CMAC_based Fault Diagnosis of Power Transformers

Wei-Song Lin[1], Chin-Pao Hung[1], Mang-Hui Wang[2]

[1]Institute of Electrical Engineering
National Taiwan University
Taipei, Taiwan, R.O.C

[2] Department of Electrical Engineering
National Chin-Yi Institute of Technology
Taichung, Taiwan, R. O. C.

E-mail: cbhong@chinyi.ncit.edu.tw

Abstract: Dissolved gas analysis (DGA) is one of most useful techniques to detect the incipient faults of power transformer. However, the identification of the faulted location by the traditional method is not always an easy task due to the variability of gas data and operational natures. In this paper, a novel CMAC_based method is proposed for the fault diagnosis of power transformers. By introducing the IEC std. 599 to generate the training data, and using the characteristic of self-learning and generalization, like the cerebellum of human being, a CMAC_based fault diagnosis scheme enables a powerful, straightforward, and efficient fault diagnosis. With application of this scheme to published transformers data, the diagnoses demonstrate the new scheme with high accuracy and high noise rejection abilities. Moreover, the results also proved the ability of multiple incipient faults detection.

Keywords: transformer fault diagnosis, dissolved gas analysis (DGA), neural network, CMAC.

1. INTRODUCTION

Power transformers are essential devices in a transmission and distribution system. Failure of a power transformer may cause a break in power supply and loss of profits. Therefore, it is of great importance to detect incipient failures in power transformers as early as possible, so that we can switch them safely and improve the reliability of power systems.

A long in-service transformer is subject to electrical and thermal stresses, which may form byproduct gases to indicate the type of incipient failure. Dissolved gas analysis (DGA) is a common practice in the incipient fault diagnosis of power transformers [1-2], which tests and samples the insulation oil of transformers periodically to obtain the constituent gases in the oil due to breakdown of the insulating materials inside. As study results indicate, corona, overheating and arcing are the three main causes for insulation degradation in a transformer [2-4]. The energy dissipation is least in corona, medium in overheating, and highest in arcing. The fault related gases include hydrogen (H_2), methane (CH_4), acetylene (C_2H_2), ethane (C_2H_6), carbon monoxide (CO), and carbon dioxide (CO_2).

In the past decade, various fault diagnosis techniques have been proposed that include the conventional key gas method, ratio method [2-5], and recently, the expert systems [6], neural network (NN) [7,8,13,14] and fuzzy logic approaches [9-12]. The conventional key gas or ratio method is based on experience in fault diagnosis using DGA data, where may vary from utility to utility due to the heuristic nature of methods and no general mathematical formulation can be utilized. The expert system and fuzzy logic approaches can take human expertise and DGA standards from the fault diagnosis system, and have been successfully applied in this field. However, there are some intrinsic shortcomings, such as the difficulty of acquiring knowledge and maintaining database, so, their effectiveness depends on the completeness and precision of expert expertise. The neural network (NN) can directly acquire experience from the training data, and exhibit highly nonlinear input-output relationships. This can overcome some of the shortcomings of expert system. However, the amount of training data must be large enough to ensure proper training. Non-training data are easy to cause wrong diagnoses. Moreover, the multiple faults diagnoses and fault anticipation abilities are still lacking.

In this paper, a novel CMAC_based method is presented for the fault diagnosis of power transformers. First, we generated the virtual training data based on the IEC std. 599 to replace the large amount actual training data. Second, we developed a CMAC_based diagnosis model and using virtual training data to train the memory weights. Finally, the proposed scheme can be used to diagnose the fault type of power transformers. With application of this scheme to published transformers data, the diagnoses demonstrate the new scheme with high accuracy and high noise rejection abilities. Moreover, the results also proved the ability of multiple incipient faults detection.

