ConF-

CONF-901155--2

DE91 002868

APPLICATIONS OF NEURAL NETWORKS IN CHEMICAL ENGINEERING - HYBRID SYSTEMS

Juan J. Ferrada
Paul A. Grizzaffi **
Irvin W. Osborne-Lee

Chemical Technology Division
Oak Ridge National Laboratory*
Oak Ridge, Tennessee 37831

**University of Southwestern Louisiana,
Lafayette, Louisiana 70404

To be presented at the National AIChE Meeting, Chicago, IL, November 12-16, 1990.

*Managed by Martin Marietta Energy Systems, Inc., under contract DE-AC05-84OR21400 with the U.S. Department of Energy.

Applications of Neural Networks in Chemical Engineering - Hybrid Systems

Juan J. Ferrada, Paul A. Grizzaffi, and Irvin W. Osborne-Lee

INTRODUCTION

Expert systems are known to be useful in capturing expertise and applying knowledge to chemical engineering problems such as diagnosis,[1] process control,[2] process simulation,[3] and process advisory.[4] However, expert system applications are traditionally limited to knowledge domains that are heuristic and involve only simple mathematics. Neural networks, on the other hand, represent an emerging technology capable of rapid recognition of patterned behavior without regard to mathematical complexity. Although useful in problem identification, neural networks are not very efficient in providing in-depth solutions and typically do not promote full understanding of the problem or the reasoning behind its solutions. Hence, applications of neural networks have certain limitations.

This paper explores the potential for expanding the scope of chemical engineering areas where neural networks might be utilized by incorporating expert systems and neural networks into the same application a process called hybridization. In addition, hybrid applications are compared with those using more traditional approaches, the results of the different applications are analyzed, and the feasibility of converting the preliminary prototypes described herein into useful final products is evaluated.

BACKGROUND

Advantages of Hybrid Systems

Manipulation of knowledge at the same level as the human expert has been an important need in many computer applications during recent years. Artificial intelligence (AI) tools such as expert systems, neural networks, and hybrid systems make this endeavor possible by emulating human expertise. An expert system typically consists of three distinct components: (1) a knowledge base (frequently a set of rules) containing all of the information relevant to a specific problem area; (2) an inference mechanism that performs reasoning that utilizes the knowledge base to resolve a given problem; and (3) a user interface that accesses information from the user and/or data bases and then provides solutions to the problem. Expert systems are best used to apply a fixed set of logical rules and related facts to a specific-domain problem.[5] Although very effective, expert systems have been less successful in applications that require the processing of raw sensory data in a way that is flexible and robust enough to deal with the real, unpredictable world.

A neural network can be defined as a computing system made up of a number of simple, highly interconnected elements that process information through dynamic responses to external inputs.[6] Information is processed by determining the value of an output signal based on the values of several input signals. After knowledge has been learned, storage is dependent on the way the neurons are connected and the way the individual neurons adjust their weights as the network learns.[7]

Two main elements constitute a neural network: processing elements and interconnections. Processing elements are the equivalents to neurons that receive the input signals and are then modified by a transfer function in order to generate an output signal. Figure 1 shows the scheme of a neural network system. The first and last layers of a network are called the input and output layers, respectively. Middle layers are referred to as hidden layers. The output signal of a particular processing element is sent to many other processing elements as input signals via the interconnections between them.

Many areas of engineering, such as fault detection and diagnosis, process control, process design, and process simulation, could benefit from systems that adapt to and learn from raw process data. Neural networking is an emerging computer discipline that seems promising for organizing and detecting patterns from unpredictable and/or imprecise inputs. Neural networks are considered clever and intuitive, primarily because they learn by example rather than by following programmed rules. Different from expert systems, a neural network does not require the user to specify a number of "if then" rules; it needs only specific examples of input values and the corresponding output values. The user normally teaches the network how to adapt to a set of specific examples while the network itself determines patterns that work for specific examples.

Both approaches to AI offer advantages and disadvantages. Expert system technology has proved to be very successful in solving problems where rules for decision-making are clear and information is reliable. Expert systems excel in explaining their lines of reasoning and producing precise recommendations; however, they become impractical when confronted with situations where heuristics are unclear or questionable. On the other hand, neural network software is now acknowledged as a viable means for reaching conclusions in situations where explicit decision rules are obscure or nonexistent, as well as in cases where information is only partially correct. Neural networks classify and associate to reach likely solutions learning from a set of examples and classifying new cases according to the similarities associated with the "memorized" patterns. However, they lack the capacity to explain their conclusions and are unable to reason in a sequential manner that results in precise conclusions.

From the above advantages and disadvantages, it is clear that there is substantial potential for synergism between the two reasoning mechanisms if, on combination of the latter, many of the weaknesses inherent in each method can be avoided and their unique strengths can be maintained or enhanced. If practical, then, hybrid systems show considerable promise as powerful tools for future applications in engineering.

Applications of Hybrid Systems

Indications of significant work in hybrid AI application research and development have been noted in the literature, although hybrid applications have not yet become widespread. Eaton Corporation (from Milwaukee, Wisconsin), a company with an interest in long-range trucking services, has developed a truck-brake-balancing application using neural networks and a knowledge based system to balance the braking force over all five axles. Brake balance has always been a factor in safety as well as in improvement of the economy of trucking operations. The system uses a suite of five neural networks and three knowledge bases, including diagnostic and preprocessing knowledge bases to render an analysis and recommendations. Neural networks with a back-propagation algorithm and expert shells with a feed-forward system were used. The system worked with plotted data that were analyzed by the neural network and established "good" or "bad" features of the brake system. The expert could look at a plot of data points from a brake drum temperature reading and determine whether a problem existed. Scans of graphs used to make these determinations formed the basis for the training sets for the neural networks--a procedure similar to that adapted for in this work. The expert system gave recommendations based on the result of the plot analysis.[8]

Hybrid systems are already being tested in robotics; one project in the developmental stage implements this technique at Oak Ridge National Laboratory. The system employs a novel self-learning feedback mechanism that allows the expert system to supervise the learning of the neural network, while also allowing the neural network to teach the expert system new rules.[9]

HNC, Inc. (a San Diego-based neurocomputing company), is developing a system that combines the learning capabilities of a neural network with the functionality of an expert system.[10] The result, Knowledge Net, is currently developed for a financial application (loan approval recommendations) but is being commercialized for more general use. Undoubtedly, other research and development efforts in hybrid applications also exist.

