APPLICATION OF NEURAL-NETWORK COMPUTING IN INTELLIGENT ALARM PROCESSING

Edward H. P. Chan
ECC, Inc.
1530 The Alameda, Suite 300
San Jose, CA 95126

ABSTRACT - A neural-network approach was used in building and testing an experimental intelligent alarm processor which would analyze the multiple alarms associated with a system problem and identify the particular problem causing these alarms. The test results, although preliminary, suggest the neural-network approach could be the long-sought answer for developing a generic intelligent alarm processor which could be implemented by utilities with minimal customization effort.

INTRODUCTION

The problem of interpreting a large number of alarms in a control center under stress condition is well known and well appreciated in the electric utility industry. [1],[2] The system operator needs information rather than the flood of alarm data. In the past, prioritizing the alarms and organizing the alarm messages have been used to reduce the number of alarms to be presented to the operator. [3] However, the operator still has to analyze, under stressful condition, the real-time data to diagnose the system problem. In the last few years, conventional Boolean algorithmic and rule-based expert system techniques have also been tested for analyzing alarms. [4],[5],[6],[7] Most of these approaches require precise definition of the power system configuration and the rules to be used in the analyses. If these approaches are applied to the analysis of system-wide alarms, the large number of system elements, and hence the large number of combinations of alarm situations that can occur, will make the knowledge base and rules extremely difficult to organize, construct, and maintain.

Furthermore, since power system network configurations and operational practices are quite different from one utility to another, what is developed for one utility will not be readily transferrable to another utility without an extremely high degree of customization effort. Therefore, a more promising approach would be desirable.

The author believes the answer could be the neural-network approach.

NEURAL-NETWORK COMPUTING

The end purpose of alarm processing is to give the system operator the correct information and perception of what problems are present in his or her system. It can be viewed as a pattern recognition/interpretation problem. The human ability to translate the symbols on this page into meaningful words and ideas is a form of pattern recognition. A system operator's task of diagnosing a system problem by analyzing a set of multiple alarms is also a form of pattern recognition.

In general, pattern recognition tasks require the ability to match sets of large amounts of input information simultaneously, and then generate categorical or generalized output. They also require a reasonable response to noisy or incomplete input (e.g., misspelled words or slightly different alarm patterns). While these tasks can be performed well by humans, attempts to handle these tasks by digital computers have only resulted in limited success.

The human brain is the most complex computing device known to man. The brain's powerful capabilities in thinking, interpreting, remembering, and problem-solving have led scientists to study and simulate the functionality of the brain with very simplified computer models. One result of such work is a computing approach different than the commonly known sequential digital computing; and it has been referred to as neural computing or neural-network computing. Although neural computing has been studied since the early 1950's, it has not received wide-spread interest until the last several years. [8]

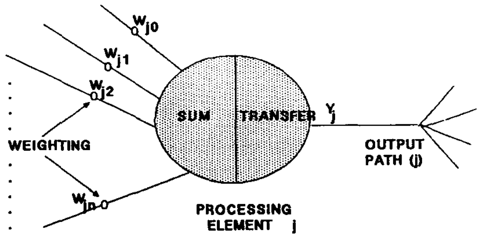
A neural network can be defined as a system that models the information processing functions of the brain.

The fundamental cellular unit of the brain is the neuron. The brain consists of tens of billions of neurons densely interconnected. Each neuron receives and combines signals from many other neurons, and if the combined signal is strong enough, it fires and produces an output signal. The output path of a neuron then splits up and connects to the input paths of many other neurons, each through a junction referred to as a synapse. The amount of signal transferred across this junction depends on the synaptic strength of the junction. This synaptic strength is what is modified when the brain learns; hence, the synapse can be considered the basic memory unit of the brain.