2. BACKGROUND ON CMAC NEURAL NETWORK

Albus proposed a neural model called CMAC (Cerebellar Model Articulation controller), which like the models of human memory, perform a reflexive processing [15]. The CMAC, in a table look-up fashion, produced a vector output in response to a state vector input. Figure 1 shows a basic configuration of CMAC network [17], where the input states are denoted by $x \in R^n$, and the output is $y \in R^m$. Through a series of mappings, include the quantization, segment addresses coding, virtual addresses concatenation, Hash coding (if needed), and sums the fired memory addresses weights to obtain an output. *The mapping processes must satisfy the similar inputs excite the similar memory addresses*, i.e. if the input states are close to (similar) in input space will have their corresponding sets of association cells overlap. For example, if x_1 and x_2 are

0-7803-7278-6/02/$10.00 ©2002 IEEE

similar (close), x_1 excites the memory addresses a_1,a_2,a_3,a_4, x_2 should excite the memory addresses a_2,a_3,a_4,a_5 or a_3,a_4,a_5,a_6. If two inputs fire up the same memory addresses, we say the similarity of the two inputs is high. Low similarity would excite fewer same memory addresses.

As described above, the CMAC architecture, every memory cell (Cerebellar cell) remembers a weight value. Through a series of mapping, every input state will fire up a group of memory cells (Fig. 1 shows the number of fired memory cells are 4). The summation of the fired memory cells weights will obtain an output. Compare the output with the desired target, then using the difference to update the fired memory cells weights (training/learning). The fired memory cells will remember the correct mapping relation for the special input state. Therefore, when the same or similar input states input again, the output will preserve the possible correct output depending on the similarity. This idea is same as the brain of human being. When we see a familiar friend, no doubt we can recognize him/her. Someday the familiar friend takes a gauze mask, we just see some feature of him/her, most situation we still can recognize him/her. As a result, the characteristic of generalization, local reflexive action and self-learning make the CMAC attractive to fault diagnosis system, especially to multiple faults and lack of fully training data system.

3. THE CONFIGURATION OF CMAC_BASED FAULT DIAGNOSIS SYSTEM

In dissolved gas analysis, the IEC codes have been used widely by the utilities. From IEC std. 599, the codes of different gas ratios and fault classifications according to the gas ratio codes are shown in Table 1 and 2. Although IEC codes are useful for fault diagnosis in transformers, but the number of code combinations is larger than the number of fault types, and "no match" may be indicated in the fault diagnosis. In this section, the CMAC_based fault diagnosis method is proposed for power transformers to solve the no match, noise, and multiple faults problem.

3.1 The development of CMAC_based fault diagnosis system

Fig.2 shows the configuration of the CMAC_based fault diagnosis system of power transformers. Refer to the IEC std. 599 the gas ratio of C_2H_2/C_2H_4, CH_4/H_2, and C_2H_4/C_2H_6 are used as the input states. The diagnosis system contains 9 parallel memory layers and every memory layer has one output node. Every memory layer remembers one fault type feature. E.g. layer 1 store the features of fault type 1 of Table 2, layer 2 stores the features of fault type 2 of Table 2, etc. Input one group gas ratio data, through a series of mapping, the input data will generate one group fired memory addresses. To sum the excited memory addresses of each layer, output node will obtain one value to express the possibility of fault type *n*. To confirm the fault type the output value will be close to 1. Multiple nodes output 1 represent the multiple fault types exist.

3.2 The training of CMAC_based fault diagnosis system

The proposed scheme using the IEC codes of Table 1 and 2 to generate the training data. Therefore the large amount real data are not necessary. For example, the gas ratio code of fault type 2 (C_2H_2/C_2H_4, CH_4/H_2, C_2H_4/C_2H_6) = (0,0,1), i.e.$C_2H_2/C_2H_4<0.1$, $0.1 \leq CH_4/H_2 \leq 1$,$1< C_2H_2/C_2H_6 \leq 3$. Therefore, using the program 1(designed by MATLAB), the virtual training data can be generated. In program 1 the step value *STEP_X* determines the resolution of training data. High resolution will cause long training time. The training data then send to the CMAC network (layer 2), through the quantization, fired addresses coding, and sums the fired memory cells weights to obtain an output. Compare the desired output 1, then the error used to tuning the fired memory weights. The details will describe as follows:

```
for C2H2_C2H4=0:STEP_1:0.1
  for CH4_H2=0.1:STEP_2:1
   for C2H4_C2H6=1:STEP_3:3
   %quantization, address coding,...
   end    end  end
```

Program 1. Training data generation

3.2.1 Quantization mapping

The input values send to the CMAC network, it is first though the quantization mapping Q to produce a quantization level output. The quantization output can be described as follows [17]

$$q_i = Q(x_i, x_{i\min}, x_{i\max}, q_{i\max}), \quad i = 1,\cdots n \qquad (1)$$

where n is the input numbers. The resolution of this quantization depends on the expected maximum and minimum inputs, x_{imax} and x_{imin}, and on the number of quantization levels, $q_{\max}$. High resolution will have good generalization ability but more memory size is needed. Since the major boundary of IEC codes are 0.1,1, and 3. Assuming the maximum quantization level $q_{\max}$ is chosen as 12, then the quantization mapping diagram can be shown as Fig. 3. Figure 3 shows that the input state between 0.1 and 1, 1 and 3 are divided five quantization levels respectively. It is note that we can change the $q_{\max}$ value depending on the resolution requirements.

