DOBD Algorithm for Training Neural Network: Part II. Application

WU Jian-yu(吴建昱)， HE Xiao-rong(何小荣)

(*Department of Chemical Engineering, Tsinghua University, Beijing 100084, China*)

Abstract：In the first part of the article, a new algorithm for pruning network—Dynamic Optimal Brain Damage(DOBD) is introduced. In this part, two cases and an industrial application are worked out to test the new algorithm. It is verified that the algorithm can obtain good generalization through deleting weight parameters with low sensitivities dynamically and get better result than the Marquardt algorithm or the cross-validation method. Although the initial construction of network may be different, the finial number of free weights pruned by the DOBD algorithm is similar and the number is just close to the optimal number of free weights. The algorithm is also helpful to design the optimal structure of network.

Key words：neural network; DOBD algorithm; Marquardt method; overfitting; pruning; training; application

CLC No.：N945.12 **Document Code**：A **Article ID**：1009–606X(2002)03–0262–06

1 INTRODUCTION

In Part I of this article, the Marquardt algorithm[1] is combined with OBD[2] and a new algorithm called Dynamic Optimal Brain Damage (DOBD) has been presented. High speed for training network is demonstrated with efficiently avoiding overfitting by pruning redundant weights at the same time. In this article, a three-layer network shown in Fig.1 with one hidden layer and only one output unit is used to test the new algorithm. The model can be described as:

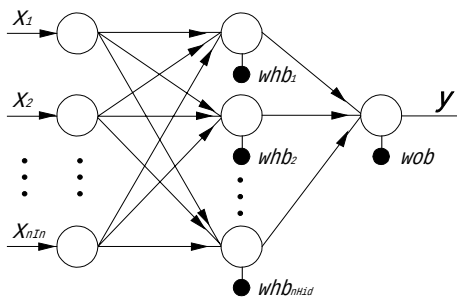

Fig.1 Neural network model of one hidden layer and one output unit

$$\min E = \sum_{i=1}^{nSam} (y^i - t^i)^2 = F^{\mathrm{T}} F ,$$

$$F = (f_1, f_2, \Lambda\ f_{nSam})^{\mathrm{T}}, \quad f_i = y^i - t^i ,$$

where y^i can be calculated with the following equations that represent the forward transmission process in the network:

$$hin_k^i = \sum_{j=1}^{nIn} \left(wh_{kj} x_j^i \right) + whb_k , \qquad hout_k^i = \frac{1}{1+\exp(-hin_k^i)}, \qquad oin^i = \sum_{k=1}^{nHid} \left(wo_k hout_k^i \right) + wob ,$$

$$y^i = \frac{1}{1+\exp(-oin^i)}, \qquad (i \in NSAM, \quad k \in NHID, \quad j \in NIN) .$$

The elements of Jacobian matrix, which are needed when using the Marquardt method and calculating sensitivities, can be calculated by

Received date: 2001-09-10, **Accepted date:** 2001-11-30

Biography: WU Jian-yu(1979–), male, native of Yangzhong city, Jiangsu Province, MS, majoring in process system engineering.

$$\frac{\partial f_i}{\partial wo_k} = y^i(1-y^i)hout_k^i, \qquad \frac{\partial f_i}{\partial wob} = y^i(1-y^i),$$

$$\frac{\partial f_i}{\partial wh_{kj}} = y^i(1-y^i)hout_k^i(1-hout_k^i)wo_k x_j^i, \qquad \frac{\partial f_i}{\partial whb_k} = y^i(1-y^i)hout_k^i(1-hout_k^i)wo_k.$$

The new algorithm is tested on several case studies and modeling the Reid vapor pressure of stabilizer gasoline, and compared with simple training by the Marquardt algorithm without pruning. The system is developed under VC++ 6.0 and all the results are got from a computer with an Inter Celeron 400 CPU.

2 CASE STUDIES

To obtain the dynamic feature of testing error during the training process, cross-validation is adopted that means to compute testing error of samples after each iteration[3, 4]. Using this method we can get dynamic curve of testing error. Within all the figures shown in this and next sections, ordinate represents the absolute error mean of samples calculated by

$$\overline{E} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - t_i)^2}.$$

CASE 1

200 training samples and 70 testing samples are generated from function $y=\sin x_1+\sin x_2+ \sin x_3+ \sin x_4$, inputs x_1, x_2, x_3, $x_4\in[-2\pi, 2\pi]$ are pseudo-random numbers generated by a C language program. The initial network has an input layer of 4 units, a hidden layer of 30 units and an output layer of 1 unit (4–30–1). The first pruning process begins after the 10th iteration and interval between two successive pruning processes is 2 iterations. MOPN is 20 and LSL is 0.03. Figure 2 illustrates the result. As shown in Fig.2, when using Marquardt without pruning, after 50 iterations testing error begins to ascend and at the 90th iteration the value reaches 0.8, which shows overfitting has occurred. While using DOBD the value of testing error is 0.26 and the value of training error is 0.19 at the 90th iteration, which shows high generalization of DOBD.

