University of Otago
Te Whare Wananga O Otago
Dunedin, New Zealand

Neuro-Fuzzy Engineering for Spatial Information Processing

Nikola K. Kasabov
Martin K. Purvis
Feng Zhang
George L. Benwell

The Information Science Discussion Paper Series

Number 96/08
June 1996
ISSN 1172-455X

University of Otago

Department of Information Science

The Department of Information Science is one of six departments that make up the Division of Commerce at the University of Otago. The department offers courses of study leading to a major in Information Science within the BCom, BA and BSc degrees. In addition to undergraduate teaching, the department is also strongly involved in postgraduate programmes leading to the MBA, MCom and PhD degrees. Research projects in software engineering and software development, information engineering and database, artificial intelligence/expert systems, geographic information systems, advanced information systems management and data communications are particularly well supported at present.

Correspondence

This paper represents work to date and may not necessarily form the basis for the authors' final conclusions relating to this topic. It is likely, however, that the paper will appear in some form in a journal or in conference proceedings in the near future. The authors would be pleased to receive correspondence in connection with any of the issues raised in this paper. Please write to the authors at the address provided at the foot of the first page.

Any other correspondence concerning the Series should be sent to:

DPS Co-ordinator
Department of Information Science
University of Otago
P O Box 56
Dunedin
NEW ZEALAND
Fax: +64 3 479 8311
email: workpapers@commerce.otago.ac.nz

Neuro-Fuzzy Engineering for Spatial Information Processing

Nikola K. Kasabov[1]
Martin K. Purvis
Feng Zhang
George L. Benwell
Computer and Information Science
University of Otago

June 1996

Abstract

This paper proposes neuro-fuzzy engineering as a novel approach to spatial data analysis and for building decision making systems based on spatial information processing, and the development of this approach by the authors is presented in this paper. It has been implemented as a software environment and is illustrated on a case study problem.

[1] Address correspondence to: Assoc. Prof. N.K. Kasabov, Department of Information Science, University of Otago, PO Box 56, Dunedin, New Zealand. Fax: +64 3 479 8311, Email: nkasabov@otago.ac.nz

1 Introduction

Neuro-fuzzy engineering has emerged as a new and very powerful technique which allows for:

- learning from data; incorporating both initial set of knowledge and data into a simple decision making framework;
- extracting knowledge from data for the sake of explanation and understanding;
- adaptive tuning of existing knowledge according to new data [1,10,11,12,13,14,15].

Neuro-fuzzy engineering nowadays is a comprehensive and robust methodology for knowledge engineering and problem solving [1].

This paper applies some already known techniques of neuro-fuzzy engineering and goes on to develop some of these methods further with particular respect to spatial information processing. An example problem, golf course suitability decision making, has been chosen for illustrative purposes and is used throughout the paper.

2 Neuro-fuzzy engineering techniques

At the centre of the neuro-fuzzy engineering techniques are artificial neural networks, or simply - neural networks (NN). Connectionist-based methods, such as neural networks, are derived from parallel and distributed computing architectures and make use of distributed, local computations in such a way that the overall system exhibits a "high-level" inferencing capabilities, such as learning, generalisation, adaptation [2,3]. NN, in particular, have the important capability of approximating any continuous function to any desired degree of precision, without the need for specifying the type or nature of the function [4]. Even relatively small neural networks can approximate polynomial functions of almost any degree, without the necessity of specifying the degree of that function prior to training the network. For this reason it can be useful to use neural networks in the initial stages of an empirical investigation, when little may be known about the nature of the spatial data set at hand [8,9,19,20].

The fuzzy systems paradigm [5], another key element of neuro-fuzzy engineering, allows for representing ambiguous, but rationale, knowledge in linguistically defined and meaningful terms. Different types of fuzzy rules and fuzzy inference methods have been explored, from simple rules with the min-max compositional inference method to more sophisticated weighted fuzzy rules with fuzzy evidential reasoning methods [5-10]. Standalone fuzzy systems have been developed for classification and decision making based on spatial data.

A fuzzy neural network (FNN) is a connectionist model which blends at a low level the neural-network and fuzzy systems paradigms. There are a variety of FNN architectures [11,12,15]; for example, the FNN model [15] facilitates learning from data, fuzzy rules extraction, fuzzy rules insertion, approximate reasoning, adaptation. This FNN uses a multi-layered perceptron (MLP) network and a backpropagation training algorithm. The general architecture consists of five layers:

1. input variables layer;
2. condition elements (fuzzy membership functions) layer;
3. rules layer;
4. action elements (output membership functions) layer, and
5. output variables layer,

as described in [1,15]. In the following experiments, partial FNNs that consist of only a condition element layer, a rule layer and an action element layer are considered. The membership functions are defined by the user. For the experiments in the next section, the membership functions are of the standard triangular type with an uniform distribution over the universe of discourse. Fuzzification and defuzzification are performed outside the structure.

One of the advantages of fuzzy neural networks is that structured information (knowledge) can be inserted and extracted from them. A FNN, after training, can be interpreted in linguistic terms. The structure of a FNN also structures the information (knowledge) representation and interpretation. Various algorithms for rules extraction from connectionist structures are discussed in [1,17]. An algorithm called REFuNN (Rules Extraction From Neural Networks) for rules extraction from a trained FNN is presented in [1,15]. Its simplified version is used in this paper. The method is based on the following assumptions: hidden nodes in a MLP capture features, rules, and groups of data; fuzzy quantisation of the input and the output variables, which is performed outside the algorithm, brings additional knowledge to the system thus improving its performance. Automatically extracted rules may require additional manipulation depending on the reasoning method applied afterwards. The algorithm uses thresholds above which network connection weights are kept and which are represented in a linguistic form as fuzzy rules. Another algorithm for rules extraction, based on a connection-masking operation, is presented here, along with a discussion of experimental results, as well.