BUILDING SIMPLE HYBRID SYSTEMS

Scope

Building a simple hybrid expert system/neural network application is not difficult, although it is somewhat more involved than using the simplest expert system or neural network shells. Certainly, hybrid systems can be quite complex, and building a system such as that currently under development by Glover and co-workers is a major undertaking, involving complicated design and programming. However, our approach is to show that much can be accomplished with expert shells and little programming. The applications described in this paper are intended to demonstrate the potential for expanding the scope of application areas of neural networks and expert systems in chemical engineering through operation and maintenance tasks using simple tools such as the shells described in the succeeding sections.

This paper focuses on chemical engineering applications with NeuroShell™ from the Ward Systems Group and VP Expert™ from Paperback Software. The selected neural network software simplifies the complexity of neural network application development by reducing the amount of effort required for setting up the network and is adapted to PC type computers. The following three layers constitute the network: input nodes, hidden nodes, and output nodes. NeuroShell limits the user to back-propagation networks of one hidden layer; however, in normal practice, three layers are usually sufficient. With three layers, the weight modification must be made not only between the second and third layers but also must be propagated back to the first and second layers.

Application Tools

The mathematics involved in neural networks, although not highly complicated, may represent an initial hurdle in programming for the novice in this field. Fortunately, commercial software packages are available. Most of the packages have no complicated systems generation or higher-level programming requirements. Packages that facilitate construction of networks without the programming complications are available for those with more experience in neural networks; however, some knowledge of how the different learning rules interact with their respective parameters is required. In any event, commercial packages permit the knowledge engineer to concentrate on the problem and to determine how neural technology can be applied to solve it, rather than on the mechanics of the network. Similarly, for expert systems, the use of a shell allows the knowledge engineer to focus on developing the logic that leads to the solution without much concern for the method by which logic can be programmed effectively.

After the neural network has been created and has learned the given sets of inputs and the desired output for a particular problem the user should need only to input a set of data to receive information that would permit solution of the problem. Accomplishment of this task requires easy keyboard entry of data points, an interface between the neural network package and an expert program, and a control program--in our case, a simple batch file. The expert program can be implemented by using an expert shell as mentioned earlier, or a programming language, such as C. In the simple hybrid systems developed here, the expert program role is to access the output from the network and provide a solution to the problem. The control program's job is to manage the interaction between the network, the expert system, and any support files or programs in a manner that allows the user to execute the entire package with only a few simple commands. The control programs used here were written in the DOS (Disk Operating System) batch programming language. Both applications were developed on IBM™-compatible personal computers using DOS.

ACTIVATED-SLUDGE TROUBLESHOOTING GUIDE

NeuroShell and VP-Expert (a commercial expert shell) were used to create an activated-sludge troubleshooting guide for a sanitary waste treatment system. During normal operations, the treatment system, consisting of a filter, a reactor, and a clarifier, receives waste through the filter and releases a clear discharge from the clarifier into a river or stream. Although the reactor controls the system, the signal denoting the status of the clarifier is the mechanism that alerts the operator of a potential problem within the reactor. Therefore, careful monitoring of the clarifier output is required. The operator can determine whether a problem exists by observing the results of settling tests on the clarifier and the appearance of the clarifier. If a problem exists, the activated-sludge troubleshooting guide can be used to identify it, provide the operator with the probable causes, and recommend the most likely solution.

A NeuroShell binary network was set up to identify a potential problem within the treatment system. A binary network was used since the data used to classify each case do not take on specific numerical values but, rather, may be described by a series of "yes" or "no" ("on" or "off") values.

To identify a problem within the system, the operator must classify a new case by entering the appearance of the clarified effluent and the results of the settling test. For each of the inputs, the operator decides if the characteristic is true ("1") or false ("0). The network then classifies the new case and responds with a "1" or a "0" for each of the output nodes. The expert system, developed using VP-Expert, can provide a solution to the problem by interfacing with the NeuroShell network.[11] The neural network classifies the specific case and passes a number to the expert system via a text file. Based on this number and the rules in its knowledge base, the program reads a specified text file that provides the operator with the problem diagnosis matching the number and then deduces the possible causes of the problem, decides what needs to be checked, and provides the most likely solution to the problem.

ROTATORY SYSTEM FAULT DIAGNOSIS

The objective of this application is to perform fault diagnosis of a simple rotatory apparatus for simulation of vibration troubleshooting. The system under analysis was a journal-bearing set up equipped with sensors (see Fig. 2) for which vibration signal tracings are available. In the application presented here, these vibration signal plots are scanned, analyzed by a neural network, and the findings passed on to an expert system for interpretation.[12] The journal-bearing apparatus was assembled using an eccentric movable mass on the shaft to create a vibration and thereby simulate different vibration faults. The vibration signal data were generated at the U.S. Naval Academy.[12]

As a machine proceeds through its service life, its parts will wear and eventually fail. Machine wear can be the result of rounding of sharp edges, shaft misalignment, damaged or loose bearings, and operations at or near a rotor whirl. During operation, these wear factors will cause an increase in the vibrational energy of the machine. This energy will be dissipated throughout the machine, placing an increase in dynamic loading on other components--especially bearings. For example, looseness may develop in journal bearings, causing increased vibrational energy in the shaft. The source of the problem can usually be determined by the change in frequency of the equipment under analysis. Among others, faults in equipment include:

- rotating member out of balance,
- misalignment and bent shaft,
- damaged rolling-element bearings,
- mechanical looseness, and
- electrically induced vibrations.

This system was tested at the U.S. Naval Academy, and frequency data such as those shown in Fig. 3 were fed in a digitized manner to a neural network. Because of the similar power spectral densities with different eccentric displacements, the neural network system had difficulty in making distinctions between the different spectra.[12]

Drawing on the Naval Academy project experience, ORNL used a slightly different approach to the problem. Rather than feeding the scan data directly to the network, this approach incorporates a preprocessing algorithm for the scan data. In scanning the frequency charts, there is a possibility of incorporating more information than in the digitized point-by-point method. However, the amount of information captured by the scanner can be overwhelming and, when fed to the neural network, can produce a very slow and limited learning process. Our network would accept only 4 cases, with a learning time of 48 hours, on a 20 mHz 80386-based personal computer. Consequently, the algorithm was devised to select significant information for input to the neural network.

The algorithm is based on sampling the frequency chart only where most of the useful information was located. This screened information, now digitized, is then read by tracing 4 parallel lines, like the ones illustrated in Figure 4, to perform a second sieving of data. The four lines are customized for the system in such a way that most of the important data points were considered. The data search algorithm produces 1076 data points per frequency chart, which are fed to the neural network.

