Figure 5(b) shows the improved BP algorithm and the traditional BP algorithm in the choice of UPS. Y-axis is the frequency fluctuation of UPS (%). X-axis is the number of algorithm running time.

As shown in figure 5(b), compare the data of each BP algorithm running ten times, the chosen UPS power frequency of traditional BP algorithm range from - 0.17% to 0.19%, and the chosen UPS power frequency of improved BP algorithm range from - 0.09% to 0.03%. It's obviously seen that the improved BP algorithm has a smaller and more stable frequency range, which is more advantageous to equipment.

V. CONCLUSION

We apply the improved BP neural network algorithm to the simulation of computer room power supply system, and use many sets of UPS power for the simulation experiment. In the experiments, improved and traditional BP algorithms are compared. The contrastive analysis of the evaluation results verifies the improved BP algorithm put forward by us has better convergence speed, less training time, and also has obvious effects of optimization and better stability on the selection of UPS power supply.

Based on this work, we will apply the improved algorithm to the more complex web services environment in the further, where has a large number of web service providers with the challenge of how to choose the fast and robust web service.

REFERENCES

[1]. Chengwen Zhang, Sen Su,et al. Genetic Algorithm on Web Service Selection Supporting QoS [J].Chinese Journal of Computers,2006,29(7):20-25.

[2]. Lijuan Long. Genetic Simulated Annealing Algorithm on Semantic Web Service Selection[D].Beijing: Beijing University of Posts and Telecommunications,2009.

[3]. Haibin Cai, Xiaohui Hu, Qingchong Lu, et al. A novel intelligent service selection algorithm and application for ubiquitous web services environment[J].Expert Systems with Applications, 2009, 36(2):2200-2212.

[4]. Maolin Tang.A hybrid genetic algorithm for the optimal constrained web service selection problem in web service compositon[J].Evolutionary Computation(CEC),2010,30(5):1-8.

[5]. Rumelhart D.E., Hinton G.E., and Williams R.J..Parall Distributed Processing-Explorations in the Microstructure of Cognition[J].MIT press, 1986, 30(5): 318-362.

[6]. Xiaoguang Liu.Research on Key Technologies of Automated Web Service Composition in Networked [D]. Shanghai: Shanghai Jiao Tong University,2008.

[7]. Yong Wang,Xiang Yi.An actor-based language to unifying web service orchestration and web service choreography [J].Computer Science and Information Processing, 2012, 8(2):1055-1060.