Comparison of Rule-Based and Neural Network Solutions for a Structured Selection Problem

JERRY J. HAJEK AND BRIAN HURDAL

Advantages and disadvantages are compared of using a rule-based paradigm versus a neural-network-based paradigm for developing expert systems involving structured selection problems. For comparison purposes, two knowledge-based expert systems were developed using the two alternative paradigms to solve the same specific problem: selection of pavement sections that would benefit most from the routing and sealing maintenance treatment. Each expert system used commercially available microcomputer software costing less than $1,000. The two programs have been compared in terms of the results achieved, software and hardware requirements, system development and programming effort, knowledge processing, how uncertainty is dealt with, and other parameters. Neural networks provide an efficient and appropriate computational tool for solving structured selection problems. They can be implemented faster and updated more easily than rule-based systems. However, neural networks do not encode knowledge in any useful form whether used for future reference, explanation of reasoning, or knowledge-based updating.

Several investigators and agencies, including the Ontario Ministry of Transportation (MTO), have developed knowledge-based expert systems to facilitate the selection of the most appropriate pavement maintenance and rehabilitation treatments (*1–3*). The selection is made from a known set of possible pavement preservation treatments using a reasoning process based on judgment and expertise. In other words, the objective of the selection process is to seek a solution to the structured selection problem by judiciously choosing the best solution from a finite set of possibilities.

Past solutions of the structured selection problem have used rule-based or "production" systems. However, structured selection problems involving many input parameters, large numbers of possible solutions, or both, require a fairly complex search strategy and consequently considerable development and programming effort (*4*). It was hypothesized that the effort to develop and program a system for solving the structured selection problem could be substantially reduced by employing an alternative neural network solution. Neural networks are designed to develop a mathematical model connecting input parameters with solutions without the need for the programmer to define the model.

J. J. Hajek, Research and Development Branch, Ontario Ministry of Transportation, Downsview, Ontario M3M 1J8, Canada. B. Hurdal, Department of Systems Design, University of Waterloo, Ontario N2L 3G1, Canada.

The objective of the research reported here was not only to test the foregoing hypothesis but also to

1. Evaluate advantages and disadvantages of the rule-based and (backpropagation) neural-network-based solutions for one specific application (considered to provide a typical example of the structured selection problem) and
2. Address more general issues of strengths and weaknesses of the two approaches and highlight generic considerations for choosing one over the other.

The two programming models, a rule-based paradigm and a neural-network-based paradigm, have been compared against the background of an existing knowledge-based expert system called ROSE (*3,5*). ROSE was designed to determine the need for one specific pavement maintenance treatment—routing and sealing (R&S) of asphalt concrete pavements in cold areas.

The availability of ROSE, developed using a rule-based system, set the stage for the comparison of the two paradigms. An alternative solution based on the neural network paradigm was developed solely for comparison purposes. For comparison purposes also, each of the two solutions was developed using a commercially available microcomputer software of similar retail value.

It should be pointed out that the terminology used here is not universally accepted (*6*). Some investigators distinguish between expert systems, which they consider part of the artificial intelligence field, and neural networks, which they do not (*7*). Others try to distinguish among procedural languages, expert systems, and neural networks by referring to them as "three principal information technologies" (*8*).

For the purposes of this paper, the term "expert system" is defined as a system that attempts to solve problems normally thought to require human specialists for their solution, a rather traditional definition. According to this definition, it does not matter which one of the various programming technologies (or their combinations) is employed to make the expert system work-conventional procedural languages, symbolic languages, if-then rules, neural networks, or generic algorithms. Ultimately all software runs on the same digital computers and all information is represented on digital computers in the same way: computers store and process information by changing state (*6*). Also, neural networks are sometimes referred to as neural nets (*9*). These two terms are interchangeable.

PROBLEM DOMAIN: ROUTING AND SEALING

A case-specific comparison of the two solution paradigms was made for the problem of selecting and recommending R&S as a maintenance treatment for cracks in asphalt concrete pavements in cold areas. Routing, often done with a carbide-tipped router, opens a crack to the width of 20 to 40 mm and a depth of approximately 10 mm. This opening, cleaned and dried by hot compressed air, is required to accommodate enough sealant (hot-poured rubberized or polymerized asphalt cement) to provide an effective seal even after the pavement contracts at low temperatures. The objective is to prevent surface water, particularly water containing deicing salts, from entering and damaging the pavement structure.

