LETTERS TO THE EDITOR

To the Editor:

In the paper titled "A Neural Network Methodology for Process Fault Diagnosis" (December 1989, p. 1993), a neural network approach was proposed for automated fault diagnosis using an example problem involving a fluidized catalytic cracking (FCC) unit. This letter demonstrates that the proposed approach to this problem is unnecessarily complicated and that better results are obtained in much less time with a simple standard method. The paper cited above also missed the point that no hidden units are required to achieve perfect recall for the set of training patterns.

The reported computational experiment involved 18 different process symptoms and 13 different process faults. Both symptoms and faults, which were either present or absent, were represented as binary data. A particular combination of symptoms would thus indicate a particular fault, and the task performed by the neural network would be to "learn" the set of associations between combinations of symptoms and faults, without any explicit information about what these combinations should be.

The neural network suggested for this case consisted of a network with 18 input nodes, each corresponding to a process symptom, 13 output nodes, each corresponding to a process fault, and a variable number, ranging from 5 to 27, of intermediate nodes (hidden units). All input nodes were connected to all intermediate nodes, and all intermediate nodes were connected to all output nodes.

In a neural network, the connections are weighted, and the "training" of the network is equivalent to estimation of the set of optimal weights, optimal in the sense that maximum discrimination among faults is obtained.

Critique. Examining the matrix of input training vectors presented in the paper (see Table 1), the structure of the associations between symptoms and faults seemed so simple that a linear pattern recognizer without intermediate nodes (hidden units) was attempted: let IN be the matrix of symptom vectors, and let OUT be the matrix of fault vectors. If a weight matrix W may be devised such that

$$\mathrm{IN} \times W = \mathrm{OUT}$$

then W may be computed simply by multiplication of the inverse of IN with OUT. From the paper, it is seen that OUT is the identity of matrix I. Thus, W is the inverse of IN (see Table 2). IN is nonsquare, so the pseudoinverse must be used:

$$W = (\mathrm{IN}^T \times \mathrm{IN})^{-1} \times \mathrm{IN}^T$$

By definition, if one of the symptom vectors is now multiplied with W, the corresponding fault vector should be the result. IN and W are presented in Tables 1–2, and it is readily verified that

$$\mathrm{IN} \times W = I$$

This means that all single faults are perfectly recalled (100% recall) and that optimal weights are identified in one iteration. For comparison, the neural network approach was reported to require more than 1,000 iterations to get above 80% recall, and the maximum recall, achieved after several thousand iterations, was reported to be slightly above 98%.

Table 1. IN Matrix

1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Table 2. *W* **Matrix:** $W = (IN^T \times IN)^{-1} \times IN^T$

$\frac{2}{3}$	$-\frac{1}{3}$	0	0	0	0	0	0	0	0	0	0	0
$\frac{1}{3}$	$\frac{1}{3}$	0	0	0	0	0	0	0	0	0	0	0
$-\frac{1}{3}$	$\frac{2}{3}$	0	0	0	0	0	0	0	0	0	0	0
0	-1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	-1	0	0	0	0	0	0
0	0	0	0	1	0	-1	0	0	0	0	0	0
0	0	0	0	0	$\frac{1}{3}$	$-\frac{1}{3}$	0	0	0	0	0	0
0	0	0	0	0	$\frac{1}{3}$	$-\frac{1}{3}$	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	$\frac{1}{3}$	$-\frac{1}{3}$	0	0	0	0	0	0
0	0	0	0	0	0	-1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	$\frac{3}{4}$	$-\frac{1}{4}$	$-\frac{1}{4}$	0	0
0	0	0	0	0	0	0	0	$-\frac{1}{4}$	$\frac{3}{4}$	$-\frac{1}{4}$	0	0
0	0	0	0	0	0	0	0	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	0	0
0	0	0	0	0	0	0	0	$-\frac{1}{4}$	$-\frac{1}{4}$	$\frac{3}{4}$	0	0
0	0	0	0	0	0	0	0	0	0	0	$\frac{2}{3}$	$-\frac{1}{3}$
0	0	0	0	0	0	0	0	0	0	0	$\frac{1}{3}$	$\frac{1}{3}$
0	0	0	0	0	0	0	0	0	0	0	$-\frac{1}{3}$	$\frac{2}{3}$

The matrix *W* does, in addition, have noteworthy structural features. It is block diagonal, with four disjoint partitions: symptoms 1–4 affect only faults 1–3; symptoms 5–11 affect only faults 4–8; symptoms 12–15 affect only faults 9–11; and symptoms 16–18 affect only faults 12–13. Less than 20% of the matrix's elements are nonzero. In a neural network where every symptom is connected to every fault, this information about problem structure is lost. In a neural network with 27 intermediate nodes, 18 input nodes, and 13 output nodes, there will be 837 weights to estimate $(18 \cdot 27 + 27 \cdot 13)$. This is about 50 times the number of observations (one observation per fault). Even with as little as five intermediate nodes, there will be more than ten times as many parameters to estimate as there are observations. In a case like this, one would normally consider looking for an alternative model structure involving fewer parameters. Using a neural network approach, it would mean that one should attempt to reduce the number of connections among nodes. The sparse structure of matrix *W* indicates that a set of four small independent parallel networks might have a better performance than one big network.