3.2.2 Segment address mapping

Each quantization level then though V mapping (Table 3) outputs A^* segment addresses, A^* the number of associated (fired) memory cells. Table 3 lists the mapping relation of quantization level and the segment address, which the quantization level q_{max} is 8 and $A^*$is 4. For example, the quantization level 3 will map a group segment addresses $[v_{11}, v_{12}, v_{13}, v_{14}] = [5,6,3,4]$.

3.2.3 Concatenation

As described above, each input state produces A^* segment addresses. The concatenation unit then concatenated these segment addresses as a virtual address. The concatenation equation can be expressed as following.

$$V_j = concat(v_{j1}, v_{j2}, \cdots, v_{jn}), \qquad j = 1,\ldots, A^* \qquad (2)$$

For example, assume the gas ratio quantization levels of (C_2H_2/C_2H_4, CH_4/H_2, C_2H_4/C_2H_6) are equal to (3,6,8), then the segment addresses generated by C_2H_2/C_2H_4 are $[v_{11}, v_{12}, v_{13}, v_{14}] = [5,6,3,4]$,by CH_4/H_2 are

$[v_{21}, v_{22}, v_{23}, v_{24}]$ =[9,6,7,8] and by C_2H_4/C_2H_6 are $[v_{31}, v_{32}, v_{33}, v_{34}]$ =[9,10,11,8]. Then the concatenation operation can be expressed as follows:

$V_1 = concat[v_{11}, v_{21}, v_{31}] = concat[5,9,9] = 010110011001\text{B}$

$V_2 = concat[v_{12}, v_{22}, v_{32}] = concat[6,6,10] = 011001101010\text{B}$

$V_3 = concat[v_{13}, v_{23}, v_{33}] = concat[3,7,11] = 001101111011\text{B}$

$V_4 = concat[v_{14}, v_{24}, v_{34}] = concat[4,8,8] = 010010001000\text{B}$

Assume

$$bitn = ceil(\log_2(q_{j\max} + A^*), \qquad (3)$$

where *bitn* is the minimum bit numbers to encode the segment address and *ceil(x)* a function rounds the elements of *x* to the nearest integers towards infinity. Then the general form of V_j can be calculated as follows:

$$V_j = \sum_{i=1}^{n} v_{ji} 2^{bitn \times (i-1)} \quad , j = 1, \cdots, A^* \qquad (4)$$

To sum the weights located at these addresses will obtain an output value.

3.2.4 Hash coding

As described above, high resolution (increase the quantization level and A^*) will have better generalization ability. An increase in the number of A^* results in concurrently increases the size of virtual address space. If the memory size beyond the limitation, Hash coding is needed. To implement CMAC network using a realistic amount of memory, Hash coding performs a many-to-one uniform random mapping to generate a physical memory address. As described in [17], Hash coding can compress the huge virtual address space into a compact amount of memory and minimize the probability of physical address collision (different inputs fire up the same association address). In this paper, we don't consider the Hash coding because of the memory size is acceptable.

3.2.5 Output mapping

The finial mapping computes the output *y* by summing the weights w_{Vj} located at the physical memory addresses. It can be described as

$$y = \sum_{j=1}^{A^*} w_{Vj} \qquad (5)$$

3.2.6 Update the weighting

During training, if the output of the CMAC does not match the desired output for a given fault type input, the fired addresses are updated using the following steepest-descent update rule:

$$w_{v_i}^{new} \leftarrow w_{v_i}^{old} + \beta \frac{y_d - y}{A^*}, \qquad i = 1, 2, \ldots, A^* \qquad (6)$$

In this equation, y_d is the desired output, *y* the actual output, and $0 < \beta \le 1$ the learning gain.