3 The case study problem

For illustrative purposes we consider an artificial problem that has been chosen for its conceptual simplicity and yet one whose "solution" is somewhat difficult for numerical modelling methods, because it is only piecewise differentiable. The problem is to determine the suitable sites for the locations of public golf courses in the South Island of New Zealand [20]. For this problem it was assumed that suitability could be determined from the observed data of mean summer temperature, mean annual rainfall, mean altitude, and distance from the nearest of four principle urban centres on the South Island. Each of the four input parameters was partitioned into five possible ranges, and the output parameter (suitability for locating a public golf course) was taken to have five possible values, ranging from 0 to 4. For each of the 153,036 1 km^2 blocks (pixels) of the South Island, a value for each of the four input parameters was determined. In order to provide an evaluation mechanism, an artificially correct "solution" was also determined for each block, based on a set of plausible, but highly non-linear rules. Figure 1 shows the distribution of these solution set points, with the darkest values (value = 4) representing the most suitable golf course sites.

In order to describe the so-called “solution” suitability, we make reference to the following six variables:

- **S** the overall suitability for building a golf course
- **sf** a numerical factor used in the calculation
- **ds** suitability associated with distance
- **ts** suitability associated with temperature
- **rs** suitability associated with rainfall
- **hs** suitability associated with altitude.

For each of the last four variables, a set of rules was constructed that dealt with the suitability for a particular attribute. For example the mean summer temperature (**temp)** rules were as follows:

If (13^o <= **temp** < 14^o), then **ts** = 4.
If (14^o <= **temp** < 15.5^o) or (12.5^o <= **temp** < 13^o),t hen **ts** = 3.
If (15.5^o <= **temp** < 16^o) or (12^o <= **temp** < 12.5^o), then **ts** = 2.
If (16^o <= **temp** < 20^o) or (11.5^o < **temp** **<** 12^o), then **ts** = 1.
If **temp** >= 20^o or **temp** <= 11.5^o then **ts** = 0.

Rules of a similar nature were established for rainfall, distance from urban centres, and altitude. The results of these separate layer analyses were combined according to the following formula for the suitability factor.

$$\mathbf{sf} = 3*\mathbf{ds} + 2*\mathbf{ts} + \mathbf{rs} + 1.5 * \mathbf{hs} - 1$$

The suitability **S** was determined as follows:

If (0 <= **sf** <= 6), then S = 0
If (7 <= **sf** <= 12), then S = 1
If (13 <= **sf** <= 18), then S = 2
If (19 <= **sf** <= 24), then S = 3
If (25 <= **sf** <= 30), then S = 4

Thus **S** could take on values from 0 (definitely unsuitable) to 4 (excellent), concerning the suitability of the land site for golf course construction.

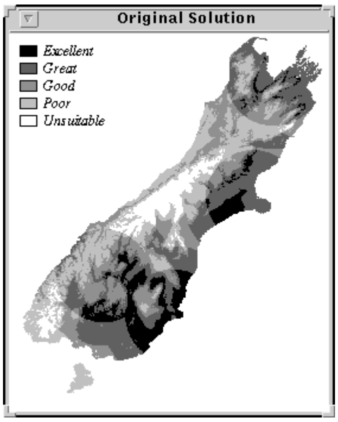

Figure 1. Solution produced by a human expert

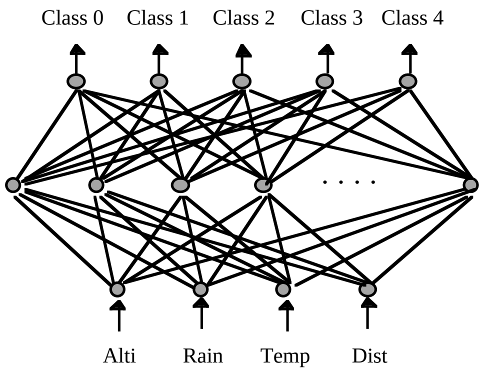

Figure 2. A three-layer MKP for the golf-course problem.

4 Neural networks for spatial information processing

A MLP NN, with 4 input nodes, 20 hidden-layer nodes, and 5 output nodes, was first trained, using only 1,000 of the 153,036 possible data blocks (Fig. 2). Despite the relatively small training set, the artificial neural network was found to provide about 80% of the correct values over the full test set of 153,036 blocks (Figure 3 and Figure 4) [20].

	100 Samples	1,000 Samples	10,000 Samples
Training epochs	1,500	890	7,900
Error after training	0.2301	0.2302	0.2309
No difference	106,782 (67.78%)	126,480 (82.65%)	127,343 (83.21%)
One class difference	45,078 (29.45%)	25,800 (16.86%)	25,362 (16.57%)
Two class difference	1,176 (0.77%)	756 (0.49%)	331 (0.22%)

Figure 3. Confusion classification table for the NN solution

As can be seen from Fig. 3, using only 1,000 of the 153,036 pixels, 82.65% of the pixels were classified correctly. Moreover, assuming that one class misclassifying is tolerable, more than 98% of the results of the NN recalling solutions will be acceptable, even if only 1000 samples are selected as training data of neural net from 153,036 possible ones. Nevertheless there is no great improvement in the solution from the case of 1,000 samples to the case of 10,000 samples [20]

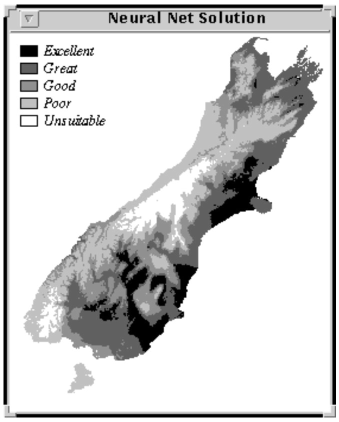

Figure 4. Testing the NN solution on the whole data set.

