Zeitschrift: IABSE reports = Rapports AIPC = IVBH Berichte

Band: 72 (1995)

Rubrik: Session 4: Artificial intelligence technology in civil engineering

Session 4

Artificial Intelligence Technology in Civil Engineering

Technologies de l'intelligence artificielle en génie civil

Technologie der künstlichen Intelligenz im Bauwesen

Leere Seite
Blank page
Page vide

A Neural Network Approach to Plant Procurement

Modèle de réseau neuronal pour l'acquisition de biens d'équipement

Einsatz eines neuronalen Netzwerks zur Baumaschinenbeschaffung

Gurdev SINGH
Lecturer
University of Leeds
Leeds, UK

Anthony MWANIKE
Post-graduate Student
University of Leeds
Leeds, UK

Antonis W.R. SLICHER
Research Scholar
University of Leeds
Leeds, UK

SUMMARY
The authors introduce a neural network model that was developed to assist in the selection of the method of plant procurement. An outline of the characteristics that make the problem suitable for neural network modelling is followed by a presentation of the procedure for developing, training and testing of the model. The paper demonstrates the feasibility and desirability of the model, but concludes that a significant amount of research work is required before a reliable tool can be realised.

RÉSUMÉ
L'article présente un modèle de réseau neuronal qui a été développé comme une aide pour l'acquisition de biens d'équipement. Après avoir présenté les différentes caractéristiques qui permettent de traiter le problème au moyen d'un modèle neuronal, l'article présente le développement du modèle, son calibrage et son utilisation. La faisabilité et les avantages de ce type de modèle sont présentés. Un travail de recherche important est cependant encore nécessaire avant qu'un outil fiable ne puisse être réalisé.

ZUSAMMENFASSUNG
Die Autoren stellen ein neurales Netzwerk-Modell vor, das entwickelt wurde um die Auswahl der Anschaffungsmöglichkeiten von Baumaschinen zu unterstützen. Einer Beschreibung der Eigenschaften, die das Problem für ein neurales Netzwerk-Modell behandelbar machen, folgen die Vorgehensweise zur Entwicklung des Modells und die Methoden des Trainings und des Testens. Der Artikel zeigt, dass das Modell realisierbar und sehr hilfreich ist, aber auch dass die Entwicklung eines verlässlichen Werkzeugs noch eine beachtliche Menge an Forschungsarbeit erfordert.

1. INTRODUCTION

Over the years the construction industry has experienced an underlying trend towards greater use of plants in the construction and maintenance processes. Several factors have encouraged this shift to mechanisation. These factors include technical improvements in construction equipment, the increasing size and complexity of construction projects, short construction periods required by clients and economic pressures. Furthermore, the increasing cost of labour has increased the relative benefits that can be derived from mechanisation. Construction is a high risk business environment where any decision that has any bearing on the financial wealth of construction firm or project demands a realistic approach to the appraisal of the options available and of the consequences of taking a particular course of action. The selection of both the plant required [1] and the method of procurement is an important element in cost and time [2, 3]. This paper introduces a neural network model for applications in the selection of plant procurement method.

There are essentially 4 methods of procuring plant:

1. Outright Purchase
2. Hire Purchase
3. Rental from a plant hire company
4. Leasing.

The choice of procurement method will have a significant effect on the financial status of any construction company and consequently on its day to day operations as will as its long term performance. The plant procurement decision can be split into two processes:

1. Technical Analysis to establish the utilisation rate, type of application and length of service for the plant.
2. Financial Assessment to establish the profitability of operating a type of plant.

The method of procurement is influenced by the former and has a profound effect on the latter.

Often the decision on the method of procuring plant is made on purely financial principles. This is not advisable for two principle reasons. First the data used to compute cash flows ensuing from the procurement and use of plant is often only a deterministic [4] estimate of the expected revenue. The reliability of such estimates is inversely proportional to the length of the period which they cover. Secondly, the outcome of the assessment is largely dependant on the discount rate used to produce the Net Present Value. Depending on the discount rate chosen it is possible to make a case for any given method and show that it is superior to any other.

Furthermore vendors offering the different options are regularly adjusting their price structures to a point where there is little difference in the annual equivalent cost between any two methods of procurement. It is therefore the rule rather than the exception for managers of construction firms to examine various qualitative factors when considering which method to use in the procurement of plant. Some of the factors that affect the procurement method are shown in Table 1. A few of the factors in this Table, like active life can be incorporated in the financial models used in analysis and thus be accounted for in the financial assessment. Other factors like the risk of technological obsolescence are difficult and often impossible to incorporate into these financial models. This hampers the use of conventional computing methods. Expert systems, which are among the most popular artificial intelligence techniques, are also not suitable for application to this problem. Their IF THEN procedures do not mirror the decision making process. Furthermore the factors shown in the table are considered simultaneously, a process that is difficult to model using traditional artificial intelligence techniques. [5].

TABLE 1: Factors Affecting the Selection of Plant Procurement Method

	FACTORS
1	Period over which plant is required
2	Type of application to which plant is to be put
3	Utilisation rate
4	Active life
5	Financial Status of company purchasing the plant
6	The magnitude of the risk of technological obsolescence
7	The probability of continuation of use
8	Annual equivalent cost, internal rate of return or Net present value
9	Urgency with which plant is required

2. NEURAL NETWORKS

A neural network is a computer model designed to emulate some of the features of the human brain. It is a non programmable adaptive information processing system capable of producing it's own algorithm in response to its environment [6,7,8]. Neural networks as shown in Figure 1 are built of numerous highly interconnected, simple processing units designed to resemble biological neurones.

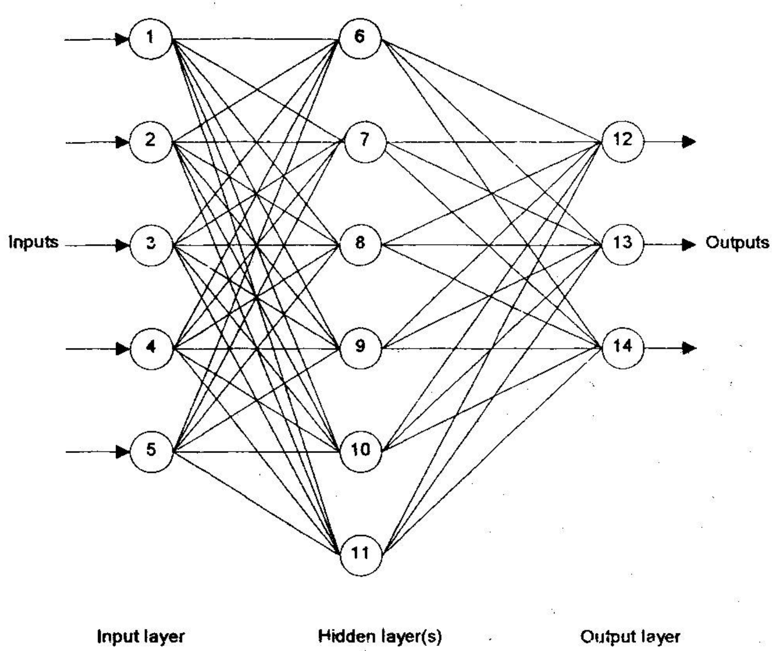


Fig. 1 Structure of a Neural Network

Each processing unit represents a specific concept: features, letters, data or an indescribable part of a larger concept [9]. All information processing is carried out by these units which only receive inputs and compute outputs which they send to neighbouring units, Neural networks are inherently parallel because the processing units can carry out their computations simultaneously [10]. Figure 1 shows how processing units are arranged in structural units called layers. Interaction between any two processing units depends on the weighting factor assigned to the connection between the two units.

Unlike standard software, a neural network is not a carefully developed series of linear instructions [11]. They are not programmed but rather they are trained using a series of input and output data. The network undergoes internal changes as it formulates the correct relationship between the input and output data. It stores what it has learnt as connection weights between processing units. The characteristic that has been the critical factor in raising interest in neural networks is this ability to learn. Other interesting and valuable characteristics of neural networks are their capacity of generalisation and induction, which enable them to give nearly correct responses when exposed to incomplete inputs. Parallelism endows neural network with processing speeds which allows them to overcome limitations imposed on conventional systems.

3. MODEL DEVELOPMENT

The first step in developing a neural network modelling is to identify the type of input that the network would require and the output desired. For the problem of selection of a plant procurement method this involves the selection of the factors that would need to be considered in order to identify the optimal procurement method. Seven factors were chosen from Table 1 to test the feasibility of a neural network approach. These factors were:

* The period over which the plant is required
* The type of application the plant is put to
* The utilisation rate
* The probability of continued use
* The company's financial status
* The risk of technological obsolescence and
* The urgency with which the plant is required

Table 2 shows the stages of development of the input and output representations used for the neural network "***NeuroPlant***". The various levels of input from the final model used for NeuroPlant was a ranking of the different methods on a scale of 0 to 1: the different methods would be ranked to show their suitability relative to one another of achieving the desired objective under given conditions. The sum of the rankings given to each method under any set of circumstances was not to exceed a value of 1. The choice of this presentation was due to two principle reasons. First, the software was more amenable to this form of representation. Second, it was our opinion that a system that ranked the methods was superior to one that gave a single output showing the optimal method. Such a system would allow the user to identify the next best method if s/he does not wish to use the optimal one.

TABLE 2(A): *NeuroPant* INPUT AND OUTPUT PATTERN

INPUT PATTERN

Requirement Period	< 1 year	1 - 2 years	2 - 5 years	5 - 10 years	
Application	Universal	Specialised			
Utilisation Rate	Very Poor	Poor	Average	> Average	High
Probability of continued use	No Possibility	Highly Possible			
Financial Status	Poor	Average	Good	Very Good	
Obsolescence Risk	Very Low	Low	Average	> Average	High
Requirement Urgency	None	Urgent			

OUTPUT: METHOD

Outright Purchase	Hire Purchase	Lease	Rental

TABLE 2(B): INITIAL INPUT AND OUTPUT PATTERN USED IN *NeuroPlant*

INPUT PATTERNS RANKINGS

Requirement Period	1	2	3	4	
Application	1	2			
Utilisation Rate	1	2	3	4	5
Probability of continued use	0	1			
Financial Status	1	2	3	4	
Obsolescence Risk	1	2	3	4	5
Requirement Urgency	0	1			

OUTPUT: METHOD

Outright Purchase	Hire Purchase	Lease	Rental

TABLE 2(C): THE FINAL INPUT AND OUTPUT PATTERN USED IN *NeuroPlant*

INPUT PATTERNS RANKING

Requirement Period	1	2	3	4	
Application	1	2			
Utilisation Rate	1	2	3	4	5
Probability of continued use	0	1			
Financial Status	1	2	3	4	
Obsolescence Risk	1	2	3	4	5
Requirements Urgency	0	1			

OUTPUT: METHOD

Outright Purchase	Hire Purchase	Lease	Rental
0	0	1	0

4. KNOWLEDGE ACQUISITION, TESTING AND TRAINING

From the very start it was realised that these will present significant problems. The selection of procurement strategy depends on a number of interrelated factors. Presented with identical information different managers arrive at different decisions. Furthermore, the same manager often makes a different decision when presented with identical input some time later. The underlying cognitive and emotive biases in humans make managers unsuitable as trainers. What is required is a training set of reliable character and adequate size. The results of the work reported here led to the realisation that a more reliable set of training data will produce a very significant improvement in the performance of the system. The next step involves the creation, through simulation, of an "expert" capable of producing any number of training sets rendered reliable by eliminating the biases. The details of this approach will be the subject of a separate paper to be published in 1994.

The network's architecture was defined using "Brainmaker" a proprietary back propagation neural network software package developed by California Scientific Software. The neural network developed had 7 neurones in its input layer, 4 neurones in its output layer and 32 neurones in the middle layer.

The neural network was trained using 33 examples. Training was done in Brainmaker and carried out on an Elonex computer with 808486 SX processor working at a speed of 25 MHz. The network converged in about 3 minutes after 425 runs. Testing was done by introducing the two categories of input patterns. The first category of inputs consisted of all the data used in training. The second category of test data consisted of examples which the network had not encountered during training.

5. DISCUSSION

The results of testing the network show that a neural network can be used to capture the relationship between qualitative factors that influence a decision. ***NeuroPlant*** was capable of producing independent outputs to patterns used in training. The network outputs had an average error of 6.1% corresponding to an average aggregated deviation of 0.064 from the desired responses. Though maximum error for the network was 20.9%, errors greater than 10% were few in number. As expected both the average and maximum error from tests on data that the network had not encountered during training was higher than the corresponding results from tests on the training data. The error was largest for input patterns where the factor representing the probability of continued use had a value of 1. The reason for this result is the dearth of training examples that had this factor taking a value of 1. Even where the factor took this value its effect on the decision was often overshadowed by the effect of the other features. When NeuroPlant was presented with examples that had an abundance of patterns where continuity was an important factor the average error was 17.5% and the maximum error was 83.3%, corresponding to aggregated deviations of 0.175 and 0.833 respectively. The average error for test data with only a few patterns where continuity of use was a crucial factor was 7.6% with a maximum error of 34.4%. The test data consisted of 12 examples. It is worthy of note that the sum of rankings given to each method by the network often exceeded the desired value of 1. This problem could not be remedied but such a system could be used to give each method a rank on a scale of 0 to 1.

The accuracy of the neural network model can be improved by:

1) Identifying the optimal size of the hidden layer and training parameters (the large maximum error indicated that the optimal size had not been identified).

2) Increasing the number of examples used in training.

3) Incorporating more factors that affect plant procurement into the model

4) Incorporating examples that are obtained from a survey of professionals with experience in plant procurement.

A valuable lesson from this study was derived through experiencing the difficulty of identifying the optimal size of the hidden layer in a network. The trial and error process used to select network architecture was made more difficult by the network's sensitivity to changes in size of the hidden layer. Furthermore the performance of the network was also dependant upon the size and quality of the training data. It will therefore be difficult to convince potential users of the reliability of neural networks until we are able to understand the working of the hidden layer and subsequently formulate rules to help select the hidden layer characteristics. Secondly the issue of optimal training size requires to be resolved. Experience shows that a network that is trained with too few examples will not learn while too many examples will hamper the network's ability to generalise. The reason for this is not well understood though it is thought that a network presented with a large training set memorised the individual patterns rather than capturing the essential relationship embodied in the data.

The study suffered from one major limitation. This was the factors used and the patterns developed for training the networks were all identified by a very small number of individuals. The networks' abilities therefore reflected the expertise and opinion of these individuals. Any further work in this area should obtain training examples from a large number of managers who are involved in the administration of plant divisions for construction companies. There is a danger that the variability in the training examples introduced by the different perceptions of the experts would make training the network more difficult. Clearly the best method of implementation would involve each interested company training the neural network on data that would represent its own perception on how the different factors affect the choice of procurement method.

6. FUTURE WORK

Anyone carrying out future work should be encouraged to investigate the feasibility of combining the neural network with two important elements:

A user interface and a spreadsheet application housing cost information.. Figure 2 shows the probable structure of the resulting hybrid system. The user interface would perform the following tasks:

1) Get information about the problem from the user

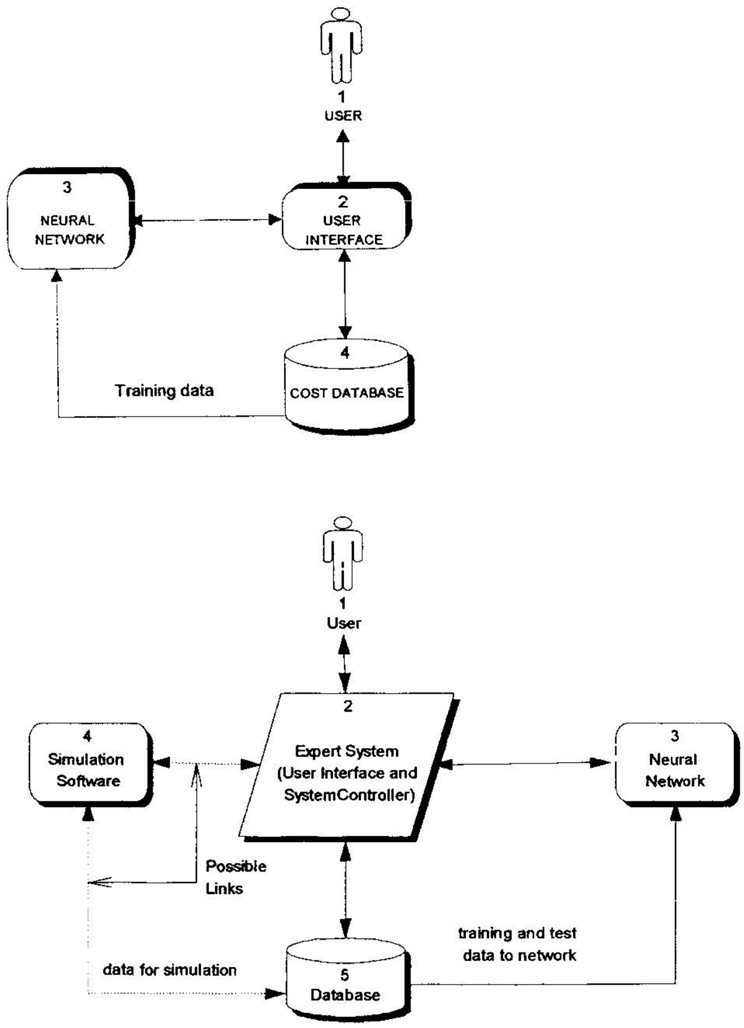


Fig. 2 Possible Structures of a Hybrid System

2) Translate the information into an input representation that the neural network can understand

3) Translate the information provided by the user and the output from the neural network into a form that the spreadsheet can use

4) Display the recommendation from the neural network into a form that the user can understand.

Once a recommendation has been given the management would wish to see the financial implications of this recommendation. This can be achieved by linking the neural network to a spreadsheet package containing a cost database for each method and a procedure for computing cash flows. The user would input various parameters like the type of plant, cost, capital allowances and other pertinent information which would be used together with the neural network's output to compute a cash flow stream. It would be advisable for future work to include an investigation on the performance of the neural network when equipped with two or more hidden layers. The results of such an investigation would allow the identification of the optimal neural network model.

7. CONCLUSIONS

The prototype presented in this paper demonstrates that a back propagation neural network can be developed to assist plant managers in the choice of a plant procurement method. Characteristics like learning and generalisation capabilities combine to make neural networks a potentially powerful decision support system. Additional work will however be required before this application can be developed into a commercially viable tool.

8. REFERENCES

1. Singh, G. and Harun, Z., An Integrated Help System for Stochastic Simulation Models of Construction Materials Handling Operations, Proceedings Civil- Comp 93, The Application of Artificial Intelligence to Civil and Structural Engineering, Edinburgh, 1993.

2. Mead, H. T. and Mitchell, G. L., Plant Hire for Building and Construction Newnes - Butterworth, 1972.

3 Singh, G. and Shaath, K. K., Monte Carlo Simulation-based DSS for Estimating and Tendering for Measured Construction Projects, Proceedings, European Simulation Conference ESM 192, York, UK, 1992.

4. Singh, G. , Risk and Reliability, Analysis on Microcomputers, Chartwell Bratt, 1987.

5. Moselhi, O., Herzagy, T. and Fazro, P., Neural Networks as Tools in Construction, Journal of Construction Engineering and Management Vol 117 No 4 December 1991. A S C E , 1991.

6. Hecht-Nielsen, R., Neurocomputer Applications Neural Computers (Ed Eclimiller, R. and Halsburg, C. V. D.) Springler - Verlag Berlin Heidelberg , 1989.

7. Davalo, E. and Naine, P., Neural Networks Macmillan Education Limited, 1991.

8. Widrow, B., Rumelhart, D.E. and Lehr, M.A., Neural Networks - Applications in Industry, Business and Science, Communications of the ACM, Vol.37, No.3, 1994.

9. Bose, A. and Allen, R. H., Rule Based Representation. Expert Systems for Civil Engineers: Knowledge Representation Ed Allen R H) A S C E, 1992.

10. Garret, J. Ghaboussi, J. and Wu, X., Neural Network Applications in Civil Engineering. Expert Systems in Civil Engineering: Knowledge Representation (Ed Allen, R. H.) A S C E, 1992.

11. Denning, J., Expert Systems: Ready to Hit the Road? Civil Engineering Vol. 62 No. 6, A S C E, 1992.

Leere Seite
Blank page
Page vide

Toward a Consistent Spatial Model

Vers un modèle spatial cohérent

Auf dem Weg zu einem konsistenten Raummodell

Xiao TANG
Computer Scientist
Univ. of Paris
Paris, France

Xiao Tang, born 1963, received her informatics engineering degree at the University of Chong Qing, China. She obtained her doctor's degree at the University of Paris VI. Since four years, Xiao Tang has studied the problem of consistency.

Khaldoun ZREIK
Prof.
University of Caen
Caen, France

Khaldoun Zreik is professor at the University of Caen, France, and researcher in Artificial Intelligence in design at Lutèce and LAIAC laboratory.

SUMMARY
This paper presents a constraint-based consistency maintenance system for a spatial model simulating a building environment perceived by an Indoor Autonomous Mobile Robot. The constraints define a reference frame for the environment model, representing the fundamental properties concerning the characteristics of spatial entities, topo-geometrical or semantic relations between them. The environment model will be declared consistent if and only if all the constraints are satisfied.