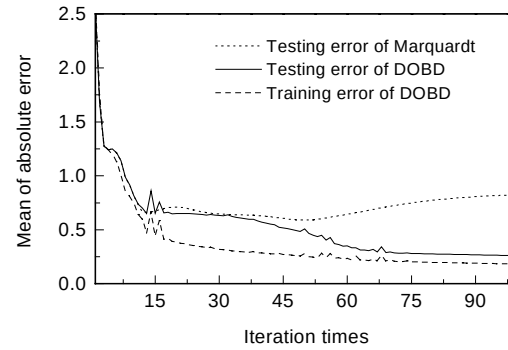

Fig.2 The training and testing results of DOBD and Marquardt method for the network with 4 inputs and 30 hidden nodes for case 1

Table 1 shows different extents to which weights are deleted by DOBD under different initial topological constructions of network and error comparison between DOBD and the Marquardt method without pruning. The criterion of complete convergence is $\|J^{\mathrm{T}}F\|<0.001$.

Table 1 Training and testing result comparison among four different networks for case 1 (90 iterations)

Topological construction of network	Initial weight number	Number of weight deleted	Number of remaining weight	Training error without pruning	Testing error without pruning	Training error with DOBD	Testing error with DOBD
4–30–1	181	122	59	0.14	0.71	0.21	0.31
4–25–1	151	90	61	0.11	0.66	0.22	0.31
4–20–1	121	55	66	0.15	0.36	0.22	0.27
4–15–1	91	41	50	0.20	0.44	0.22	0.26

CASE 2

200 training samples and 70 testing samples are generated from function

$$y=\frac{\exp(x_1\ln\frac{x_2}{x_5})x_4}{x_2+x_3+x_4}+\frac{x_8}{\ln x_1}-\exp\left\{\frac{x_3}{x_4[1+\exp(x_7/40)]}\right\}+x_3\ln x_6\ln x_2+\frac{x_4\sin x_3}{3}+\frac{x_8x_5}{x_6},$$

where $x_1, x_2, \ldots, x_8\in[20, 70]$ are pseudo-random numbers. The topological construction of the network is 8–10–1. The first pruning process begins after the 8th iteration and the interval between two successive pruning processes is one iteration. MOPN is 20 and LSL is 0.02. There are totally 61 weights deleted from initial 101 ones at the end of the training process.

As shown in Fig.3, although no obvious phenomenon of overfitting appears by Marquardt without pruning, error of testing samples descends slowly. While using DOBD, it only takes about 35 iterations to converge near to local minimal point where training error is 15.0 and testing error is 18.4.

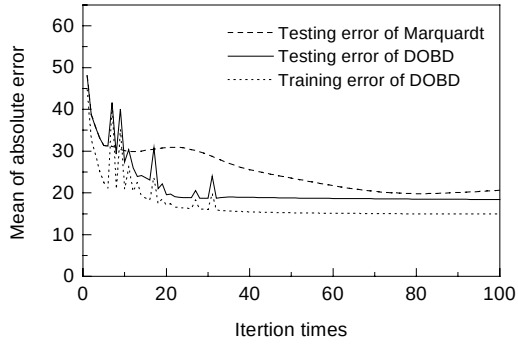

Fig.3 The training and testing results of DOBD and Marquardt method for the network with 8 inputs and 10 hidden nodes for Case 2

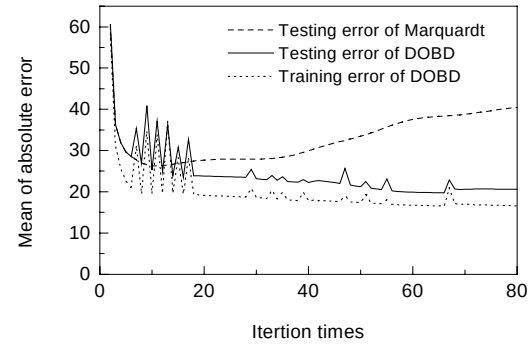

Fig.4 The training and testing results of DOBD and Marquardt method for the network with 8 inputs and 15 hidden nodes for Case 2

If changing the topological construction of the network to 8–15–1 and keeping other conditions unchanged, there are totally 106 weights deleted from initial 151 ones at the end of the training process and the result is illustrated in Fig.4.

As shown in Fig.4, when using Marquardt without pruning, after 10 iterations testing error begins to ascend and at the 80th iteration the value reaches 40, indicating obvious overfitting. While using DOBD the value of testing error is 20 and the value of training error is 17 at the 80th iteration. The difference between these two errors is only 3, which confirms the DOBD can avoid overfitting efficiently.