3.2.7 Noise rejection

The quantization and segment mappings give the CMAC the ability to generalize (produce similar outputs in response to similar inputs). Continuous variations in input values translate into discrete variations in input quantization levels. As described in 3.2.3 , if the quantization level of C_2H_2/C_2H_4 input change by 1(3→4), then the segment address mapping $[v_{11}, v_{12}, v_{13}, v_{14}]$ from [5,6,3,4] change to [5,6,7,4]. All virtual address segments remain the same except v_{13}, which shifts from 3 to 7. Consequently, outputs associated with neighboring input quantization levels will have three of four virtual addresses in common because only one address will have changed, V_3 shifts from *concat*[3,7,11] to *concat*[7,7,11] (assuming the other two input levels remain constant). Therefore, even though the noise added to the input, the CMAC outputs still can preserve most of the correct features. Note that for a given input quantization mapping, an increase in A^* results in an increase in the amount of shared weights between neighboring input/output pairs. This will increase the generalization ability and improve the noise rejection.

3.2.8 Convergence

For a supervised learning system, the convergence is confirmed [16]. In this paper, we assume the memory size is large enough and without using the Hash coding. The collision will not happen and the convergence is guaranteed.

3.2.9 Learning performance evaluation

Assuming the i^{th} node (i=1,...,9) of Fig.2 outputs 1 represents system with fault type *i*. The training data number n_g generated by program 1 can be calculated as following equation:

$$n_g = fix(\frac{0.1-1}{step\_1} + 1) \cdot fix(\frac{1-0.1}{step\_2} + 1) \cdot fix(\frac{3-1}{step\_3} + 1) \qquad (7)$$

where *fix*(*x*) function rounds the elements of *x* to the nearest integers towards zero.

Let

$$E_i = \sum_{j=1}^{n_g} (y_{ij} - 1)^2 \qquad i = 1, \ldots, 9 \qquad (8)$$

where *i* subscript represents the i^{th} memory layer. Then we can stop the training phase when $E_i < \varepsilon$, ε is a positive small number.

3.3 Diagnosis algorithm

As described above, the diagnosis algorithm summarized as follows:

3.3.1 Off-line training phase

Step 1. Build the architecture of CMAC_based fault diagnosis system, including three input states, nine layers memory and nine output nodes.

Step 2. Specify the quantization level $q_{\max}$, learning gain β, and the amount of network generalization A*.

Step 3. Generate the virtual training data via IEC code of Table 1 and 2.

Step 4. Quantization, fired addresses coding, and sums the fired memory cells weights to obtain an output.

Step 5. Update the fires memory cells weights using eq. (6).

Step 6. Does the training data finish? No, go to step 3. Yes, next step.

Step 7. Learning performance evaluation. If $E_i < \varepsilon$, stop training and save the memory weights. Otherwise, go to step 3.

Step 1 to 7 is off-line mode. The training time maybe shorter just few seconds or longer more than few hours depending on the data resolution, q_{max}, A^*, and the selection of gas ratio range (PENTIUM iii500, using MATLAB programming). Fortunately, the off-line mode just only needs to run one time. Generally, long training time will obtain better and more exact weights, just like the learning mode of human being.

3.3.2 On-line mode

Finish the off-line training mode, then the diagnosis system can be used to diagnose the fault type of transformers.

Step 8. Load the last saved memory weights and specify the threshold value (e.g. $\eta = 0.9$).

Step 9. Input the gas ratio data that to be diagnosed.

Step 10. Quantization, fired addresses coding, and sums the fired memory cells weights to obtain an output. If the output larger than a specified threshold value, then the fault type is confirmed.

Step 11. If the diagnosis is correct, go to step 12. Otherwise, go to step 13.

Step 12. Does the new gas ratio data to be diagnosed? Yes, go to step 9. No, go to step 14.

Step 13. Update the fired memory cells weights, then go to step 12.

Step 14. Save the memory weights and exit.