RÉSUMÉ
Un système de maintien de la cohérence est présenté pour un modèle de l'environnement spatial simulé où naviguent des robots mobiles autonomes. Ce système est basé sur des contraintes qui forment une référence de cohérence. Ce sont des propriétés concernant les relations géométriques, topologiques et sémantiques existantes entre les entités spatiales. Le modèle de l'environnement est dit cohérent si, et seulement si toutes ces contraintes sont satisfaites.

ZUSAMMENFASSUNG
Der Beitrag stellt ein auf Nebenbedingungen beruhendes System vor, das die Konsistenz in die Simulierung eines Raummodelles wahrt, wie es von einem autonomen mobilen Roboter für Einsätze im geschlossenen Räumen wahrgenommen wird. Die Nebenbedingungen definieren einen Bezugsrahmen, der die unabdingbaren Eigenschaften räumlicher Grössen und ihren topologischen und semantischen Beziehungen repräsentiert. Das Raummodell wird nur dann als konsistent bezeichnet, wenn alle Nebenbedingungen eingehalten sind.

1. INTRODUCTION

The problem of consistency maintenance in Artificial Intelligence is not a new one. In the domain of knowledge-based systems, the consistency maintenance is a fundamental issue [1] ; in the domain of CSP (Constraints satisfaction problem) [2], the process of solving consists in searching a solution with which all the constraints are consistent ; etc. The notion of consistency depends in fact on the domain in consideration.

The consistency maintenance in our context concerns a model of a simulated spatial environment for IAMRs.

An IAMR can perform intelligent motion planned [7]. The planning of robot's trajectory and the pattern recognition during the motion depend directly on the representation model of its spatial environment. The model must provide necessary information about the real world at different levels : geometric, topological, semantic, etc.

A spatial environment can not always be completely and precisely described, and the change in the environment perceived by an IAMR may be falsified (by the noise of its vision sensors for example). For this raison, the consistency maintenance of the environment model is necessary.

In order to control the environment model's consistency, we develop a consistency maintenance system PROVE which possesses a referential truth model built on three types of constraints : constraints-to-be-propagated (Cp), constraints-to-be-verified (Cv), and constraints-to-be-verified-and-propagated (Cvp). Each constraint is a inviolable numerical or symbolical relation existing between spatial entities such as "each space must have an opening which make it accessible (directly or non) from the exterior".

Two cases are checked. Firstly the initial model will be verified before supplied to the robot : we verify the structure of each spatial object, the relation between different objets, etc.

Secondly, any change in the environment will be controlled by PROVE system : it will be propagated throughout the model by using the constraints to be propagated. If the change and its induced result satisfy all the constraints to be verified, it is considered as consistent and the environment model may be updated. Otherwise, the change will be refused and an analysis procedure, based on a mechanism inspired from ATMS (Assumption-based Truth Maintenance Systems) [4], will be engaged to detect the origins of the anomaly.

2. ENVIRONMENT MODELLING

The IAMR's environment considered is a built space represented on two dimensions and half (as illustrated in the figure 1), which can be decomposed into several levels : <Environment> -> <Spaces, Walls, Openings> -> <Facets> -> <Ridges> -> <Points>. If we consider facets and openings as elementary entities, the structure hierarchy of the environment showed in figure 1 can be represented by the figure 2. Thus several levels of information must be taken into account : numerical at low level to represent the geometry of spatial entities ; symbolical at high level to represent the topology of spatial entities.

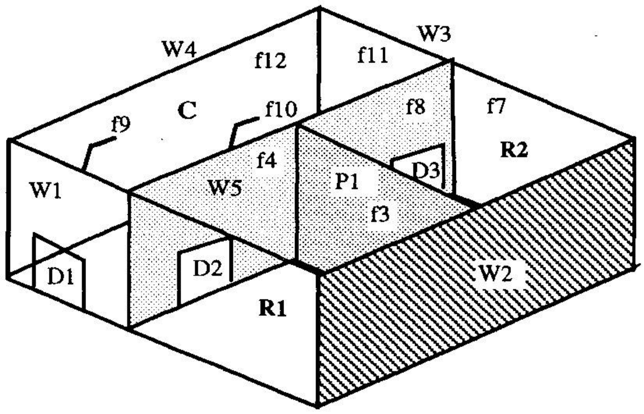


Figure 1. An illustrative building environment

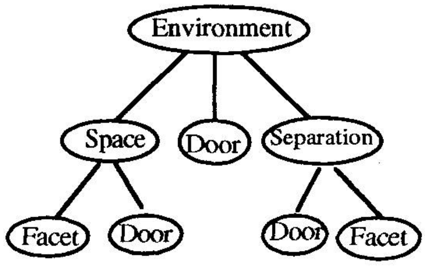


Figure 2. Structure hierarchy

Each spatial entity (facet, door, space or wall) possess some properties like "coordinate", "length" and some topo-geometrical relations with other entities like "adjacent" between two rooms, etc.
We have adopted the object-oriented representation, owing to their encapsulation and inheritance faculties [6, 10], to represent the spatial entities.
Each entity is described by its intrinsic properties, like the width, length of a space, the height of a door ; and by the topo-geometrical relations with other entities, like composition relation between a composite entity and its components, adjacent relation between two spaces, etc.
Each spatial entity is represented by an object with own identifier and attributes. Three categories of attributes have been distinguished :

- terminal attributes being peculiar to an object (size, area of a room, ...) ;
- constituent attributes pointing to the component objects of a composite object (facets and doors of a room, ...) ;
- topological attributes representing the link between objects ("space-linked" of a door, "furniture-contained" of a room, ...).

We define a special attribute "Graph" associated with each composite object in order to link it with its structure graph.
For example the state of Room R2 in the figure 1 can be described as below :

R2 = (<Width 3.0>, <Length 4.0>, <Facets (f1, f2, f3, f4)>, <Doors (D2)>, ..., <Graph Gr2>)

Spatial entities of the environment are described by the following object classes : Facet, Door, Separation, Space. Separation has two sub-classes : Partition and Wall ; Space has also two sub-classes : Room and Corridor.

3. TOWARD A CONSISTENT ENVIRONMENT MODEL

An other type of relations existing between the spatial entities, allow us to define some restrictions on the state or structure of spatial entities, such as for example "the area of a room is equal to the product of its width and its length", "each space must be accessible from the exterior", "the length of the room R2 must be lower than 3 metres", etc. We consider these relations as constraints which may concern, on the one hand, the intrinsic properties of spatial entities, and some arbitrary functional requirements on the other hand.

These constraints constitute a truth model allowing us to control the state consistency or the structure consistency.

3.1 Truth Model Description

Constraints of consistency

In our context, the constraints of consistency concern all the geometrical relations, topological relations and the semantics of the environment's entities, as the examples in the following :

- the constraint of object's position : an object can not be present on the left and on the right of another at the same time, etc. ;
- the constraint of object's structure : a door is always incorporated in a wall ; each space must have a door such that with which this space can be directly or indirectly accessible from the exterior ; etc. ;
- the algebraic relation : the area of a space is the multiplication of its width and its length ; etc. ;
- the deduction : if a door exists between two spaces, then a passage exists between the two spaces, etc. ;
- the exigency : the machines room must have a door such that its width ranks above 2.0 metres ; etc.

Classification and representation of the constraints

The constraints in the truth model differ from those of CSP [11, 3, 2, 9], in that they concern the variables having infinite value domains (the constraint's variables in the CSP have generally the discrete and finite value domains). The constraints can be classified in three types according to the manner in which they are employed : constraints-to-be-propagated to propagate an objects change ; constraints-to-be-verified-and-propagated to refine previous propagation and constraints-to-be-verified to verify objects consistency.

To describe the different constraints, we use a constraint representation formalism, based on the sorted first order predicate logic [12]. The predicates of form "Attribute(entity, value)" are used to present the triplets <entity, attribute, value>.

<u>Constraints-to-be-propagated (Cp)</u>

The Cp constraints are some directed relations between numerical and symbolic data. They allow to add new values to entity's attributes, to remove an object and even lead to the creation of entirely new objects.

A Cp is represented as follows:

$$\langle Quant_{prem} \mid T_{prem} \rangle \{\langle Quant_{prem} \mid T_{prem} \rangle\}$$
$$\langle C_{prem} \rangle \Rightarrow \{\langle Quant_{conc} \mid T_{conc} \rangle\} \langle C_{conc} \rangle \; [\; \mid \mid R]$$

where:

$\langle Quant_{prem} \rangle$ represents the quantifier ($\forall$ or $\exists$) of a variable "prem" in a premise ;

$\langle T_{prem} \rangle$ represents the type of "prem" ;

$\langle C_{prem} \rangle$ represents the body of constraint premises ;

$\langle C_{conc} \rangle$ represents the constraint's conclusion ;

<R> represents a formula that solves the constraint on the variables in the premises and the

conclusion.

Examples :

Cp1: $(\forall x \mid Space(x)) (\forall y \mid Space(y)) (\forall z \mid Door(z))$

$Bound(z, x,y) \Rightarrow (\exists p \mid Passage(P)) (Space\text{-}between(P, x,y) \wedge Access(P, z))$

Cp2 : $(\forall x \mid Space(x))\ Length(x, L) \wedge Width(x, w) \Rightarrow Area(x, a) \mid\mid a=l*w$

The constraint Cp1 describes that a passage exists between two spaces if there is an open access between them. The constraint Cp2 says that we can get the space area from it's length and width.

Equivalent constraints

An algebraic relation between n variables can be represented by n calculating formulas, we will define n equivalent constraints Cp to represent all the formulas. However, only one of them can be triggered in a reasoning cycle, to avoid a dead loop. To do this, we introduce a particular predicate : Deduce(O, A, o1, a1, ..., oi, ai) which says that the value of attribute "A" of object "O" is the calculating result of the value of the attribute a1 of object o1, ..., and the value of attribute ai of object oi. Then the relation represented by Cp2 can be defined by three equivalent constraints and one of which can be represented as below :

Cp2' : $(\forall x \mid Space(x))\ Length(x, l) \wedge Width(x, w) \wedge \neg Deduce(x, Length, x, Area, x, Width)$
$\wedge \neg Deduce(x, Width, x, Area, x, Length) \Rightarrow Area(x, a) \wedge Deduce(x, Area, x, Length, x, Width) \mid\mid a=l*w$

Constraints-to-be-verified(Cv)

A Cvi$\in$Cv is one or several directionless relations between objects. It gives a Boolean value and is represented as follows:

$<Quant_{prem} \mid T_{prem}> \{<Quant_{prem} \mid T_{prem}>\} <C_{prem}> \mid\mid <P>$

where: <P> represents a set of predicates that use the variables from the premises as their arguments, then they will be satisfied by these variables. It may be one or several formulas (linked by "And") of comparison between the variable and its reference (as in Cv1) , or one or several geo-topological relations to be verified (as in Cv2 and Cv3).

Examples:

- the area of a room is equal to the product of its width and its length :

Cv1: $(\forall x \mid Room(x))\ Length(x, l) \wedge Width(x, w) \wedge Area(x, a) \mid\mid a=l*w$

- a rectangular room consists of four closed facets and of a door at least :

Cv2: $\forall x\ Room(x) \mid\mid Facets(x, F) \wedge Doors(x, D) \wedge Cardinality(F, 4) \wedge$
$Length\text{-}higher(D, 1) \wedge Closed(F)$

- each space must be accessible from exterior

Cv3 : $(\forall x \mid Space(x)) \mid\mid Accessible(x)$

The first consists in controlling the attribute "area" of a room, the second consists in controlling the global structure of a room, and the last consist in controlling a topological relation between each space and the exterior which is a special space.

In the Cv1, when one of the three variables is a calculating result of the two others, by the constraint

Cp, this constraint is automatically satisfied. To avoid the redundancy, we introduce also the predicate Deduce(O, A, o1, a1, ..., oi, ai), the Cv1 is transcribed as :

Cv1': $(\forall x \mid \text{Room}(x))\ \text{Length}(x, l) \wedge \text{Width}(x, w) \wedge \text{Area}(x, a) \wedge \neg\text{Deduce}(x, \text{Length}, x, \text{Area}, x, \text{Width})$
$\wedge\ \neg\text{Deduce}(x, \text{Width}, x, \text{Area}, x, \text{Length}) \wedge \neg\text{Deduce}(x, \text{Are}, x, \text{Width}, x, \text{Length})\ \|\ a=l*w$

Constraints-to-be-verified-and-propagated (Cvp)

A Cvpi∈Cvp is a combination of the two preceding types of constraints. It is a constraint to be verified (Cvpvi), associated with other constraints to be locally propagated (Cvppi). The constraints to be propagated (Cvppi) are triggered when the verified constraints (Cvpvi) are unsatisfied.

Examples:

- Cvp1:

Cvpv1: $(\exists R3 \mid \text{Room}(R3))\ \text{Area}(R3, a)\ \|\ a=\text{"Fixed"}$

Cvpp1: $(\exists R3 \mid \text{Room}(R3))\ \text{Area}(R3, a) \wedge \text{Width}(R3, w) \Rightarrow \text{Length}(R3, l)\ \|\ l=a/w$

$(\exists R3 \mid \text{Room}(R3))\ \text{Area}(R3, a) \wedge \text{Length}(R3, l) \Rightarrow \text{Width}(R3, w)\ \|\ w=a/l$

This example indicates that if the surface of Room R3 must remain fixed, its length must be modified if its width is modified, and its width must be modified if its length is modified.
The satisfaction of a Cvp begins with the evaluation of Cvpv ; if the Cvpv is unsatisfied a Cvpp will be chosen. This will be repeated until the Cvpv is satisfied.

3.2 Consistency Maintenance of the Environment Model

Two types of consistency control may be considered : static consistency control and dynamic consistency control.
The static consistency control concerns the checking of a stable environment model with respect to the constraints-to-be-verified Cv. We must control : if each object is a such object designated by the nom of its class (its proprieties and its structure) ; if the geo-topological relations between different objects are right, etc.
The dynamic consistency control takes place when any entity is changed. A change may be a real change or the noise of IAMR's vision sensors. It may be consistent itself, but its induced results may be not consistent. So, to control the consistency of a change, we verify in the first if the initial modification is consistent, and then we propagate the modification by using the constraints-to-be-propagated and verify if the induced results are consistent.

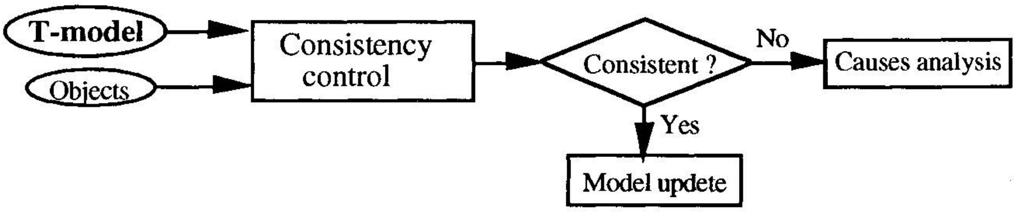


Figure 3. Objects consistency control schema

If the change and the results of its propagation satisfy all the constraints of T-model, we consider that this change is consistent and the environment model may be updated, otherwise we don't accept this change and it triggers its analysis module to detect the anomaly's causes (figure 3).

4. CONSISTENCY MAINTENANCE SYSTEM : PROVE

The consistency maintenance system PROVE which realises the above ideas is implemented on a SUN workstation and developed with the object language AIRELLE (a super-layer on the Le-lisp 15.2 of the INRIA) [8].

4.1 Constraint Representation

Different constraints are represented by three object classes : "Constraint" for Cp, "Constraint-V" for Cv, and "Constraint-VP" for Cvp.

The first and the second classes have three principal attributes : "Premise" representing a constraint's premise (the left part of the connector "=>" of Cp or "| |" of Cv), "Conclusion" representing a constraint's action (the right part of the connector), "Condition" which represent an optional trigger condition.

The third class is a combination of the two first classes, and it has two attributes : "Constraintv" which points to a constraint Cv, "ConstraintP" which points to a list of constraints Cp.

A constraint object may be presented in the form of a list composed of three parts, each of which begins by a reserved key word : "If" for the premise, "Such-as" for the trigger condition, and "Then" for the conclusion.

Constraint = <((If premise1 premise2 ...) (Such-as condition) (Then action1 action2 ...))>

Premise

A premise is represented by a triplet : <Object attribute value> as :

((Class x) Attribute (type y))

Where "Class" is the sort of x which limits the value domain of x, "Attribute" is an attribute of x, "type" is the attribute value type, and y is a variable representing the value of the attribute.

When the object or the attribute value is known, the premise becomes as following :

(O Attribute (type y)) or ((Class x) Attribute V)

Condition

A condition is a LISP-like logical expression such as : (equal x y).

Action

A constraint's actions have different representations as below :

(=> A Attribute V) for modifying of an attribute's value

(O Remove A) for removing "A" instance of "O" class

((O Create A) (=> A Attribute V)
{(=> A Attribute V)}) for creating a "A" instance of "O" class

(>= X V) for comparing a "X" variable with its "V" reference

Where : "=>" is the message sending function ; "A" is a known object ; "Attribute" is a real object attribute (but is not a variable) ; "O" is an object class ; "V" is the value of an object attribute, which can be a constant or formula ; "Remove" and "Create" are two reserved words. The first formula assigns "V" value to "Attribute" attribute of "O" object.

Examples :

The Cp1 and Cv1' can be rewritten as :

Cp1 : ((if ((Door D) Space-linked (ens S)))
(Then ((Passage 'Create P) (=> P 'Space-between S) (=> P 'Access D))))

Cv1 : ((If ((Space s) Length (float l))
((Space s) Width (float w))
((Space s) Area (float a)))
(such-as (and (and (not (Obtain s Length s Width s Area)) (not (Obtain s Width s Length s Area)))
(not (Obtain s Area s Width s Length))
(Then (= s (* l w))))

Cv3 : ((if ((Model M) Space (ens S)))
(Then ((mapc S) (Verify-Accessible var-inter)

Where the "Obtain (O A o1 a1 ... oi ai)" function is a procedure verifying if the value of A attribute of O object is the calculating result of the value of a1 attribute of o1 object, ..., and the value of ai attribute of oi object. "ens" is a data type of AIRELLE. "(mapc S)" showing that the following action will be executed in an ensemble of objects (as a "broadcast") (we do not use the method "applique" which allows to do also a "broadcast", in order to easily recuperate the action result).
The equivalent constraints are marked by an attribute "C-equivalent". For example the "C-equivalent" value of Cp2' is "Cp3, Cp4", the one of Cp3 is "Cp2', Cp4", and the one of Cp4 is "Cp2', Cp3".
The constraints and the corresponding objects are linked by the sort of each constraint which presents in the premise of a constraint.

4.2 PROVE's Running

The PROVE system contains four principal modules as illustrated in figure 4. When an object modification is received, in the first PROVE propagates, using the constraints Cp, this modification throughout the model to induce all its effects, then it refines this propagation by using the constraints Cvp, and finally the consistency of the modification and its induced results will be verified with respect to Cv. Therefore the propagation process of a modification is non-linear.

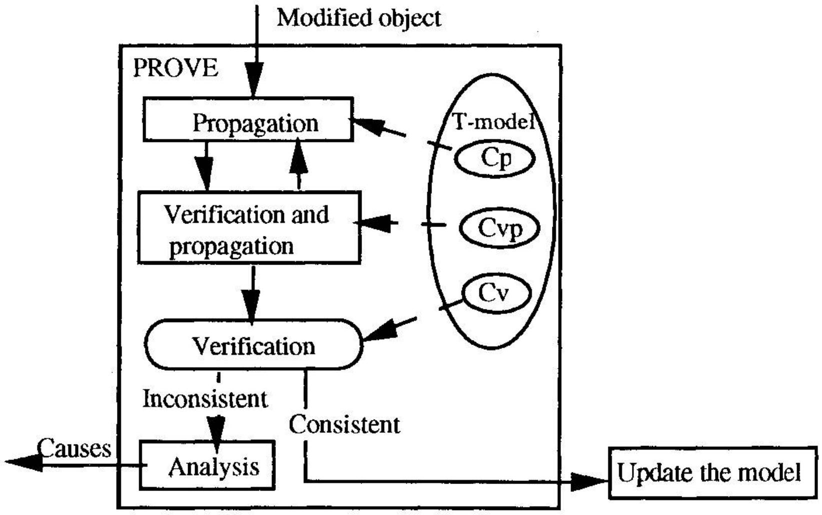


Figure 4. Structure of the consistency maintenance system

The **Propagation** module triggers Cp constraints in the order determined by the priority assigned to each constraint, using the "forward-chaining" control structure with the "width-first" search strategy. The modified objects or the new created objects are stored in a special objects class called DMODIF. This mechanism prevents from operating directly on the model : if the original change is considered unacceptable, the model will remain unaffected.

The **Verification and propagation** module proceeds in two stages. It begins by verifying the Cvpv of each Cvp constraints; if they are satisfied, the module pass to the next one.