5 Fuzzy neural networks for spatial information processing

A FNN, as described in [1,15], was then used for the same task discussed in Section 4. Five linguistic values: *unsuitable*, *poor*, *good*, *great*, and *excellent*, were created for describing the output (decision) variable - the suitability level. They are presented as five fuzzy membership functions, depicted in Figure 5.

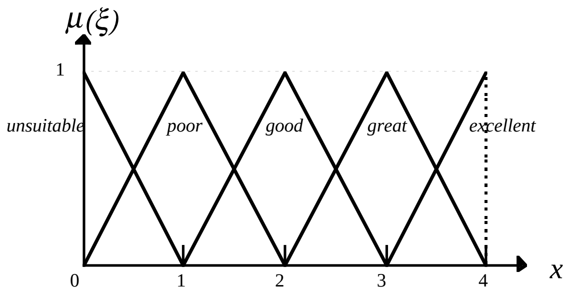

Figure 5. Membership function of suitability

All the input variables: altitude, rainfall, temperature, and distance, are represented as five fuzzy values each as illustrated in Figure 6.

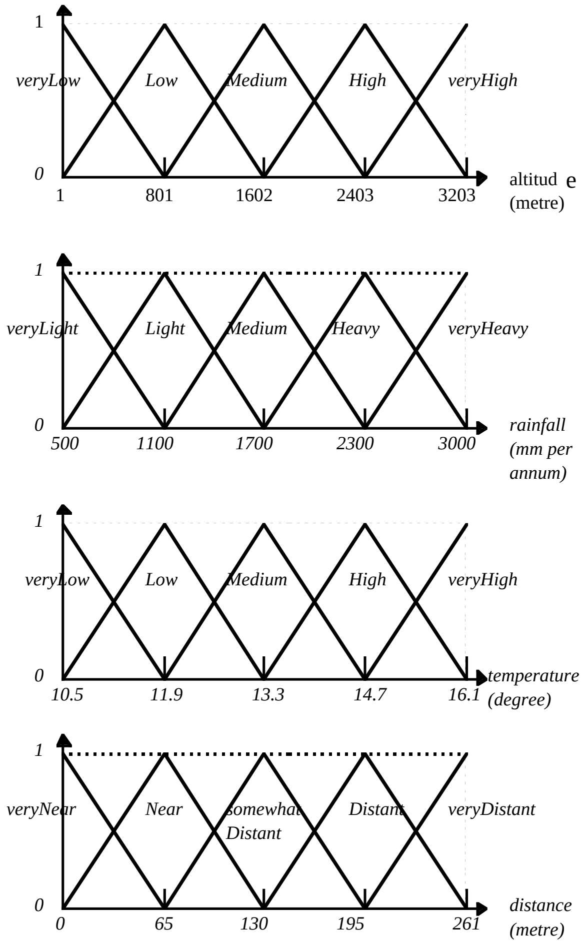

Figure 6. Membership functions for the input variables

To match the linguistic input values, a three layer FNN was created with 20 input nodes (each of which is associated with a membership function of the input variables), 20 hidden nodes, and 5 output nodes to calculate the output membership degrees (Figure 7) . The same number of (now appropriately fuzzified) samples, as those in the previous section, were taken from the 153,036 data examples. The FNN was trained with the 1,000 fuzzified samples using the backpropagation algorithm.

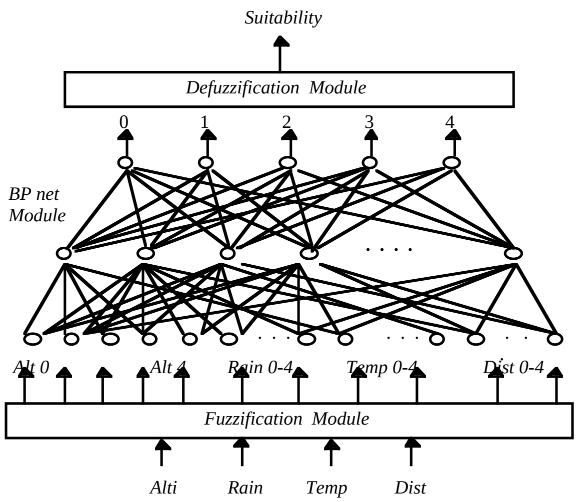

Figure 7. The partial-FNN architecture with the fuzzification and defuzzification procedures

The FNN was tested again over the entire data set, illustrated in Figure 8. Its evaluation is shown in Figure 9. It can be seen that the generalisation ability of the FNN was found to be better than that of the neural network solution given in the previous section.

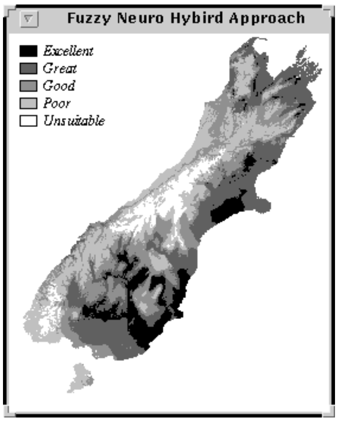

Figure 8. Solution of fuzzy neuro hybrid system

Training epochs	400
Error after training	0.1140
No difference	131,060 (85.64%)
One class difference	21,729 (14.20%)
Two classes difference	247 (0.16%)

Figure 9. The test results for the FNN solution

Using the FNN not only provides a better solution, but it also makes possible the extraction of underlying classification rules. Appropriate fuzzy inference methods can then be applied over the extracted rules, thus making possible to use a fuzzy rule-based system for the classification task. This issue is discussed in the next section.