4. CASE STUDIES AND DISCUSSIONS

4.1 Tested data diagnosis

To demonstrate the effectiveness of the proposed CMAC_based fault diagnosis method, twenty power transformer DGA results of from references [7-9] are tested. The detailed gas data are shown in Table 4, where the AFC, IEC and CMC express the actual fault type, the diagnoses of the IEC method and the proposed CMAC_based scheme, respectively. Through 10 iterative times training or learning performance evaluation ($E_i < 0.01$), the memory weights mapping distribution drawing is shown in Fig. 4. Figure 4, similar to the cerebellum of human being, maps the feature of each fault type on a special memory layer. The larger difference of each layer distribution represents easier to diagnose the exact fault type. Similar distribution plot of each layer means the diagnoses with the multiple fault types easily. Table 5 lists the related parameters of CMAC network. Table 6 shows the detailed outputs of each node. The last column is the diagnosis fault types considering the threshold value $\eta = 0.9$ or 0.95(bold type). Fault type no. 4 and 5 are medium temperature and high temperature thermal fault respectively. Therefore fault type no. 4 and 5 are similar, and the diagnoses of the 4th, 7th, 8th, 11th, and 13th data contain the multiple faults of no. 4 and 5 and the possibility of fault type no. 5 is higher than no. 4. Observe the IEC code of Table 2, the codes of fault type no. 8 and 9 are overlap somewhat. Consequently, the diagnoses of the 5th, 6th, 12th and 14th data output the multiple fault of no. 8 and 9 are reasonable. Note that the 10th and 19th have no matching codes for diagnosis by the IEC method, but the results of the proposed method still detect the possible faults type. Moreover, the gas ratio of C_2H_2/C_2H_4 in the 5th, 6th data are far away the virtual training data bound (the bound of C_2H_2/C_2H_4 is 3, virtual training data take to 6, and the 5th, 6th data are 14, 15.94, respectively). The diagnoses still diagnose the fault type no. 9. The 20th data diagnosed by the IEC method only obtained the fault type no. 4, but the proposed CMAC_based scheme confirmed the multiple faults type no. 4 and 5. The output values of node 4 and 5 are equal to 1 exactly. In short, the proposed method provided the most possible fault type diagnoses and never lost the actual fault type. It is proved that the proposed scheme not only diagnoses the main fault types of power transformers but also provides useful information for future fault trend analysis.

4.2 On-line training (learning)

If the wrong fault type happened in the diagnosis process, the on-line training is proceeded to re-train the memory weights. The update rule is same as the equation (6). But since the characteristic of the proposed scheme is to provide the most possible fault type diagnosis, we don't use the anti-excited scheme to update the redundant fault type. Increase the threshold value η or just consider the maximum output value will filter the redundant fault type naturally.

4.3 Noise rejection test

To test the noise rejection ability of the proposed method, we added ±5% to ±50% random noise to the input states, i.e. added $\pm(5\sim 50)\% \times rand(1)$ noise to the input states, where $rand(1)$ is normal distribution random function between 0 and 1. Table 7 shows the detailed output of each node for input node with ±10% noise. Table 8 shows the diagnoses for different percentage noise added. In most test data the diagnoses still output correct fault type even though the noise over 50%. Table 9 shows the diagnoses performance with different percentage noise, the last column are the IEC method results. As observed from Table 8 and 9, it proved the proposed method with high noise rejection ability and capable of multiple faults detection.

5. CONCLUSION

This paper presents a novel CMAC_based fault diagnosis method for power transformers. Using the characteristic of generalization, local reflexive action and self-learning ability, the proposed scheme achieves at least the following merits: 1) Don't require the actual data to train the CMAC network and high diagnosis accuracy is obtained. 2) Detect the main fault type and provide useful information for future fault trends and multiple faults analysis. 3) High noise rejection ability. 4) Suit to non-training data and associate the most similar fault type. 5) Don't require any expert experience to train the CMAC network. The tested data demonstrate the success of proposed scheme. However, how to design an optimal memory size, quantization level, and associated cells numbers to obtain more efficient application are our future work and under studying.

6. REFERENCE

[1] J. J. Kelly, "Transformer fault diagnosis by dissolved-gas analysis," *IEEE Trans. on Industry Applications*, Vol. 16, No. 6, pp. 777-782, Dec. 1980.

[2] R. R. Rogers, "IEEE and IEC codes to interpret faults in transformers, using gas in oil analysis," *IEEE Trans. on Elect. Insul.*, Vol. 13, No. 5, pp. 349-354, 1978.

[3] IEC Publication 599, Interpretation of the analysis of gases in transformers and other oil-filled electrical equipment in service, First Edition, 1978.

[4] *IEEE Guide for the Interpretation of Gases Generated in Oil-Immersed transformers*, IEEE Std. C57.104-1991.