If a Cvpi∈Cvp (its Cvpvi) is not satisfied the module immediately propagates a corresponding Cvppi constraint. This propagation may modify some data that retriggers the "Propagation" module and starts so a new operating loop. The looping continues until either the Cvp constraints are satisfied or some stop conditions are verified (for example when there are Cvp constraints that cannot be satisfied by the model during its processing state).

The **Verification** module consist in verify the Cv constraints. If all the constraints are satisfied the change signalled by the original message is considered acceptable and the model will be updated.

If a Cvi∈Cv constraint is unsatisfied, the change will be not accepted and an anomaly will be declared.

The **Analysis** module makes use of a mechanism inspired from ATMS [4], to identify the set of primary causes (objects initially modified) of each anomaly. The "Analysis" module treats as hypotheses the modifications contained in the message that triggers PROVE. It receives couples of the form (O, J) from the "Propagation" module, where O is a modified object and J the justifications for the modification. Since the object modifications in the original message are treated as hypotheses, the objects justify themselves. For each modified object the module computes a label L representing the hypotheses that have brought about the modification, by constructing a derivation tree using the concerned justifications. A hypothesis is self-labelled by definition and the modified objects are noted as combinations of (O, J, L).

5. CONCLUSION

We have presented a consistency maintenance system PROVE for a spatial object model in the context of our research project : IAMR environment simulation.

We qualify this system as a constraint-based system. Three types of constraints have been classified for the propagation (Cp, Cvp) of an objects change as well as for the objects verification (Cv). However, the constraints in our context are different from the ones in CSP. On the one hand, the domain of variable value of our constraints is rather infinite than finite and discrete. On the other hand, the Cp constraints don't exist explicitly in CSP. Generally the constraints treated in CSP may be classified as Cv and Cvp constraints (for example, in the geometric constraint engine [5], only the Cvp constraints have been treated).

The integration of hypothetical reasoning, to analyse the origins of unsatisfied constraint provides the possibility of a collaboration with IAMR's module vision to identify an object perceived by its sensors.

REFERENCES

[1] M. AYEL et M.C. ROUSSET : La cohérence dans les bases de connaissances. Cepad, 1990.

[2] A. DECHTER, R. DECHTER : Belief maintenance in dynamic constraint networks. Proc. AAAI 88.

[3] J. JATTAR et J.L. LASSEZ : Constraint logic proramming. Proc of the 14th ACM POPL Symposium, Munich, West Germany, 1987

[4] J. de KLEER : An assumption-based TMS. Artificial intelligence. vol. 28, 1986.

[5] G. A. KRAMER : A Geometric Constraint Engine. Artificial intelligence. vol. 58, 1992.

[6] G. MASINI et Al. : Les langages à objets : langages de classes, langages de frames, langages d'acteurs. InterEdition, 1989.

[7] A. MEYSTEL : Autonomous Mobile Robots - Vehicles with Cognitive Control. World Scientific, 1991

[8] A. NICOLLE-ADAM : Le métalangage AIRELLE. Revue d'Intelligence Artificielle, vol 4, n°1, Hermes, 1990.

[9] F. ROSSI : Constraint satisfaction Problems in logic programming. SIGART news letter, Oct. 1988.

[10] M. STEFIK et D. G. BOBROW : "Object-Oriented Programming : Themes and Variations". The AI magazine, Vol. 6.1, 1986.

[11] G.J. SUSSMAN et G.L. STEELE : CONSTRAINTS - A language for expressing almost hierarchical descriptions, Artificial Intelligence, 14, 1980, P.1-39

[12] A. THAYSE, A. BRUFFAERTS etc, Approche logique de l'intelligence artificielle. vol.3, Dunod, 1990.

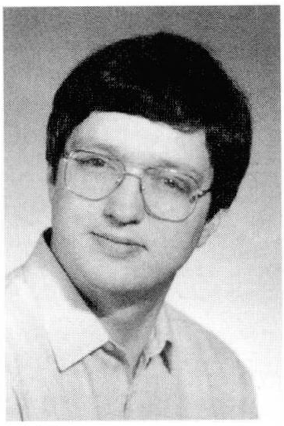
Modelling of the Oedometer Test by Neural Networks

Modélisation d'un essai oedométrique avec réseaux de neurones

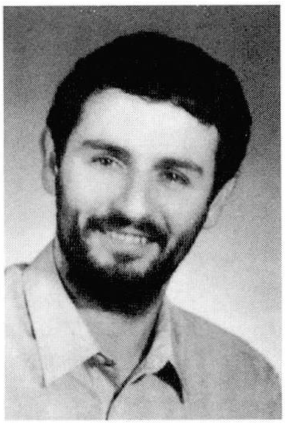
Modellierung von Oedometerversuchen mit neuronalen Netzen

Janko LOGAR
Civil Eng.
University of Ljubljana
Ljubljana, Slovenia

Goran TURK
Civil Eng.
University of Ljubljana
Ljubljana, Slovenia



Janko Logar, born in 1962, received his civil engineering degree in 1985 and M.Sc. degree in 1990 at the University of Ljubljana. His main area of interest is constitutive modelling and numerical methods in geotechnical engineering.



Goran Turk, born in 1963, received his civil engineering degree in 1987 and M.Sc. degree in 1990 at the University of Ljubljana. His research topics are: heat conduction and rheology of fresh concrete, reliability analysis, sensitivity analysis and neural networks.

SUMMARY
Constitutive modelling of non-linear material such as soil is a very difficult task. The development of artificial intelligence offers a new possibility in this field. A feed-forward neural network used in predicting the oedometer loading curve was trained by 40 oedometer curves obtained from tests made on marshland soil. The neural network was tested on 6 oedometer curves which were not included in the learning process. The oedometer curves and the clay parameter were evaluated. Good agreement between measured and predicted values was obtained, which implies that the neural networks can serve as an effective constitutive model as well as predicting sample behaviour.

RÉSUMÉ
Il est très difficile d'établir la relation entre contraintes et déformations pour des matériaux non linéaires, tels que les sols. Le développement de l'intelligence artificielle offre, avec les réseaux neuronaux, une nouvelle possibilité dans le domaine des modèles constitutifs. Les prévisions de la courbe oedométrique ont été établies avec un chargement, basé sur apprentissage, de 40 essais oedométriques du sol du Marais de Ljubljana. Les résultats ont été vérifiés avec 6 essais, non incorporés dans le processus de l'apprentissage. La bonne concordance entre les résultats expérimentaux et calculés permet d'utiliser le réseau neuronal comme modèle constitutif pour prévoir le comportement de l'échantillon.

ZUSAMMENFASSUNG
Das Aufstellen konstitutiver Beziehungen ist für nichtlineare Materialien, z.B. Erdstoffe, keine einfache Aufgabe. Die Entwicklung der künstlichen Intelligenz bietet als eine neue Möglichkeit den Gebrauch neuronaler Netze. Das Lernen von 40 Oedometerversuchen mit Erdstoffen aus dem Moor von Ljubljana ergab die Resultate für die Verdichtungskurve. Das neuronale Netz wurde mit 6 Oedometerversuchen, die nicht in den Lernprozess eingeschlossen waren, geprüft. Die Verdichtungskurven und die Parameter des Cam-Clay-Models wurden verglichen. Die Übereinstimmung der Messungen mit den Voraussagen des neuronalen Netzes zeigt, dass es als ein konstitutives Modell dienen kann, und die Voraussage des Erdprobenverhaltens ermöglicht.

1. INTRODUCTION

Oedometer tests are widely used for the determination of the compressibility characteristics of soils. For this reason a large amount of test data is available. We believe that the development of artificial intelligence enables us to convert these old files to active knowledge by using neural networks. Some results of the first part of our study on the use of artificial neural networks as a knowledge-based constitutive model for uniaxial soil behaviour are presented in the paper.

2. DATA BASE

During the last thirty years extensive in-situ and laboratory testing of Ljubljana marshland soils has been performed, mainly for the purpose of road and highway construction in very difficult soil conditions. Ljubljana marshland subsoil typically consists of two very soft silty and clayey layers underlain by stiffer silts or clays including sand and/or gravel. From the constructional point of view, the upper, up to 9 m thick, normally consolidated silty soil layer of very high compressibility (MH) is the most critical. The results of 46 oedometer tests made on samples of this silty soil were available. In addition to the oedometer curve $\sigma'-e$ (σ' is the effective vertical stress, e is the void ratio), for each sample the following characteristics are known: initial void ratio, e_0, natural water content, w_0, liquid limit, w_L, plastic limit, w_P, plasticity index, I_P, consistency index, I_C, specific weight, γ, and depth z from which the sample was taken. The extreme and the average values of these characteristics are shown in Tab. 1. Consistency characteristics of soil samples are presented in the plasticity chart (Fig. 1).

Parameter	Minimum	Maximum	Average
e_0	1.191	4.989	2.66
w_0 [%]	68.2	206.3	100.8
w_L [%]	55.0	155.9	92.1
w_P [%]	37.0	82.6	48.9
I_P [%]	11.4	81.0	43.2
I_C	-2.97	0.47	-0.40
γ [kN/m^3]	12.8	15.9	14.5
z [m]	1.25	8.15	3.9

Table 1 The extreme and the average values of soil characteristics.

Oedometer curves, characterized by z, w_0, w_L and w_P, are given by 5 to 7 pairs of test stress σ' and corresponding void ratio e (Fig. 2). Most samples were tested in the stress range from 0 to 160 kPa. The maximum stress applied to any of the 46 samples was 300 kPa. Based on the collected data the table of data was formed as an input to the neural network training algorithm. Each row of the table corresponds to one point of one particular oedometer curve and represents an input-output pair. The following data were provided for each input-output pair:

z depth from which the sample was taken,
w_0 natural water content,

w_0 **natural water content,**
w_L liquid limit,
w_P plastic limit and
σ' testing effective vertical stress

as input parameters and

e corresponding void ratio as the output parameter.

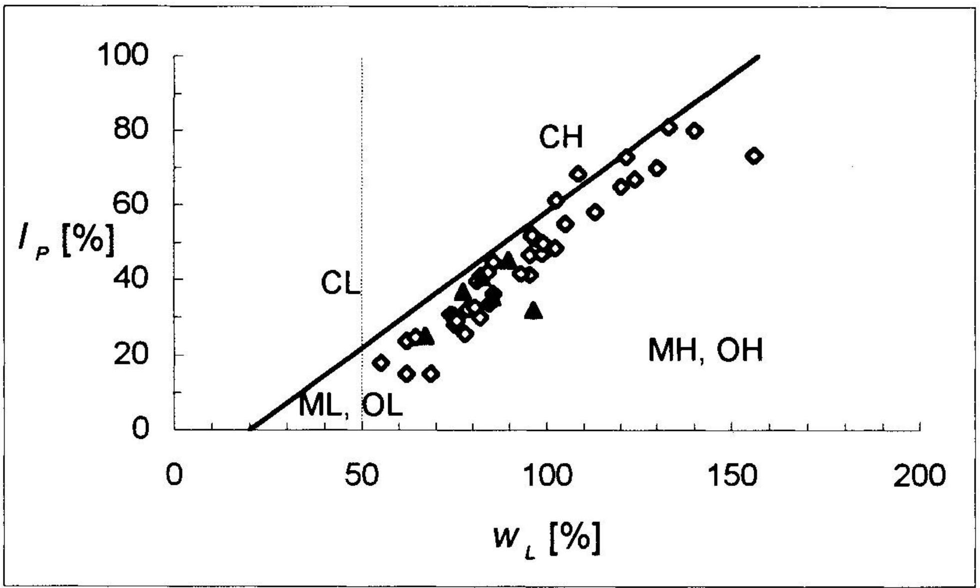


Fig. 1 Consistency characteristics of samples in plasticity chart.

Parameters I_P, I_C and γ were not included in the data base since they may be evaluated from the natural water content and consistency limits by the following equations:

$$I_P = w_L - w_P$$

$$I_C = \frac{w_L - w_0}{I_P}$$

$$\gamma = \frac{(\gamma_s + \gamma_w e_0 S_r)}{1 + e_0} = \frac{(1 + w_0)\gamma_w S_r}{(S_r \gamma_w / \gamma_s + w_0)} \approx \frac{(1 + w_0)\gamma_w}{(\gamma_w / \gamma_s + w_0)}$$

where γ_s, γ_w, S_r are the specific weight of solid particles, the specific weight of water, and the degree of saturation, respectively. The degree of saturation S_r for the studied material is approximately equal to one.

The geological stress $\sigma'_p = \int_0^z \gamma(\zeta) d\zeta$ was substituted by the depth of sample z, since the estimation of the depth is considerably more accurate than the estimation of σ'_p due to the unknown exact distribution of γ. This substitution was feasible because the samples of the treated soil were taken from the upper layer and because the water table is practically at the surface, which makes the distribution of γ similar for all boreholes.

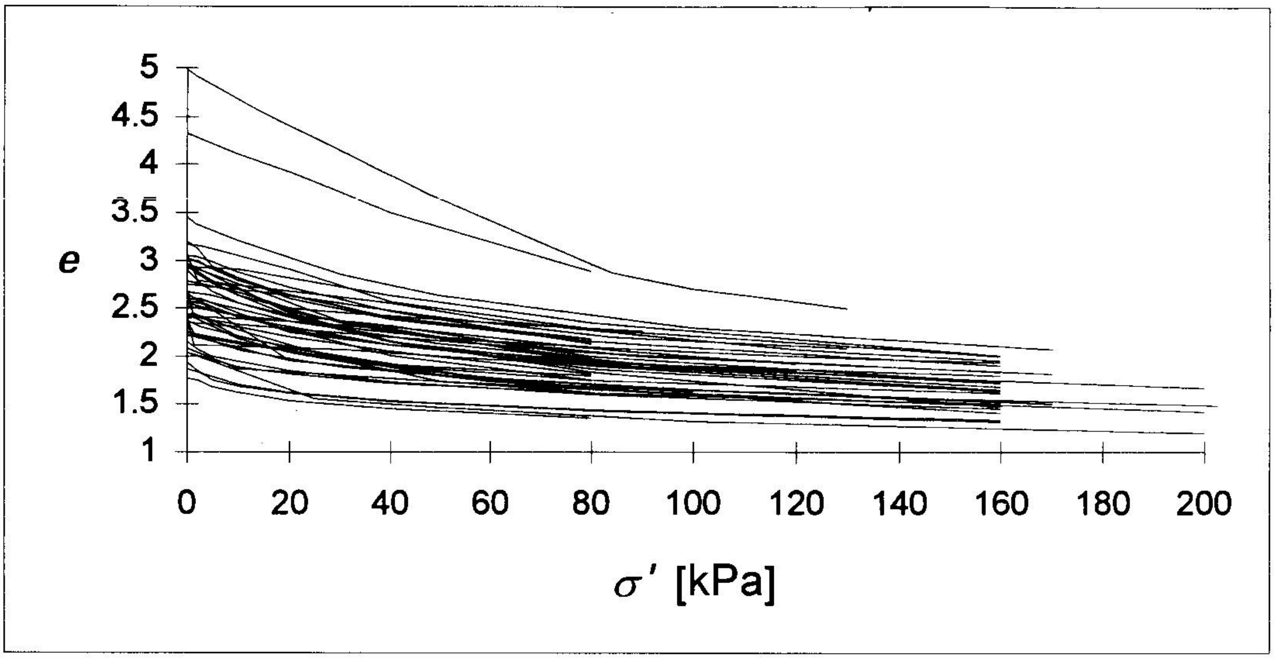


Fig. 2 Oedometer curves used in this study.

The available data were split into two parts. Forty oedometer tests, represented by 203 input-output pairs, were used as training data set. The remaining six oedometer tests were selected to test the performance of trained neural networks.

The preliminary calculations had shown that predicted oedometer compression lines did not follow the well known behaviour of soil sample in uniaxial compression conditions. For stresses greater than the pre-consolidation pressure the compression line is usually a straight line (if plotted in the logarithmic scale). The oedometer compression lines predicted by the neural network did not exhibit such behaviour. Their plots in the logarithmic scale were curved from zero to the final stresses (Fig. 3). To improve the solution, a number of additional data points were linearly interpolated between the measured values in the straight part of oedometer compression lines. This extended data base consisting of 408 input-output pairs was used for the training of neural networks.

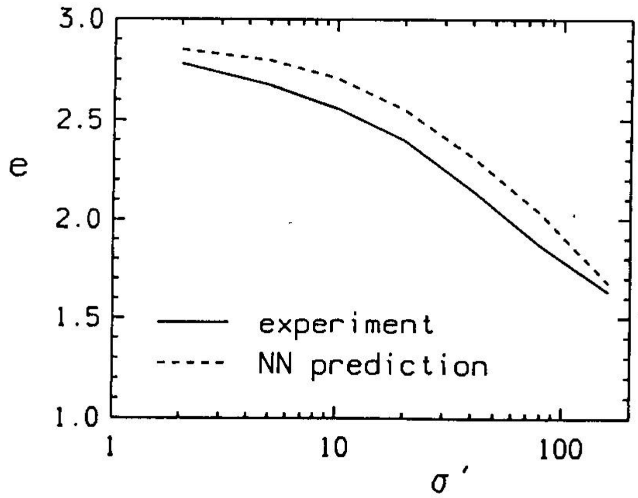


Fig. 3 Observed and preliminary determined oedometer compression lines.

3. NEURAL NETWORKS

An approximation of an unknown mapping $f(\mathbf{X})$ of several variables is sought. There are two possible ways, how to solve the problem:

- Traditionally the problem is solved in two steps. Firstly, an approximation function $g(\mathbf{X},\mathbf{C})$ is chosen. Subsequently, unknown parameters $\mathbf{C}$ are evaluated by the least squares method.
- Alternatively, the neural network may be used to approximate the unknown mapping. For this purpose a feed-forward neural network is usually used.

The general characteristics of neural networks which naturally act as associative memory, are able to generalize, and are highly fault tolerant, imply that the alternative is often better than the traditional method. Moreover, if there are many independent variables, the choice of the approximation function is a very difficult task. As a result, several authors ([1], [3], [5], [8], [10], [12]) reported on their successful use of neural networks in the approximation of functions. It has been proven that any continuous mapping can be approximated by a network with at least one hidden layer [4], [6].

In our research the feed-forward neural network was used. One or several hidden layers contained different number of neurons. A typical neural network is shown in Fig. 4. The activation function was chosen to be a sigmoidal with no bias term

$$\mathrm{sig}(x) = \frac{1}{1+\mathrm{e}^{-x}}$$

In training procedure a classical back-propagation algorithm was used [11]. In order to improve the efficiency of learning, the adaptive step size algorithm [7] or simulated annealing may be used [9], [13]. To avoid a possible over-learning for some experimental points, the experiments for which the error was lower than a specified fraction of allowed error were excluded from the learning.

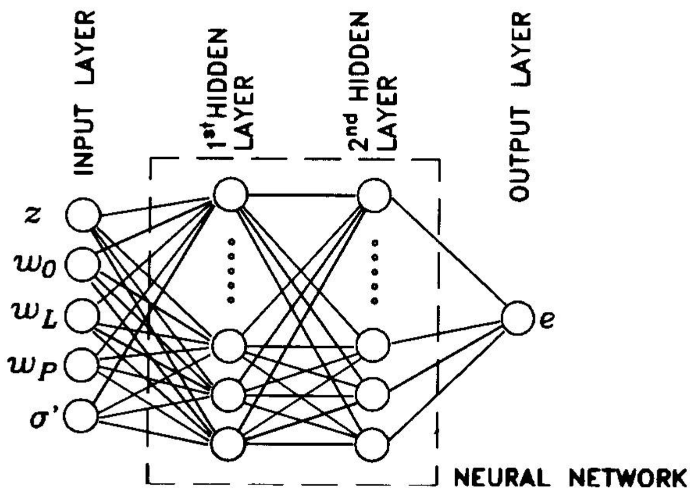


Fig. 4 Feed forward neural network

4. RESULTS

Neural networks of different size were tested. The number of hidden layers was taken to be 1, 2 and 3. Although the networks having two or three hidden layers learned the mapping rapidly, the preferred number of hidden layer is one. Networks with two and three hidden layers lacked the ability to generalize. Therefore, only the networks with one hidden layer and different number of neurons were used.

Firstly, the network was trained using 408 training input-output pairs. After the networks had been trained, the success of the training was tested using a set of testing data. One of the oedometer curves was eliminated from the testing set, since the error was notably larger than in the other five oedometer curves. It was later found that this had occurred because of an evident experimental error. Maximum and average normalized errors in predicting the void ratio are shown in Tab. 2. Considering the accuracy that is usually met in geotechnical engineering, the errors are relatively small. However, the results were considerably improved if the augmented learning was performed. When using the basic learning, the oedometer curve was predicted from z, w_0, w_L and w_P only. Additional data could be obtained from the classification tests, i.e. e_0, which represents the void ratio at the beginning of the test ($\sigma' = 0$). Therefore, the training of the network using this additional input-output pair (z, w_0, w_L, w_P, $\sigma' = 0$ and e_0) may proceed. The performance of the neural network trained by the augmented training was better. The maximum error decreased in most cases from about 30% to a little more than 10%. Similarly, the average error decreased from more than 10% to less than 5%.