The preprocessing steps are effective--the learning process for 6 known cases takes about 5 hours, after which the learning process for the system has been able to classify successfully to within 80 percent of certainty.

The neural network was linked to an expert program written in the C programming language. This choice was made for several reasons. First, the use of an expert shell, such as VP-Expert has been demonstrated in the first application. In this application, programming the simple expert system in C provided more flexible screen graphics and interfacing than the shell. In addition this option resulted in a compiled program that ran faster and eliminates one licensing requirement for distribution.

PROCEDURAL INSIGHT

The diagnostic process requires several steps, as shown in Fig. 5. First, the plot derived from the frequency chart is subjected to a scanning device. For the sake of uniformity, the output file created by the scanner is then sized (i.e., reduced) to a standard size by the user/operator. [Note, however, that this size is not exact; the standard size is precise only to within 2000 bytes of data.] Of the scanner formats available, MS windows format proved to be the most convenient; hence scan data are saved under the ".msp" format. The scan file contains the text representation of the graphics data read by the scanner. At this point, the ".msp" file is sent to a conversion program, written in C, which prepares the data for input to the preprocessor algorithm.

The conversion program first skips the initialization and title portion of the scanner file. Next, it reads the file, one character at a time. Each character is then converted into its ASCII decimal code, which is transformed into a text representation of the bits using binary numbers. (Bits are binary pieces of information that translate into "1" for "yes" or "0" for "no".) Initially, this text representation was sent to the neural network for classification; however, such an approach eventually proved to be infeasible because of excessive learning time and memory requirements. This situation prompted the development of an algorithm that captures only the most vital points in each text representation file. This algorithm is analogous to actually drawing four lines, each parallel to the other, at specific positions on the plot. An example of this search algorithm is illustrated in Fig. 4. The values of the individual "bits," which lie on each line and represent interceptions and noninterceptions, are then written to a file that can be sent to the neural network for classification. Since each line is 269 "bits" long and there are six possible classifications of faults, the neural network is made up of 1076 input neurons and 6 output neurons.

Because of a memory limitation, the network could only employ 35 neurons in the hidden layer. This limitation is due to the commercial shell, which can only access conventional memory (640,000 bytes) and not the extended or expanded memory available with the 80386 processor.

Each classified case represents a specific maintenance problem with the rotatory equipment. For example, when a new case is presented to the system and classified as closer to case 1 previously learned, it signifies that the vibrations may come from a worn bearing or a slight misalignment of the shaft. The cure for this problem is to accurately check the bearings and shaft and replace the worn parts. Every output of the neural network represents a different rotatory equipment fault.

Before sending the newly transformed file to the neural network, the network must be trained via solution of known sample cases. The data for these sample cases are presented to the network for learning by using the same process as described for converting test cases. Once the network has been trained and the problem case has been sent for classification, the network makes its best attempt at classifying the problem case according to what the network has already "learned." The result of this classification is then sent to an expert system, written in the "C" programming language, which issues recommendations according to the type of fault reported by the neural network.

The user is shielded from some of the previous steps. By using a batch file, he/she does not need to enter the operating system (DOS in this case) after the scanning/sizing procedure. (Currently, the user must first scan and size the data to be processed. In future applications, however, the sizing step may be eliminated by the use of a flat-bed scanner.) The user then invokes the batch file, which performs the conversion of the scanned chart to text representation and "draws" the lines through the data, writing the results to an output file. Next, the neural network is called to classify the processed file. The user selects the proper menu item, and classification occurs. Once the neural network is exited, the batch file sends the output file to the expert system for recommendations. An example of a consultation process is shown in Figs. 6 and 7. Figure 6 shows the interaction process required for inputting the scanned file for the vibration chart (the user supplies the name of the scanned file). The scan data are then converted to binary format, and an output file is produced and submitted to the preprocessing program, which will apply the four-parallel lines search algorithm to the converted scan file. Next, the system will access the NeuroShell package and request a classification.

Figure 7 shows the step that the system requires to exit the NeuroShell package once a classification has been made, followed by the interaction process to determine the appropriate recommendations to solve the fault problem.

The expert system will interact with the user and, in some cases, will request additional information from the user to establish an appropriate set of recommendations. Other cases will require the user to take some action and return to the program for more suggestions regarding improvement of the rotatory machine if faults are detected.

The batch file has been incorporated into this package to reduce the chance of error by an experienced user, as well as to keep the operation simple (and, therefore, attractive) for the novice user. The batch file itself is simply a means to execute several DOS commands by invoking only one file. This particular batch file contains calls to (1) the conversion/rescanning program, (2) the NeuroShell binary with the problem file "curve," and (3) the expert system. A complete, user-friendly software package has been developed through the use of this file. Figure 8 illustrates the files and procedures accessed by the batch file.

RESULTS

This paper has presented hybrid applications combining a neural network commercial package, or "shell", with an expert system. The engineering examples demonstrate the feasibility of building hybrid systems of neural networks and expert systems.

The troubleshooting guide, discussed as an example, clearly demonstrates that a neural network and a conventional expert system can be used to complement each other. The neural network adds speed, plus the capability to handle fuzzy or imprecise information. The expert system, handling the interpretation of the diagnostic results and providing recommendations, lets the system retain the capability for giving explanations to the user.

The most time-consuming steps in building a hybrid system of this type are those of training the network and building the knowledge base. In this application, the network took 2 to 3 hours to learn the sample cases. Moreover, achievement of the desired error level often requires adjustment of network learning parameters, followed by another training session for the network. The time required to develop the knowledge base depends greatly on the complexity of the logic required to solve the problem. In this application, the logic involved in interpreting network output and providing intuitive recommendations was not complex; construction of the knowledge base required only 1 day.

The fault diagnosis example for the rotatory machine shows how two different types of AI tools can be combined to provide a more complete solution to a problem that either tool alone could handle only partially. The neural network, in this example, can only diagnose the nature of the problem causing the measured vibrations, but this cannot be accomplished by the expert system alone. It has been shown that the network can accommodate different types of data without degrading the accuracy of its answers. On the other hand, the expert system is required to convert the diagnosis to a recommended course of action--a substantial time-saving feature of the system.

Also, additional tools (e.g., external programs) can be folded into such an integrated application, removing many limitations of the stand-alone expert system or neural network. Thus, the integrated nature of a hybrid system has merits beyond the simple combination of network and expert system features.