6 Rules extraction and fuzzy reasoning for spatial systems

Three methods for rules extraction from the trained FNN were investigated. The motivation for these three methods was to examine the performance and appropriateness of different rule structures with respect to spatial information processing. In each of the three cases the extracted rules were interpreted by applying various fuzzy reasoning methods as explained later in this section. Classification test results are presented and compared with the results obtained in the previous two sections.

The first rules extraction method used was the REFuNN [15] algorithm. A "zeroing" operation was performed on the FNN, and only the connection weights which were greater in value than a given threshold were kept. Thus, in this case, negative connection weights were not retained. Fuzzy rules with numerical coefficients of importance and confidence factors were then extracted, using the retained weights. An example of such a rule with its numerical coefficients is shown below (all input values range among A, B, C, D, or E, with A being the lowest value and E being the highest value) (see Figure 10):

if <ALTITUDE is B 6.216> and <RAINFALL is D 4.852> and <TEMPERATURE is A 5.302> and <DISTANCE is B 6.68>
then <SUITABILITY is B 6.218>

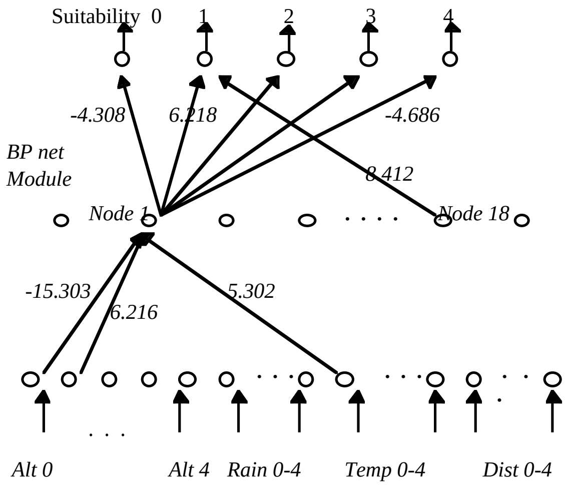

Figure 10. Connection weights of a trained FNN are Interpreted as fuzzy rules by rules extraction algorithms

In order to apply min-max compositional or other methods for fuzzy inference over simple fuzzy rules, the weighted rules extracted above can be converted into simple rules by simply ignoring the numerical coefficients attached to them. For example the above given rule is converted into the following one:

If <ALTITUDE is B> and <RANFALL is D> and <TEMPERATURE is A> and <DISTANCE is B>
then <SUITABILITY> is B>.

The conversion was made by retaining only those rule elements that have coefficients of importance above a certain threshold value (which may be set according to the problem at hand). Appendix A gives a sampling of the rules extracted by the above rules extraction method when a threshold of 4 is used. When these simple fuzzy rules are evaluated using a min-max compositional fuzzy inference method, the performance is not as good as the previous ones - only 40% of the South Island blocks (pixels) were categorised correctly and 50% are off by one from the correct value. This is a reflection of the inevitable loss of information during the conversion of the extracted weighted rules into simple "flat" rules.

Although the rules derived from the above approach were relatively simple, the method can yield a large number of rules. In the above case there were 254 rules extracted when the REFuNN algorithm was used. For the extracted rule set to be convenient for spatial analysis professionals, a smaller number of rules would be desirable. In order to arrive at a smaller number of rules and achieve better inference performance from extracted rules, a second rules extraction and fuzzy inference method was developed that used an evidential reasoning approach as explained below:

- First, all the node connections in the fuzzy neural net that had a weight value below a certain threshold value (0 in this case) were constrained to be zero. In other words, all negative weighted connections were set to zero for this experiment. Then the fuzzy neural net was retrained under this constrained condition. This first step could be repeated, if necessary.

- Then fuzzy rules are extracted from this neural network. Rule components were only derived from node connections that had weights above another chosen threshold value (4 in this case).
- For the inference procedure, the overall degree of matching for the left-hand side of each rule is calculated, which is a weighted sum of the membership values to which input data belong to all its antecedent elements. A rule fires if and only if the overall matching degree of its antecedent part is positive.
- Then the degree to which each of the output membership functions is inferred collectively by all the rules is determined by calculating a weighted sum of all the confidence factors associated with that output membership function from the activated rules.

This is illustrated in the following example.

Suppose, the altitude of a block is 267.8 metre, the rainfall is 2,400 mm/annum, the temperature is 10.5 degree, and the distance is 260 metre. The membership function values (&) of the input variables to which these data belong was found to be:

$\mu_{veryLow}$ (Altitude) = 0.3,
μ_{Low} (Altitude) = 0.7,
μ_{Heavy} (Rainfall) = 0.9,
$\mu_{veryLow}$ (Temperature) = 1.0,
μ_{High} (Temperature) = 0,
μ_{Near} (Distance) = 0,
$\mu_{somewhatDistant}$ (Distance) = 0.

Then, overall degree of matching of the left-hand side of the exemplar rule from Figure 10 may be calculated by the following summation:

-15.303(0.3_+ 6.216(0.7 + 4.852(0.9 + 5.302 = 10.69,

which is positive, so all the rules that contain this left-hand side will fire. When this is done, an overall degree of < Suitability is *Poor* > is calculated as 6.218 + 8.412 = 14.63, which is positive again. Therefore, the membership function value μ_{Poor} (Suitability) = 1, where the defuzzification method has been employed to obtain the final crisp solution.