No. of neurones	Basic learning		Augmented learning	
	Δe_{max} [%]	$\overline{\Delta e}$ [%]	Δe_{max} [%]	$\overline{\Delta e}$ [%]
30	31.1	13.9	11.0	3.6
35	20.9	8.9	12.5	4.3
40	36.3	15.8	15.8	4.8
45	31.2	14.4	12.6	4.2
50	38.1	16.9	10.5	3.9
60	39.4	15.7	13.0	5.7
75	33.3	13.2	22.0	3.8

Table 2 The maximum and the average error in void ratio prediction on a testing set.

The analysis of the soil behaviour including in-elasticity and consolidation is often performed by the finite element method. One of the most successful material models is the Cam-Clay model. One of the parameters (λ) of the model is the slope of the normal compression line in the ($\ln \sigma' : e$) plane.

Parameter λ was evaluated from the actual testing curves and the curves obtained by neural networks. The normalized errors are larger than in the case of void ratio. However, it appears that the largest errors correspond to extremely small values of λ. Thus, the absolute error of λ is relatively small. The normalized errors of λ for different number of neurons in the hidden layer are shown in Fig. 5. Bars represent the maximum and lines the average error values. In this case the error of λ does not exceed 12.0% and the average error is 6.5% for the network with 35 neurones in the hidden layer.

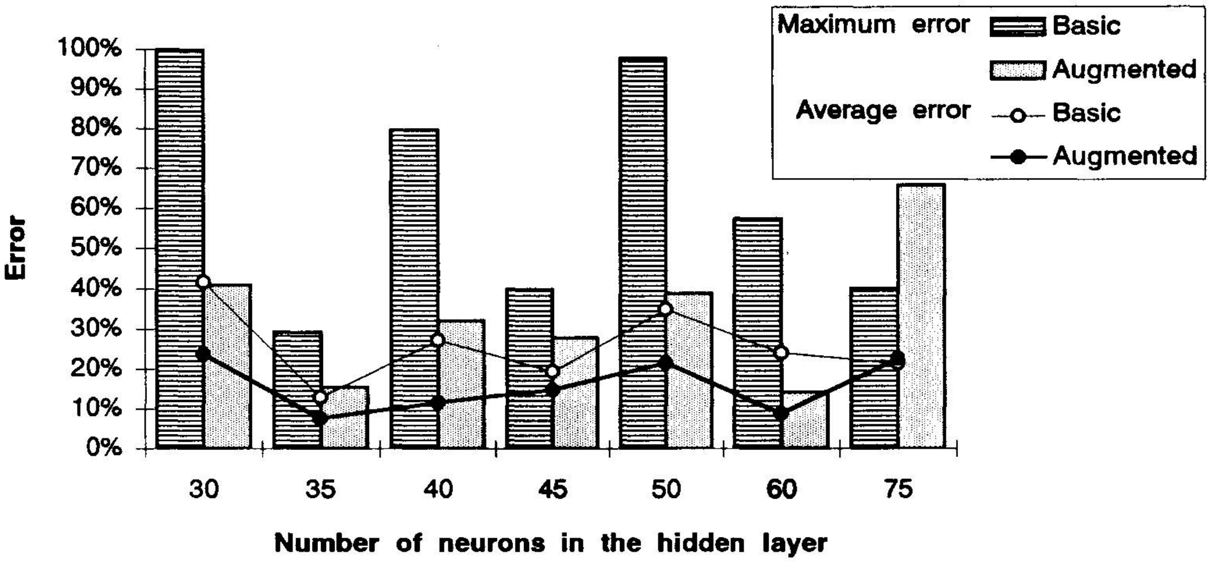


Fig. 5 Normalized errors of λ.

The overall performance of the neural network was optimal if the number of neurons in the hidden layer was 35. The actual oedometer curves and those obtained by the neural network with 35 neurons in the hidden layer are shown in Fig. 6. It is evident that in the case of the oedometer curve which was excluded from the testing set the discrepancy occurs because of an experimental error. Hence, the neural network may also be used as a warning unit embodied in the testing device. When the difference between the expected (neural network) and the measured values of void ratio is too large, the warning ought to be issued.

5. CONCLUSIONS

The neural networks were trained to approximate the oedometer curves obtained from the database of 46 laboratory experiments. Forty of them were employed as a training set, while the remaining six made up the testing set. The basic and the augmented training were performed.

The results have shown that the prediction of the void ratio is very reliable. The errors were generally lower than 15% for the basic training and lower than 5% for the augmented training. The error was larger in the case when the Cam-Clay parameter λ was predicted. However, if λ is to be used in a constitutive model, all available data (including testing data set) should be used in the training procedure.

One possible source of the error stems from the fact that the experiments had been performed by three different laboratories. Every laboratory has its own characteristics, and due to those differences the errors, which have to be eliminated, occur. Some of the gross errors of the measurement have already been found and eliminated by the neural networks.

The data concerning the unloading part of the oedometer tests were not as complete as those for loading. After the data are obtained, the network will be trained for that part, too. It is our goal to use a neural network as a constitutive model, replacing the explicit models, such as Cam-Clay, used in FEM codes today.

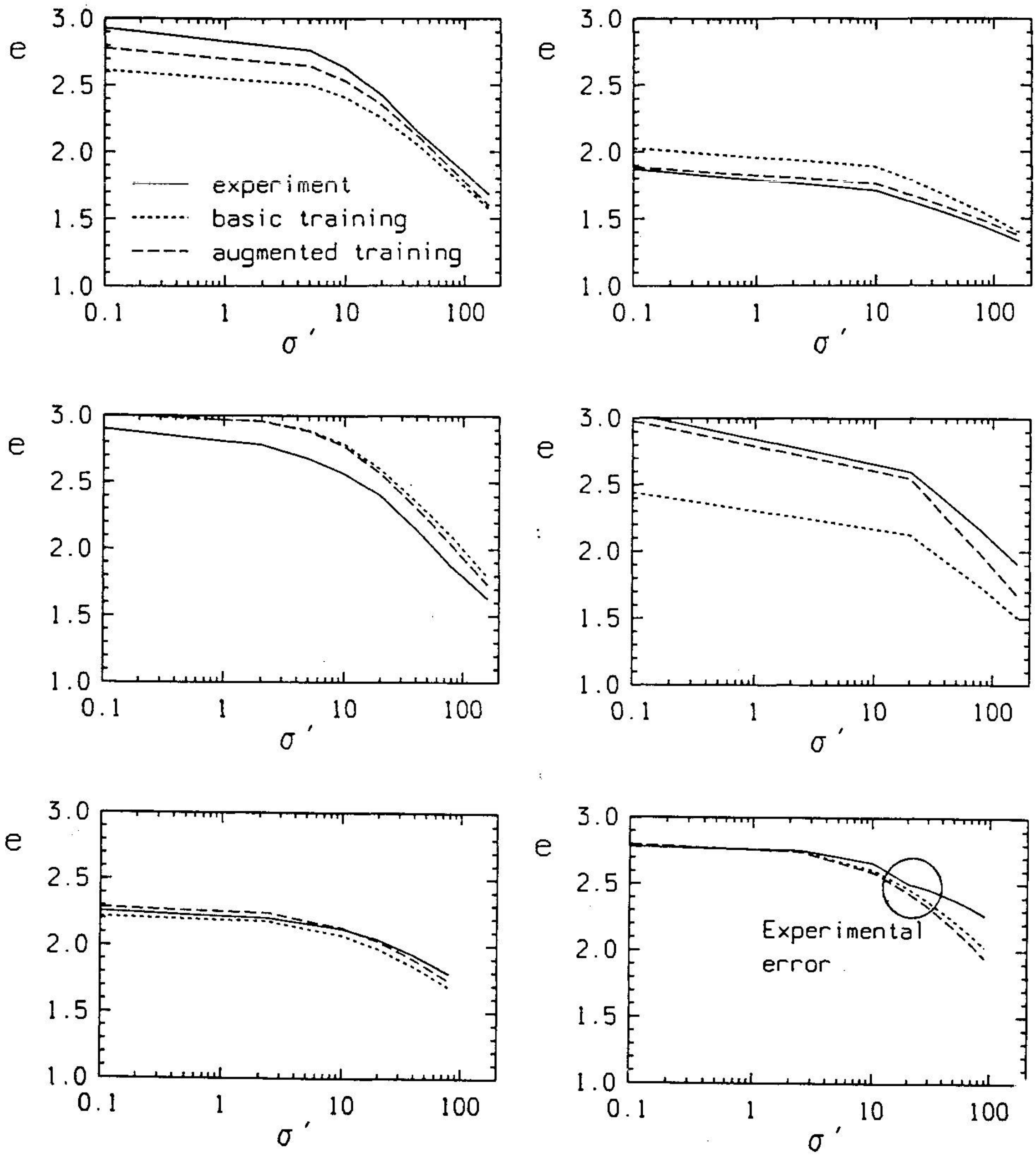


Fig. 6 Oedometer curves of the testing set.

ACKNOWLEDGEMENT

The experimental data for this research work were obtained from the records of three Slovenian soil mechanics laboratories: Geological Institute of Ljubljana, Institute of Materials and Structures and Department of Civil Engineering and Surveying. Their contribution is gratefully acknowledged. The authors wish to thank Mrs. Ana Gaberc, M.Sc., for the selection of suitable oedometer test results and for numerous helpful discussions regarding the details of oedometer testing and the presentation of test results.

The research was partially sponsored by the Ministry of Science and Technology of the Republic of Slovenia.

REFERENCES

1. G. CAMMARATA, S. CAVALIERI, A. FICHERA, L. MARLETTA, Neural Networks versus Regression Techniques for Noise Prediction in Urban Areas. WCNN '93, Vol. 1, pp. 237-240, 1993.
2. M. CAUDILL, C. BUTLER, Naturally Intelligent Systems. The MIT Press, Cambridge, MA, 1991.
3. I. FLOOD, N. KARTAM, Neural Networks in Civil Engineering. II: Systems and Application, Journal of Computing in Civil Engineering. ASCE, Vol. 8, No. 2, 149-162, 1994.
4. K.-I. FUNAHASHI, On the Approximate Realisation of Continuous Mappings by Neural Networks. Neural Networks, Vol. 2, pp. 183-192, 1989.
5. J. GHABOUSSI, J.H. GARRETT JR., X. WU, Knowledge-Based Modelling of Material Behaviour with Neural Networks. Journal of Engineering Mechanics, ASCE, Vol. 117, No. 1, pp. 132-153, 1990.
6. K. HORNIK, M. STINCHCOMBE, H. WHITE, Multilayer Feed-forward Networks are Universal Approximators. Neural Networks, Vol. 2, pp. 359-366, 1989.
7. J. JANAKIRAMAN, V. HONAVAR, Adaptive Learning Rate for Increasing Learning Speed in Backpropagation Networks. WCNN '93, Vol. 4, pp. 378-381, 1993.
8. T. POGGIO, F. GIROSI, Networks for Approximation and Learning. Proceedings of the IEEE, Vol. 78, No. 9, pp. 1481-1497, 1990.
9. W.H. PRESS, S.A. TEUKOLSKY, W.T. VETTERLING, B.P. FLANNERY, Numerical Recipes. Cambridge University Press, Cambridge, MA, 1992.
10. J.L. ROGERS, Simulating Structural Analysis with Neural Network. Journal of Computing in Civil Engineering, ASCE, Vol. 8, No. 2, 252-265, 1994.
11. D. E. RUMELHART, J.L. MCCLELLAND, Parallel Distributed Processing, Vol. 1: Foundations. The MIT Press, Cambridge, MA, 1986.
12. Z.P. SZEWCZYK, P. HAJELA, Damage Detection in Structures Based on Feature-Sensitive Neural Networks. Journal of Computing in Civil Engineering, ASCE, Vol. 8, No. 2, 163-178, 1994.
13. YIP, Y.H. PAO, A Fast Universal Training Algorithm for Neural Networks. WCNN '93, Vol. 3, pp. 614-621, 1993.

Leere Seite
Blank page
Page vide

A Knowledge-Based Systems Building Tool

Un outil pour la construction de systèmes à base de connaissance

Ein Shell-Programm für die Konstruktion von Expertensystemen

Silvia BRUNO
Civil Eng.
Univ. di Roma "La Sapienza"
Rome, Italy

Carlo GAVARINI
Prof. of Structural Mechanics
Univ. di Roma "La Sapienza"
Rome, Italy

Antonio PADULA
Researcher
Univ. di Roma "La Sapienza"
Rome, Italy

Federico VITTORI
Civil Eng.
Univ. di Roma "La Sapienza"
Rome, Italy

SUMMARY

The increasing interest in Artificial Intelligence as a powerful aid in solving civil engineering problems has suggested the realisation of a domain-independent tool for knowledge-based systems construction. The paper describes a fuzzy inference engine, which has been developed in order to build knowledge bases and to perform evaluations. Knowledge acquisition issues and approximate reasoning techniques are also illustrated and discussed.

RÉSUMÉ

L'intérêt croissant pour l'intelligence artificielle comme aide puissante dans la résolution des problèmes de génie civil, a suggéré le développement d'un outil bien adapté à la construction de systèmes à base de connaissances, indépendamment du domaine d'application. L'article décrit la réalisation d'un moteur inférentiel flou, qui peut être utilisé soit pour la création de bases de connaissance, soit pour obtenir des évaluations. Les problèmes les plus importants dans l'acquisition des connaissances et les techniques de raisonnement flou sont aussi illustrés et discutés.

ZUSAMMENFASSUNG

Das wachsende Interesse an der Verwendung künstlicher Intelligenz bei der Lösung von Problemen im Bauwesen hat die Realisierung eines vom Wissensgebiet unabhängigen Shell-Programms für die Konstruktion von auf wissensbasierten Systemen nahegelegt. Für diese Shell wurde eine sogenannte fuzzy-Inference Maschine realisiert. Fragen der Wissensakquirierung und der verwendeten Techniken werden aufgezeigt.

1. INTRODUCTION

Nowadays engineering problems are seen in terms of decision, management and prediction; solutions are seen in terms of faster access to more information and of increased aid in analyzing, understanding and utilizing the information that is available, and in coping with the information that is not. Aiming at modelling knowledge, the methodologies and concepts of Artificial Intelligence, as embodied today in the field of knowledge-based expert system, can potentially provide tools for dealing with these two elements, large amounts of information coupled with large amounts of uncertainty, which constitute **complexity**, the ground of many major engineering problems.

As pointed out in [1], a pressing need to improve the capability to acquire, assimilate, and codify knowledge that currently exists only in the form of personal engineering experience, judgements, and heuristics has arisen. Knowledge acquisition has thus become a crucial area for insuring continued progress in the development and application of expert systems.

A prevailing enthusiastic view is perhaps based on the increasing number of successful expert systems. However, problems in the development, maintenance, and enhancement of expert systems may severely restrict their integration into operational settings. There is no doubt about the fact that many of these problems involve faulty knowledge-base development methodologies. In fact, the developmental area most often cited as the "bottleneck" in expert system development is **knowledge acquisition**, i.e. the process of extracting and translating expert-level knowledge into rules that become the heart of an expert system.

The tool that is presented in this paper is intended to aid the domain expert in introducing his expertise, in form of rules, in a computer system, thus preventing the above mentioned problems from arising.

2. KNOWLEDGE ACQUISITION

Knowledge acquisition is concurrently referred to as the most important aspect of expert system development and the most problematic. It alternately has been tagged "knowledge extraction", "knowledge elicitation", and "knowledge acquisition". It refers to the "transfer and transformation of problem-solving expertise" from a knowledge source into rules to implement in a computer program.

The major knowledge acquisition stages are:

• **conceptualization:** it involves specifying how the primary concepts and key relationships among the concepts in the domain are depicted and related by domain experts;

• **formalization:** it requires mapping the recognized concepts, subtasks, relations, into formal representation mechanisms;

• **implementation:** it involves carrying formalized knowledge into an executable computer program connected with an inference engine. The primary goal of this stage is to develop a *prototype system*, which allows developers to test out design and representation mechanism decisions using only a small subset of the complete knowledge-base rules;

• **testing:** it requires that the prototype system be evaluated as to the efficacy of the system's formalization. To enable appropriate testing, developers must investigate and select an appropriate test scenario or problem set. Once the chosen scenario has been applied to test the system, results from testing are used to revise the prototype. Common revisions may include reformulating initial concepts, refining knowledge representation schemes and interrelationships, etc.

3. A KNOWLEDGE-BASED SYSTEMS BUILDING TOOL (KSBT)

AI techniques, supported by appropriate mathemathical frameworks, has successfully been applied to solving **knowledge-intensive problems**, (i.e., complex problems for which neither algorithms nor data are explicitely known) and, among them, structural engineering problems, notably in the field of seismic engineering. Uncertainties and ambiguities involved in structural performances have usually been treated by means of probability theory. However, as complexity often arises in engineering problems, and most decisions are made with a shortage of numerical evidence and depend on informed opinion, some uncertainty which are not random in nature may play important roles in the vulnerability and risk assessment of structures. Those uncertainties might be called *subjective uncertainties*, since they can be evaluated only by an engineer's experience and judgement.

Aiming to fulfil the potential offered by AI methodologies, a shell has been developed in order to provide domain experts with the possibility to build expert systems, directly within a friendly environment. The utilization of this tool for knowledge construction is completely domain-independent. As professional judgements are often expressed as verbal statements (e.g., "the structure is *moderately damaged*" or "the quality of the masonry is *poor*"), with an intrinsic vagueness or *fuzziness* which eludes the usual ordinary set representation, both traditional reasoning methodologies and *fuzzy logic* can appropriately be adopted in the representation of the domain knowledge and in the implementation of the inferential engine. *Approximate reasoning* in particular has been recognized to offer the proper mathematical support to dealing with such descriptive words or phrases. While traditional reasoning is mainly based on the manipulation of *symbols* representing arbitrary objects in the domain and on matching techniques (*symbolic elaboration of information*), approximate reasoning can deal with the *meaning* of propositions, thus being characterized by the ability to perform a *semantic elaboration of information*.

A brief overview of the basic concepts related to the latter methodology is provided in the following section.

3.1 Fuzzy logic and approximate reasoning

A *fuzzy logic* FL, that is, a logic based on *fuzzy set theory*, may be viewed, in part, as a fuzzy extension of a nonfuzzy multi-valued logic, i.e. a logic whose truth values are represented by real numbers in the interval [0, 1] (usually the *standard Lukasiewicz logic* L_1), which constitutes a *base* logic for FL. Truth values in FL are fuzzy subsets of the unit interval with linguistic labels such as *true*, *false*, *not true*, *very true*, *quite true*, *more or less false*, etc. The truth value set of FL is assumed to be generated by a context-free grammar, with a semantic rule assigning each linguistic term a *meaning* represented by a fuzzy subset of [0, 1].

One of the appealing features of fuzzy logic is its ability to deal with approximate causal inferences. According to Zadeh, *approximate* or, equivalently, *fuzzy reasoning* can be informally defined as a process by which a possibly imprecise conclusion is deduced from a collection of imprecise premises. More specifically, given an inference scheme "IF A THEN B" involving fuzzy propositions expressed in natural language, it is possible from a proposition A' that *matches only approximately* A, to deduce a proposition B' *approximately similar* to B, through a logical interpolation called *generalized modus ponens*. Such an inference is impossible in ordinary logical systems. The definition of a *possibility distribution* provides a natural basis for the representation of the meaning of fuzzy propositions, allowing its numerical computation and quantitative treatment (*quantitative fuzzy semantics*). Retranslation of possibility distributions in natural language can be accomplished by *linguistic approximation* procedures. Systematization of the use of words or sentences in a natural language for the purpose of an approximate characterization of the values of variables and their interrelations is accomplished by the concept of a *linguistic variable* (see [13]).

3.2 Use of the generalized *modus ponens*

The concept of a generalized *modus ponens* provides the basis for approximate deductions, allowing subjective judgements, once assigned a meaning and translated into linguistic values through a linguistic approximation procedure, to be treated as linguistic variables within a formal model of fuzzy inference.

Approximate inferences are often of the form:

knowledge: IF x is A THEN y is B
fact: x is A'

---

approximate conclusion: y is B'

($A, A' \in U$; $B, B' \in V$; U, V universes of discourse).

The *fuzzy conditional proposition* "IF x is A THEN y is B" represents a certain relation between A and B. A *translating rule* translates a fuzzy conditional proposition into a fuzzy relation in $U \times V$.

The inference mechanism that has been realized is based on implication coupled with Zadeh's compositional rule of inference (*max-min composition*). Use of approximate reasoning with fuzzy logic has involved the determination of an appropriate rule for implication among those commonly occurring in literature. The axiomatic approach proposed in [2] has been taken into account.

Of the most common implication relations, the so-called *arithmetic rule*, given as

$$
\begin{aligned}
\mathrm{R_a}(A, B) &= (\overline{A} \times V) \oplus (U \times B) \\
&= \int_{U \times V} (1 \wedge (1 - \mu_A(u) + \mu_B(v)))/(u,v)
\end{aligned}
$$

($u \in U$; $v \in V$).

is the only one meeting certain desirable prerequisites, postulated in order to assure an intuitive understanding of the nature of fuzzy deductions . This rule has thus been widely accepted, as it has appeared to satisfy intuition in many applications.