In general, R&S is recommended as a preventive pavement maintenance treatment. R&S should be done before the initially formed single pavement cracks deteriorate (ravel, branch out into multiple cracks, or, in the case of transverse cracks, become stepped). Nevertheless, it is not usually practical to perform R&S on hairline cracks. If only a few cracks are suitable for R&S, the operation may not be economically worthwhile. Conversely, if the cracking is extensive, it is usually better to resurface the entire pavement rather than to perform R&S on it.

In addition to the amount and width of cracks, R&S decisions also depend on crack type, pavement serviceability, pavement structure and age, presence of other pavement distresses (ravelling, flushing, rutting, etc.), and the existence of pavement maintenance treatments (*5*). Altogether, there are about 40 different variables and factors influencing R&S decisions.

The economic significance of the R&S treatment has been evaluated by Joseph (*10*). For significant benefits of the treatment to be realized, the pavement sections must be selected for cost-effectiveness and the R&S applications must be well executed. Judicious and timely selection of such sections is the subject of the expert system solution described here.

RULE-BASED SYSTEM

As stated previously, ROSE provides problem solutions using the if-then rules and serves as a benchmark for comparison with the neural network solution. The following steps form the basic procedure for the development of rule-based systems (Figure 1).

Problem Analysis and Definition

Recommending R&S treatment was formulated as the selection from a set of numbers 0, 1, 2, 3, . . . 10 that indicate the desirability of R&S. Definite rejection of R&S is indicated by 0, whereas 10 means that R&S is a highly desirable and cost-effective treatment.

Detailed Knowledge Acquisition

The development of a rule-based system requires detailed knowledge of the problem domain. It is often necessary to select, evaluate, or combine different points of view and to

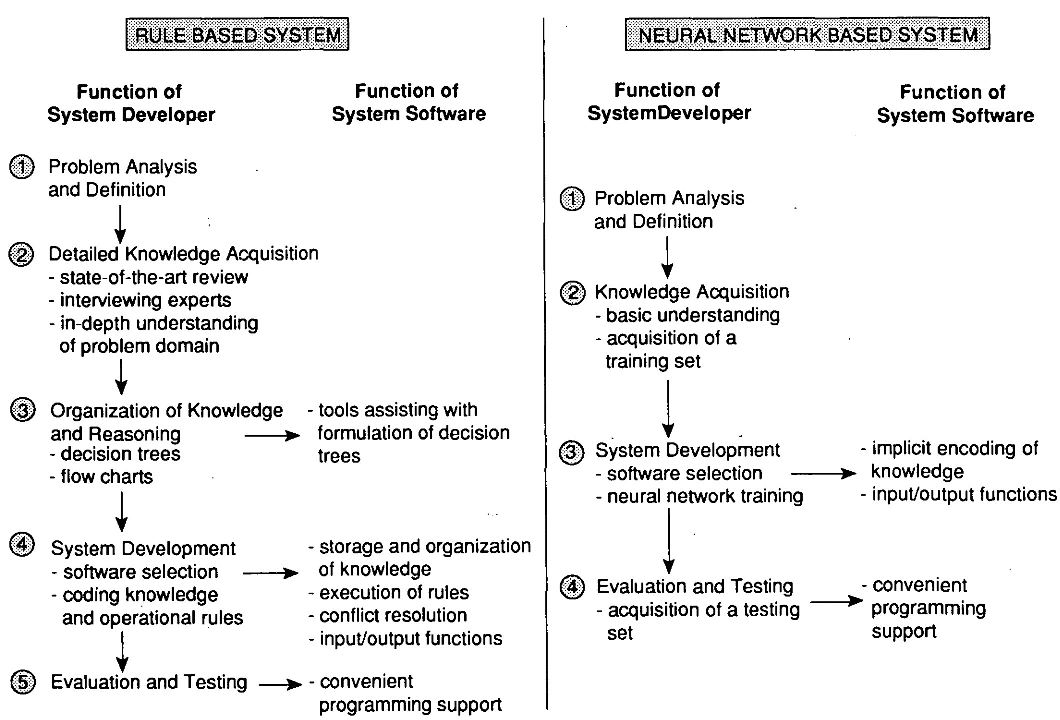

FIGURE 1 Comparison of basic functions by system developer and system software for rule-based and neural network systems.

provide conflict resolution when necessary. Knowledge for ROSE was acquired from written sources and by interviewing experts using an iterative process (*5*).