3 INDUSTRIAL APPLICATION

We present a model to estimate the stabilizer gasoline RVP (Reid vapor pressure) for a FCC (Fluidized catalytic cracking) unit in a refinery[5,6]. Figure 5 illustrates the flow chart of FCC. Further refining of crude gasoline produced from catalytic cracking is underwent in a stabilizer which is a rectifying column with operating pressure at 10~15 kPa. Deethanized gasoline is input from the middle part of the column. Stabilized gasoline is from the bottom and liquefied gas from the top. Sometimes non-condensable gas is sent out to stabilize the pressure of the column. Because change in operating condition can make a significant difference on the vapor pressure of gasoline, on-line analysis and

forecast are needed to get acceptable gasoline and optimize the operating condition. Moreover, it is favorable to control gasoline quality in response to the market. However, mathematic and physical models cannot meet the industrial requirement. Neural network is an alterative method according to the industrial record data of control parameters of the stabilizer column. Eight most important control factors are selected as the inputs of the network and RVP to be forecast is the output. Part of the data is shown in Table 2.

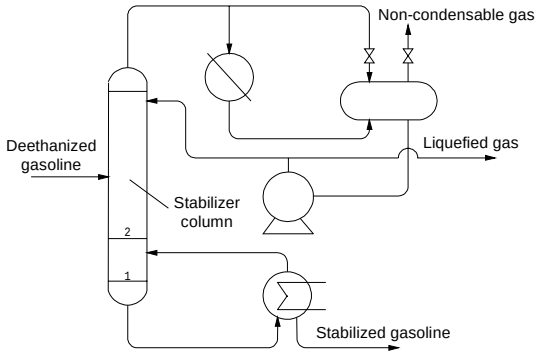

Fig.5 Flow chart of gasoline stabilization in the FCC process

Table 2 Part of training samples for RVP model

Feed flow (t/h)	Feed temp. (^oC)	Bottom temp. (^oC)	Temp. of vapor from reboiler (^oC)	Top temp. (^oC)	Top pressure (MPa)	Reflux Flow (t/h)	Reflux temp. (^oC)	RVP (kPa)
80.0	140.0	165.0	170.0	54.0	9.00	24.0	33.0	41.0
120.0	133.0	158.0	165.0	49.0	10.00	33.0	34.0	50.0
96.0	131.0	159.0	163.0	55.0	9.00	25.0	34.0	60.0
90.0	130.0	156.0	161.0	53.0	10.50	24.0	36.0	64.0
…	…	…	…	…	…	…	…	…
89.0	125.7	152.8	158.3	56.2	9.80	37.8	37.4	68.0
77.0	125.2	153.1	158.2	51.0	9.80	31.5	36.5	75.0
68.0	123.8	151.6	157.8	51.3	10.00	36.2	34.6	82.0

264 training samples and 70 testing samples are selected from industrial data. The topological construction of the network is 8–10–1. The first pruning process begins after the 6^{th} iteration and the interval between two continual pruning processes is one iteration. MOPN is 10 and LSL is 0.1. There are totally 51 weighs deleted from initial 101 ones at the end of the training process.

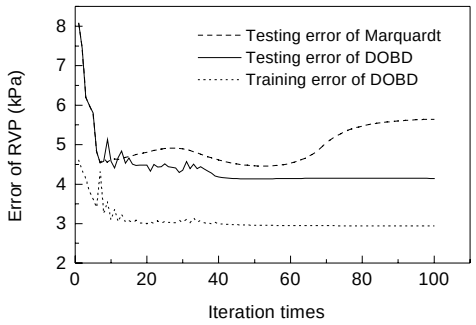

Fig.6 The training and testing results of DOBD and Marquardt method for the network

As shown in Fig.6, DOBD works well on industrial data. The testing error for DOBD is obviously smaller than that of Marquardt without pruning. However, because of the noise in industrial data, there is a difference about 1 kPa between the testing and training error.

Table 3 shows the testing results of the two different methods. For DOBD the result comes from the data of the 100^{th} iteration. For cross-validation the result comes from the smallest testing error as shown at the 55^{th} iteration in Fig.6. It is required for industrial production that the steam pressure error should be in the range of ±5 kPa with confidence level above 85%. The result of DOBD has met the requirement.

Table 3 Testing result for FCC modeling

	Cross-validation	DOBD
Average of absolute error	4.46	4.14
Sample number with absolute error over 5 kPa	19	10
Data with absolute error over 5 kPa (%)	25.0	13.2

4 RESULT ANALYSIS

It is confirmed by the above three examples that DOBD can avoid overfitting caused only by over-complex construction of network. In all the examples, the testing error of DOBD is obviously smaller than that of the Marquardt method without pruning and is also close to the training error. Moreover, there is also some improvement got from DOBD compared with cross-validation and no additional testing samples are needed, which is necessary for cross-validation. Thus DOBD is much more useful in dealing with problems with small number of samples.