It is noted that the arithmetic rule is based on the implication rule of Lukasiewicz logic, i.e.:

$$a \rightarrow b = 1 \wedge (1 - a + b), \quad a, b \in [0, 1].$$

Inferences of the form:

knowledge: IF x_1 is A_1 AND x_2 is A_2 AND ... AND x_n is A_n THEN y is B
fact: x_1 is A_1' AND x_2 is A_2' AND ... AND x_n is A_n'

---

approximate conclusion: y is B'

($A_i, A_i' \in U_i$, $i = 1, 2, ..., n$; $B, B' \in V$),

have been translated into a fuzzy relation in $U_1 \times U_2 \times ... \times U_n \times V$ by an *extended arithmetic rule*, defined as

$$\mathrm{R_a}(A_1, A_2, ..., A_n; B) = (A_1 \cap A_2 \cap ... \cap A_n \times V) \oplus (U_1 \times U_2 \times ... \times U_n \times B)$$

$$= \int_{U_1 \times U_2 \times \ldots \times U_n \times V} (1 \wedge (1 - (\mu_{A_1}(u_1) \wedge \mu_{A_2}(u_2) \wedge \ldots \wedge \mu_{A_n}(u_n)) + \mu_B(v))) / (u_1, u_2, \ldots, u_n, v)$$

$(u_i \in U_i, i = 1, 2, \ldots, n; v \in V)$.

It can be shown (see for example [12]) that *the consequence B' is given as the union of the consequences B_i' of ordinary fuzzy reasoning* such that

knowledge: IF x is A_i THEN y is B
fact: x is A_i'

---

approximate conclusion: y is B_i' $\quad (A_i' \circ \mathrm{R_a}(A_i; B))$

Approximate reasoning gives therefore the possibility to cover in a satisfactory way a given domain of knowledge, by means of a relatively small number of rules (fuzzy in nature, and consequently with overlapping regions of applicability).

3.3 Description of KSBT

3.3.1 The inference engine

The shell whose implementation is in progress is intended to the construction of *rule-based fuzzy systems*, i.e. systems directly encoding structured knowledge in the numerical framework introduced in the previous part of this section. Such systems map input fuzzy sets A to output fuzzy sets B. They stores separate fuzzy rules and in parallel fires each of them to some degree for each input (Fig. 1). Outputs B_m' are first obtained as consequences of each of the fired rules, and suitable decisional criteria are then adopted in order to determine the result of the inference process, which is finally assigned a meaning through linguistic approximation.

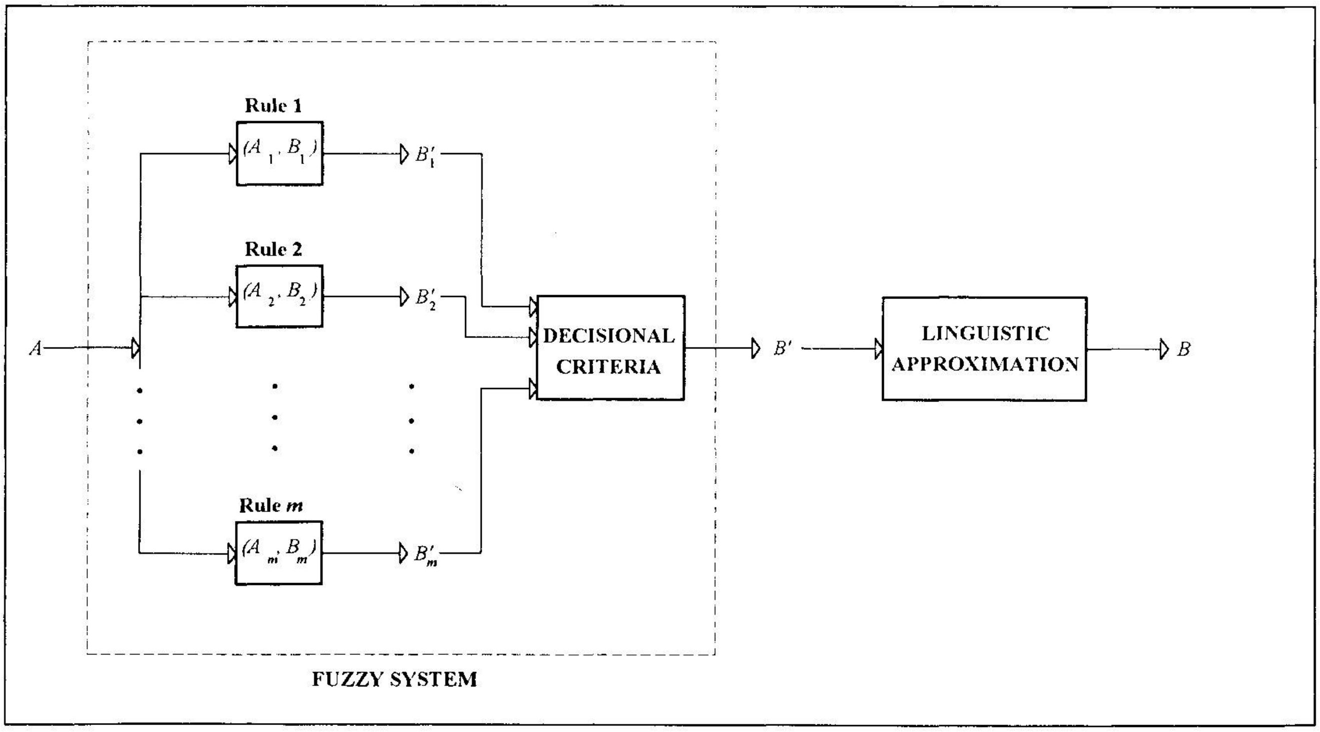


Fig. 1 Fuzzy-system architecture.

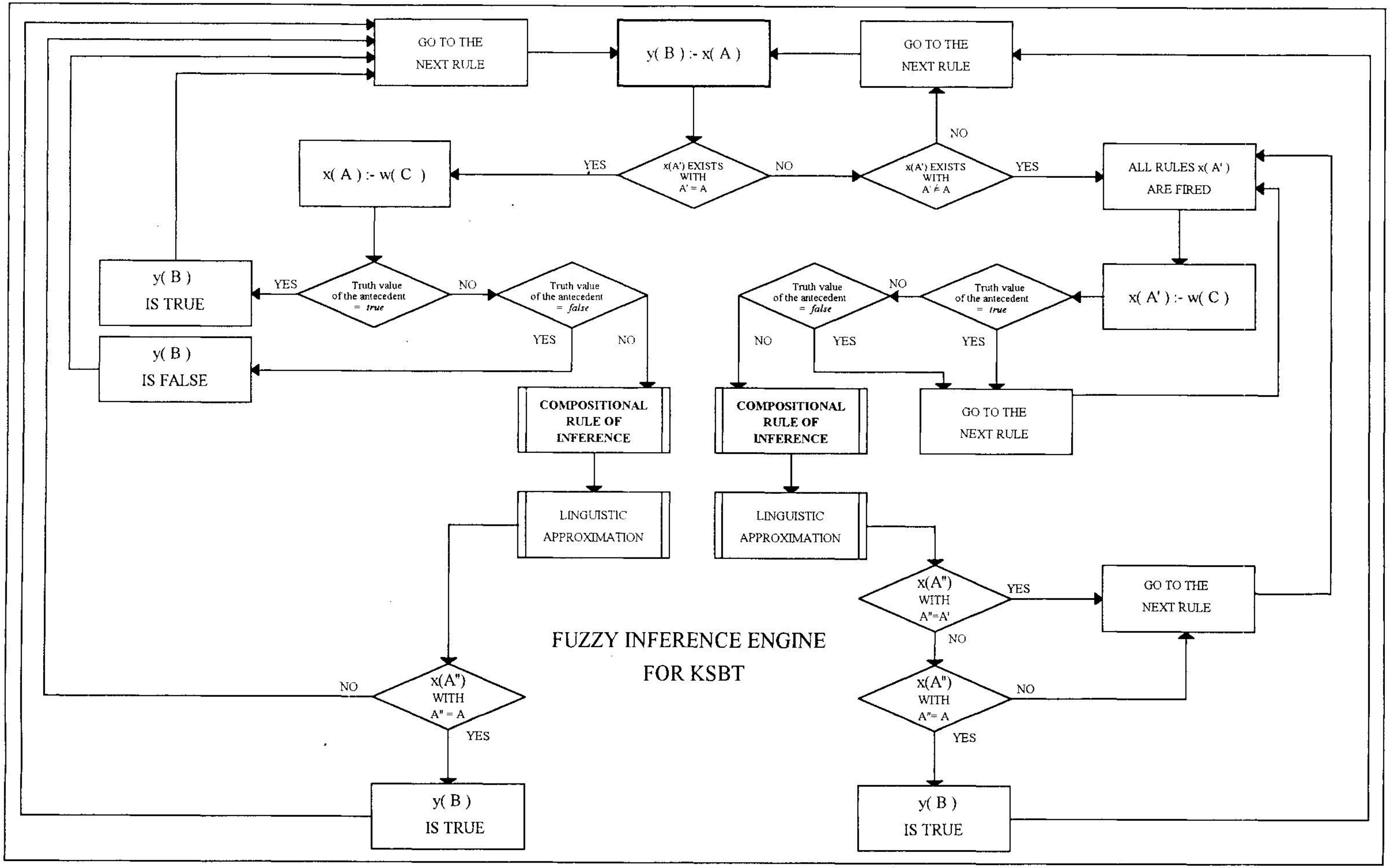


Fig. 2 Fuzzy inferential mechanism

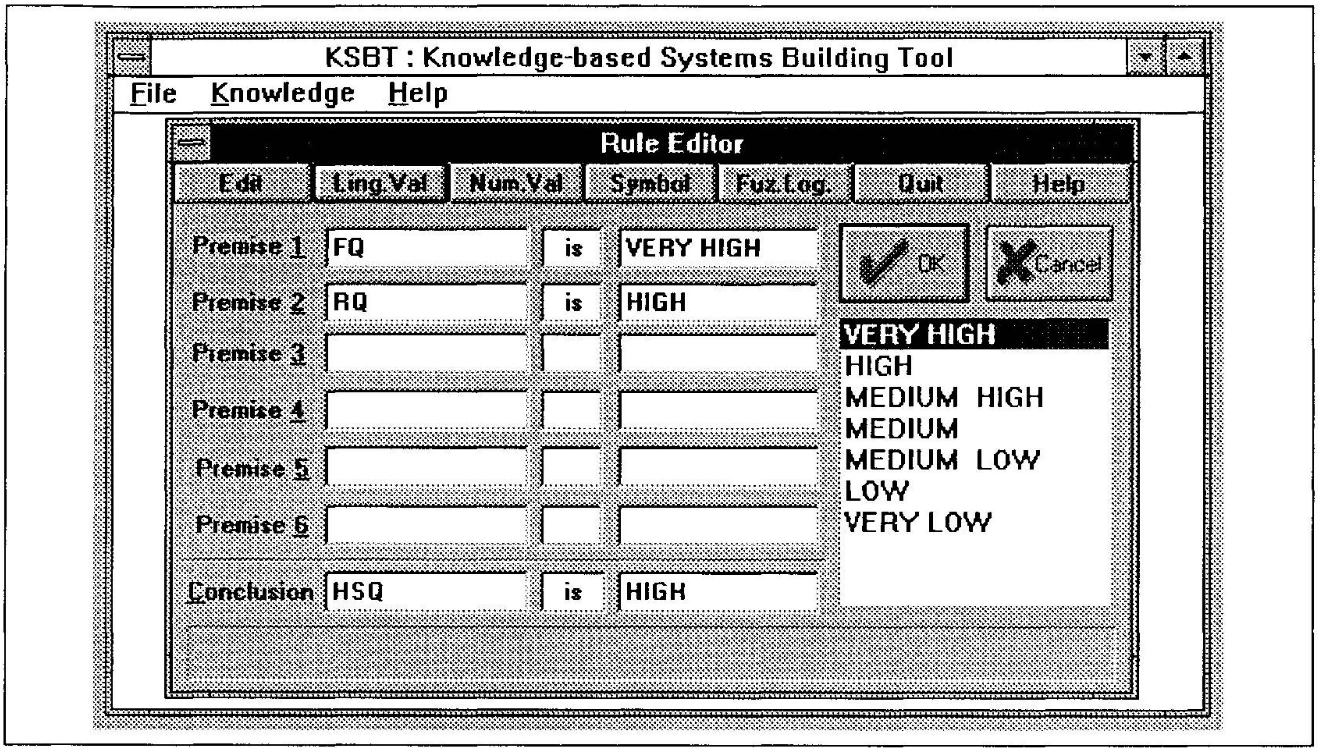


Fig.3 KSBT Rule Editor

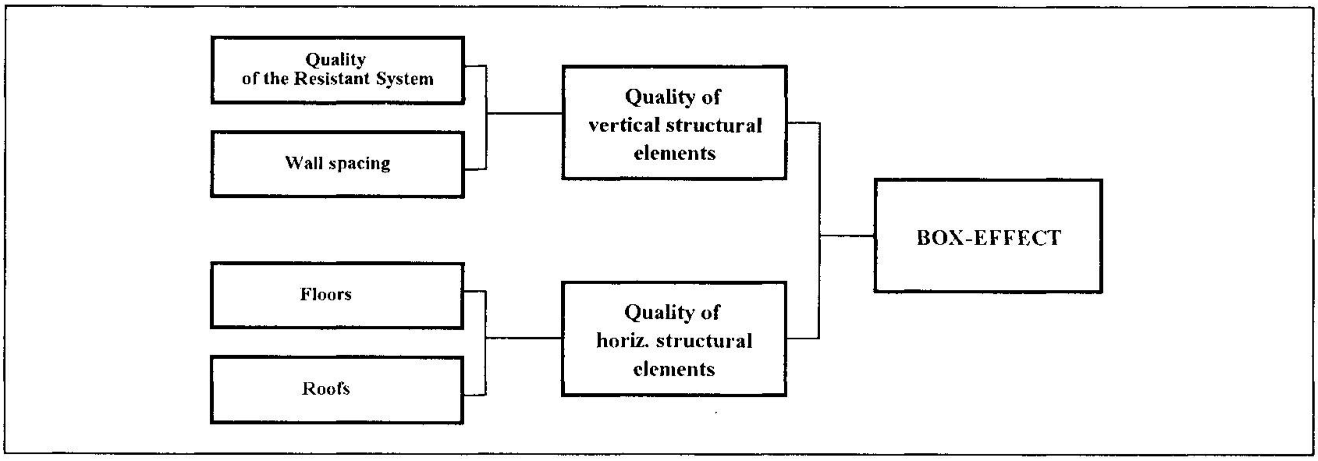


Fig.4 *Box-effect* evaluation

FQ \ RQ	VL	L	ML	M	MH	H	VH
VL	VL						
L	VL			L		ML	
ML	L		ML			M	
M				M	M	MH	
MH			M		MH		
H		MH		H			
VH						VH	VH

Fig.5 Linguistic matrix for HSQ assessment

Performing evaluations is accomplished by means of a *fuzzy inference engine*. The backward chaining mechanism is represented in Fig. 2. As it can be seen, symbol matching is used in case of coincidence of facts with antecedents of the rules, while the compositional rule of inference is activated in case of approximate matching between facts and antecedents. The compositional rule of inference module can also be utilized for building fuzzy knowledge bases.

3.3.2 Knowledge bases development

An environment for **writing**, **testing** and **using knowledge-based expert systems** has been developed, in order to assist domain experts in conveying their knowledge on a specified domain in form of rules. The following activities are supported:

- antecedent and conclusion variables identification;
- linguistic or numerical values specification;
- fuzzy or non-fuzzy rules definition and modification;
- fuzzy rule bases assisted development;
- use of knowledge bases for performing simulations and evaluations.

Both fuzzy and non-fuzzy rules are simply edited in a *MS-Windows* environment; furthermore, *fuzzy rules can be defined resorting to a tool which allows fuzzy knowledge bases construction assisted by fuzzy logic*.

Rule editing is performed through the *MS-Windows* dialog box shown in Fig. 3. Multiple antecedents composed of up to six fuzzy or non-fuzzy propositions linked by the logic connective AND can be specified. Addition and modification of fuzzy rules is at present accomplished with the use of a built-in vocabulary of terms in natural language.

The tool for supporting fuzzy knowledge base construction is intended to help the user in minimizing the number of fuzzy rules to define. It actually evidentiates the region of applicability of the rules themselves, thus preventing addition of redundant rules. Whenever the user decides to open the assisted session, the possibility distributions corresponding to the specified linguistic values are manipulated through the Lukasiewicz rule and the *max-min* composition. All possible valid inferences are subsequently obtained by activating the rules introduced in the knowledge base with input linguistic values varying within the whole predefined term set. The approximate conclusions (which, though mathematically correct being obtained by fuzzy calculus, may not represent properly the portion of the domain under consideration) are submitted to users' acceptance. Furthermore, a control module checks if new rules or results of valid inferences preserve the internal consistency of the knowledge base.

4. APPLICATIONS

Shell validation is being carried out in developing an expert system for seismic vulnerability assessment for masonry buildings. The process of fuzzy knowledge base construction can then be illustrated referring to the evaluation of the so-called "box effect", which might be identified as a sub-task of the main problem of vulnerability assessment. The box effect is defined as an estimation of the capability of a masonry building to behave as an effective earthquake-resistant structural system, with sufficiently rigid floors well connected to the vertical walls so as to prevent unresisted out-of-plane bending of the walls themselves. Variables involved in the box effect evaluation can be combined as shown in Fig. 4.

Representing this portion of knowledge of the domain of interest can be achieved following the steps listed below:

a. Let "IF the quality of floors is *very high* AND the quality of roofs is *high* THEN the quality of horizontal structures is *very high*" be the first rule to introduce in a new knowledge base. The user first formalizes it by associating identification symbols to the antecedent and consequent variables (a possible choice might be FQ, RQ, and HSQ, respectively), and selecting from a drop-down menu the linguistic values assumed by the variables within the rule itself.

b. At that stage, once the edited rule has been accepted, the user can either immediately define a second rule, or select the appropriate option for opening the fuzzy logic assisted work session. In this case, the compositional rule of inference module is activated, and the accepted rule is fired with those combinations of linguistic values for the antecedent variables leading to significant inferences. The current content of the knowledge base and the resulting inferences are displayed in two separate windows, so that the user can easily verify the correct behaviour of the system under construction when activated with possible different inputs.

c. The resulting inferences can either be accepted or rejected by the user. Resuming the example, the system proposes "IF FQ is *very high* AND RQ is *very high* THEN HSQ is *high*" as a first possible inference, obviously unacceptable for the problem at hand. The user refuses the suggestion, and is then asked to enter a new rule, with the same linguistic values for the antecedent variables FQ and RQ as in the rejected inference, and an appropriate linguistic value for the conclusion variable HSQ. The fuzzy value *unknown* has to be inserted, in case of ignorance of the value assumed by the conclusion variable, given the specified values of the antecedent variables. The new rule is then defined as follows: "IF FQ is *very high* AND RQ is *very high* THEN HSQ is *very high*"

d. The system goes back to step **b**, now considering a knowledge base made up of two rules.

e. The following possible inference is submitted to user's acceptance: "IF FQ is *very high* AND RQ is *medium high* THEN HSQ is *high*". As judged satisfactory by the user, this inference can be listed in the appropriate box.

f. The system goes back again to step **b**, still considering a knowledge base composed of two rules.

The session ends as soon as the domain of the problem at hand has sufficiently been covered by the rules in the knowledge base. A possible (not necessarily the only satisfactory one) final set of rules for HSQ asessment is represented in Fig. 5 by a 7×7 linguistic matrix. For what has been pointed out in the previous sections, this pattern ensures that an output (*unknown* included) can be associated to each possible pair of linguistic values for the input variables. For example, as seen above, appropriate rules are fired by the compositional rule of inference module in order to obtain the acceptable value *high* for HSQ from the fuzzy input values *very high* and *medium high* for FQ and RQ, respectively.

Writing rules with argument others than FQ and RQ for the antecedent variables, and HSQ for the consequent variable requires restarting from step **a**.

5. CONCLUSION

The main purpose of the tool presented in this paper is to provide domain experts, which usually neither are knowledge engineers nor have software developers capabilities, a friendly support for developing expert systems. Both fuzzy and non-fuzzy knowledge can be formalized and utilized, by means of an inference engine performing symbolic or semantic elaboration of information, according to the nature of the current input.

Implementation is being carried out by an appropriate utilization of *C++* and *PDC-Prolog* in a *MS-Windows* environment. Further development of the tool for fuzzy rule bases construction will lead to the possibility for expert users to modify existing membership functions, to add new ones, to adjust the inference mechanism by introducing suitable fuzzy logic operators. The implementation of an explanatory module in form of hypertext system is in progress.

Although the present stage of realization of the shell mainly focuses on the management of fuzzy information, extensions of its usability should be provided by integrating different mathematical frameworks for dealing with uncertainty, in an attempt to reproduce more efficiently the characteristics of real world situations.

REFERENCES

1. ARCISZEWSKI, T., ROSSMAN, L.A. (eds.), *Knowledge Acquisition in Civil Engineering*, ASCE, 1992

2. BALDWIN, J. F., Axiomatic Approach to Implication for Approximate Reasoning with Fuzzy Logic, *Fuzzy Sets and Systems*, **3**, 193 - 219, 1980.