A modified version of REFuNN was developed where negative connections not excluded from the resulting rules, and the number of rules equals the number of hidden nodes. The negative weights are represented in by using "not" in the rule. A representative rule is the following one:

if <ALTITUDE is A 7.938> or
<ALTITUDE is not B 10.530> or
<ALTITUDE is not C 5.442> or
<RAINFALL is not C 4.294> or
<RAINFALL is not D 4.457> or <TEMPERATURE is C 5.214> or
<TEMPERATURE is not D 5.638> or <DISTANCE is not C 10.168> or
<DISTANCE is not D 8.900>
then <SUITABILITY is not C 12.2237> and <SUITABILITY is not D 7.177> and
<SUITABILITY is E 8.946>

This second approach produces more complicated rules, but fewer in number than the first approach: there were now only 20 rules extracted for the sample golf course problem. It is our conjecture that this rule set would be more meaningful, and hence more practically useful, to spatial analysis professionals than the first rule set. When the evidential inference method was applied with the 20 fuzzy rules extracted for a threshold of 4, the results shown in Fig.11 were obtained. This represented an improved performance over that of the first method, i.e. 61% correct (61% in the category of "no difference" between the fuzzy inference classification and the "correct" classification). In general, the lower the extraction threshold, the more fuzzy rule components are extracted from a trained FNN and the more accurate is the solution achieved. The fuzzy inference solution over the rules for a threshold of 2 is pictured in Figure 12.

No difference	93,402 (61.03%)
One class difference	43,147 (28.19%)
Two class difference	10,766 (7.04%)
Three class difference	3,844 (2.51%)
Four class difference	1,877 (1.23%)

Figure 11. The results after applying fuzzy evidential reasoning over fuzzy rules extracted with the use of the modified REFuNN algorithm for a threshold of 2.

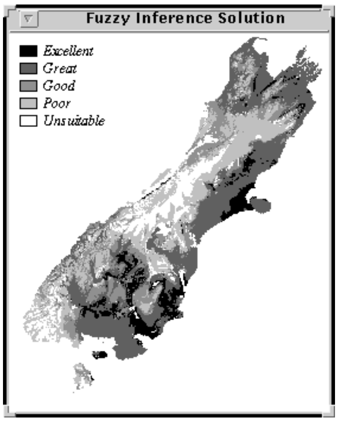

Figure 12 .Test results after applying fuzzy evidential reasoning over fuzzy rules extracted with the use of the simplified REFuNN algorithm for a threshold of 2.

Although the solution of the fuzzy rule based approach above is not as precise as that of the NN or the FNN ones, 89% classification accuracy is achieved if one class error of misclassification is tolerated.

In order to generate a fuzzy rule set that involved simpler rules (fewer rule components) than that produced by the above method, a third fuzzy rule extraction approach was developed. It is called "LEave the Strongest COnnections and their Neighbouring ones" (LESCON). According to this method the FNN was trained as before, with negative weighted connections set to zero. Then the FNN was retrained with only the strongest input node connections and its neighboring connections being retained. All other input node connections were constrained to be zero during this last retraining stage. When rules are extracted each hidden node represents a single fuzzy rule, which has only the strongest connection from each fuzzy input variable as well as its neighbouring connections, represented in its antecedent part. This will result in fewer components in the antecedent part of the rule than the above-described second approach. A set of 20 fuzzy rules, extracted in this fashion is shown in Appendix B. A representative rule from this approach was

if <Altitude is not A 16.482> or
<Rainfall is not C 2.186> or
<Rainfall is E 2.423> or
<Temperature is A 12.592> or <Temperature is B 5.095> or
<Distance is not C 8.566> or
<Distance is D 5.019>
then <Suitability is A 13.469> and <Suitability is not B 19.502>

This rule is relatively easy to interpret, since it essentially says that if the Altitude is not very low or the Rainfall is very high or the Temperature is low or the Distance is relatively great, then the Suitability is very low. The rules derived by this LESCON approach were tested on the entire data set and resulted in 56% of the pixels classified "correctly" (Fig. 13). Although the overall inference performance wasn't quite as good as the second method, the simpler structure of these rules were thought to be potentially more valuable for practical use.

No difference	86,367 (56.4%)
One class difference	39,749 (26.0%)
Two class difference	13,196 (8.6%)
Three class difference	7,176 (4.7%)
Four class difference	6,548 (4.3%)

Figure 13. Test results (confusion table) for the evidential reasoning method applied on fuzzy rules extracted by using the LESCON method.

7 Conclusion

The paper presents a novel approach, *neuro-fuzzy engineering*, to spatial data analysis and to building decision making systems based on spatial information processing. It affords the possibility that the system under construction can learn from data, perform approximate reasoning, extract rules from the data, and explain the underlying rules of the solution to the spatial problem. Three fuzzy rule development approaches were described in the context of a golf course suitability decision making problem and experimental results were presented. It is important to emphasize that the extraction of rules is valuable only to the degree to which the extracted rules are meaningful and comprehensible to human observers. Three rule-extraction and inference approaches were developed that had differing degrees of complexity and inference performance, and the decision as to which one is superior can only be made by spatial analysts for a given application. The neuro-fuzzy engineering approach seeks to *engineer* appropriate rule extraction processes for given application tasks in spatial analysis, and this can only be ultimately accomplished with the collaborative participation of spatial information professionals to provide appropriate feedback. Further research aims it is planned to combine the neuro-fuzzy engineering techniques with the traditional geographic information systems, known as GIS, in order to combine the excellent visualisation and statistical analysis features of GIS with the neuro-fuzzy engineering techniques for sophisticated spatial information processing.

