A Membership Function Selection Method for Fuzzy Neural Networks

Qingqing Zhou, Martin Purvis and Nikola Kasabov

Computer and Information Science, University of Otago, PO Box 56, Dunedin, New Zealand
Email: mpurvis@commerce.otago.ac.nz

Abstract

Fuzzy neural networks provide for the extraction of fuzzy rules from artificial neural network architectures. In this paper we describe a general method, based on statistical analysis of the training data, for the selection of fuzzy membership functions to be used in connection with fuzzy neural networks. The technique is first described and then illustrated by means of two experimental examinations.

1 Introduction

A fuzzy neural network blends elements of fuzzy and neural network computations into a single connectionist architecture [Horikawa *et al*, 1992; Kasabov, 1993, 1996; Purvis *et al*, 1997]. In general terms, two different types of neural-fuzzy hybrids have emerged: one that uses neural networks to derive the parameters of a fuzzy system, and another that provides an implementation of a fuzzy system within a neural network architecture. It is the latter type hybrid that is discussed here.

Neural-fuzzy systems can be used to derive solutions involving the neural network's advantages of model-free learning, good generalization, and powerful non-linear mapping capabilities. The primary strength obtained from the fuzzy logic component is a system that can both be initialized by the existing semantic knowledge, and have structured information (knowledge) extracted from it in an interpretable format. Thus the "black box" difficulty, which is widely regarded as a principle weakness of neural networks, can be lessened.

There are a series of steps to implement a basic neural-fuzzy system: (1) convert real-valued data into a fuzzified representation; (2) train the fuzzified information with a neural network; and then (3) de-fuzzify the result to produce real values of the desired output. After the system is trained to satisfaction, fuzzy rules can be extracted from the trained neural network.

1.1 The FuNN network

We employ a neural-fuzzy model, called FuNN [Kasabov 1993; Kasabov, 1996], which consists of five layers: an input variable layer; a condition element (fuzzification) layer; a rule layer; an action element (de-fuzzification) layer; and an output layer. In Figure 1, each of the two inputs can be fuzzified in the condition layer by representing the degree of their membership in fuzzy sets. When the FuNN fuzzy neural network is trained, backpropagation training algorithm is used to adjust either the weights of the middle layers (layers 2, 3, and 4) or the weights of all the layers, including the fuzzification and defuzzification layers.

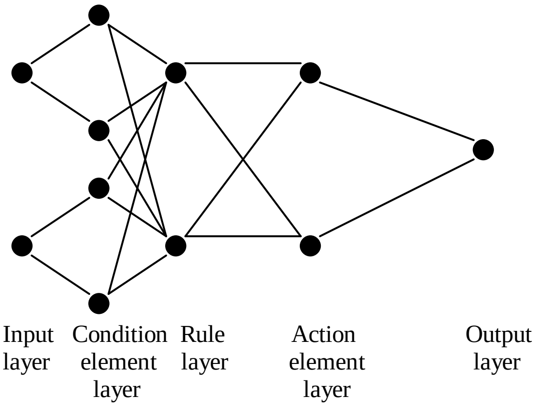

Figure 1. The FuNN architecture

2 Fuzzy Membership Functions

The neurons in the input layer of FuNN architecture represents the input variables as crisp values. These values are fed to the condition element layer that performs fuzzification. Building the proper membership functions (MFs) is a key in the fuzzification process. Several approaches for building and adapting membership functions are proposed [Kasabov 1993; Kasabov 1996; Hauptmann *et al*, 1995; Horikawa *et al*, 1992]. Three methods for producing the membership functions employed by FuNN are as following:

2.1 A fixed center-based membership approach

A center-based MFs approach can be implemented by using three-point triangular membership functions. The triangles are completed with the minimum and maximum points attached to adjacent centers, or shouldered in the case of the first and last membership functions. The membership functions are spaced equally according to the minimum and maximum values of the input data. (Figure 2(a))

With these triangular membership functions each input value will belong to no more than two fuzzy sets, and their membership degrees will always sum to one. Though this approach is straightforward, the division into equally spaced membership functions may be naïve and inappropriate for some data sets.