For example, in digitizing power spectrum density information (magnitude versus frequency), the neural network was not able to make a clear distinction from one case to another. However, after receiving information produced by a scanner, the neural network was able to produce a much more accurate classification. The major difficulty in applying this method is that too much information is created when a scanner is used. Every pixel is recorded by the scanning process, and many of these pixels represent blank spaces that add no useful data for the classification step. The magnitude of the neural network under these conditions grows enormously (25,000 inputs, for example), the learning time for such a neural network is excessive (48 h) and nonpractical, and the number of examples to learn (only four) is low. The solution of customizing the problem, by eliminating the region of non-useful baseline data and then applying the algorithm of four parallel straight lines to the curve, has proved to be very satisfactory for classification purposes. However, this remedy could not have been accomplished without integrating an external program into the application an option made possible through hybridization.

CONCLUSIONS

Hybrid systems involving both types of AI tools are beginning to appear more frequently in the engineering world. The creation of a hybrid system appears to be a solution that capitalizes on the advantages of both neural networks and conventional expert systems. The future application of these computer tools in the chemical engineering field is both a promising and an innovative area.

Both of the applications presented here, the activated-sludge troubleshooting guide and fault diagnosis in rotatory equipment, illustrate the successful integration of a neural network and an expert system into useful tools. The commercial packages selected for use in this work are simple, inexpensive, and easy to learn. Nevertheless, the neural network shell presents a few limitations in comparison with conventional expert systems. Networks will not access other data bases and combine with other external procedures as easily as will conventional expert systems. A trace of the reasoning that the network follows to arrive at a specific conclusion is not possible with neural networks. The combination of networks with expert systems alleviates these problems. Addition of a control program, moreover, has been observed to improve flexibility, capability, and user-friendliness.

ACKNOWLEDGEMENT

One of us (P.A.G.) gratefully acknowledges support from the Department of Energy, Division of University and Industry Program, Office of Energy Research, as a Science and Engineering Research Program participant.

REFERENCES

1. T. S. Ramesh and J. F. Davis, "CATCRACKER: An Expert System for Process and Malfunction Diagnosis in Fluid Catalytic Cracking Units," Annual Meeting of the American Institute of Chemical Engineers (AIChE), San Francisco, CA, November 1989.

2. J. E. Bassler, "Expert System for Control of Plant Operations in Real Time," Spring Meeting of the AIChE, New Orleans, LA, March 1989.

3. J. T. Robinson and P. J. Otaduy, "An Application of Object Oriented Programming to Process Simulation," Summer National Meeting of AIChE, Denver CO, August 1988.

4. J. J. Ferrada and B. R. Rodgers, "Application of Expert System to Hazardous and Radioactive Waste Management," DOE Model Conference, Oak Ridge, TN, October 1988.

5. J. C. Hoskins and D. M. Himmelblau, "Neural Network Models of Knowledge Representation," in Process Engineering, Computers and Chemical Engineering, Vol. 12, No. 9/10, pp. 881-90, 1988.

6. M. Caudill, "Neural Networks Primes," Part 1, AI Expert, December 1987.

7. C. W. Engel and M. Cran, "Pattern Classification. A Neural Network Competes with Humans," PC AI (May/June 1990).

8. A. Beard, "Marking Hybrid Systems," AI Week, Vol. 7, No. 11, 1990.

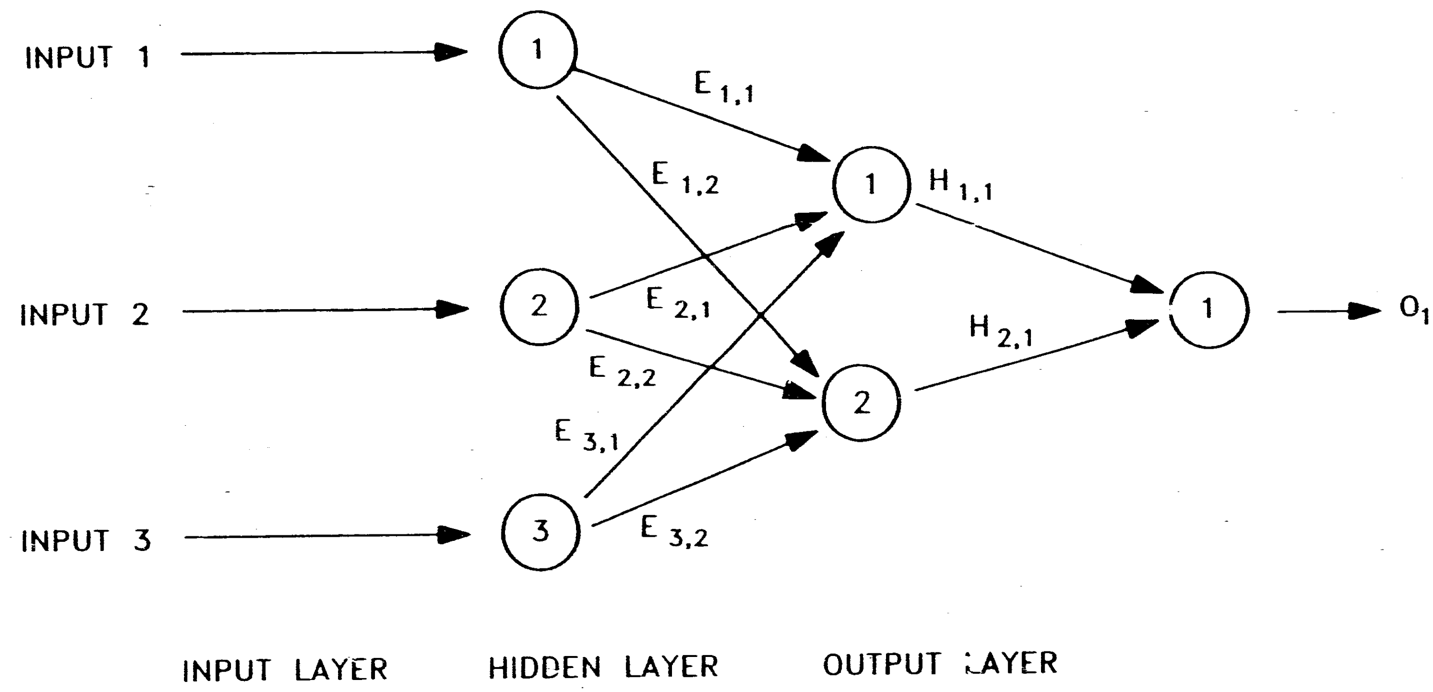
9. C. W. Glover and P. Spelt, "Hybrid Intelligent Perception System. Proposal for Seed Money," Oak Ridge National Laboratory, Oak Ridge, TN, 1988.