Organization of Knowledge and Reasoning

Desirability of R&S is influenced by approximately 40 numerical variables, which are routinely collected and stored in the MTO pavement management data bank. Thirty of these variables describe severity and density of 15 pavement surface defects (*5*). The organizational task was to develop a logical procedure for utilizing the values and interrelationships of all these variables and converting them into one variable: the desirability of R&S. This was accomplished by developing a general decision model in the form of a flow chart. A more typical representation of knowledge for structured selection problems is usually provided by decision trees (*2*,*4*).

System Development

The programming was done in the form of if-then rules using an EXSYS expert system development package (*11*). An example of an EXSYS rule is shown in Figure 2. EXSYS has a user-friendly interface, and the rule formulation and coding were greatly assisted by the EXSYS editing program and inference mechanism. Although the if-then rule program is not strictly a procedural program, the rules cannot be arranged in an arbitrary order regardless of their context. Indeed, the main means of controlling the user's interface and program execution is through the arrangement of knowledge rules and facts and specifically created operational (strategy control) rules. Consequently, rule-based programs may require considerable programming effort. Specifically, ROSE required about 3 months of development and programming. This and other attributes, which will be discussed later, are summarized in Table 1.

```
Rule Number:  48
IF:
         The severity of half, full, and multiple
         transverse cracking is moderate
and      The density of half, full and multiple
         transverse cracking is throughout
THEN:
         [BASE]  IS GIVEN THE VALUE 3
and      [CRACK EXTENT] IS GIVEN THE VALUE [CRACK EXTENT]
         + 0
and      routing and sealing is governed by crack extent
and      amount of half, full and multiple transverse
         cracking is too many
and      [TOTAL] IS GIVEN THE VALUE [TOTAL] + 3
```

FIGURE 2 Example of ROSE rule coded in EXSYS.

TABLE 1 Comparison of Rule-Based and Neural Network Attributes for R&S Problem

	Solution	
Attribute	Rule Based	Neural Network Based
Software Used	EXSYS Pro. Cost: $795.00	BrainMaker Pro. Cost: $795.00
Hardware Used	IBM compatible microcomputer	IBM compatible microcomputer
Documentation	608 page user's guide	422 page user's guide
Linkage with Software	Access to dBase, Lotus. Can call external programs during execution.	Access to dBase, Lotus, Excel. Execution can't be interrrupted to call external programs.
Development Effort	1 week of knowledge acquisition plus 3 months of development and programming.	1 day of knowledge acquisition plus 3 weeks of development.
System Size	About 360 rules constituting the main program.	148 training facts. No programming required.
Explanatory Capabilities	Can explain reasoning by recalling applicable rules. Logic path is known and can be followed.	Very limited explanatory capabilities. Reasoning path is unknown.
Other Computational Features	Excellent input/output features. Versatile command language. User created help and explanation files. Direct interaction with other programs.	Good screen editor but limited input/output features. Easy to learn.
Knowledge Encoding	Detailed encoding of knowledge base is mandatory.	Encoded knowledge cannot be accessed.
Knowledge Updating	A good understanding of the rules is required to make any substantial changes.	Easy: no in-depth knowledge required. However, success of retraining is not guaranteed.
Dealing with Uncertainty	Uncertainty can be associated with both inputs and outputs, and can be quantified.	Can handle both input and output certainties by using "fuzzy" inputs and outputs. Uncertainty is difficult to quantify.
Implementation Result	Both solutions provide comparable results in comparable computational time.	

Evaluation and Testing

Taking advantage of the editing features and the inference engine supplied by EXSYS, ROSE was calibrated, tested, and verified on approximately 100 pavement sections.

NEURAL-NETWORK-BASED SYSTEM

The computational procedure referred to as the neural network derives its name from biological neural systems. These systems or networks organize billions of basic cells, called neurons, into a highly functional organ—the brain. It is claimed that neural networks attempt to model associative reasoning and pattern matching of the human brain, and neural network technology has been inspired by studies of the brain and the nervous system (*8,12*). However, at present, neural networks only model the process that connects input data with output data by exploiting the capacity of computers to perform an iterative series of rapid numerical calculations.