2.2 Manual adjustment of MF centers

For triangle-shaped MFs, the centers can be adjusted manually. In this case a rigid partitioning is often used to establish a region within which each center can move but not cross. Generally, manual alterations only adjust the membership functions slightly and in some cases must rely on human experience and knowledge.

2.3 Modified backpropagation training algorithm for adjustment of MFs

This algorithm [Horikawa *et al*, 1992; Hauptmann *et al*, 1995; Kasabov 1993; Kasabov *et al*, 1997] allows changes to be made to MFs, subject to constraints necessary for retaining semantic meaning. In the FuNN structure, the fuzzification layer and the defuzzification layer changes their input connections based on simple and intuitive formulas. These changes reflect the concepts represented by the layers and must satisfy the constrains imposed on the membership functions.

2.4 Genetic algorithms for adaptation of MFs

Genetic algorithms (GAs), which are based on the principles of genetics and natural selection, are widely used as powerful search techniques. Their use for adaptation of fuzzy membership functions has been shown to lead to more effective solutions than manual alteration. The work carried out in [Mang *et al*, 1995] proposed an algorithm that uses two factors (shift factor and shrink factor) to make small changes to the width and center positions of the membership functions. A similar approach [Kasabov *et al*, 1997] has been taken in the GA module of FuNN, where again fixed, arbitrary boundaries have been set up to limit the amount of possible adjustment. In this case only the centers need to be represented in the chromosome of the GA modules, speeding up the adaptation process and possibly reducing spurious local minima in the approach used by [Mang *et al*, 1995]. The use of GAs, however, can be computationally expensive. In the following, we describe an alternative approach based on the c^2 statistic to build membership functions before the FuNN structure is created and trained.

3 MFs Based on the Chi2 Approach

The goal of Chi2-based membership approach is to choose the optimal membership functions via a statistically-based algorithm that can make neuro-fuzzy computation more efficient. The Chi2-based membership approach performs automatic discretisation of the data, which can lead to an appropriate selection of the number and widths of the membership functions.

3.1 Chi2 algorithm

The Chi2 algorithm [Setiono, 1997] is a general algorithm that uses the c^2 statistic to discretise numeric attributes and achieve feature selection. It conducts a significance test on the relationship between the value of an attribute and the categories of neighboring classifications and consists of two phases. In the first phase data values for a given attribute are sorted and associated with an interval (initially, the number of intervals equals the number of distinct values of an attribute). Then the c^2 value is calculated for each pair of adjacent intervals, and adjacent intervals are merged if their c^2 values fall below a certain value. This is done for each attribute and the merged intervals now represent a discretisation of the data set. As intervals are merged, inconsistencies can appear (identical inputs yielding different outputs). The above process is repeated until an inconsistency rate, d, is exceeded in the discretised data.

The second phase of the Chi2 approach is a finer process of the first phase. Each attribute takes turns for merging. If during this process an attribute has been merged to a single interval, then it will not be involved in further merging.

3.2 Constructing MFs via the Chi2 algorithm

The merged intervals for a given attribute (input) from the Chi2 algorithm (phase 1 or phase 2) determine the number and the widths of the membership functions. Four-point trapezoidal membership functions are used that cause each input value to belong to a maximum of two membership functions, the membership degrees for which will always sum to one. The boundaries of each membership function are decided as follows: the smaller interval is chosen from each pair of adjacent intervals. Then the half (or quarter) size spaces of these smaller intervals are calculated. The fuzzy boundaries are obtained by setting those spaces on both sides of each interval boundary.