10. PC AI, "Knowledge Net, HNC's Neural Net/Expert System Symbiosis," PC AI, September/October 1990.

11. J. J. Ferrada, N. Cucaizella, and I. W. Osborne-Lee, "Expanding the Power of Expert Systems Application with Neural Networks," National AIChE Meeting, San Diego, CA, August 1990.

12. M. Gordon, U.S. Naval Academy, Personal Communication, 1990.

ORNL DWG 90A-1336



$E_{I,J}$: TRANSFORMED INPUT TO INPUT LAYER FROM NEURON I TO NEURON J

$H_{I,J}$: TRANSFORMED INPUT TO HIDDEN LAYER FROM NEURON I TO NEURON J

O_I: OUTPUT FROM NEURON I

Fig. 1. Schematic conceptualization of a neural network.

ORNL DWG 90A–1335

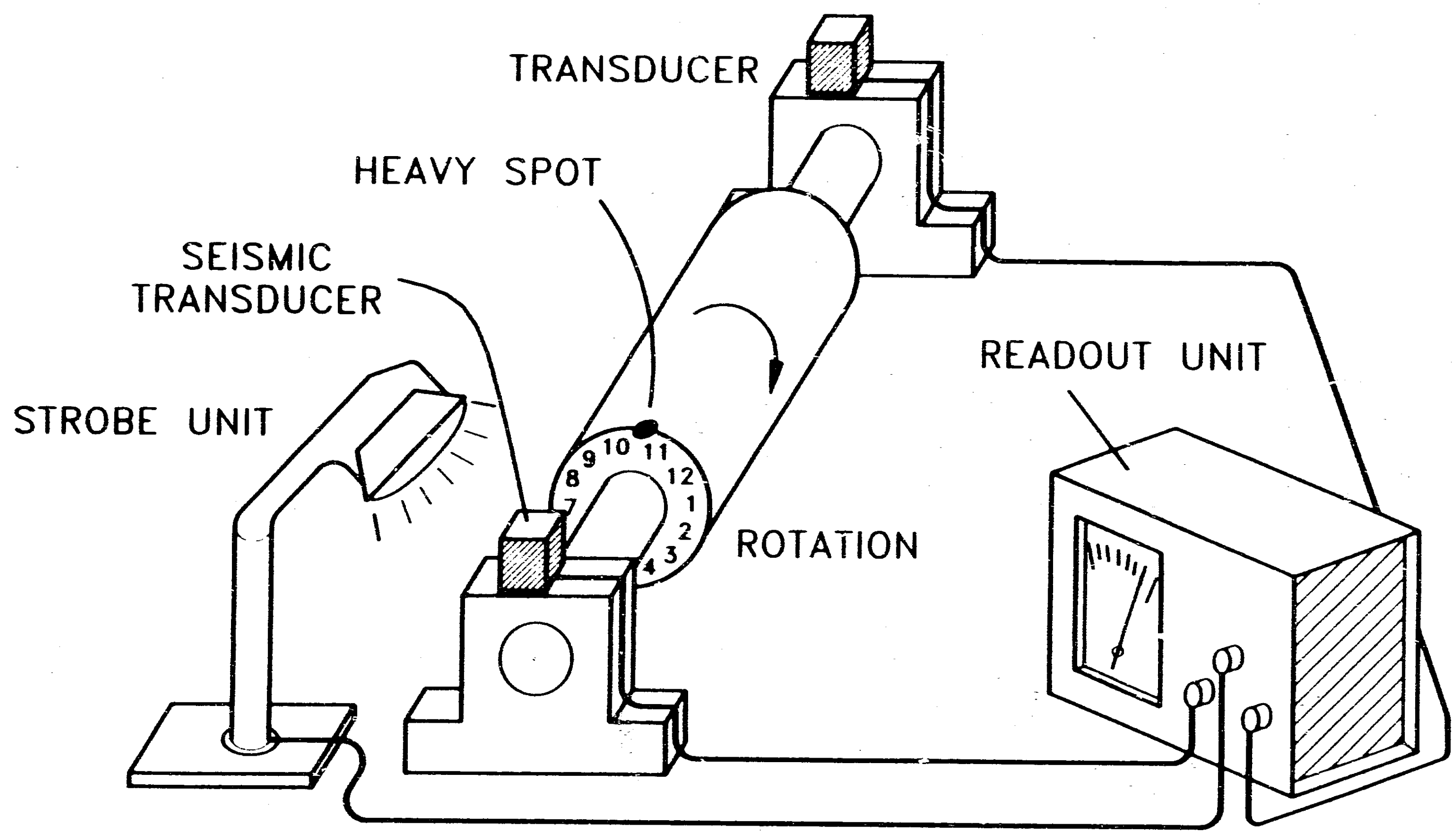


Fig. 2. Rotatory equipment equipped with sensor outputs suitable for input to a neural network.