The basic unit of a neural network is a processing element, sometimes also called a PE, cell, or node. Its function is analogous to the biological neuron: it combines (typically sums) the inputs and produces an output in accordance with a transfer function

CH2747-4/89/0000-0246$1.00© 1989 IEEE

(typically a threshold function). A neural network consists of many processing elements joined together in several layers, with the output of one processing element connected to the input paths of other processing elements through connection weights. These connection weights are analogous to the synaptic strengths in the biological neural connections, and they are the "memory" elements of the network. Since each connection has a corresponding weight, the summation done by each processing element can be treated as a weighted sum. [9], [10] A processing element is illustrated in Figure 1. A typical 3-layer network is shown in Figure 2.



$$Y_j = f(I)$$
$$I = \Sigma\, w_{ji} x_i$$

FIGURE 1: PROCESSING ELEMENT OF A NEURAL NETWORK

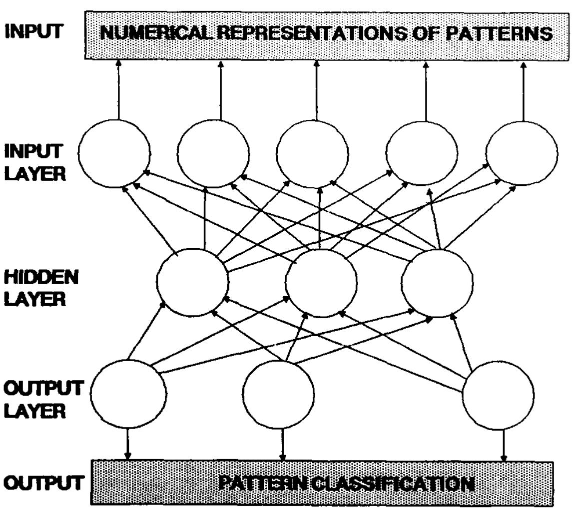


FIGURE 2: STRUCTURE OF A TYPICAL NEURAL NETWORK

An important characteristic of neural network is its ability to "learn". Unlike the rule-based expert systems where knowledge is made explicit in the form of rules, neural networks generate their own rules by "learning" from examples. Learning (or being "trained", which is analogous to being "programmed" in digital computers) is achieved through an adaptive process: when presented repetitively with the input and the corresponding desired output, the neural network organizes itself internally, gradually adjusting itself to achieve the desired input/output mapping. This ability to learn and build unique structures for a particular problem without requiring explicit rules and extensive human effort, makes neural networks especially useful in pattern recognition applications in which rules are too complex to define and maintain. In the case of interpreting power system alarms, where system configurations, alarm patterns, and system problems are difficult for each utility, this very ability to learn holds the key to the building of a generic intelligent alarm processor.

Another important characteristic of neural networks is the way information is stored. The memory of a neural network is both distributed and associative. A unit of knowledge, represented, for example, by an input/desired-output pair, is distributed across all memory units in the network; and it shares these memory units with all other units of knowledge stored in the network. Neural network memory is also associative, in the sense that, if the network is presented with a partial input, the network will choose the closest match in its memory, and generate an output which corresponds to a full input. This distributed and associative nature of neural network memory leads to reasonable response when presented with incomplete, noisy or previously unseen input. Because of these properties, neural- network systems are most suited to many pattern recognition tasks, more so than the traditional algorithmic or rule-based expert systems.

CONCEPTUAL APPROACH

Although neural networks have been used successfully in such applications as handwritten Japanese, English, French, and numeral character recognition, signature verification, radar target identification, machine-part identification, heart-beat monitoring and classification, etc., [8], [11], [12], their application in power system in general and in alarm processing in particular, has not been reported. However, the basic principles behind these applications can be used to build a neural-network-based Intelligent Alarm Processor (IAP) which will have the special properties and promising capabilities described in the previous paragraphs, namely:

- its ability to learn from examples, without the intensive effort required in defining explicit rules
- its ability to function with slightly different inputs or noisy inputs
- its adaptability by various utilities with minimal customization effort.

The concept of a neural-network-based IAP is based on the following conjectures:

1. For a power system in a given system state, a particular system problem will result in a particular pattern of alarms. Therefore, by correctly recognizing a particular pattern of alarms, the corresponding system problem which causes these alarms can be identified.

2. Most of the multiple-alarms situations involve large number of relay actions; and the relay protection scheme, once set, does not change day-to-day. Therefore, the expected relay actions can be viewed as the basic characteristics or recognizable "signature" of a system problem. [13]

3. The pattern of alarms will likely appear with "noise", possibly due to equipment problems, incorrect relay settings, or miscalibrated metering devices. In other words, the expected pattern of alarms may not always appear exactly as expected.