Acknowledgment

The reported work was conducted under a PGSF (Public Good Science) research grant UOO509, funded by the New Zealand Foundation for Research, Science and Technology.

References

[1] Kasabov, N., "Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering", MIT Press, Cambridge, Massachusetts, 1996.

[2] D Rumelhart and J McClelland, Parallel Distributed Processing: Explorations in the Micro-structure of Cognition, Vol 1 and 2, Cambridge: MIT Press, 1986.

[3] D Rumelhart, J McClelland, and R Williams, "Learning representations by backpropagating errors", Nature, Vol 323, 1986, pp. 533-536.

[4] G. Cybenko, "Approximation by Superposition of Synosuidal Function", Mathematics of Control, Signals, and Systems, 2 (1989), pp. 303-314.

[5] L A Zadeh, "Fuzzy sets", Information and Control, Vol 8, 1965, pp. 338-353.

[6] M Sugeno and G T Kang, "Structure identification of fuzzy model", Fuzzy Sets and Systems, Vol 28, 1988, pp. 15-33.

[7] T Takagi and M Sugeno, "Fuzzy identification of systems and its applications to modeling and control", IEEE Transactions on Systems, Man, and Cybernetics, Vol 15, No 1, 1985, pp. 116-132.

[8] Bezdek, J.C., Hall, L.O. and Clarke, L.P., "Review of MR Image Segmentation Techniques Using Pattern Recognition", Medical Physics, 20(4), 1993: 1033-1048.

[9] Bezdek, J.C., "A Review of Probabilistic, Fuzzy, and Neural Models for Pattern Recognition", Journal of Intelligent and Fuzzy Systems, 1(1), 1993: 1-25.

[10] Kasabov, N.K., "Hybrid Connectionist Fuzzy Production System: Towards Building Comprehensive AI", Intelligent Automation and Soft Computing, 1(4), 1995: 351-360.

[11] Horikawa, S.-I., Furuhashi, T. and Uchikawa, Y., "On Fuzzy Modelling Using Fuzzy Neural Networks with the Back-Propagation Algorithm", IEEE Transactions on Neural Networks, 3(5), 1992: 801-806.

[12] Gupta, M. and Rao, D.H. On the principles of fuzzy neural networks, Fuzzy Sets and Systems, 61(1), 1994: 1-18.

[13] N K Kasabov, "Learning fuzzy production rules for approximate reasoning with connectionist production systems", in: Artificial Neural Networks 2, North Holland, Lawrence Erlbaum, 1993, pp. 337-345.

[14] N K Kasabov, "Learning fuzzy rules through neural networks", Proceedings of the first New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems, 1993, pp. 137-140.

[15] Kasabov, N.K., "Learning Fuzzy Rules and Approximate Reasoning in Fuzzy Neural Networks and Hybrid Systems", Fuzzy Sets and Systems, Special Issue, 1996:1-19.

[16] R Andrews, J Diederich, and A B Tickle, "A survey and critique of techniques for extracting rules from trained artificial neural networks", PQUTNRC-95-01-02, Queensland University of Technology, 1995.

[17] L M Fu, "Rule generation from neural networks", IEEE Transactions on Systems, Man, and Cybernetics, Vol 24, No 8, 1994, pp. 1114-1124.

[18] S. Openshaw, "Developing Appropriate Spatial Analysis Methods for GIS", in Geographical Information Systems, Vol. 1 Principles, D. J. Maguire, M. F. Goodchild, and D. W. Rhind (eds.), Longman Scientific, Essex, (1991) pp. 389-402.

[19] X. Zhuang and B. A. Engel, "Classification of Multispectral Remote Sensing Data Using Neural Networks vs. Statistical Methods", in Proceedings of the International Winter Meeting of the American Society of Agricultural Engineers, Chicago, (1990).

[20] G. L. Benwell, N. Kasabov, M. K. Purvis, F. Zhang, B. R. Mclennan, and S. Mann, "Spatial Analysis with Artificial Neural Networks", in Conf. Proc. Eight Australian Joint Artificial

Intelligence Conference, Proceedings of the Workshop on AI and the Environment, Australian Defence Force Academy, Canberra, Australia, 1995, pp. 43-52.
[21] M K Purvis, N K Kasabov, F Zhang, and G L Benwell, “Connectionist-based methods for knowledge acquisition from spatial data”, Proceedings of the IASTED International Conference on Advanced Technology in the Environmental Field, Gold Coast, Australia, 1996, pp. 151-154.

Appendix A. Sampling of fuzzy rules extracted from a trained FNN for the golf-course problem by using the REFuNN method for a threshold of 4.