A basic neural network consists of three layers of interconnected nodes called neurons (Figure 3). Input-layer neurons receive data from the user; output-layer neurons send information to the user. The middle (hidden) layer of neurons receives signals from all the neurons in the input layer and has the option of sending signals to all the neurons in the output layer. The computer is programmed to calibrate the strength of the signals transmitted within the network by an iterative process until the output neurons yield desired results. Mathematical formulation of this process is available in several sources (*12,13*).

The basic procedure used for development of the neural network solution consisted of the following steps (Figure 1).

Problem Analysis and Definition

Problem formulation was the same as that for the rule-based system. Input-layer neurons were identified with 40 input variables and output-layer neurons with the R&S desirabilities on a scale from 0 to 10.

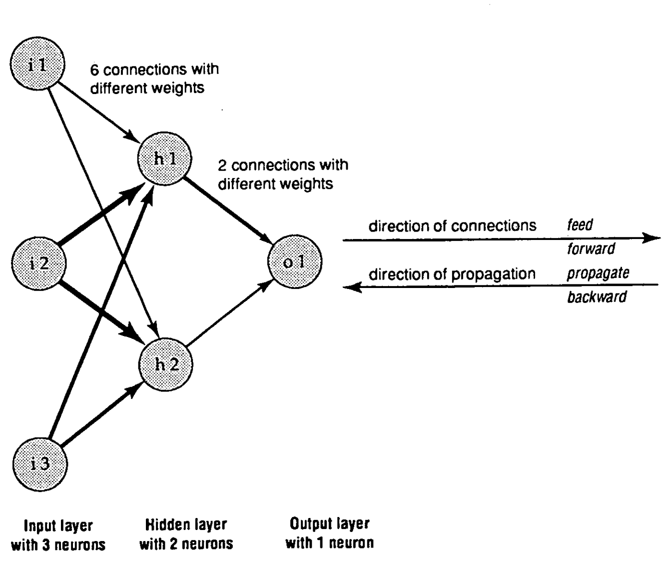

FIGURE 3 Simple three-layer neural network.

Knowledge Acquisition

Neural networks do not require (and often cannot utilize) detailed knowledge of the problem domain. For example, the knowledge of the influence of asphalt concrete thickness on R&S desirability, when all other variables are held constant, is required for the development of the rule-based system but not for the neural-network-based system. However, knowledge is always beneficial for building and training (calibrating) neural networks. The development of neural networks requires at least two pieces of knowledge on the part of the developer: (*a*) factors or variables that are likely to influence the results and (*b*) recognition of the validity of the results. Neural networks can only make predictions based on "experience"—on previous linkages between input and output sets.

Knowledge acquisition consisted mainly of acquiring all relevant input data for a random sample of 148 pavement sections. The sections were obtained from the pavement management data bank and were assumed to provide a wide variety of data across all ranges of values. The R&S desirability for the 148 sections was determined by ROSE because ROSE was already available, is considered reliable, and uses the same input data. The resulting 148 input-output pairs were used as the neural network training set.

System Development

The neural network software used, BrainMaker Professional 2.0 (*14*), is a representative neural network software designed for general use. It forms a complete system for designing, building, training, testing, and running neural networks and was considered to be an appropriate corresponding counterpart for EXSYS. Figure 4 shows two typical BrainMaker menus: a startup menu for creation of training sets and a run menu for training and testing of neural networks.

The bulk of the neural network development involved training the network to provide acceptable results or outputs for all 148 cases in the training set. To determine the outputs, BrainMaker utilizes a supervised training scheme called backpropagation. The backpropagation algorithm compares the current network output with the desired output for a given training fact. If the difference is unacceptable, the weights of connectors leading from the input layer to the output layer (Figure 3) are adjusted to produce an improved output. This algorithm is employed for all other facts in the training set and repeated for the whole training set until some predetermined training criteria are met, at which time the network is said to converge. For example, the total error across the whole training set is below a certain limit. Although this iteration is performed automatically, the user can influence the process or observe it by specifying several parameters.

Transfer (Activation) Function

When a neuron receives inputs (i.e., outputs from other neurons), it calculates its output using an activation function. In

NetMaker Professional v2.0	
Read in Data File	Ctrl-F
Manipulate Data	Ctrl-M
Create BrainMaker File	Ctrl-C
Go to BrainMaker	Ctrl-B
Save NetMaker File	Ctrl-S
Exit NetMaker	Ctrl-Z

NetMaker Start-up Menu/List

Train Network	^T
Continue Training	^C
Test Network	
Run Trained Network	
Get Next Fact	^G
Erase Network Input	^E
Hypersonic Train	

BrainMaker Run Menu/List

FIGURE 4 NetMaker startup menu and BrainMaker run menu.

the simplest models, this is just the weighted sum of its inputs. The transfer function used in this application was a sigmoidal or an S-shaped function, which has asymptotic approaches at the high and low ends of input values.