3. BOOSE, J., GAINES, B. (eds.), *Knowledge Acquisition Tools for Expert Systems*, Academic Press, London, 1988.

4. BRUNO, S., C. GAVARINI, A. PADULA, Vulnerability and Damage Assessment of Structures Before and After an Earthquake: Use of Expert Systems and Fuzzy Set Theory, 1st Workshop of the European Group of Structural Engineering Applications of Artificial Intelligence (EG-SEA-AI), Lausanne, March 21-22 1994.

5. BRUNO, S., C. GAVARINI, A. PADULA, Seismic Evaluation of Buildings: a Global AI Approach, 2nd Franco-Italian Symposium of Earthquake Engineering: *Strengthening and Repair of Structures in Seismic Areas*, Nice, October 17-19, 1994.

6. DUBOIS, D., H. PRADE, *Fuzzy Sets and Systems: Theory and Applications,* Academic Press, New York, 1980.

7. FENVES, S. J., Expert systems: expectations versus reality, IABSE Colloquium: *Expert Systems in Civil Engineering*, Bergamo, 1989.

8. GAVARINI, C., Towards the Systematic Use of Expert Systems in Seismic Risk Reduction, Proceedings of the 10th WCEE, Madrid, July 1992.

9. KLIR, G. J., T.A. FOLGER, *Fuzzy Sets, Uncertainty, and Information*, Prentice-Hall, Englewood Cliffs, N. J., 1988.

10. KOSTEM, C. N., M. L. MAHER, *Expert Systems in Civil Engineering*, ASCE, 1986.

11. MCGRAW, K.L., K. HARBISON-BRIGGS, *Knowledge acquisition: principles and guidelines*, Prentice-hall, Englewood Cliffs, N. J., 1989.

12. MIZUMOTO, M. B., Multifold Fuzzy Reasoning a Interpolative Reasoning. In: *Fuzzy Sets, Neural Networks, and Soft Computing*, edited by R. R. Yager and L. A. Zadeh, ITP, VNR, NY, pp. 188-193, 1994.

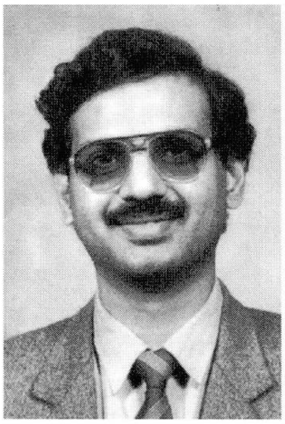
13. YAGER, R. R., S. OVCHINNIKOV, R.M. TONG, H.T. NGUYEN (eds.), *Fuzzy Sets and Applications: Selected Papers by L.A. Zadeh*, John Wiley & Sons, New York, 1987.

A Rebar Corrosion Decision System Using Machine Learning

Aide à la décision concernant la corrosion d'armatures par l'emploi d'apprentissage-machine

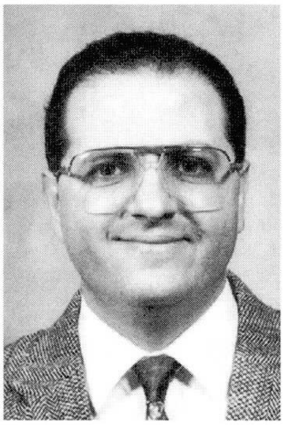
Eine Entscheidungshilfe zur Bewehrungsstahlkorrosion unter Verwendung von Maschinenlernen

Srinath NAGARAJA
Research Assist.
Kansas State University
Manhattan, KS, USA



Srinath Nagaraja, born 1966, received his BS and MS degrees from Bangalore Univ., India, in civil and structural eng. He is pursuing his PhD degree in civil eng. at Kansas State Univ. His area of concentration is expert systems application to structural and bridge engineering.

Hani G. MELHEM
Assist. Prof.
Kansas State University
Manhattan, KS, USA



Hani G. Melhem, born 1958, earned his M.S. and Ph.D. from the Univ. of Pittsburgh. He joined Kansas State Univ. in 1991 as an Assistant Professor. His areas of interest include fuzzy logic and machine learning applied to structural engineering problems.

SUMMARY
The infrastructure around the world is seriously affected by the problem of concrete rebar corrosion. This research consists of the application of a concept of Artificial Intelligence - Inductive Learning - to build a rebar corrosion expert system using machine learning programs. The resulting system could become a tool for assessing the extent of corrosion in a structure and predicting its service life. The structural examples that can be handled by the system include bridges and bridge components.

RÉSUMÉ
Dans le monde entier, les infrastructures sont mises sérieusement en danger par la corrosion des barres d'armature. Dans le cadre d'un projet de recherche, le concept de l'étude inductive par l'intelligence artificielle a été appliqué dans un système expert pour l'étude de la corrosion. Celui-ci permet d'estimer l'évolution de la corrosion dans un ouvrage d'art et de prédire la durée de vie de l'ouvrage. Il y a lieu de citer des applications possibles pour les ponts et les éléments de ponts.

ZUSAMMENFASSUNG
Weltweit sind Infrastrukturbauten durch das Problem der Bewehrungsstahlkorrosion ernstlich gefährdet. In einem Forschungsprojekt wurde das Konzept des induktiven Lernens aus der Künstlichen Intelligenz in ein Korrosions-Expertensystem übertragen. Damit können das Ausmass der Korrosion im Bauwerk bestimmt, und seine Nutzungsdauer vorhergesagt werden. Mögliche Anwendungsbeispiele sind Brücken und ihre Bauteile.

1. INDUCTIVE LEARNING APPROACH

The most difficult task in expert system development is knowledge acquisition. Recent research in artificial intelligence has resulted in several machine learning techniques that allow automatic generation of the knowledge base, the main component of an expert system. This enables merging information of both qualitative and quantitative nature using available experimental data and case studies. These techniques are supported by well formulated theories and learning-type algorithms. The Information theory is one of the most popular.

1.1 Inductive Learning and its Requirements

The successful use of induction requires having the appropriate examples with corresponding attributes, a set of classes, and a suitable induction algorithm (Hart, 1986). The examples or training set form the basis of the induction process. An incomplete or inadequate set of examples is likely to result in poor rules. The attributes are the characteristics which describe the examples and enable comparison to be made between the different examples. Some of the characteristics form the factors which influence the rules. The classes represent the decisions or classifications made by the domain experts. The inductive algorithm is the method which the computer program uses to induce rules from the training set. A summary of various inductive learning techniques and their applications has been done at Kansas State University [1].

1.2 C4.5 Machine Learning Algorithm

The most important algorithm that has been used in the C4.5 programs[2] is the process of generating an initial decision tree from a set of training cases. The evaluating criteria for the ID3 algorithm which is the forerunner of C4.5 is called *gain*, defined by Eqn. 3 below. Suppose S is set of cases, let $freq(C_i,S)$ stand for the number of cases belonging to class C_i. The notation for denoting the number of cases in a set S is given as $|S|$. To find the expected information from a message pertaining to class membership, we sum over the classes in proportion to their frequencies in S, giving

$$info(S) = -\sum_{j=1}^{K} \frac{freq(C_j,S)}{|S|} \times \log_2\left(\frac{freq(C_j,S)}{|S|}\right) \text{ bits.} \qquad (1)$$

When applied to a set of training cases, *info(T)* measures the average amount of

$$info_X(T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} \times info(T_i) \qquad (2)$$

information needed to identify the class of a case in T. (This quantity is also known as the *entropy* of the set S). Considering a similar measurement after T has been partitioned in accordance with the n outcomes of a test X, the expected information requirement can be found as the weighted sum over the subsets, as The quantity

$$gain(X) = info(T) - info_X(T) \qquad (3)$$

measures the information that is gained by partition T in accordance with the test X. Although *gain criterion* gives quite good results, it has a serious deficiency-it has a strong bias in favor of tests with many outcomes. The bias inherent in the gain criterion can be rectified by a kind of normalization in which the apparent gain attributable to tests with many outcome is adjusted. The *gain ratio criterion* selects a test to maximize *gain ratio*, subjected to the constraint that the information gain must be large-at least as great as the average gain over all the tests examined.

2. CORROSION OF REBAR IN CONCRETE

Corrosion of steel in concrete structures exposed to chloride has become a major problem for bridge and structural engineers. This problem is evidenced by large scale premature failures of reinforced concrete structure in a fraction of their design life. Many of these failures have been attributed to the corrosion of the reinforcing steel and are sufficiently severe to require refurbishing or replacement of the structure [3].

When steel is exposed to an aerated alkaline solution corresponding to that found in the pores of well formulated concrete (whose pH is about 13), it corrodes to form a solid corrosion product. This product (an iron oxide) forms continuously, adherently and coherently on the metal surface and serves to stifle any further corrosion. The corrosion mechanism can be schematically represented as in Fig. 1. There is a large amount of data pertaining to corrosion of rebar in concrete. The data contains a large number of variable dependencies and cannot be processed manually.

3. DEVELOPMENT OF LEARNING SYSTEM USING CORROSION DATA

A decision system for the assessment of corrosion of steel bars in reinforced concrete structures is being developed at Kansas State University[4]. The system is composed of two main modules (Fig. 2). The first module determines the final decisions, namely, the degree of corrosion and type of remedial action. The second module predicts the serviceability life of the structure. The latter takes into consideration the important aspects of deterioration rate of rebar, the acceptable limit of deterioration, and the loss in load carrying capacity of the structure, in addition to several other important factors. The system is developed and implemented with different types of data, both numeric and symbolic in nature.

Several attributes or corrosion variables have been chosen to support the knowledge base structuring. These are broadly classified as material factors, structural factors, and environmental factors. These factors and their attributes are shown in Fig. 3. Data supporting these attributes form the basis for pattern recognition in the machine learning process. Besides the data are pre-processed using the CORRODE (Version 1.0) in order to get the decision pertaining to the servicibility life module[5].

4. CONCLUSION

Using earlier prediction models which are based on experimental results, various investigators have found it difficult to estimate the corrosion intensity in a structure. At the present time there are no reliable methods available which could use the existing prediction models. Even the most accurate test results have some degree of uncertainty. In order to remove anomalies in corrosion intensity values, machine learning techniques can be applied to understand and model the serviceability life. Using such techniques, a decision system for Rebar corrosion has been proposed and is presently under development. This type of system could help in an on-site monitoring of corrosion intensity of damaged structure.

REFERENCES

1. MELHEM H.G. and NAGARAJA S., Applications of Machine Learning in Civil Engineering. Submitted to Civil Engineering Systems, 1994.

2. QUINLAN R., C 4.5: Programs for machine learning. Morgan Kaufmann Pub., 1993.

3. TONINI, D. E. (ed.), Chloride corrosion of steel in concrete. STP-629, ASTM Publ., 1976.

4. NAGARAJA, S., A Rebar Corrosion Decision System using Machine Learning. Ph.D. Dissertation (in preparation).

5. PURVIS R., BABAEI K., CLEAR K.C AND MARKOW M.J., Life-cycle Cost Analysis for Protection and Rehabilitation of Concrete Bridges Relative to Reinforcement Corrosion. SHRP-S-377 Report, Strategic Highway Research Program, Washington D.C, 1994.

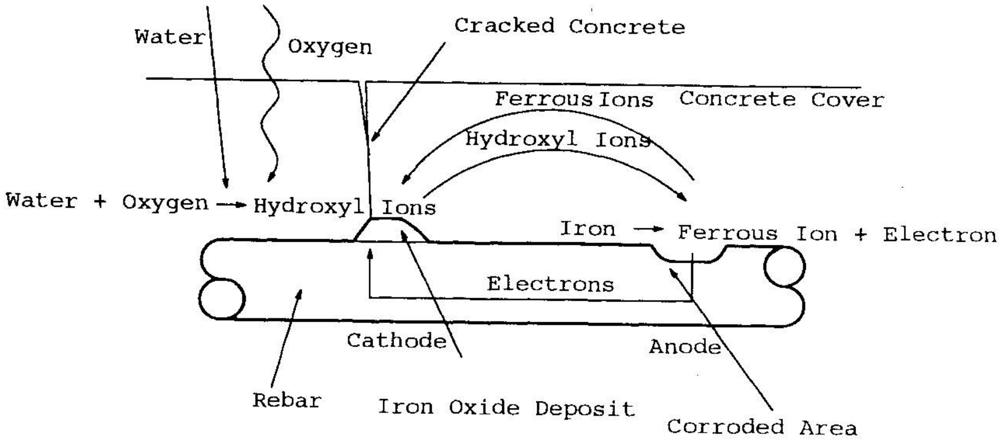


fig. 1 Corrosion Mechanism

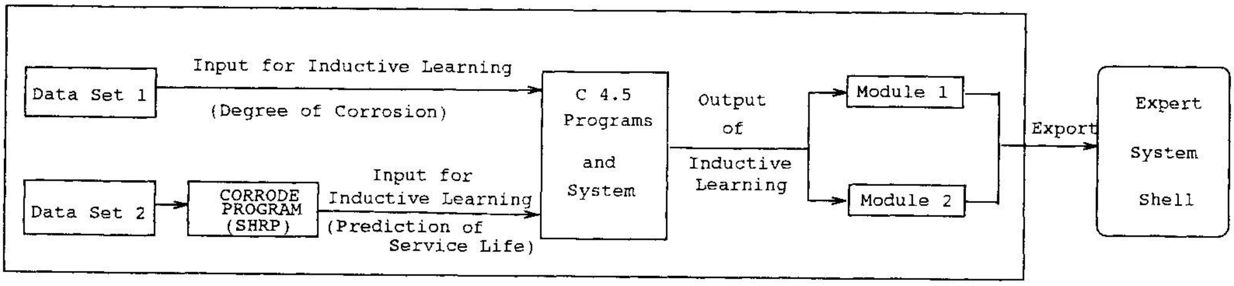


fig. 2 Architecture of System

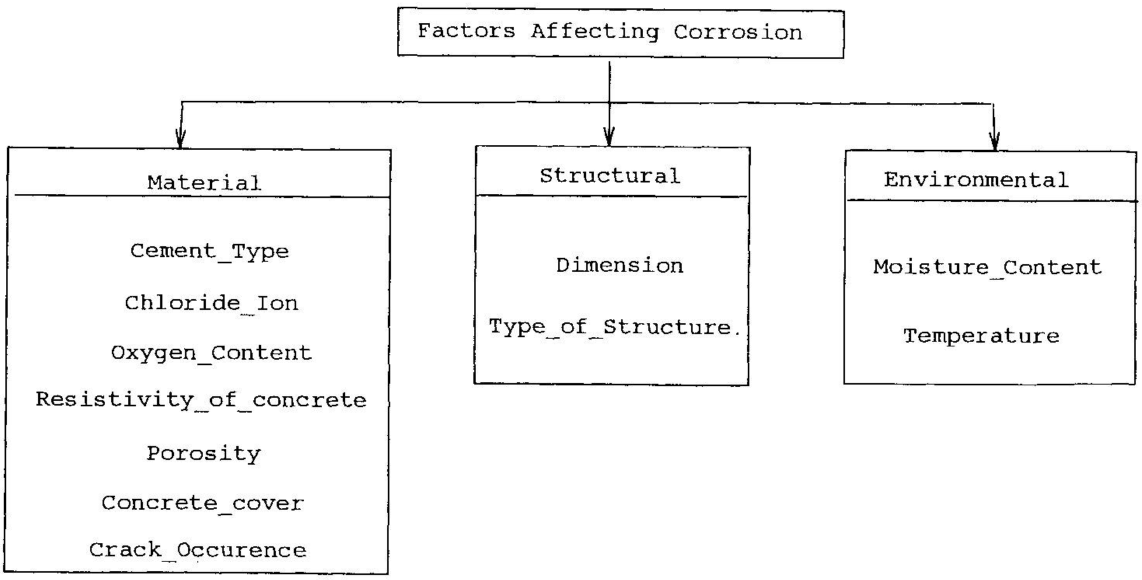


fig. 3 Factors Affecting Corrosion

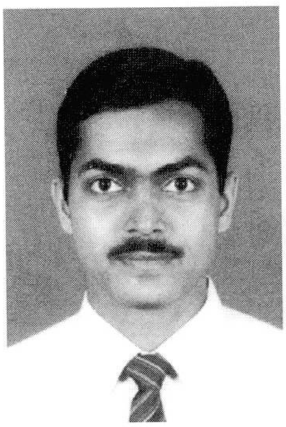
Neural Networks for Damage Detection in Steel Railway Bridges

Réseaux neuronaux pour la détection de dommages dans les ponts-rails métalliques

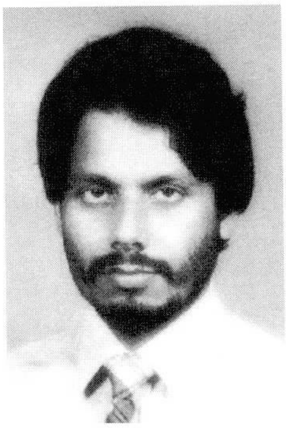
Neuronale Netze für die Feststellung von Schäden an Eisenbahnstahlbrücken

S. V. BARAI
Research Scholar
Indian Inst. of Science
Bangalore, India

P. C. PANDEY
Assist. Professor
Indian Inst. of Science
Bangalore, India



S. V. Barai, born 1964, BE and ME in civil eng. from MSU, Baroda. Currently he is doctoral student at the Indian Inst. of Science, Bangalore. His fields of research are Artificial Intelligence and Artificial Neural Networks applications in structural engineering.



P. C. Pandey, born 1948, received his BE (Civil) at BIT, Ranchi, ME (Struct) at IISc and Ph.D. at Liverpool. He specialises in Computational Mechanics, Artificial Intelligence and Artificial Neural Networks applications in structural engineering.

SUMMARY
The paper presents Artificial Neural Networks developed for typical steel railway bridges for the purpose of damage detection. Multilayer perceptrons have been used for generating the architecture for the bridges of different configurations. The back propagation algorithm has been adopted for training the network with simulated damage states. The training pairs have been generated using a standard finite element program. The weights of the trained networks have been stored and can be used as a knowledge source independently. It is demonstrated that the trained networks have practical relevance.

RÉSUMÉ
Des réseaux neuronaux sont développés pour la détection de dommages dans les ponts-rails métalliques. Des perceptrons à couches multiples ont été employés pour produire l'architecture de ponts divers. L'algorithme à rétro-propagation est utilisé pour entraîner les réseaux avec des états de dommages simulés. L'entraînement est produit avec un programme standard d'éléments finis. Les poids des réseaux entraînés sont mémorisés et peuvent être utilisés indépendamment comme source de connaissance. Ces réseaux entraînés sont utilisables dans la pratique.

ZUSAMMENFASSUNG
In der vorliegenden Arbeit werden künstliche neuronale Netze vorgestellt, die zwecks Feststellung von Schäden an typischen Eisenbahnstahlbrücken entwickelt worden sind. Gelayerte Perzeptronen wurden zum Netzaufbau für Brücken verschiedenartiger Konfigurationen eingesetzt. Der Rückfortpflanzungsalgorithmus wurde herangezogen, um das Netz mit simulierten Schadenzuständen zu trainieren. Die Trainierpaare wurden mit Hilfe eines Standard-Finite-Element-Programms erzeugt. Die Gewichte der angelernten Netze wurden gespeichert, und sie können selbständig als Kenntnissquelle benutzt werden. Es wird nachgewiesen, dass diese angelernten Netze praxisrelevant sind.

1. THE NETWORK RECALL PROCESS IN DAMAGE IDENTIFICATION

System identification techniques [1,2] can be extended to structures for systematic damage detection and evaluation. Structural identification is a process for constructing a mathematical description of a physical system when both the input and the corresponding output parameters are known. The recent emergence of *Artificial Neural Networks (ANN)* can be explored as an alternative tool for identification exercise in such situations. The multilayer perceptron derived from single-layer perceptron have been used only by a few investigators so far in the structural damage identification [3-7]. This paper presents the application of Multilayer Perceptron in identification of damage in truss members of various configurations of bridge structures typically shown in Fig. 1 and advocates that the trained networks can serve as knowledge source in damage assessment paradigms.

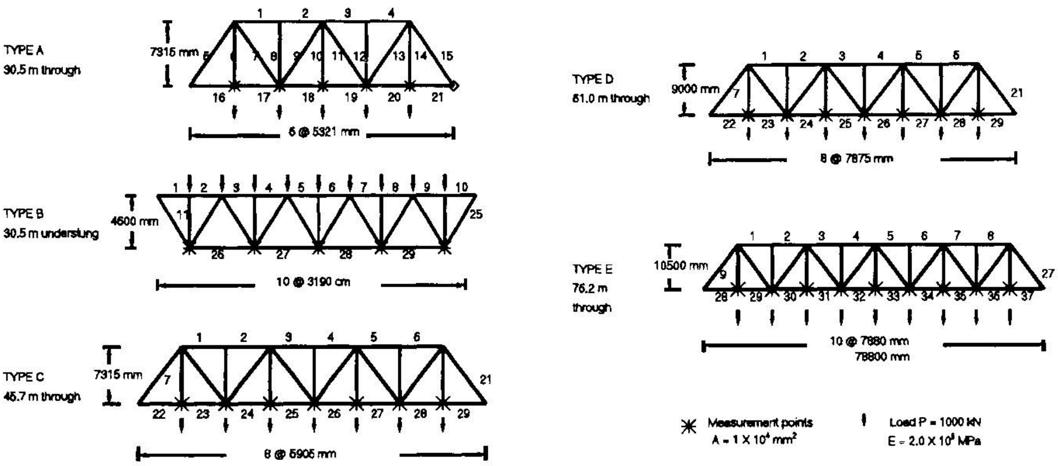


Figure 1 : Configurations of steel railway bridges

The ANN are typically characterized by three main phases.