[5] E. Dornenburg, and W. Strittmater, "Monitoring oil cooling transformers by gas analysis", Brown Boveri Review, 61, pp. 238-274, 1974.

[6] C. E. Lin, J. M. Ling, and C. L. Huang, "An expert system for transformer fault diagnosis using dissolved gas analysis", *IEEE Trans. on Power Delivery*, Vol. 8, No. 1, pp. 231-238, Jan. 1993.

[7] Y. Zhang, X. Ding, Y. Liu and P. J. Griffin, " An artificial neural network approach to transformer fault diagnosis", *IEEE Trans. on PWRD*, Vol. 11, No. 4, pp. 1836-1841, Oct. 1996

[8] Z. Wang, Y. Liu, and P. J. Griffin, "A combined ANN and expert system tool for transformer fault diagnosis", *IEEE Trans. on Power Delivery*, Vol. 13, No. 4, pp. 1224-1229, Oct. 1998.

[9] G. Zhang, K. Yasuoka, and S. Ishii," Application of fuzzy equivalent matrix for fault diagnosis of oil-immersed insulation", *Proc. of 13th Int. Conf. On Dielectric Liquids (ICDL'99)*, Nara, Japan, pp. 400-403, July 20-25, 1999.

[10] Q. Su, C. Mi, L. L. Lai, and P. Austin, " A Fuzzy Dissolved Gas Analysis Method for the Diagnosis of Multiple Incipient Faults in a Transformer," *IEEE Trans. on Power Systems*, Vol. 15, No. 2, pp. 593-598, May 2000.

[11] J. J. Dukarm, "Transformer oil diagnosis using fuzzy logic and neural networks", 1993 *Canadian Conference on Electrical and Computer Engineering*, Vol. 1, pp. 329-332, 1993.

[12] H. T. Yang, C. C. Liao, J. H. Chou, "Fuzzy learning vector quantization networks for power transformer condition assessment", *IEEE Trans. on Dielectrical Insulation*,Vol. 8, No. 1, February 2001.

[13] T. Yanming, Q. Zheng, "DGA based insulation diagnosis of power transformer via ANN", *Proceeding of the 6th conference on properties and applications of dielectric materials*, pp.133-136, 1999.

[14] K. F. Thang, R. K. Aggarwal, D. G. Esp, A. J. McGrail," Statical and neural network analysis of dissolved in power transformers", *IEE Dielectric Materials, Measurements and Applications Conference Publication* No. 473, 2000.

[15] J. S. Albus, "A new approach to manipulator control: the cerebeller model articulation controller (CMAC)[1] , *Trans. ASME J. Dynam., Syst., Meas., and Contr.*, vol. 97, pp.220-227, 1975

[16] Y. F. Wong, A. Sideris, "Learning convergence in the cerebellar model articulation controller", *IEEE Trans. on Neural Network*, vol. 3,no. 1, pp. 115-121, 1992.

[17] D. A. Handeiman, S. H. Lane, and J. J. Gelfand, "Integrating neural networks and knowledge-based systems for intelligent robotic control" , *IEEE Control System Magazine*, pp. 77-86, 1990.

Table 1 IEC gas ratio codes

Ranges of the gas ratio	Codes of gas ratio		
	$\frac{C_2H_2}{C_2H_4}$	$\frac{CH_4}{H_2}$	$\frac{C_2H_4}{C_2H_6}$
<0.1	0	1	0
0.1-1	1	0	0
1-3	1	2	1
>3	2	2	2

Table 2 Fault types according to the gas ratio codes

Fault no.	Fault type	$\frac{C_2H_2}{C_2H_4}$	$\frac{CH_4}{H_2}$	$\frac{C_2H_4}{C_2H_6}$
1	No fault	0	0	0
2	<150℃ Thermal fault	0	0	1
3	150℃~300℃ Thermal fault	0	2	0
4	300℃~700℃ Thermal fault	0	2	1
5	>700℃ Thermal fault	0	2	2
6	Low energy partial discharges	0	1	0
7	High energy partial discharges	1	1	0
8	Low energy discharges	1 or 2	0	1 or 2
9	High energy discharges	1	0	2