Learning Rate

Learning rate influences the amount of adjustment to the connection weights between successive iterations. For example, a learning rate of 0 means that the weights never change, no adjustments are made, and "learning" never takes place. The application used a default learning rate of 1, which guarantees convergence if convergence is possible. Only limited experimentation was done with higher learning rates because the total training period of about 2 hr required for the learning rate of 1 was acceptable.

Number of Hidden Neurons

The number of hidden neurons can significantly affect the training and performance of neural networks. BrainMaker was set to automatic neuron selection.

Training Tolerance

Training tolerance indicates the range for which outputs are considered correct. The best results for the application were achieved at around 10 percent tolerance. The higher tolerance settings resulted in disorderly predictions. If the tolerance is set too low, the network runs the danger of memorizing rather than predicting outputs and requires excessive computer time. The training period was about 2 hr at 10 percent tolerance on a Compaq Deskpro with a 387 numeric coprocessor.

Number of Training Facts

A relatively low number of training facts (148) was used. Although there were 40 input variables and some of these variables were categorical ones, many variable combinations do not occur [e.g., high pavement age and high Pavement Condition Index (PCI)]. Also, there are dominant variables (such as the PCI) that may overshadow the influence of marginal variables. Finally, the objective was to create a workable neural network solution for comparative purposes only.

Sequence of Training Facts

Neural network solutions often converge most effectively when the training facts are in a specific order, which is often a random order as used in this application.

Diagnosing Problems

BrainMaker has the capacity to display histograms of the weight matrixes placed on the connections between neurons in the input and hidden layers or in the hidden and output layers (Figure 5). For example, the histogram in Figure 5, obtained for the final (trained) network, shows that there are about 26 neuron connections with the value of 3 between the input and hidden layers. Histograms are useful for assisting in evaluating overall network performance. The histogram in Figure 5 resembles a bell-shaped curve and has a lot of spare capacity for weights in the higher ranges (-8 to -3 and $+3$ to $+8$).

Evaluation and Testing

Neural network testing is done by giving the network information not available before and observing the results. The common way of doing this is to reserve about 10 percent of the training set for testing. In this application, testing was done using a random set of 20 additional pavement sections.

An integral part of evaluation and testing of an expert system is to study how the system reaches its conclusion. Rule-based systems provide a clear trail of rules that fully explain how the system works. The explanatory capability of rule-based expert systems is particularly useful for the prediction of unusual cases. Unlike rule-based systems, neural networks

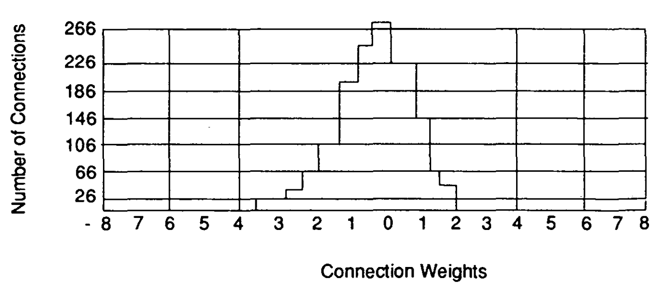

FIGURE 5 Histogram of weight matrix between input and hidden layers.

do not rely on causal relationships and existing expertise and attempt to model only the process by which the inputs become outputs. Consequently, neural networks provide limited opportunities for purposes of causal analysis.

Some knowledge of the relative importance of input factors used by the network can be obtained by examining connection weights (9). For example, BrainMaker software indicates the importance of input variables using neuron sensitivity graphs. Some insights are also provided in terms of "neuron activity" by a series of little bar graphs called thermometers. Figure 6 shows 40 such thermometers (for the 40 input variables) obtained for the first section of the testing set (pavement test section 1 in Table 2). For this particular case, variable PCI has a relative influence on the output (R&S desirability of 1.676) equal to ⅝, and variable density of machine patching (dmcp) has no influence on the output.