To conclude, the FCC example presented by Venkatasubramanian and Chan (1989) is not suited to demonstrate the applicability of neural networks. The structure of this example problem is so simple that the use of a neural network with hidden units cannot be justified. The proposed neural network approach to the FCC example problem yields poorer results than the simple standard method, mentioned above, both with respect to the recall of training patterns and with respect to the computational effort required.

Literature cited

Venkatasubramania, V., and K. Chan, "A Neural Network Methodology for Process Fault Diagnosis," *AIChE J.*, **35**, 1993 (1989).

Kristian M. Lien
Laboratory of Chemical Engineering
The Norwegian Institute of Technology
N-7034 Trondheim, Norway

Reply:

Dr. Lien's observation that the FCCU example is simple enough to lend itself to other alternative approaches is valid. The theme of the paper, however, was to examine this problem using recent advances in multilayer neural networks based on the back-propagation learning algorithm. Dr. Lien's observation about linearly separable patterns is actually a result quite well-known among neural-network researchers (e.g., see Rumelhart and McClelland, 1986). One of the motivations for looking at the FCCU problem was that it had been examined before in the context of rule-based expert systems (Venkatasubramanian, 1988) and it thus offered us an opportunity to compare the neural-network-based approach with that of the knowledge-based system approach, as discussed in the paper. The other motivation was to examine the aspects of generalization, probabilistic inferences for incomplete and/or uncertain symptom data, and these are more readily handled by the single-hidden layer back-propagation network.

It should be noted that the accuracy measure used in the paper is a measure of how close the actual output of the network is to the desired output. It is not a direct measure of the percentage of cases correctly or incorrectly diagnosed. The accompanying table indicates the outputs of a 15 hidden-unit network during recall, after it had been trained for 8,000 time steps (see Table 1). This network has an overall accuracy of 99%. However, this does not imply that the network failed to provide correct diagnosis in 1% of the cases tested. In each case, as seen from the table, the output of the node that signals a fault (e.g., output node 1 in the first data set) is quite close to the desired output of 1.0. At the same time, the outputs of the other output nodes are close to their desired output of zero. Thus, the network is successful in correctly identifying malfunctions in all the cases studied: i.e., 100% "diagnostic accuracy." The accuracy measure merely shows quantitatively whether or not we are making any "progress" on further training. All the accuracy plots should be viewed in this light.

Table 1. 15-Hidden-Unit Network Trained for 8,000 Time Steps during Recall: Actual vs. Desired Output

Fault No.	o/d*	Output Node Number 1	2	3	4	5	6	7	8	9	10	11	12	13
F_1	o	**0.980**	0.012	0	0	0.009	0	0	0	0.005	0.007	0.001	0.008	0.01
	d	**1.00**	0	0	0	0	0	0	0	0	0	0	0	0
F_2	o	0.014	**0.975**	0.018	0.010	0	0	0.004	0.005	0	0.007	0.010	0	0
	d	0	**1.00**	0	0	0	0	0	0	0	0	0	0	0
F_3	o	0.001	0.017	**0.980**	0.003	0.002	0.008	0	0	0.006	0.003	0	0.010	0
	d	0	0	**1.00**	0	0	0	0	0	0	0	0	0	0
F_4	o	0	0.010	0.005	**0.979**	0.005	0.001	0.013	0.002	0	0.003	0.010	0.011	0.002
	d	0	0	0	**1.00**	0	0	0	0	0	0	0	0	
F_5	o	0.008	0	0.004	0.006	**0.978**	0.002	0.017	0.001	0.009	0.003	0	0.001	0.002
	d	0	0	0	0	**1.00**	0	0	0	0	0	0	0	0
F_6	o	0.001	0	0.005	0	0.004	**0.983**	0.013	0.002	0	0.003	0.005	0.003	0.007
	d	0	0	0	0	0	**1.00**	0	0	0	0	0	0	0

*o = actual output of the network; d = desired output of the network.

Literature cited

Rumelhart, D. E., and J. L. McClelland, eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: 1. Foundations,* MIT Press, Cambridge, MA (1986).

Venkatasubramanian, V., *CATDEX: An Expert System for Diagnosing a Fluidized Catalytic Cracking Unit,* G Stephanopoulos, ed., Cache Case-Studies Series on Knowledge-Based Systems in Process Engineering, Vol. 1, 41, Austin, TX (1988).

Venkat Venkatasubramanian
King Chan
Department of Chemical Engineering
Purdue University
West Lafayette IN 47907