if <Altitude is C> and <Rainfall is A> and <Temperature is D> and <Distance is A>
then <Suitability is A>
else
if <Altitude is C> and <Rainfall is A> and <Temperature is D> and <Distance is D>
then <Suitability is A>
else
if <Altitude is C> and <Rainfall is D> and <Temperature is D> and <Distance is A>
then <Suitability is A>
else
if <Altitude is C> and <Rainfall is D> and <Temperature is D> and <Distance is D>
then <Suitability is A>
else
if <Altitude is B> and <Rainfall is E> and <Temperature is A>
then <Suitability is A>
else
if <Altitude is C> and <Rainfall is E> and <Temperature is A>
then <Suitability is A>
else
...
if <Altitude is B> and <Rainfall is D> and <Temperature is A> and <Distance is C>
then <Suitability is B>
else
if <Altitude is B> and <Rainfall is D> and <Temperature is B> and <Distance is C>
then <Suitability is B>
else
if <Altitude is C> and <Rainfall is D> and <Temperature is A> and <Distance is C>
then <Suitability is B>
else
if <Altitude is C> and <Rainfall is D> and <Temperature is B> and <Distance is C>
then <Suitability is B>
else
if <Altitude is B> and <Rainfall is E> and <Temperature is A>
then <Suitability is B>
else
if <Altitude is C> and <Rainfall is E> and <Temperature is A>
then <Suitability is B>
else
if <Altitude is D> and <Rainfall is E> and <Temperature is A>

```
  then <Suitability is B>
else
...
if  <Temperature is C> and <Distance is D>
  then <Suitability is C>
else
if  <Altitude is B> and <Rainfall is D> and <Temperature is A> and <Distance is C>
  then <Suitability is C>
else
if  <Altitude is B> and <Rainfall is D> and <Temperature is B> and <Distance is C>
  then <Suitability is C>
else
if  <Altitude is C> and <Rainfall is D> and <Temperature is A> and <Distance is C>
  then <Suitability is C>

else
...
if  <Altitude is B> and <Rainfall is B> and <Temperature is B> and <Distance is B>
  then <Suitability is D>
else
if  <Altitude is B> and <Rainfall is B> and <Temperature is B> and <Distance is D>
  then <Suitability is D>
else
if  <Altitude is B> and <Rainfall is B> and <Temperature is E> and <Distance is A>
  then <Suitability is D>
else
if  <Altitude is B> and <Rainfall is B> and <Temperature is E> and <Distance is B>
  then <Suitability is D>
else
if  <Altitude is B> and <Rainfall is B> and <Temperature is E> and <Distance is D>
  then <Suitability is D>
else
if  <Altitude is B> and <Rainfall is D> and <Temperature is A> and <Distance is A>
  then <Suitability is D>
else
if  <Altitude is A> and <Temperature is C> and <Distance is B>
  then <Suitability is E>
else
if  <Altitude is A> and <Rainfall is B> and <Distance is B>
  then <Suitability is E>
else
if  <Altitude is A> and <Rainfall is E> and <Distance is B>
  then <Suitability is E>
else
if  <Altitude is B> and <Rainfall is D> and <Temperature is D> and <Distance is A>
  then <Suitability is E>
else
if  <Altitude is B> and <Rainfall is D> and <Temperature is D> and <Distance is D>
  then <Suitability is E>
```

else
if <Altitude is B> and <Rainfall is E> and <Temperature is D> and <Distance is A>
then <Suitability is E>
else
if <Altitude is B> and <Rainfall is E> and <Temperature is D> and <Distance is D>
then <Suitability is E>
else
if <Altitude is C> and <Rainfall is D> and <Temperature is D> and <Distance is A>
then <Suitability is E>
else
if <Altitude is C> and <Rainfall is D> and <Temperature is D> and <Distance is D>
then <Suitability is E>
else
if <Altitude is C> and <Rainfall is E> and <Temperature is D> and <Distance is A>
then <Suitability is E>
else
if <Altitude is C> and <Rainfall is E> and <Temperature is D> and <Distance is D>
then <Suitability is E>

Appendix B. Fuzzy rules extracted from a trained FNN on the golf-course data with the use of the "LEave the Strongest COnnections and the Neighbouring ones" (LESCON) method

if <Altitude is A 13.0273> or <Altitude is not B 15.9051> or <Rainfall is E 10.4085> or <Temperature is not B 4.09406> or <Distance is C 8.75057> or <Distance is not D 4.44167>
then <Suitability is A 8.89592> and
<Suitability is not B 10.754> and <Suitability is C 9.3568> and <Suitability is not D 7.86719>
else
if <Altitude is A 4.95228> or <Altitude is not B 6.76817> or <Rainfall is not A 3.95766> or <Rainfall is B 4.79682> or <Temperature is not A 3.30265> or <Distance is B 8.26505>
then <Suitability is not A 10.7605> and
<Suitability is C 6.73752> and <Suitability is not D 8.50128> and <Suitability is E 8.21429>
else
if <Altitude is A 7.86952> or <Altitude is not B 9.91048> or <Rainfall is A 2.15575> or <Temperature is C 7.31038> or <Temperature is not D 6.64119> or <Distance is not A 3.78694> or <Distance is not B 2.52337> or <Distance is not C 4.76734>
then <Suitability is not C 8.37734> and
<Suitability is not D 6.67695> and <Suitability is E 7.9864>
else
if <Altitude is A 9.01384> or <Rainfall is A 9.84078> or <Temperature is not B 5.09546> or <Temperature is C 7.70795> or <Temperature is D 6.67762>
then <Suitability is not A 4.25216> and
<Suitability is not B 14.6772> and
<Suitability is not C 11.5201> and <Suitability is D 14.46>
else
if <Altitude is B 12.7054> or <Altitude is not C 18.9117> or <Rainfall is not C 5.37072> or <Rainfall is D 10.3822> or <Rainfall is not E 4.8775> or <Temperature is A 2.2537> or <Temperature is not B 3.05944> or <Distance is A 10.6352> or <Distance is not B 3.9751>
then <Suitability is not B 8.13483> and
<Suitability is D 6.13101>
else
if <Altitude is B 2.79218> or <Altitude is C 11.8224> or <Rainfall is D 3.71634> or <Rainfall is E 4.06815> or <Temperature is A 11.1077> or <Distance is A 3.76611> or <Distance is B 11.9971> or <Distance is not C 10.8761>
then <Suitability is not C 17.2116> and
<Suitability is D 8.30443> and <Suitability is not E 13.1925>
else
if <Altitude is B 3.39278> or <Altitude is C 4.72182> or <Rainfall is D 8.02081> or <Rainfall is E 3.04538> or <Temperature is D 20.8153> or <Distance is not C 12.6237> or <Distance is D 2.34545>
then <Suitability is not A 14.0995> and
<Suitability is B 16.6715> and <Suitability is E 8.62711>
else
if <Altitude is B 7.93521> or <Rainfall is not A 7.05954> or <Rainfall is B 10.2588> or <Rainfall is C 2.29922> or <Temperature is not B 2.60861> or <Temperature is C 8.36302> or <Temperature is not D 3.42525> or <Distance is C 5.45352>
then <Suitability is not A 5.54203> and <Suitability is not B 14.0267> and <Suitability is D 14.2643> and
<Suitability is not E 18.2878>
else