Table 3 Quantization level and segment address mapping

Segment address

11								v_3
10							v_2	v_2
9						v_1	v_1	v_1
8					v_4	v_4	v_4	v_4
7				v_3	v_3	v_3	v_3	
6			v_2	v_2	v_2	v_2		
5		v_1	v_1	v_1	v_1			
4	v_4	v_4	v_4	v_4				
3	v_3	v_3	v_3					
2	v_2	v_2						
1	v_1							
	1	2	3	4	5	6	7	8

Quantization level

Table 4 Tested gas data of transformer and dianoses by different method

No.	H_2	CH_4	C_2H_6	C_2H_4	C_2H_2	AFC	IEC	CMC
1	14.7	3.7	10.5	2.7	0.2	1	1	1,6
2	345	112.3	27.5	51.5	58.8	8	8	8
3	181	262	41	28	0	3	3	3
4	173	334	172	812.5	37.7	5	5	4,5
5	127	107	11	154	224	9	9	8,9
6	60	40	6.9	110	70	9	9	8,9
7	220	340	42	480	14	5	5	4,5
8	170	320	53	520	3.2	5	5	4,5
9	27	90	42	63	0.2	4	4	4
10	565	53	34	47	0	8	N	1,6,8
11	56	286	96	928	7	5	5	4,5
12	200	48	14	117	131	9	9	8,9
13	78	161	86	353	10	5	5	4,5
14	32.4	5.5	1.4	12.6	13.2	9	9	8,9
15	980	73	58	12	0	6	6	1,6
16	160	130	33	96	0	2	2	2,8
17	650	53	34	20	0	6	6	1,6
18	95	110	160	50	0	3	3	1,3
19	300	490	180	360	95	4	N	4
20	200	700	250	740	1	4,5	4	4,5

Table 5 CMAC network parameters

Learning times	q_{max}	A^*	bitn	Memory size	η	ε	β	Step_1 Step_2	Step_3
10	12	10	5	32768	0.9	0.01	0.9	0.04	0.1

Table 6 Detail outputs of CMAC_based method (no noise)

No.	Each node output									diagnoses
	1	2	3	4	5	6	7	8	9	
1	1.001	0.500	0.473	0.232	0.035	0.928	0.892	0.335	0	1,6
2	0.333	0.333	0.210	0.155	0.137	0.285	0.517	0.964	0.733	8
3	0.749	0.666	1.000	0.693	0.449	0.428	0.394	0.529	0.198	3
4	0.166	0.500	0.526	0.922	1.000	0	0	0.544	0.515	4,5
5	0	0.250	0	0.116	0.206	0	0.063	0.963	0.977	8,9
6	0.169	0.667	0.105	0.309	0.412	0.142	0.144	1.021	1.017	8,9
7	0.166	0.583	0.526	0.924	1.001	0	0	0.628	0.634	4,5
8	0.166	0.500	0.526	0.922	1.000	0	0	0.544	0.515	4,5
9	0.166	0.333	0.894	1.000	0.792	0	0	0.386	0.265	4
10	0.918	0.833	0.419	0.155	0	1.000	0.892	0.922	0.402	1,6,8
11	0.166	0.333	0.526	0.923	1.000	0	0	0.386	0.265	4,5
12	0	0.333	0	0.155	0.274	0	0.317	0.964	0.982	8,9
13	0.166	0.500	0.526	0.922	1.000	0	0	0.544	0.515	4,5
14	0	0.333	0	0.155	0.274	0	0.317	0.964	0.982	8,9
15	0.918	0.333	0.419	0.155	0	1.000	0.892	0.335	0	1,6
16	0.334	1.000	0.369	0.694	0.724	0.142	0.144	0.920	0.899	2,8
17	0.918	0.500	0.419	0.155	0	1.000	0.892	0.524	0.132	1,6
18	0.916	0.583	0.999	0.386	0.173	0.571	0.534	0.426	0.064	1,3
19	0.499	0.583	0.736	0.924	0.829	0.285	0.265	0.736	0.511	4
20	0.166	0.333	0.578	1.000	1.000	0	0	0.386	0.265	4,5

Table7 Detail outputs of CMAC based method with 10% noise(partial result)