4. In cases where a definitive conclusion cannot be made due to noisy inputs, presenting a few probable causes of the alarms will still be extremely helpful to the system operators.

It should be observed that Item 1 above can be elaborated as follows: for a power system in different system states, the same system problem may result in different patterns of alarms. These different system states may include different system generation and load levels, and even different network configurations. In other words, a particular system problem (e.g., a bus fault) may manifest itself in different alarm patterns. The task of associating these different alarm patterns with the same system problem (e.g., a bus fault) is analogous to the character-recognition task of recognizing the different handwritten versions of a particular character written by different persons.

Based on these conjectures and observations, the basic concept in this neural-network approach for intelligent alarm processor can be summarized as follows:

1. Create a list of system contingency cases which reflect, as much as possible, all system contingencies that can occur.

2. For each system contingency, identify, through the relay protection scheme and load flow studies, the alarms which can to be expected under different system conditions and configurations. This constitutes the training set for the neural network.

3. Train the neural network with the training set.

4. The neural network will then be ready to interpret alarms by recognizing the incoming alarm patterns, with or without noise.

This concept is illustrated in Figure 3. It should be pointed out that it is likely that not all the system contingencies would be anticipated initially; and consequently, the neural network will not recognize such unanticipated contingencies. However, a neural network can "learn" incrementally when a new case is input.

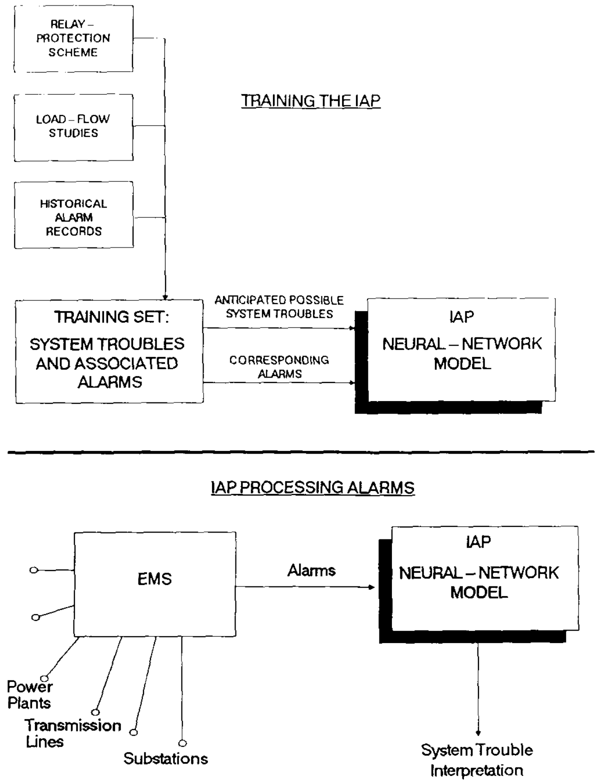


FIGURE 3: THE BASIC CONCEPT OF USING NEURAL-NETWORK FOR INTELLIGENT ALARM PROCESSING

PROOF-OF-CONCEPT TESTING

To prove the feasibility and viability of this approach, an experimental Intelligent Alarm Processor (IAP) was built by carrying out the following steps:

1. Construct a simple power system model for testing purposes.

2. Generate a training set consisting of input/output pairs of system problems and corresponding multiple-alarms. As mentioned above, this can be easily obtained by knowing the relay protection scheme and by running contingency study cases (multiple load-flow runs). The end results would be, for each possible system fault or trouble, the expected system alarms.

3. Enter the training set into the neural network to train the network.

4. Enter test sets of alarms to verify the corresponding system problems or system faults are correctly identified.

5. Introduce missing alarms as well as additional alarms to verify the network's ability to make correct or close identification in the presence of noise.

6. Observe the number of multiple alarms that can be replaced with one single IAP message diagnosing the system problem.

Two tests were performed. They are described in the following paragraphs.