COMPARISONS

The BrainMaker neural network solution was designed to replicate the R&S desirabilities determined by the EXSYS rule-based system. The results obtained by the two alternative solutions for the random sample of 20 pavement sections are summarized in Table 2, from which several observations can be made.

1. For lower desirabilities, in the 0 to 5 range, there are some substantial differences between the two solutions.
2. For higher desirabilities, in the 6 to 10 range, the results provided by the two solutions are quite similar. Of the six occurrences, two are identical, three are 1 point apart, and one is 2 points apart (6 versus 4).
3. In practice, only the ranking of sections with the higher desirabilities is important, because only these sections are actually considered for the R&S treatment and, usually, only the sections with highest desirabilities receive the treatment.
4. Recommendations expressed on the scale of 0 to 10 are probably more detailed than necessary. The expert is likely to express his or her recommendations using only three or four categories [e.g., priority for R&S is none (0–3), low (3–5), medium (6–7), or high (8–10)].

The results for several other evaluation parameters are described in Table 1. Overall, it is concluded that the two alternative solutions are comparable and that the neural network solution is considerably easier and faster to develop.

On the basis of this application and previous experience, the following additional comments are offered.

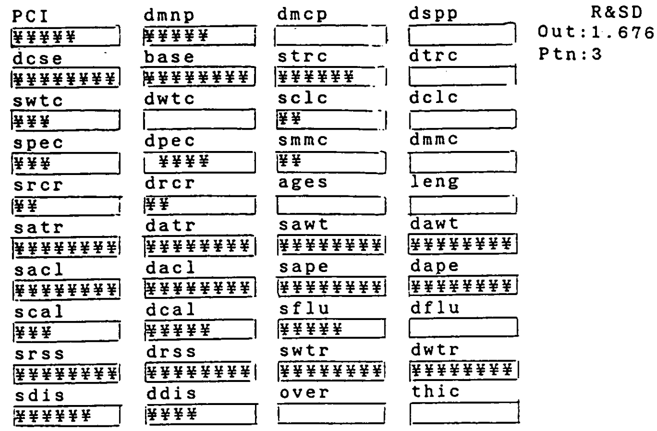

FIGURE 6 Thermometer display obtained during network testing. [See section 1 in Table 2; *Out: 1.676* is output, predicted number; *Ptn: 3* is pattern, actual (or ROSE) number. Abbreviations above thermometers denominate input-layer variables.]

Explanatory Capabilities

Unlike neural networks, rule-based systems can explain how the system reached its conclusions. However, the explanation provided automatically by the rule-based systems only identifies the (chain of) applicable rules used by the system. The rules themselves, even if fully defined by the explanatory facility, are often quite cryptic and may require further explanation or translation to be useful to many users (Figure 1). Explanatory facilities of the rule-based system can be expanded and enhanced, but this requires additional programming.

Knowledge Encoding and Recall

Several different viewpoints can be advanced on this issue. Rule-based systems demand (and enable) detailed encoding of the domain knowledge. However, this knowledge is not really readily accessible to the user. Nevertheless, a typical user is usually not interested in minute details; he or she is interested principally in the results and their reliability and only then, to some degree, in the main features behind the program.

The need to develop knowledge rules for the rule-based system has several positive consequences. The need may provide motivation to finally capture and declare rules, identify discrepancies, and develop precise guidelines. Once known,

TABLE 2 Comparison of R&S Desirabilities

Pavement Section	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Target Value for R & S (ROSE)	3	6	0	0	3	5	8	2	6	5	6	8	4	5	8	5	0	1	5	5
Value Obtained by Neural Network	2	7	3	0	2	3	8	3	7	3	4	7	3	1	8	0	5	1	3	5

Notes: All results are rounded and reported as integers.
10 represents the highest R & S desirability, 0 is least desirable.

rules can be translated into different computer languages and utilized by different hardware and software systems. The existence of the rules allows other experts to supplement or correct the knowledge base.

Neural networks are particularly useful when there is no effective way to explain reasoning, models or algorithms are unavailable, or there is no interest in generating models.

Updating of Programs

Both rule-based or neural network programs must be updated by someone who knows the specific programming or development environment (e.g., EXSYS, BrainMaker). Neural network updating by increasing the size of the training set is quite efficient and simple compared with updating rule-based systems, in which the context of the rules may also be important. However, neural network updating may require a considerable amount of training time and additional, perhaps scarce, training facts.