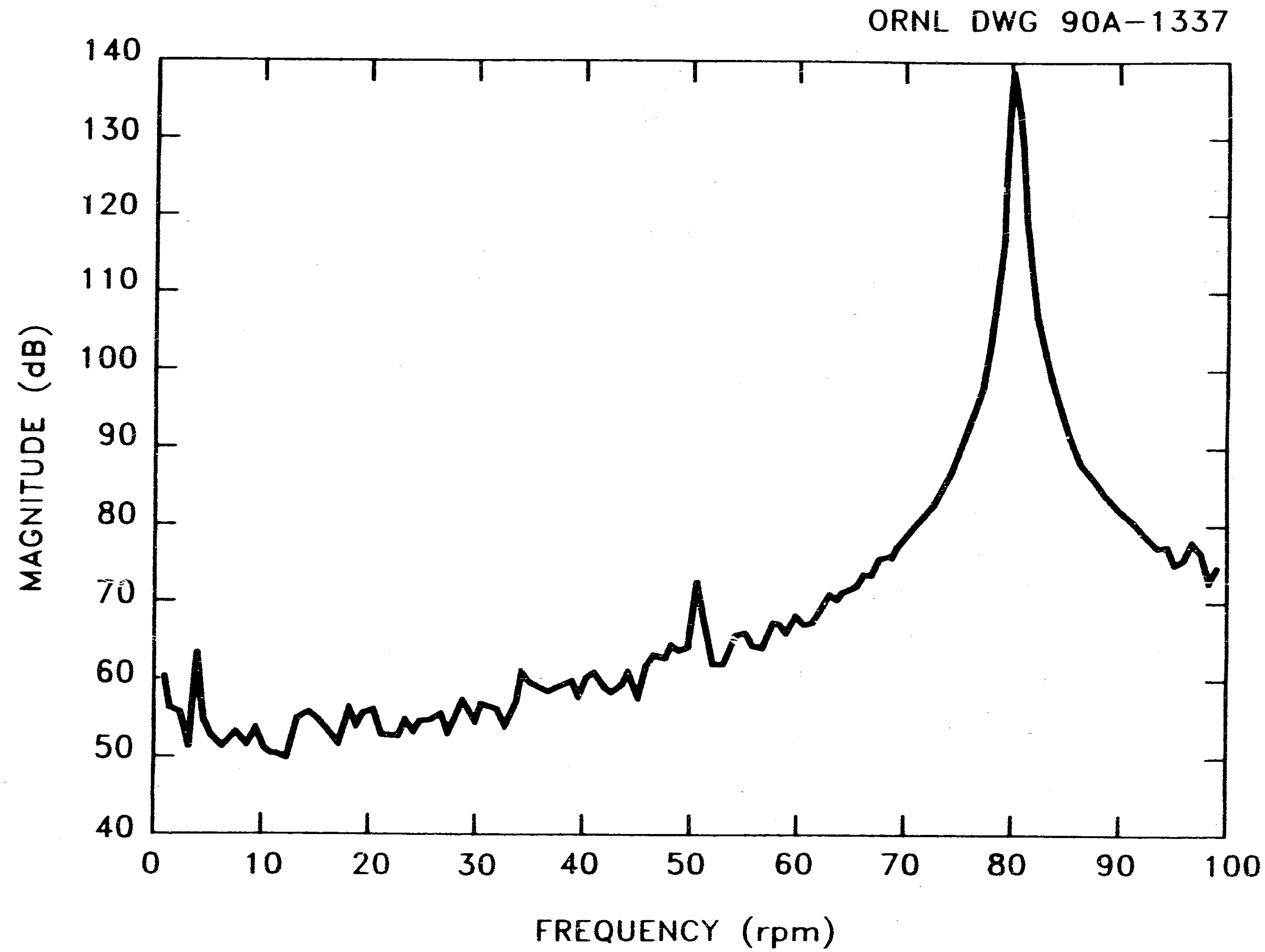


Fig. 3. Power spectrum density for fault diagnosis of rotatory equipment.

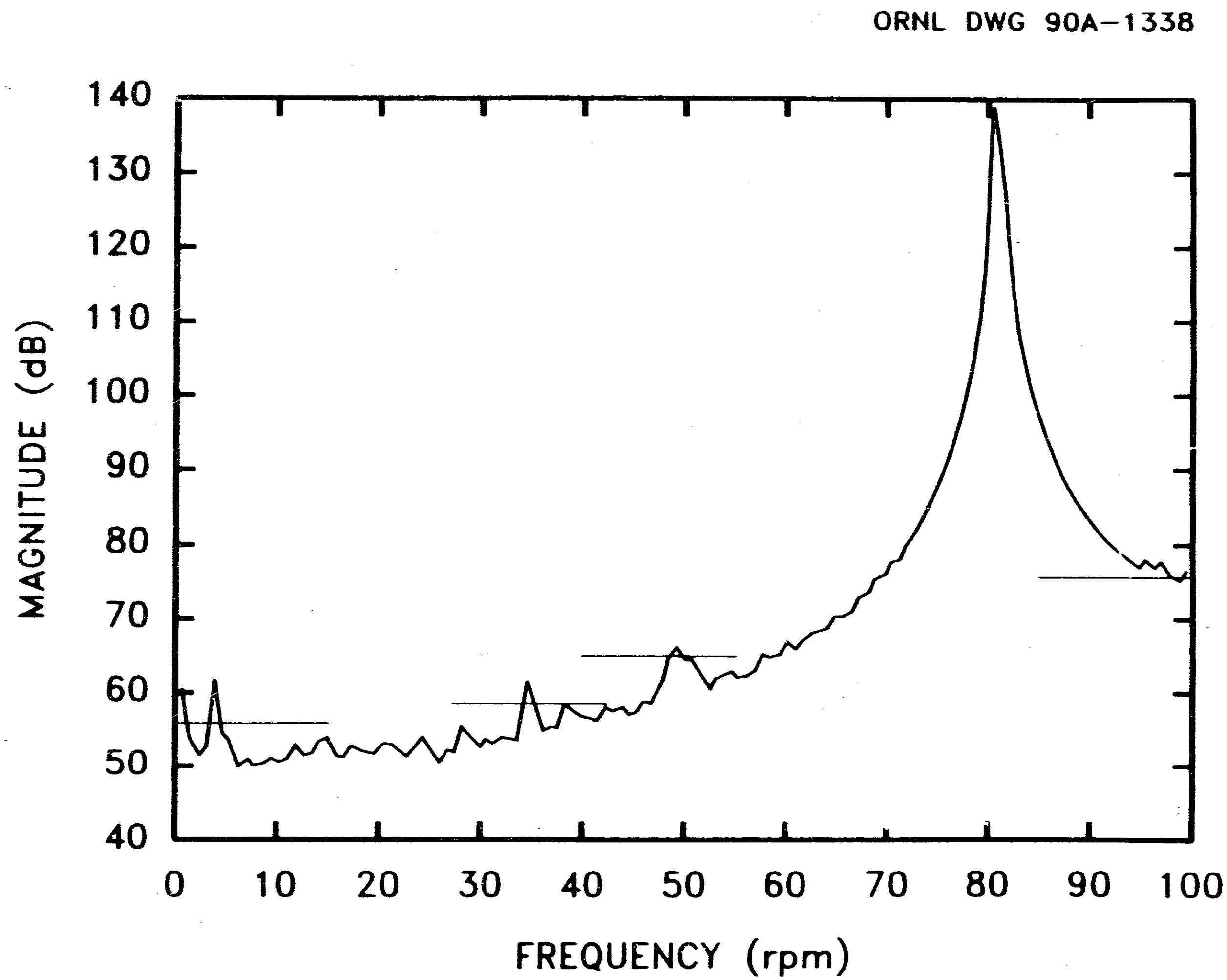


Fig. 4. Limiting the sensory data to make the "learning" process by the neural network more manageable.