Test Case 1

A 115 kV/12 kV distribution substation depicted in Figure 4 was used for the test. The substation has twelve 12 kV feeders and two banks tapping off a 115 kV circuit. (Breaker 162 is normally closed.) There is one spare 12 kV breaker, operating through the auxiliary bus, to facilitate feeder breaker maintenance. Since there are some customer-generation on the feeders, the feeders are tripped in case of a bus fault. For each feeder, the following are monitored:

- Breaker Position
- Momentary Detect (Open to Close)
- Phase Overcurrent (Relay Target)
- Ground Overcurrent (Relay Target)
- Timer (Relay Target)
- Instantaneous (Relay Target)
- Auxiliary (Connect to Aux. Bus)

Each feeder is equipped with a recloser at approximately the half way point. For the purpose of locating the fault, the section covering 90% of the length between the substation and the recloser is called Section A, and the rest of the feeder is called Section B.

By working with the utility staff engineer responsible for the relay protection of the substation, a list of system problems which could reasonably happen was generated; and for each of these system problems, the corresponding alarms which would be reported were delineated, as shown in Figure 5. An "X" means an alarm is detected; and a blank means no alarm is detected. Note that some system problems manifest themselves in more than one alarm pattern (e.g., 1101 Feeder Fault, depending on which breaker is serving Feeder 1101, and Bus Fault, depending on whether the two bus sections have been tied and if some breakers are already open).

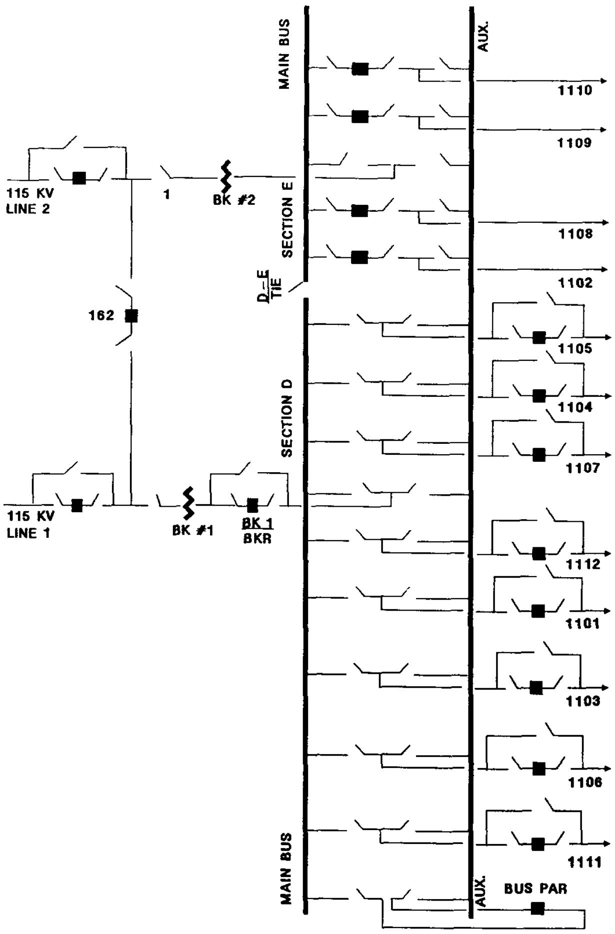


FIGURE 4: DALY CITY 115/12 kV SUBSTATION USED IN TEST CASE 1

SYSTEM PROBLEM #	1		2		3		4		5		6....		61			62	63	64	65
PROBLEM DESCRIPTION / ALARMS	FDR 1101 3Ø/L-L FAULT A	FDR 1101 3Ø/L-L FAULT A (1)	FDR 1101 L-G FAULT A	FDR 1101 L-G FAULT A (1)	FDR 1101 3Ø/L-L FAULT B	FDR 1101 3Ø/L-L FAULT B (1)	FDR 1101 L-G FAULT B	FDR 1101 L-G FAULT B (1)	FDR 1101 FAULT /CLEAR/RECLOSE	FDR 1101 FAULT /CLEAR/RECLOSE	FDR 1102 3Ø/L-L FAULT A	. . .	BUS FAULT (2)	BUS FAULT (3)	BUS FAULT (4)	XFORMR 1 FAULT	XFORMR 2 FAULT	LOSS OF 115 #1	LOSS OF 115 #2
1. 1101 BKR POS	X		X		X		X				X		X		X				
2. 1101 MOMENT'Y									X	X									
3. 1101 PHASE	X				X				X		X								
4. 1101 GRD			X				X			X									
5. 1101 TIME					X		X			X									
6. 1101 INSTANT.	X		X						X		X								
7. 1101 AUX, BUS		X		X		X		X											
8. 1102 BKR POS																			
9. 1102 MOMENT'Y																			
10. 1102 PHASE																			
11. 1102 GRD																			
12. 1102 TIME																			
13. 1102 INSTANT.																			
14. 1102 AUX. BUS																			
⋮																			
85. BUS// BKR POS.		X		X		X		X											
86. BUS// MOMENT'Y																			
87. BUS// PHASE		X				X													
88. BUS// GRD				X				X											
89. BUS// TIME						X		X											
90. BUS// INSTANT.		X		X															
91. BUS// AUX. BUS																			
92. BK #1 BKR													X		X	X			
93. BKR 162														X	X	X	X		
94. BKR 172																X		X	
95. BKR 182														X	X		X		X
96. 115KV LINE 1 MW																		X	
97. 115KV LINE 2 MW																			X
98. BK #1 MW																X			
99. BK #2 MW																	X		