if <Altitude is C 12.6486> or <Altitude is not D 2.91195> or <Rainfall is A 7.56551> or <Rainfall is B 3.92754> or <Temperature is not B 7.54877> or <Temperature is C 9.28581> or <Temperature is not D 5.83064> or
<Distance is not C 16.1327> or <Distance is D 14.878>
then <Suitability is not D 9.42645> and
<Suitability is E 9.46109>
else
if <Altitude is C 16.5925> or <Rainfall is not C 3.59343> or <Rainfall is D 5.31406> or <Rainfall is E 3.43175> or <Temperature is A 6.2262> or <Distance is C 10.5317> or <Distance is D 3.94971>
then <Suitability is not A 4.86968> and
<Suitability is C 10.0662> and <Suitability is not D 12.9877>
else
if <Altitude is C 8.00641> or <Rainfall is D 5.90299> or <Rainfall is not E 7.75102> or
<Temperature is not C 4.89196> or <Temperature is D 3.67105> or <Distance is not C 7.0164> or
<Distance is D 6.24405>
then <Suitability is A 7.57558> and
<Suitability is not B 6.01196> and <Suitability is C 4.61644>
else
if <Altitude is D 7.63874> or <Rainfall is D 9.42996> or <Temperature is A 16.2979> or
<Temperature is not B 5.35385> or <Distance is C 6.51662> or <Distance is D 21.6948>
then <Suitability is B 11.0054> and
<Suitability is not C 15.6962>
else
if <Altitude is not A 15.7708> or <Rainfall is C 2.15311> or <Rainfall is D 7.01743> or <Rainfall is E 3.41303> or <Temperature is A 6.47468> or <Temperature is B 5.23355> or <Distance is not B 5.29901> or <Distance is C 5.72783> or <Distance is not D 3.80905>
then <Suitability is B 9.29823> and <Suitability is C 13.9202> and <Suitability is not D 24.3266> and
<Suitability is not E 4.68894>
else
if <Altitude is not A 16.4821> or <Rainfall is not C 2.18582> or <Rainfall is E 2.42311> or
<Temperature is A 12.5918> or <Temperature is B 5.09485> or <Distance is not C 8.56606> or
<Distance is D 5.0194>
then <Suitability is A 13.4692> and
<Suitability is not B 19.5022>
else
if <Altitude is not A 17.2851> or <Altitude is B 2.50714> or <Altitude is C 3.07074> or <Rainfall is E 5.79224> or <Temperature is not B 4.34055> or <Distance is not A 6.03842>
then <Suitability is A 8.94499> and <Suitability is B 4.34786> and <Suitability is not C 12.5205>
else
if <Altitude is not A 3.44423> or <Altitude is B 4.48606> or <Rainfall is E 6.84339> or <Distance is not C 3.34067> or <Distance is D 5.2221>
then <Suitability is C 5.80954> and
<Suitability is not D 10.0862> and
<Suitability is not E 5.09162>
else
if <Altitude is not B 5.05389> or
<Temperature is not B 10.0301> or <Temperature is C 3.47939> or <Distance is not C 8.3466> or
<Distance is D 14.976>

then <Suitability is not B 6.9778> and
<Suitability is C 12.4682> and <Suitability is not D 11.9576> and <Suitability is not E 4.95697>
else
if <Altitude is not B 5.12584> or <Altitude is not C 3.33649> or <Rainfall is not B 7.44466> or <Rainfall is C 3.05647> or <Temperature is A 3.81249> or <Distance is C 3.08615> or <Distance is D 6.32319> or <Distance is not E 2.72201>
then <Suitability is A 10.4042> and
<Suitability is not B 15.1288> and <Suitability is C 8.72826>
else
if <Altitude is not B 5.66248> or <Rainfall is not A 3.60335> or <Rainfall is not B 2.65114> or <Temperature is A 3.12132> or <Temperature is B 4.72171> or <Temperature is not C 16.06> or <Distance is C 3.55934> or <Distance is D 17.3116>
then <Suitability is A 5.93074> and <Suitability is B 6.8969> and <Suitability is not C 20.0265> and
<Suitability is not D 6.59029>
else
if <Altitude is not B 7.70367> or <Altitude is C 8.46831> or <Rainfall is not D 9.59137> or <Rainfall is E 7.71643> or <Temperature is not B 4.55086> or <Temperature is not D 5.29569> or <Distance is not B 3.7394> or <Distance is C 9.33156> or <Distance is not D 9.40808>
then <Suitability is not A 7.84688> and <Suitability is B 7.35638> and <Suitability is D 7.76816> and
<Suitability is not E 8.05753>