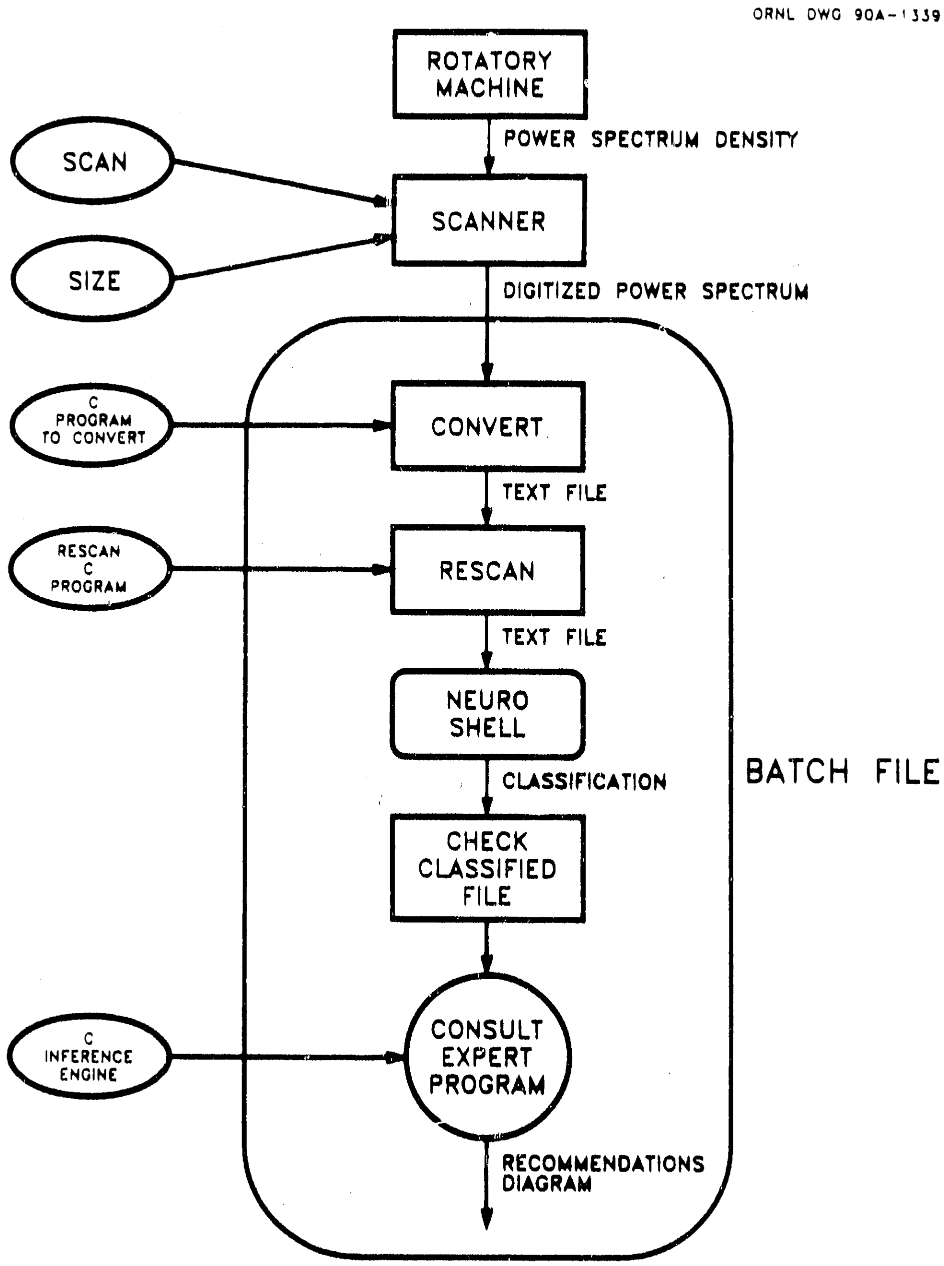


Fig. 5. Hybridized neural network/expert system programs linked by the batch file.

ORNL DWG 90A-1340

```
*********************************************
*** Scanner File Conversion Program ***
*********************************************
```

```
Enter the name of the file to be converted
(minus the file extension) pttst
```

```
INPUT FILE:  pttst

CONVERSION FROM ASCII TO BINARY...PLEASE WAIT
```

```
INPUT FILE:  pttst
OUTPUT FILE:  curve.ats

RESCANNING IN PROGRESS...ONE MOMENT PLEASE
```

```
Memory available is 70,590 bytes = 68k; problem is curve
```

```
=============== Main Menu Options ===============
  Define the characteristics...
  Enter characteristics for sample cases...
  Learn the sample cases (develop a model of the problem)
  Classify new cases according to the problem model...
  Print the characteristic definitions
  Print the characteristics of sample cases
  Print the characteristics of classified cases
  Select from Advanced Options menu...
  Quit this program
```

```
Case number: 1

=========== Identify Characteristics to Classify ===========
  1
  2
  3
  4
  5
  6
  7
  8
  9
  10
  11
  12
  13
  14
  15
  16
  17
  18
  19
  20

Esc-main menu   ↑↓-scroll   ← 2 -clear   F3-reclassify
```

Fig. 6. Classification of scanned information.