Dealing with Uncertainty and Missing Data

Neural networks have greater generalization ability and can include uncertainty implicity as part of the training set. Rule-based systems cannot deal with situations that are not covered by the rules. Also, the rules require an exact linguistic match between the names of variables in the rules and the names used by the user for inputting data.

Unusual Cases

Neural networks require special training to accommodate special or unusual cases, and an adequate solution is not guaranteed. Rule-based systems can handle unusual cases using explicit rules.

CONCLUSIONS AND RECOMMENDATIONS

The following conclusions were drawn and recommendations were made:

1. The two alternative solutions, the EXSYS rule-based solution and the BrainMaker neural network solution, yield comparable results.
2. Neural networks provide an efficient and appropriate computational tool for solving structured selection problems (*a*) that do not require detailed encoding of causal relationships, (*b*) for which detailed knowledge is unavailable, or (*c*) that are not of interest to the users.
3. Neural networks would benefit from development of techniques for interpreting their inner workings in terms of causal relationships. Some limited tools exist, such as analyzing connection weights by graphing neuron sensitivity, but they are far from satisfactory. At present, neural networks are reliable pattern matchers and not much more.
4. Since rule-based and neural network solutions exhibit strengths and weaknesses in different areas and supplement each other, their combination in one software system or their use for one application would be advantageous.

ACKNOWLEDGMENT

The authors gratefully acknowledge the inspiration provided by Roshdy H. M. Hafez, Department of Systems and Computer Engineering, Carleton University, Ottawa.

REFERENCES

1. S. G. Ritchie, C-I. Yeh, J. P. Mahoney, and N. C. Jackson. Development of an Expert System for Pavement Rehabilitation Decision Making. In *Transportation Research Record 1070*, TRB, National Research Council, Washington, D.C., 1986, pp. 96–103.
2. K. T. Hall, C. M. James, M. I. Darter, and S. H. Carpenter. Expert System May Aid CPR Field Work, Planning. *Roads and Bridges*, April 1988, pp. 35–39.
3. J. J. Hajek, G. J. Chong, R. C. G. Haas, and W. A. Phang. Can Knowledge-Based Expert System Technology Benefit Pavement Maintenance? In *Transportation Research Record 1145*, TRB, National Research Council, Washington, D.C., 1988, pp. 37–47.
4. J. J. Hajek. Structured Selection Problem in Pavement Rehabilitation. In *Applications of Artificial Intelligence in Engineering* (G. Rzevski and R. A. Adey, eds.), Computational Mechanics Publications, Southampton, United Kingdom, 1991, Vol. 6, pp. 389–403.
5. J. J. Hajek, R. C. G. Haas, and W. A. Phang. ROSE: A Knowledge-Based Expert System for Routing and Sealing. *Proc., Second North American Pavement Management Conference*, Toronto, Nov. 1987, pp. 2.301–2.314.
6. D. L. Parnas. Why Engineers Should Not Use Artificial Intelligence. *Information*, Vol. 26, No. 4, Jan. 1988, pp. 234–246.
7. F. Zahedi. An Introduction to Neural Networks and a Comparison with Artificial Intelligence and Expert Systems. *Interfaces*, Vol. 21, No. 2, April 1991, pp. 25–38.
8. *An Overview of Neural Computing*. Neuralware, Inc., Pittsburgh, Pa., n.d.
9. G. D. Garson. Interpreting Neural-Network Connection Weights. *AI Expert*, April 1991, pp. 47–51.
10. P. Joseph. *Is Crack Sealing in Flexible Pavement a Worthwhile Maintenance Procedure*? Internal Report. Ministry of Transportation, Ontario, Canada, Feb. 1992.
11. *EXSYS Professional, Advanced Expert System Development Software*, Ver. 2.1. EXSYS Inc., Albuquerque, N.M., 1988.
12. J. Lawrence and S. Luedeking, eds. *Introduction to Neural Networks: Computer Simulations of Biological Intelligence*, 3rd ed. California Scientific Software, Grass Valley, May 1991.
13. M. S. Kaseko and S. G. Ritchie. Pavement Image Processing Using Neural Networks. *Proc., Second International Conference on Applications of Advanced Technologies in Transportation Engineering*, Minneapolis, Minn., Aug. 1991, pp. 238–243.
14. *Brainmaker Professional: User's Guide and Reference Manual*, 2nd ed. California Scientific Software, Grass Valley, Dec. 1990.