NOTES
(1) Operated on Aux Bus
(2) Section D
(3) Section E
(4) Section D & E TIED

FIGURE 5: SYSTEM PROBLEMS AND ASSOCIATED ALARM FOR TEST CASE 1

These system- problems/alarms sets constitute the training set. The training set is then inputted into the IAP neural network as shown in Figure 6. After training the IAP neural network, each of the alarm patterns was input into the IAP.

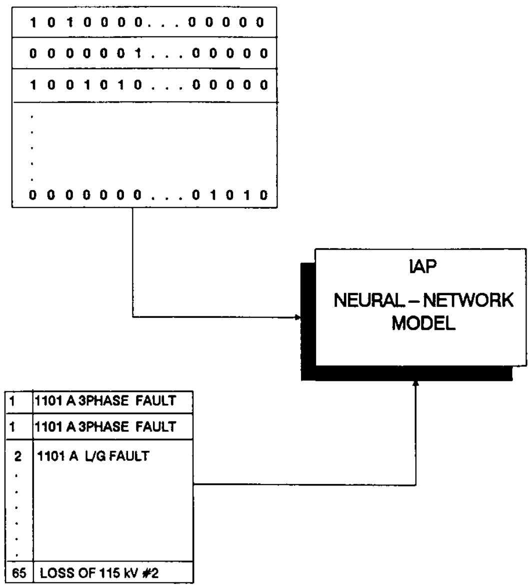


FIGURE 6 – TRAINING THE IAP IN TEST CASE 1

Test Case 2

A second test case was set up using the IEEE 30-bus model. Using the same approach as in Test Case 1, 72 bus-fault and line-fault cases were set up as the possible system problems in this 30-bus network. A set of 112-bit alarm patterns were generated, each of which represent the particular combination of reported circuit breaker trip, associated with a particular system fault. The IAP neural network was then trained with these input/output pairs of system faults and associated alarms.

Once trained, the IAP was tested using the alarm input patterns as well as "noisy" sets of alarms, made by randomly changing the status of a few alarms in each set.

Observation

1. The IAP correctly identified all cases of system problem in Test Case 1 and Test Case 2 when there is no noise (no missing alarm). Therefore, its performance is at least as good as the human system operator and a typical rule-based system when there is no erroneous alarms.

2. The number of alarm messages associated with each of these system problems ranges from two (e.g., loss of 115 kV line) to fifteen (e.g., bus fault). With the IAP, each system problem can be reported with only one message, diagnosing the problem.

3. When noises were introduced in the alarm input patterns (purposely omitting some alarm inputs), the following results were observed:

   a. The IAP still correctly identified the source problem in some cases.

   b. For those cases in which the IAP neural network was "confused", the IAP indicated the most likely source problems and the associated probabilities.

   c. The IAP is most successful in identifying those cases with associated alarm patterns which are significantly different than the other alarm patterns. A missing alarm in this pattern will not make it look like any one of the other patterns.

4. For some cases, a missing input would result in a wrong identification. However, it must be noted that a system operator, given such incomplete data, could reach the same wrong conclusion.

5. The "most likely" problems suggested by the IAP, in the event of missing data, may be totally unrelated in a power system sense. This is because the IAP only picks the most closely related alarm pattern.