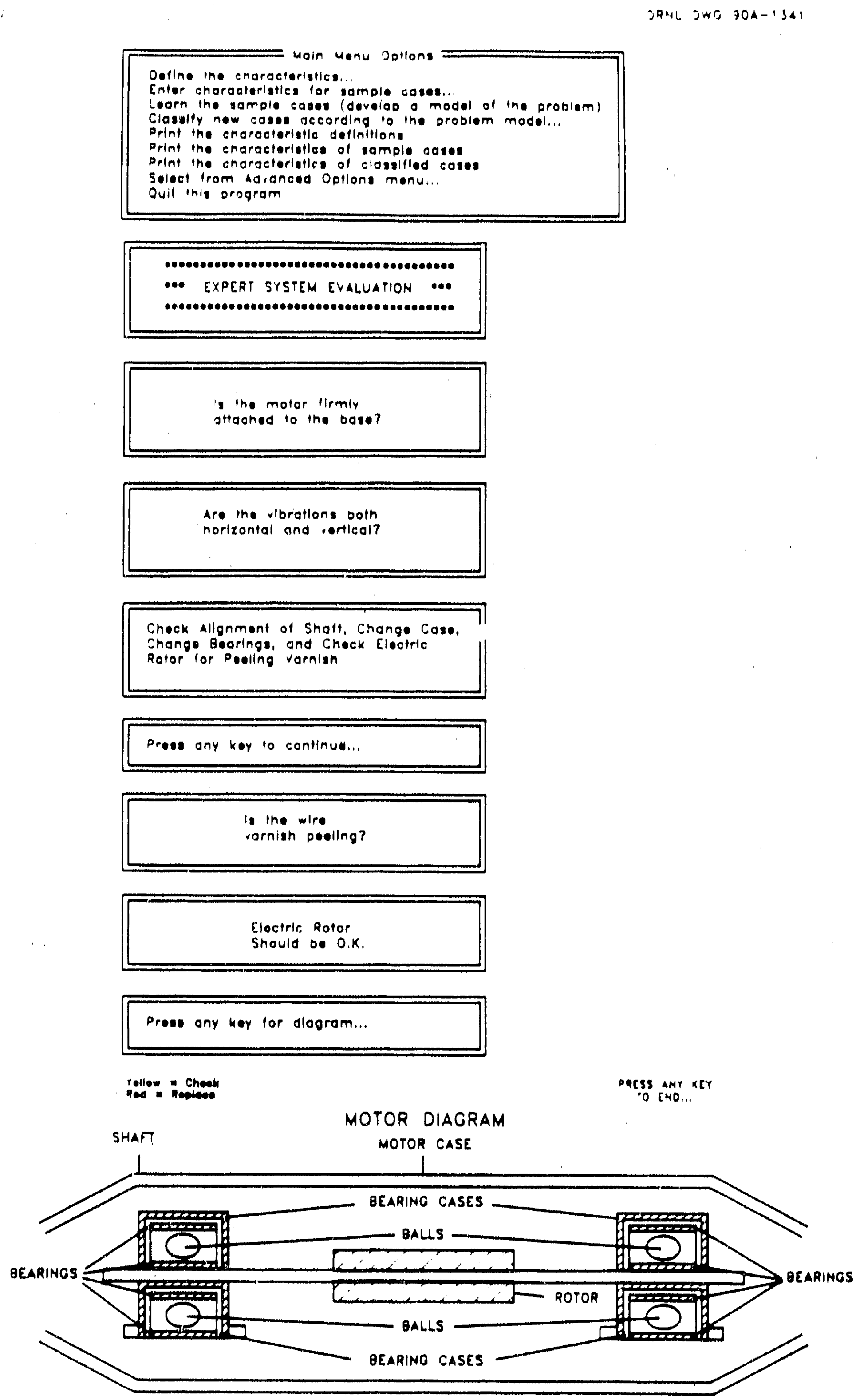


Fig. 7. Expert system interaction process.

ORNL DWG 90A-1342

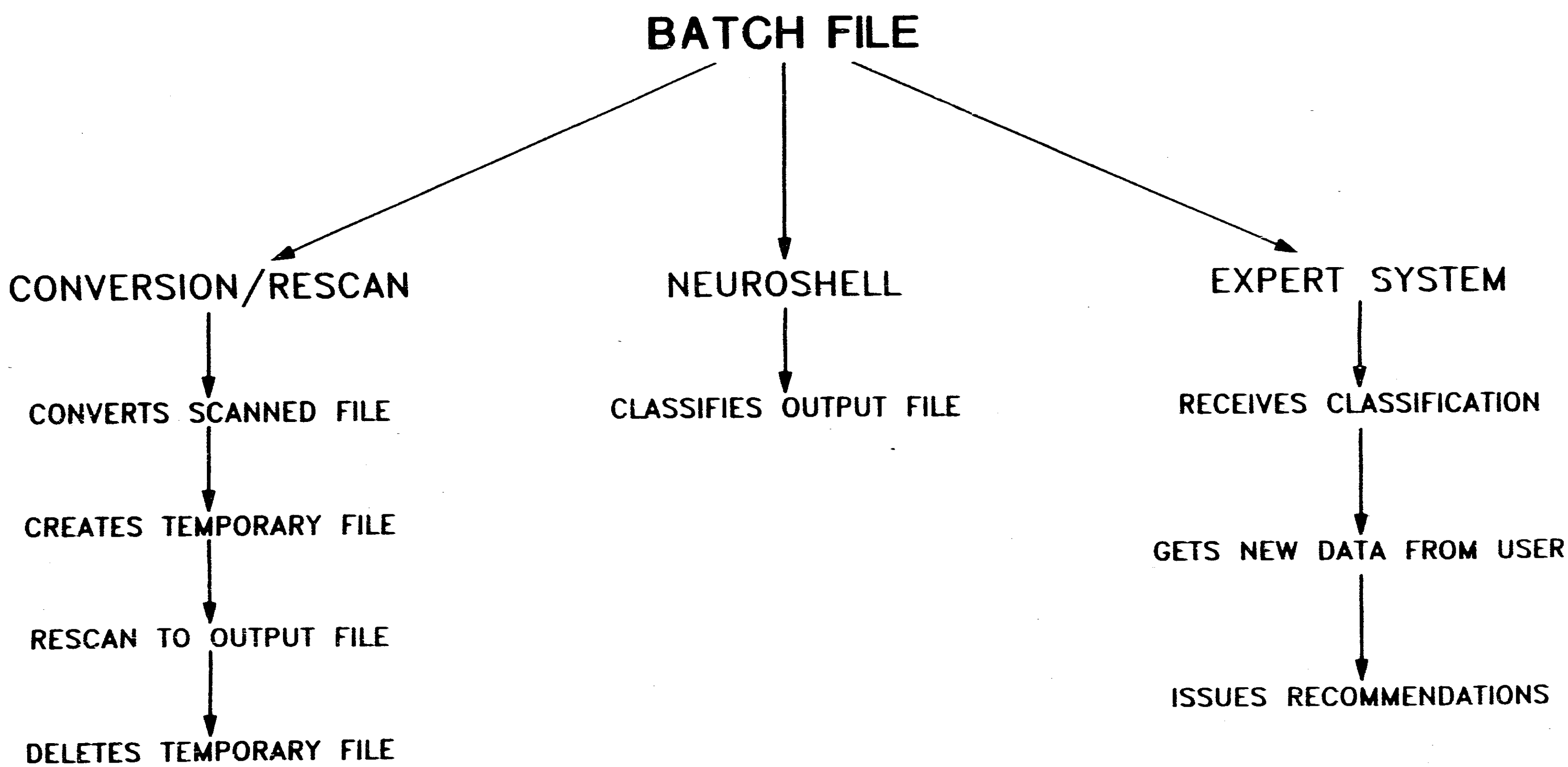


Fig. 8. Batch file for the fault diagnosis hybrid system.

END

DATE FILMED

12 / 05 / 90