[1] Network Design : The design of network architecture constitutes the determination of network topology (Number of hidden units per layer, number of hidden layers etc.) and training parameters (learning parameter η, momentum parameter α, error tolerance etc.) which are usually arrived at by trial and error in the Generalized Delta rule /backpropagation learning algorithm [8].

[2] Network learning : Training / Learning phase of the network involves determining the weight updating function and error corrections.

[3] Network recall : Recall constitutes the phase establishing network operation such as propagating rule and transfer function.

General guidelines for network simulation for bridge structures investigated in [7] is adopted here. In the present study, the network recall process using stabilized weight matrices is demonstrated to serve as independent knowledge source and its modularity in implementation in the paradigm for damage detection in railway bridges. After selecting a suitable architecture and training examples, the network is trained using backpropagation learning rule. In the course of training, once the network reaches some converged set of weights for imposed error tolerance, the weight matrices and the threshold values are stored for future use as *stabilized weight matrices* and *threshold parameters*. These stabilized weights are used further independently in the network recall process as re-training of the network is not required. This helps in minimizing computing effort needed in exhaustive time consuming training exercise.

A program based on the recall algorithm has been coded independently in C and implemented on VAX 8810 in order to demonstrate the usage of stabilized weight matrices of the trained neural network as independent knowledge source. The developed program uses the file containing the information regarding number of layers used, number of input neurons, number of neurons in internal layers (Hidden neurons), number of output neurons, learning parameter, η, momentum parameter, α, Average System Errors (ASE) and stabilized weights etc. for regenerating the results.

The typical training patterns for generating the weight matrices for the recall process were obtained as follows: In a structure the change in stiffness of the members (or cross sectional properties) gets reflected in its response (e.g. displacements). Let us call a member with reduced stiffness as a damaged member, which is reflected through the change in the member areas. From the measured response it would be possible to identify the damaged members through the stiffness changes. In order to generate the training patterns, all the members are assumed to be damaged in turn. A total of 40 training patterns consisting of simulated damaged states and structural response were generated with the help of finite element analysis and stored in files and additional 10 testing patterns were stored separately for verification of the trained networks after proper normalization. The normalization is essential while using sigmoid function, in order to have the input-output training patterns values between 0 and 1. These normalized training patterns presented to the network help in faster convergence. For normalization of the input-output pair, an interface program has been developed. This exercise of training the network was repeated for all the configurations of the bridges shown in Fig. 1.

2. ARCHITECTURE OF THE NETWORK AND THE LINKING OF WEIGHTS FOR THE BRIDGE TRUSSES

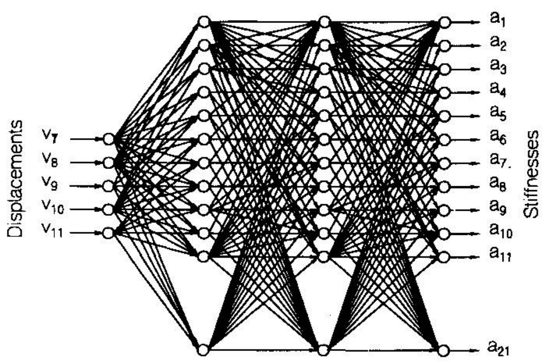


Figure 2 : Typical 5-(21-21)-21 architecture

In this example the input nodes are displacements measured only at a few selected locations in the structure and the output nodes are the cross sectional areas of members. A typical architecture (Fig. 2) adopted here is identified as **input nodes (n) - (hidden nodes per hidden layer (m)) - output nodes (p)** . For example of 5- (21-21)-21 architecture, (21-21) are two hidden layers with 21 hidden units per layer while 5 and 21 are input nodes and output nodes respectively. The architectures thus identified for various bridge configurations are given in Table 1. In these architectures, the learning parameter $\eta = 0.9$ and the momentum parameter $\alpha = 0.7$ based on earlier recommendations was found suitable [7].

Bridge type	Architecture $\eta = 0.9$ $\alpha = 0.7$	Training Samples	Testing Samples	Iterations	CPU time taken on VAX8810 for training in minute:seconds
Type A	5-(21-21)-21	40	10	875	14:46.17
Type B	5-(29-29)-29	40	10	1688	50:38.26
Type C	7-(29-29)-29	40	10	1294	44:26.17
Type D	7-(29-29)-29	40	10	1068	33:09.30
Type E	9-(37-37)-37	40	10	6458	311:41.92

Table 1 : ANN study : Structural damage detection in bridge structures, ASE = 0.001

3. RESULTS AND DISCUSSIONS

The identification performance of the network for various bridge configurations have been illustrated in Table 1 and Fig 3. The following observations are made.

(1) The trial study was carried out considering the Average System Errors (ASE) as 0.1, 0.01 and 0.001. It was found that the ASE taken as 0.001 gave acceptable results considering the time required for convergence and accuracy in training as well as in testing the patterns. (2) Trained networks were able to identify the damaged member clearly for the testing patterns presented at enquiry stage. (3) The CPU time required for training the networks on VAX8810 was less than 60 minutes in each case except for the bridge type E (Table 1). For bridge type E, the training time was too high which may be due to the network size. (4) From the Fig. 3, it has been observed that in all the cases presented during the recall phase, the average errors in the predicted values of the cross sectional area were less than 0.8%. This can be considered as a reasonably good performance of the network. (5) The network can be used identifying for new patterns in the range of training samples. (6) Time taken in recall is almost instantaneous, hence,

the benefit of using trained network is obvious in field applications. (7) The exercise demonstrates the

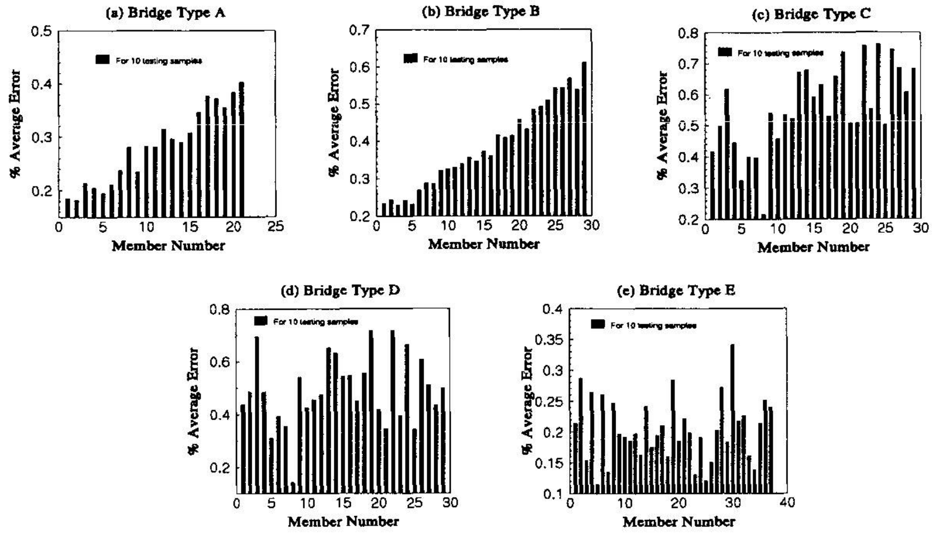


Figure 3 : Recall performance of neural networks for bridge configurations

ease of implementation of the recall process which should be quite attractive to engineers involved in damage detection, but do not want to get involved deeply with the ANN.

4. CONCLUSIONS

The paper presents standard ANN architectures for damage detection in typical steel railway bridges. For demonstration purpose, the networks have been trained with simulated damage states in typical railway steel bridges with static response. The stabilized weights of ANN with recall algorithm can be integrated with a suitable damage assessment paradigm as an independent knowledge source. Thus, the user need not be concerned with training the network again and again. This can be further extended for simulating dynamic response. This simulation may be appreciated in field applications as measured data is needed only at few chosen locations in the structure, which can minimize in-situ response measurement problems. It can be argued that the ANN has a strong potential for applications in performance monitoring and damage detection in bridge truss structures.

REFERENCES

1. YAO, J.T.P., *Safety and reliability of existing structures*, Pitman Advanced Publishing Program. 1985

2. HAJELA, P. and SOEIRO, F.J., " Recent developments in damage detection based on system identification method ", *Structural Optimization*, 2, pp: 1-10. 1990

3. WU, X., GHABOUSSI, J. and GARRETT, J.H., Use of neural networks in detection of structural damage, *Computers and Structures*, 42(4), pp: 649-659, 1992

4. SZEWCZYK, Z. P. and HAJELA, P., *Neural networks based damage detection in structures*, Technical Report, RPI, Troy. 1992

5. ELKORDY, M.F., CHANG, K.C. and LEE, G.C., " Neural networks trained by analytically simulated damaged states ", *Jl. of Computing in Civil Engineering, ASCE*, 7(2), pp:130-145, 1993

6. MASRI, S.F., CHASSIAKOS, A.G. and CAUGHY, T.K., "Identification of nonlinear dynamic systems using neural networks", *Transaction of ASME, Jl. of Applied Mechanics*, 60(3), pp:123-133, 1994

7. BARAI, S.V. and PANDEY, P.C., " Performance of generalized delta rule in artificial neural networks in structural damage detection ", Under review, 1994

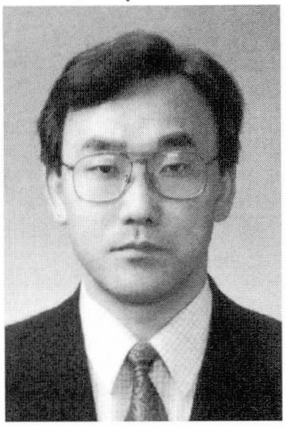
8. RUMELHART., D.E., HINTON, G.E. and WILLIAMS, R.J., " Learning internal representations by error propagation" in *Parallel Distributed Processing : Explorations in the Microstructures of Cognition. Vol.1: Foundations* by Rumelhart, D.E. and McCelland, J.L.(Eds.), pp:318-362, MIT press, Cambridge, MA, 1986

Concrete Bridge Rating Expert System with Machine Learning

Système expert avec apprentissage-machine pour l'évaluation de ponts en béton

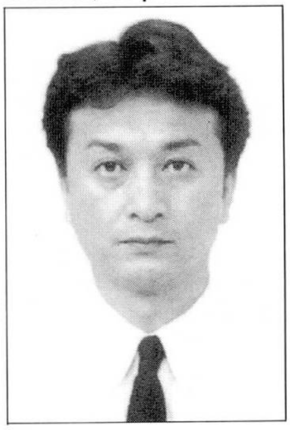
Ein Expertensystem mit Maschinenlernen zur Einstufung von Betonbrücken

Ayaho MIYAMOTO
Assoc. Prof.
Kobe University
Kobe, Japan



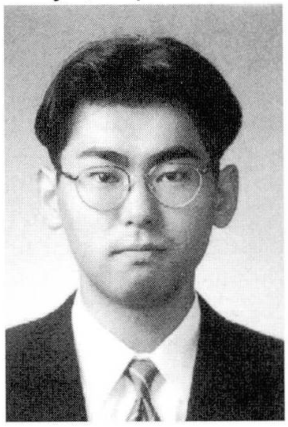
A. Miyamoto, born 1949, received his Dr of Eng. degree from Kyoto Univ. in 1985. His recent research activities are in the area of structural safety assessment on concrete bridges and establishment of design concept for concrete structures under impact loads.

Moriyoshi KUSHIDA
Sub Manager
Kurimoto Ltd
Osaka, Japan



M. Kushida, born 1954, received his Master of Eng. degree from Kobe Univ. in 1979. He is attached to the Struct. Eng. Div. of Kurimoto Ltd, which is now working on long span bridges. He is also interested in serviceability assessment procedures for existing bridges.

Kazuya KINOSHITA
Civil Eng.
Obayashi Corp.
Tokyo, Japan



K. Kinoshita, born 1969, received his Master of Eng. degree from Kobe Univ. in 1994. He is engaged in investigation of systematisation of bridge rating techniques, and also has interest in expert systems for serviceability rating of concrete structures.

SUMMARY
The aim of this study is to develop a concrete bridge rating expert system with machine learning, employing a combination of neural network and bi-directional associative memories. Introduction of machine learning into this system facilitates knowledge base refinement. By applying the system to an actual bridge in service, it has been verified that the employed machine learning method using results of questionnaire surveys of bridge experts is effective for the system.

RÉSUMÉ
Le but de l'étude était de réaliser un système expert avec apprentissage-machine pour l'évaluation de ponts en béton, combinant un réseau neuronal avec des mémoires associatives bidirectionnelles. L'introduction de l'apprentissage-machine augmente la qualité de la base de connaissance. L'application de ce système au cas réel d'un pont en exploitation démontre l'efficacité de l'introduction de l'apprentissage-machine, utilisant les informations fournies par les spécialistes.

ZUSAMMENFASSUNG
Ziel der vorliegenden Studie ist die Entwicklung eines Expertensystems zur Brückenbeurteilung, das ein neuronales Netzwerk mit zweiseitigen assoziativen Gedächtnissen kombiniert. Die Einführung des Maschinenlernens vereinfacht die Verfeinerung der Wissensbasis. Durch Anwendung des Systems auf eine in Betrieb stehende Brücke wurde festgestellt, dass die Lernmethode mittels strukturierten Erhebungen unter Brückenfachleuten wirkungsvoll ist.

1. INTRODUCTION

The authors have been working for some time on the development of a Concrete Bridge Rating Expert System[1,2] that can evaluate the serviceability of concrete bridges on the basis of knowledge and experience acquired from domain experts. The final goal of the present system is to evaluate the structural serviceability of bridges on the basis of the specifications of target bridges, environmental conditions, traffic volume, and other subjective information such as one obtained through visual inspection. The inference mechanism in the system first selects a membership function (Π function parameters) defined in the knowledge base on the basis of the knowledge acquired from domain experts to achieve the lowest level subgoals of the diagnostic process. The inference mechanism then combines the subgoals with a higher level subgoals according to Dempster's rule of combination and repeats this process[1]. In evaluating the serviceability which is evaluated by a combination of "load carrying capability" and "durability" of a target bridge, which is the final goal of the expert system, the inference mechanism performs fuzzy mapping considering the degree of influence and the degree of confidence, and outputs the result of serviceability evaluation of the bridge accordingly. It has become known, however, that for certain types of inputs, the system sometimes outputs inconsistent results because the relevant knowledge accumulated in the system is incomplete[3]. It is no easy task, however, to refine the knowledge base in the system while maintaining the integrity of the system. Consequently, the procedure of knowledge base management needs to be simplified by introducing machine learning into the expert system.

In this study an inference system combining the neural network[4] and the bidirectional associative memory (BAM)[5] was constructed as part of the Concrete Bridge Rating Expert System. The results of questionnaire surveys conducted on domain experts during field tests were used as teacher data (objective criteria) to give the system the ability to learn and verify the effectiveness of the learning method.

2. SYSTEM DESCRIPTION

Fig. 1 shows the configuration of the expert system. The knowledge base, the inference engine, the associative memory and the submodels (neural network models) of the system are constructed on a personal computer (NEC PCH98 U100), and the learning module runs on a UNIX workstation (SONY NEWS). The expert system is all written in *C* language.

Fig. 2 illustrates the inference process of this system. As a first step, the system asks a series of basic questions for the lowest level subgoals regarding the specifications of the bridge, environmental conditions, traffic volume, the conditions of cracks, etc. and asserts the answers from the user as fact clauses. Then the system searches all relevant fact clauses according to the inference rules. The system asks new questions if the message number for a found fact clause is "q" and

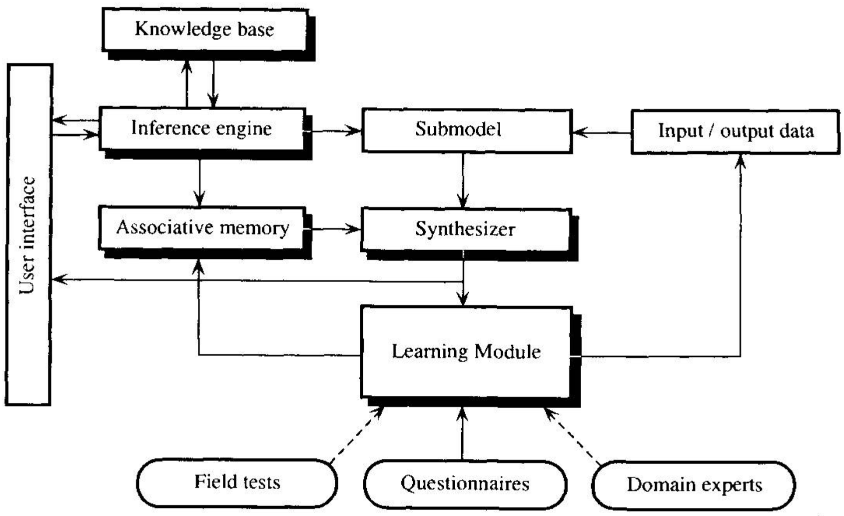


Fig. 1 System Configuration(New System)

asserts the responses to those questions as new fact clauses. If the found clause has a numeral, the system outputs a corresponding message. When having found all relevant facts by repeating this forward-chaining inference, the system moves on to the stage of associative memory and neural network inference. First, the associative memory determines the degree of match of the antecedent and calculate the weight for the consequent in the associative memory. The system then combines the outputs obtained here with the outputs from the consequent neural network model to give a diagnosis. The diagnosis given here is actually a set of soundness indicators calculated as the probabilities of the five possible conditions, namely, safe, slightly safe, moderate, slightly dangerous, dangerous. The system can also evaluate bridges with respect to the need for repair or strengthening and the remaining service life of both of the floor slab and the main girder. If a diagnosis is not a proper one, input/output data (teacher data) is modified on the basis of such information as the results of questionnaire surveys so as to refine the knowledge base according to the back propagation algorithm. After that, the neural networks are run again to output a diagnosis reflecting the modification. If the diagnosis is a proper one, the certainty for the corresponding rule is altered and the Concrete Bridge Rating Expert System returns to the startup menu.

3. VERIFICATION OF EFFECTIVENESS OF THE EXPERT SYSTEM

3.1 Comparison with Previous System[1,2]

The newly developed Concrete Bridge Rating Expert System was used to evaluate the serviceability of an actual bridge. The results of evaluation by this system were compared with results obtained from the previous system to verify the reliability of the acquired initial knowledge.

The bridge evaluated here was a reinforced concrete T-girder bridge[6] that had been constructed with relatively poor execution of work. The main girder then had flexural cracks, shear cracks and cracks due to the corrosion of reinforcing bars. Particularly cracks due to corrosion were rather wide, and water leakage, free lime and spalling of cover concrete around those cracks were

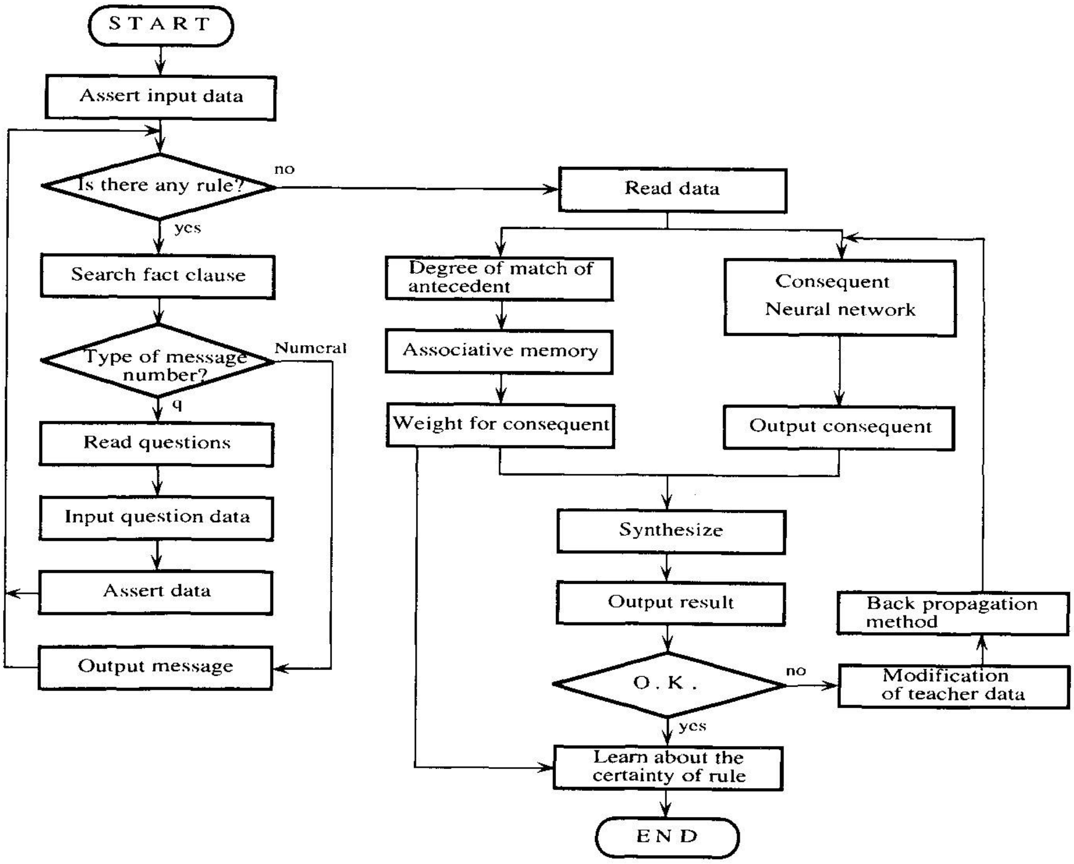


Fig. 2 Inference Process in Inference Engine of the Concrete Bridge Rating Expert System(New System)

noticeable. Tables 1 and 2 show the results of evaluation of the main girder of this bridge by the previous (original) system and the new system. Except for the subgoals for the design of the main girder, the original system and the new system gave similar results(see Tables 1 and 2). With respect to the design of the main girder, since the original system combines membership functions to higher level subgoals following Dempster's rule of combination in the inference process, the degree of uncertainty tends to increase as the inference process progresses. This resulted in the inconsistency of showing two peaks, namely at "slightly dangerous" and "slightly safe" (see Table 1). By contrast, the new system gave results somewhat centering around a single peak (see Table 2). Errors contained in these results need to be corrected through neural network-based learning. In the other respects the results obtained from the original system and the new system showed fair agreement (see Tables 1 and 2).
From above, it can be concluded that the new system has acquired the knowledge of the original system very accurately. Various problems found in diagnoses given by the system[3], however, indicate that the knowledge needs to be refined.