The adoption of the Marquardt algorithm has greatly improved convergence speed of DOBD. As shown by the three examples, generally after 100 iterations the error curve tends to stabilize, which means that the convergence process is very close to certain local minimal point. For OBD, however, network is trained by the traditional steepest descent method and it often takes hundreds of iterations to complete training. Moreover, reduplicate training also diminishes computing efficiency. It is expected that applications of DOBD to on-line optimization or control may be possible because of its high training speed.

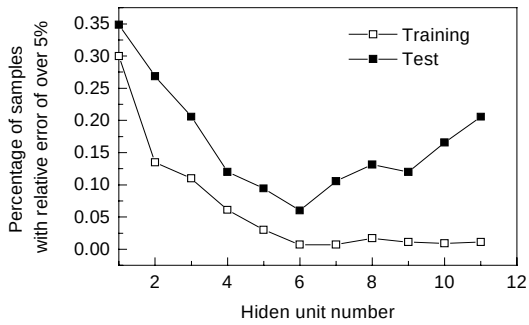

Fig.7 Relationship between overfitting and hidden unit number for Case 2

DOBD is also useful in looking for the optimal network construction. As shown in Case 1, it can be drawn from Table 1 that although there are different initial topological constructions, the final number of free weights remained is about the same. It is also the same with Case 2, for the network of 8–10–1, the initial number of weights is 101, after pruning 61 redundant weighs the final number of weights remained is 40. For the network of 8–15–1, the initial number of weights is 151, after pruning there is 45 weights remained and for the network of 8–20–1 it is 52. These facts suggest that the optimal network of Case 2 should have 5 hidden units. To test this supposition, the relation between overfitting and number of hidden units for Case 2 is illustrated in Fig.7, where abscissa represents the number of hidden unit and ordinate represents the fraction of samples with relative error above 5%. The criterion of complete convergence is $\|J^{\mathrm{T}}F\|<0.01$ and Fig.7 is the average of five groups whose initial weights are selected randomly. When the number of hidden unit is below 6, it is too simple for the network to simulate the mapping relation. When the number is above 6 overfitting happens. Thus it can be drawn that the network with 6 hidden units is optimal, which is close to the result of 5 hidden units forecast from about 50 remaining weights.

5 CONCLUSION

Network with one hidden layer and only one output unit have been constructed and trained with the DOBD algorithm. Two case studies and an industrial application are presented to confirm the efficiency of avoiding overfitting caused by over-complex network construction and the high convergence speed for DOBD, which is analyzed in Part I of the article. Moreover, DOBD also has significance on looking for the optimal network construction by selectively deleting redundant weights.

NOTATION:

NIN	Set {1, 2,⋯, *nIn*}
$NHID$	Set {1, 2,⋯, *nHid*}
$NSAM$	Set {1, 2,⋯, *nSam*}
J	Jacobian matrix of the error function with respect to weights
hin_k^i	Summed input value of the kth hidden unit with respect to sample *i*
$hout_k^i$	Output value contributed by the kth hidden unit with respect to sample *i*
$nHid$	Number of uints of middle hidden layer
nIn	Number of units of input layer
$nSam$	Number of training samples
oin^i	Summed input value of output unit with respect to sample *i*
t^i	Real output for sample *i*
wh_{kj}	Weight between the kth hidden unit and the jth input
whb_k	Threshold of the kth hidden unit
wo_k	Weight between the kth hidden unit and the output
wob	Threshold of output
x_j^i	Value of the jth input of sample *i*
y^i	ANN calculational output for sample *i*

REFERENCES:

[1] Hagan M T, Menhaj M B. Training Feedforward Networks with the Marquardt Algorithm [J]. IEEE Transaction on Neural Networks, 1994, 5(6): 989–993.

[2] Le Cun Y, Denker J S. Optimal Brain Damage [A]. Touretzky D S. Advances in Neural Information Processing(2) [C]. Denver: Morgan Kaufmann, 1990. 598–605.

[3] Reed R. Pruning Algorithms—A Survey [J]. IEEE Transactions on Neural Networks, 1993, 4(5): 740–747.

[4] Lutz P. Automatic Early Stopping Using Cross Validation: Quantifying the Criteria [J]. Neural Networks, 1998, 11(4): 761–767.

[5] ZHAO X G, HE X R, CHEN B Z. Establishing Quality Models for Oil Products by Using Neural Networks [J]. Petroleum Processing, 1993, 24(9): 9–14 (in Chinese).

[6] LIN S X. Petroleum Refining Engineering [M]. Beijing: Petroleum Industry Press, 1988. 109–112 (in Chinese).