6. The effort required to train the IAP is very reasonable. In carrying out the Test Case 1, the "knowledge engineering" required to identify the possible system problems and associated alarms took one hour, entering the data and preparing the training set took two hours, and training the neural network took ten hours on an IBM PC/386. Conducting the Test Case 2, including the generation of contingency cases and case runs, and the training of the neural-network, took approximately twenty hours.

7. Once trained, the time it took the IAP to recognize an alarm pattern (execution time) is in the neighborhood of one second on an IBM PC/386, which is fast enough for real-time application.

8. With special neural network hardware becoming available [14], [15], the training time and execution time can be further reduced.

CONCLUSION

The neural-network is a viable and very promising approach in building an intelligent alarm processor. This could be the basis for developing a generic intelligent alarm processor for easy implementation by all utilities.

When there is no missing alarm data, the experimental Intelligent Alarm Processor correctly interpreted 100% of the system problems.

When there is missing alarm data, and if such missing data will not result in an alarm combination identical to one associated with another system problem, the experimental IAP successfully made the right guess. This is similar to the behavior of a human system operator.

In the test cases, a very simple alarm bit-pattern structure was used as the identification pattern. This was chosen because of implementation considerations: such alarm bit-patterns are easily obtainable from a typical Energy Management System. However, pattern recognition based solely on such bit-pattern structure is susceptible to missing data. Although a human system operator is also susceptible to the same missing alarms and may draw the same wrong conclusions, improvements can be made.

Further work can be done to characterize the system problems/contingencies. Such characterizations include the order the alarms are reported, the magnitude of the limit violations, the values of certain important system parameters, and the behavior of the alarms over a certain time period. A network of several neural networks may be constructed to capture the different system problem characteristics and the time-sequential significance of the alarms to draw more definitive conclusions.

ACKNOWLEDGEMENT

The author would like to express his appreciation for the diligent and enthusiastic effort by two ECC colleagues, Martin G. Delson and Frances M. Cleveland, in setting up and the making the test case runs. The author would also like to thank Mr. Jim Flanagan of Pacific Gas & Electric Company in serving as the "expert" for Test Case 1.

REFERENCES

[1] Rod Griffith. "The Alarm Excess Problem: The User's Viewpoint", The 9th Biennial Real-time Monitoring and Control Workshop, October 1986.

[2] W. Prince, B. Wollenberg, D. Bertagnolli. "Survey on Excess Alarms", IEEE/PES Winter Meeting 1989.

[3] H. Amelink, A. M. Forte, R. P. Guberman. "Dispatcher Alarm And Message Processing", PICA, 1985.

[4] "Combinatorial Sensing Capability", Chapter from EPRI Report EL-81-5-LD, April 1981.

[5] Bruce Wollenberg. "Feasibility Study For An Energy Management System Intelligent Alarm Processor", PICA, 1985.

[6] Sarosh N. Talukdar, Eleri Cardozo, Ted Perry. "The Operator's Assistant-An Intelligent, Expandable Program For Power System Trouble Analysis", PICA, 1985.

[7] F. Hein. "Summary of CIGRE-Workshop on Alarm Handling in Control Centers", Paris, September 1986.

[8] "Nerves of Silicon", Discovery Magazine, February 1988.

[9] Stephen Grossberg. "Neural Networks and Natural Intelligence", MIT Press, 1988

[10] David Rumelhart, James McClelland, and the PDP Reasearch Group. "Parallel Distributed Processing", MIT Press, 1988.

[11] Teuvo Kohonen. "The Neural Phonetic Typewriter", Computer, Vol. 21, No. 3, March 1988.

[12] Kunihiko Fukushima. "The Neural Network for Visual Pattern Recognition", Computer, Vol. 21, No. 3, March 1988.

[13] F. Hein. "Expert System Using Pattern Recognition by Real-Time Signals", CIGRE Paper SC39-10, 1986.

[14] R. M. Kuczewski, M. H. Meyers, W. J. Crawford. "Neurocomputer Workstations and Processors", ICNN Conference Proceeding, 1987.

[15] Hans P. Graf, Lawrence D. Jackel, Wayne E. Hubbard. "VLSI Implementation of a Neural Network Model", Computer Vol. 21, No. 3, March 1988.