No.	Each node output									diagnoses
	1	2	3	4	5	6	7	8	9	
1	1.001	0.500	0.473	0.232	0.035	0.928	0.892	0.335	0	1,6
3	0.749	0.666	0.999	0.693	0.449	0.428	0.394	0.621	0.198	3
4	0.166	0.500	0.526	0.922	1.000	0	0	0.544	0.515	4,5
5	0	0.333	0	0.116	0.206	0	0.063	0.963	0.977	8,9
11	0.166	0.333	0.526	0.923	1.000	0	0	0.386	0.265	4,5
18	0.916	0.583	0.999	0.386	0.173	0.571	0.534	0.426	0.064	1,3
19	0.499	0.583	0.736	0.770	0.829	0.285	0.265	0.736	0.511	N
20	0.166	0.333	0.578	1.000	1.000	0	0	0.386	0.265	4,5

Table 8 Diagnoses of ±5%~±50% random noise added using CMAC_based method (partial results)

No.	Diagnoses for different percentage noise						
	5%	10%	15%	20%	25%	30%	50%
1	1,6	1,6	1,6	1	1,6	1,6	1
2	8	8	8	8	8	8	8
3	3	3	3	3	3	3	3
13	4,5	4,5	4,5	4,5	4,5	4,5	4,5
14	8,9	8,9	8,9	8,9	8,9	8,9	8,9
15	1,6	1,6	6	1,6	1,6	1,6	1,6
16	$\boxed{8}$	$\boxed{8}$	2,8	$\boxed{8}$	2,8	$\boxed{N}$	$\boxed{8}$
18	1,3	1,3	1,3	1,3	1,3	$\boxed{1}$	$\boxed{1}$
19	4	$\boxed{N}$	4	4	$\boxed{N}$	4	$\boxed{N}$
20	4,5	4,5	4,5	4,5	4,5	4,5	4,5

Table 9 Diagnosis performance comparison

Noise (%)	CMAC_based method	IEC method
	$\eta = 0.9$	
± 0%	100%	87.5%
± 10%	90%	87.5%
± 20%	95%	82.5%
±30%	90%	77.5%
±50%	85%	72.5%

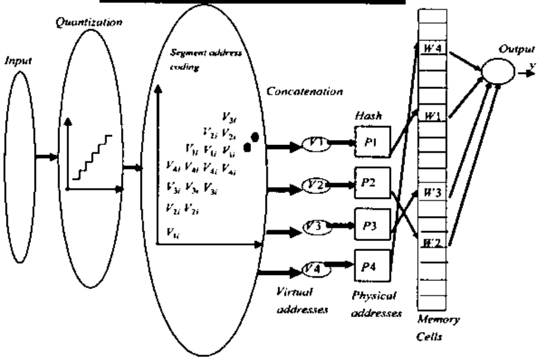

Fig. 1 The schematic of CMAC neural network

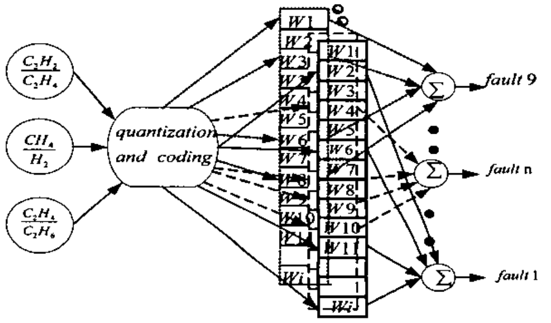

Fig.2 The configuration of CMAC_based fault diagnosis system of power transformers

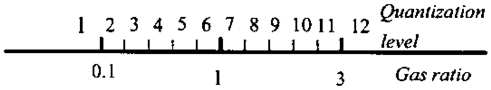

Fig.3 Quantization mapping

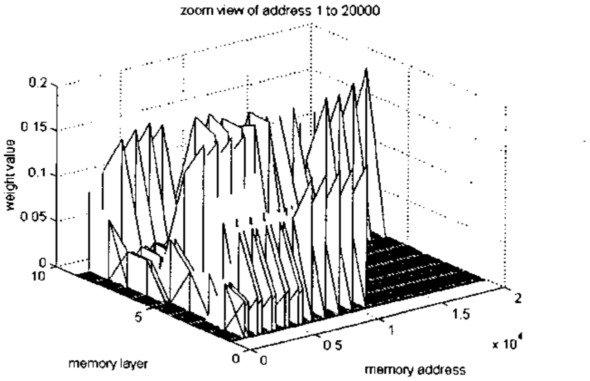

Fig.4 Weights distrubtion of memory layers