3.2 Refinement of Knowledge Based on Results of Questionnaire Survey

In this section, the refinement of knowledge in the consequent neural network is performed based on teacher data (objective criteria) obtained from questionnaire surveys on domain experts.
The results of the questionnaire surveys were divided into five categories, each corresponding to 20 points on a scale of 100. These categories were related to the probability of states ranging from dangerous to safe output from the system, and the data thus obtained was used as teacher data (objective criteria). By use of this data, the knowledge was refined according to the back propagation algorithm. Of the evaluation items for the reinforced concrete T-girder bridge mentioned earlier, shown below is the process of knowledge refinement for the subgoals related to cracks in the main

Table 1 Evaluation of RC T-Girder Bridge by Original System

Judgment factor	Mean soundness score	Danger	Slightly danger	Moderate	Slightly safe	Safe
Design	47.0	0.151	**0.273**	0.161	**0.371**	0.044
Execution of work	17.4	0.338	0.605	0.056	0.000	0.000
Service condition	76.0	0.000	0.000	0.167	0.644	0.189
Flexural crack	60.9	0.000	0.173	0.447	0.191	0.189
Shear crack	60.9	0.000	0.186	0.435	0.186	0.194
Corrosion crack	39.6	0.260	0.457	0.016	0.051	0.217
Whole damage of girder	50.9	0.127	0.285	0.293	0.106	0.189
Load-carrying capa. of girder	56.2	0.099	0.185	0.293	0.210	0.213
Durability of girder	49.8	0.146	0.308	0.136	0.244	0.166
Serviceability of girder	51.7	0.161	0.245	0.201	0.221	0.173

Table 2 Evaluation of RC T-Girder Bridge by New System

Judgment factor	Mean soundness score	Danger	Slightly danger	Moderate	Slightly safe	Safe
Design	64.7	0.001	0.019	0.319	**0.564**	0.097
Execution of work	24.8	0.330	0.608	0.055	0.005	0.002
Service condition	70.0	0.014	0.026	0.144	0.575	0.241
Flexural crack	58.4	0.035	0.240	0.260	0.202	0.264
Shear crack	58.7	0.033	0.313	0.145	0.202	0.307
Corrosion crack	25.1	0.283	**0.698**	0.006	0.006	0.007
Whole damage of girder	52.0	0.148	0.222	0.208	0.227	0.195
Load-carrying capa. of girder	65.8	0.001	0.038	0.300	0.493	0.168
Durability of girder	52.6	0.058	0.165	0.394	0.352	0.031
Serviceability of girder	58.9	0.019	0.109	0.343	0.464	0.065

girder (see Table 2). Table 3 shows the results of the questionnaire surveys (teacher data) regarding cracks in the main girder. Table 4 shows the results of evaluation by the system after the knowledge refinement of the consequent neural network based on the teacher data shown in Table 3. Figs. 3 and 4 show, in the form of membership functions, the results of evaluation regarding corrosion cracks and flexural cracks in the main girder before the knowledge refinement, the teacher data used in knowledge refinement, and the results of evaluation after the knowledge refinement.

As shown in Tables 2 and 4, and Fig. 3, the corrosion cracks in the main girder were judged "slightly dangerous" before knowledge refinement, while after the knowledge refinement they were judged "slightly safe". As for the flexural cracks in the main girder, the results of evaluation before the knowledge refinement showed more or less even distribution over the soundness scale, while those after knowledge base refinement shows a peak in the "moderate" range. This indicates that the degree of uncertainty decreased as a result of knowledge refinement (see Tables 2 and 4, and Fig. 4).

From above, it can be concluded that the accuracy of knowledge refinement in this system was considerably high, evidencing the effectiveness of the learning method of the system. In cases, however, where results of questionnaire surveys are used as teacher data (objective criteria), the reliability of the questionnaire results themselves becomes an important consideration. Teacher data might even be inconsistent to the extent of prohibiting knowledge refinement. It is desirable, therefore, that indicators related with more objective data obtained from reliable sources, such as field tests, be used as teacher data.

Table 3 Example of Teacher Data on Level of Cracks in Girder used for Knowledge Base Refinement

Judgment factor	Danger	Slightly danger	Moderate	Slightly safe	Safe
Flexural crack in girder	0.000	0.154	0.538	0.308	0.000
Shear crack in girder	0.000	0.308	0.231	0.231	0.231
Corrosion crack in girder	0.077	0.154	0.308	0.385	0.077

Table 4 Example of Output after Refinement of Knowledge on Level of Cracks(Girder)

Judgment factor	Mean soundness score	Danger	Slightly danger	Moderate	Slightly safe	Safe
Flexural crack	54.5	0.029	0.179	**0.427**	0.268	0.098
Shear crack	57.6	0.017	0.305	0.208	0.223	0.247
Corrosion crack	51.7	0.090	0.217	0.279	**0.345**	0.070

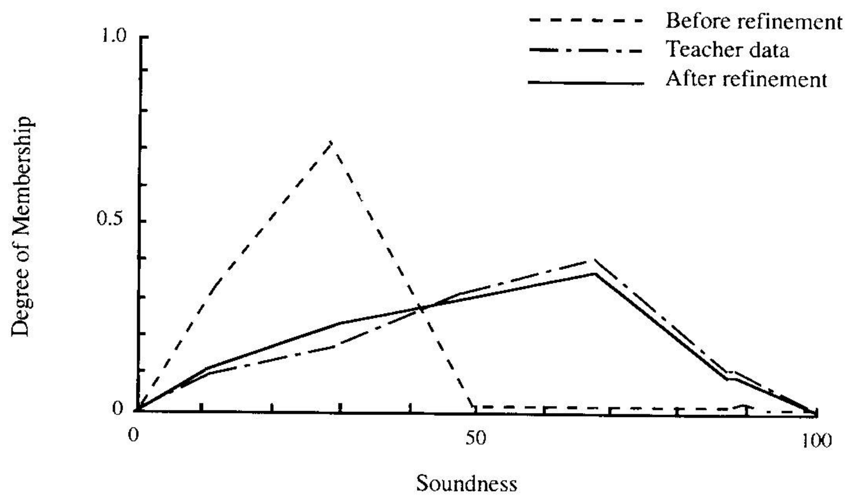


Fig. 3 Comparison of Outputs on Corrosion Cracks in Girder

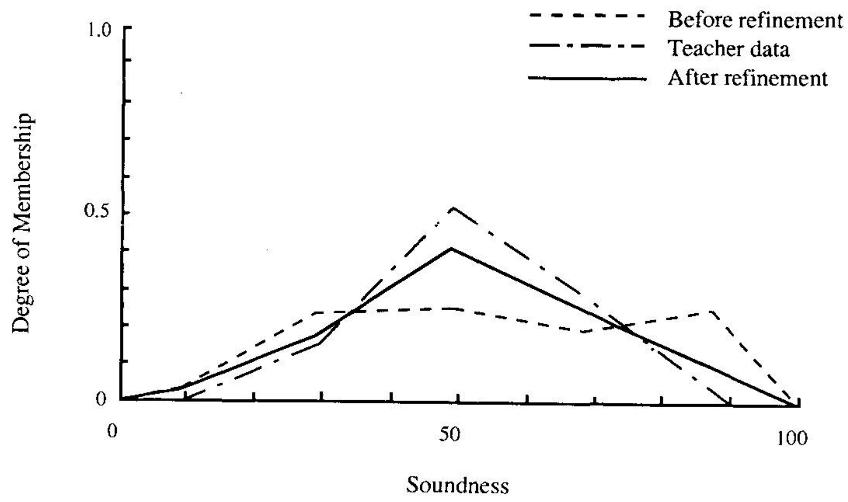


Fig. 4 Comparison of Outputs on Flexural Cracks in Girder

4. CONCLUSIONS

In this study a Concrete Bridge Rating Expert System with Machine Learning has been developed. Using neural networks, the developed system facilitates the modification of the knowledge base based on data such as results of questionnaire surveys conducted on domain experts. Independent neural networks constructed for individual rules help prevent the inference mechanism from becoming a black box. The time required for learning can also be reduced because the learning process involves only the networks concerned. The results of this study can be summarized as follows:

(1) As a method of modifying the knowledge base of the Concrete Bridge Rating Expert System, a learning method based on the neural network has been presented. And a new inference process similar to the conventional fuzzy inference has been developed by combining the neutral network and associative memory.

(2) The Concrete Bridge Rating Expert System was applied to the girder of an actual bridge to verify the results of evaluation. Good agreement between the results obtained from the original system and the new system confirmed that the knowledge for the new system was successfully acquired from the original system.

(3) The knowledge base was refined using neural networks on the basis of the results of questionnaire surveys on domain experts. Good results achieved as a result of knowledge base refinement evidences the effectiveness of the learning method in the system.

In order to enhance the reliability of the expert system, it is necessary to refine the knowledge base through application to more bridges. It is also necessary to clearly define the relationships between the outputs of the system and field data (e.g., linking numerical analysis programs) instead of relying solely on information obtained from visual inspection.

REFERENCES

1. Miyamoto, A., Kimura, H. and Nishimura, A.: "Expert System for Maintenance and Rehabilitation of Concrete Bridges", Proc. of IABSE Colloquium on Expert Systems in Civil Eng., Bergamo, pp. 207-217, 1989.
2. Miyamoto, A., Masunari, I. and Nishimura, A.: "Development of Expert System for Serviceability Rating of Concrete Bridges and Its Practical Application", J. of the Society of Materials Science, Vol. 40, No. 450, pp.256-262, Nov. 1991.
3. Miyamoto, A., Morikawa, H., et al.: "A Method of Inductive Machine Learning in Knowledge Base for Bridge Rating Expert System", J. of Structural Engineering, Vol. 37A, II, pp. 643–654, March 1991.
4. Asou, H.: "Information Processing using Neural Network", Sangyo Tosho, 1988.
5. Kosho, B.: "Adaptive Bidirectional Associative Memories", Applied Optics, Vol. 26, No. 23, Dec. 1987.
6. The Construction Engineering Research Institute Foundation.: Report on Serviceability Evaluation of the Old Aoki-bashi Bridge, March 1990.

Image-Based Analysis of Evolution by Using a Neural Network

Réseau neuronal pour l'analyse de l'évolution avec base en images

Entwicklungsanalyse durch Bildauswertung mittels eines neuronalen Netzwerks

Rosana SATORRE
Computer Scientist
Univ. of Alicante
Alicante, Spain

Rosana Satorre, born in 1967, received her Computer Science Degree at the Universidad Politécnica de Valencia. Today she is teaching at the Universidad de Alicante.

SUMMARY

A system is developed to automatically detect changes in parameters among several measures taken at different times. It is assumed that the variables characterise the system which has been studied regardless of the nature of these variables. A neural network is used which learns a sequence of events and marks a representative set of changing parameters. The neural network recognises whether the new set of parameters, representing the actual state of the object being studied, falls in the admissible range, or if these values are not suitable. In this paper, the parameters are the pixels of images (photographs) and a system is proposed to check the evolution of the surfaces by using a video-camera.

RÉSUMÉ

Un système est développé pour la détection automatique des variations de paramètres lors de différentes mesures faites à divers moments. Il est admis que les variables caractérisent le système étudié mais que celui-ci ne dépend pas de la nature de ces variables. Un réseau neuronal apprend une séquence d'événements et établit une formule représentative de paramètres variables. Le réseau neuronal reconnaît si la nouvelle formule de paramètres représentant l'état actuel de l'objet à étudier se trouve dans le domaine de valeurs admissibles ou non. Dans cet article, les paramètres sont les pixels des images, dont l'évolution est contrôlée par magnétoscope.

ZUSAMMENFASSUNG

Es wird ein System vorgestellt, das automatisch die Veränderung von Zustandsgrössen entdeckt, die in verschiedenen Messungen zu unterschiedlichen Zeiten ermittelt wurden. Dabei wird vorausgesetzt, dass das System durch diese Grössen charakterisiert ist, aber nicht von ihrem Wesen abhängt. Ein neuronales Netzwerk lernt aus der Abfolge von Ereignissen und bezeichnet einen repräsentativen Satz sich ändernder Zustandsgrössen. Später soll es auch erkennen können, ob ein neuer Satz, der den aktuellen Zustand beschreibt, in den zulässigen Bereich fällt oder nicht. Im vorliegenden Fall sind die Zustandsgrössen Bildpixel, deren Entwicklung auf den Oberflächen durch eine Videokamera geprüft werden.

1. INTRODUCTION

In civil engineering many efforts are being oriented to carry out quality controls and defects finding, during the construction process as well as when the construction is finished, referring to the constructions development and maintenance.

These diagnosis tasks need, in general, laborious and expensive methods, specially in singular building.

This work develops a method to detect changes (alterations in structural joints, fissures and other defects, normal or abnormal movements of the elements, etc.) from sequences of images.

The kernel of the system is a type RHI artificial neural network, which has the property to extract characteristics that leave us discriminate among several patterns in a set. It can be used in two ways:

- **To detect changing characteristics**: A set of photographs of the same scene taken in several times are used in the learning process. The significant fragment that results will be located inside the set of pixels that have changes among the photographs; that is, in the portions of the scene where fissures, shifts, deformations, etc. are produced.

- **To detect abnormal changes**: to detect unusual changes (not due to temperature,wind, etc.), the system consists of two phases:

- Learning: this is the same described in the precedent paragraph. The learning process will be a preprocessing operation.

- Recalling: In future the actual photograph will be used to ask the system. If there are abnormal changes, the response will be corrupted.

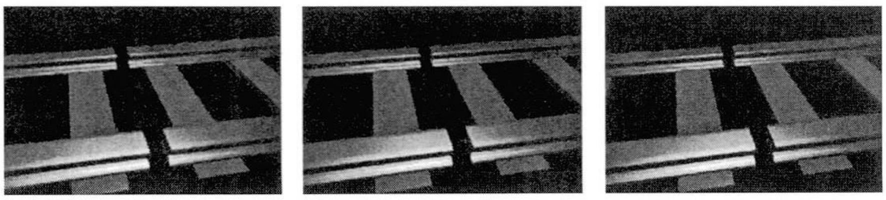
The RHI considers the components of each pattern which allow us to distinguish one pattern from the other and through these components, the model can infer, at least partially, the associated pattern to a given one. New components of this given pattern are then inferred in new iterations.

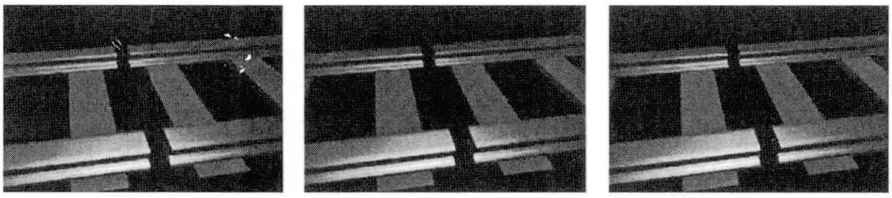
Total recall is not always possible, however, the model can indicate which components can not be obtained in an incremental manner.

Other artificial neural network models [1,2,3,7,8,9,10,11,12,13,14] use all the componente of the patterns and, as a consecuence, they can not used to extract the changing pixels in a set of images.

2. EXAMPLE

In order to demostrate the efectiveness of the proposed system, we have simulated the movement of a joint in a railway. We have taken 80 events that correspond to different positions to make a learning set of a RHI neural network.





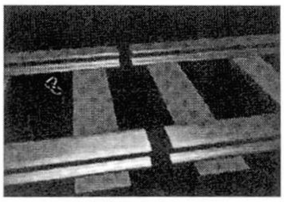
Several elements of the learning set that correspond to different positions of the railway.

The significant fragment of the RHI contains 75 pixels which positions are the zones where are the changes in the scene.

Significant fragment indicating the changing joint in the railway.

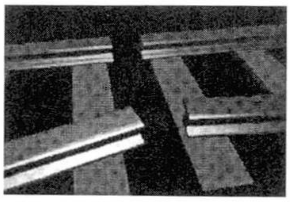
2.1 Detection of abnormal changes

This phase corresponds to maintenance tasks. In future, if a new photograph will be used to ask the RHI, the response will be either one of the corresponding to correct position of the railway if the railway is right or a corrupted response if the railway should an abnormal deformation.

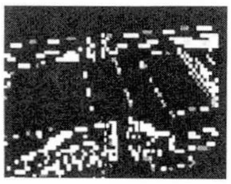


Scene showing the actual situation (normal movement)

Response given by the RHI neural network



Scene showing the actual situation (abnormal movement)



Response given by the RHI neural network

3. CONCLUSIONS

A prototype has been developed and an example is showed about the evolution of a surface: The RHI tells us which areas of the scene have suffered changes. Besides, the system could inform us about abnormal evolutions of the surface. This work could be of great interest as a system for helping on diagnostic based on superficial changes (or other) and in automated monitoring of a construction, in maintenance tasks.

REFERENCES

1. CARPENTER, G. A. & GROSSBERG S., ART2: Self-Organization of Stable Category Recognition Codes for Analog Input Patterns. Applied Optics, 26, pp. 4919-4930, 1987
2. FUKUSHIMA, K. & MIYAKE, S., Neocognitron: A New Algorithm for Pattern Recognition Tolerant of Deformations and Shifts in Position. Pattern Recognition, vol. 15, nº 6, pp 455-469, 1982
3. GARCÍA-CHAMIZO, J.M.; CRESPO-LORENTE, A., Orthonormalized Associative Memories. Proceedings of the IJCNN, Baltimore, vol. 1, pp. 476-481, Jun. 1992
4. GARCÍA CHAMIZO, J.M. & CRESPO LORENTE, A., Incremental Heteroassociative Networks. IFIP Congress '92, Madrid, Sept. 1992
5. GARCÍA CHAMIZO, J.M.; CRESPO LORENTE, A. & RIZO ALDEGUER, R., Output Pattern Recalling Aided by Themselves: Incremental Heteroassociative Networks. Proceedings of the IJCNN, Beijing, Nov. 1992
6. GARCÍA CHAMIZO, J.M.; SATORRE CUERDA, R., Estudio del Modelo RHI Generalizado a Dominios Discretos y al caso continuo. Depto. T.I.C.; Report interno R.E. 578, U. Alicante, Oct. 1993
7. HECHT-NIELSEN, R., Applications of Counterpropagation Networks. Neural Networks 1, pp. 131-139, 1988
8. HOPFIELD, J. J., Neurons with Graded Response Have Collective Computational Properties Like Those of Two State Neurons. Proc. National Academy of Sciences 81, pp. 3088-3092, 1984
9. IBARRA PICÓ, F. & GARCÍA CHAMIZO, J. M., A Generalized Bidirectional Associative Memory with a Hidden Orthogonal Layer. ICANN'94, Sorrento, May, 26-29, 1994.
10. KOHONEN, T.; MÄKISARA, K.; SIMULA, O. & KANGAS, J.. EDITORS, Self-Organization and Associative Memory. Springer-Verlag, 2nd edition, 1988
11. KOSKO, B., Bidirectional Associative Memories. IEEE Trans. on Systems, Man, Cybernetics, vol. 18, nº 1, Jan. 1988
12. PAO, Y., Adaptive Pattern Recognition and Neural Networks. Addison-Wesley Publishing Company, Inc., 1989
13. TORREGROSA, V.; GARCÍA CHAMIZO, J.M.; RIZO, R. & CRESPO LORENTE, A., Modelo Neuronal de Memoria Asociativa Ortonormalizada Adaptativa. Actas V Conferencia CAEPIA'93, Madrid, Nov., 1993
14. ZURADA, J. M., Artificial Neural Systems. West Publishing Company, 1992