Example

We illustrate how the Chi2-based membership approach works using the Pima Indians Diabetes database, which contains 768 patterns. Each pattern is described using eight numeric attributes: number of times pregnant, body mass index etc. The class value is either tested positive or negative for diabetes. The data set is split randomly into two sets, with 576 patterns used for finding the proper membership functions based on the Chi2 algorithm and the rest for test. Here the two stages (phase1 and phase2) of Chi2 processing are shown to demonstrate the behavior of the algorithm. The inconsistency rate d was set at 5%. With the phase1 stage the data was discretised such that all eight attributes have the same minimum significance level (SigLevel = 0.99, c^2=6.63), and the number of inconsistencies was kept under the inconsistency threshold of 28 values (28 = 576 * 5%). The phase2 stage then proceeds until no further attribute value merging is possible without sacrificing discriminating power. Table 1 shows the intervals and c^2 values after initialization for the first attribute, "number of times pregnant". The results for this attribute at the ends of the two stages are shown in Table 2. With the c^2 threshold 6.63, for example, four intervals (discrete values) are needed for the first attribute: [0, 1) -> 1, [1, 3) -> 2, [3, 7) -> 3 and [7, ＋∞) -> 4. The membership functions of this input attribute will be illustrated in the next section.

Table1. The initial intervals and c^2 values for number of times pregnant.

Int	c^2		Int	c^2
0	6.34		9	0.61
1	0.02		10	0.02
2	2.51		11	0.07
3	0.16		12	0.29
4	0.35		13	2.06
5	0.94		14	0.1
6	2.45		15	0.1
7	0.65		17	
8	0.01			

Table 2. The intervals and c^2 values for attribute "number of times pregnant" after Phase1 and Phase2. The c^2 thresholds are (a) 6.63 and (b) 20.83.

Int	c^2
0	7.98
1	11.29
3	12.03
7	

(a)

Int	c^2
0	28.38
7	

(b)

4 Experiments

In this section, we employ a FuNN fuzzy neural network, as described in the first section, to test different training and adaptation strategies of the membership functions for two example applications. The Pima Indians diabetes data set is used in the first experiment. The second experiment is based on the problem of determining suitable sites for golf courses in the South Island of New Zealand.

4.1Pima Indian Diabetes Data Set

MFs based on fixed center-based approach

576 patterns were randomly chosen from the original data set for training , and the rest (192 patterns) were for testing. All eight input variables are represented as five fuzzy values. The membership functions for "number of times pregnant" are illustrated in Figure 2(a). To match the linguistic input values, a three layer fuzzy neural network was created with 40 input nodes (each of which is associated with a fixed membership function of the input variables), and 5 output nodes for the de-fuzzification module. The backpropagation algorithm is used to train the middle layers with the 576 fuzzified samples. The test results are shown in Table 3(a).

MFs based on Chi2 approach

The same number of training and testing samples as those in the previous section were taken from the data set for use with the Chi2 approach. The number of membership functions for each input variable are different here, for example 2 membership functions for "number of times pregnant" and 10 membership functions for "body mass index". The membership functions "for number of times pregnant" are shown in Figure 2(b). We use the same fuzzy neural network architecture as in the previous section with 40 input nodes, 20 hidden nodes and 5 output nodes. Its evaluation is shown in Table 3(b).

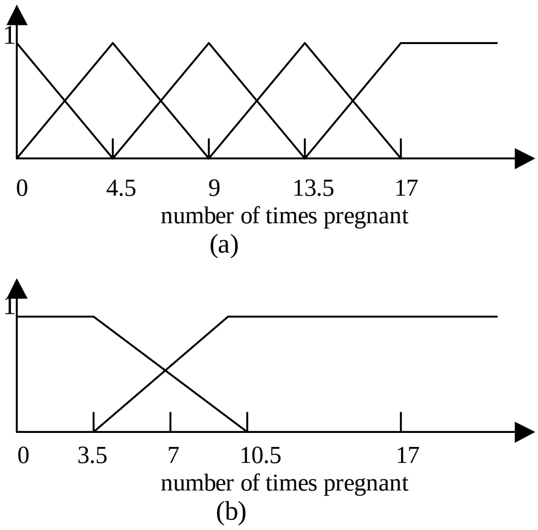

Figure 2. MFs for the number of times pregnant by (a) center-based and (b) Chi2-based approach.

Table 3. The test result for Pima Indian Diabetes data set (a) MFs produced by center-based approach (b) MFs produced by Chi2-based approach.

Training epochs	100
Error after training	0.1826
No difference	146 (76.04%)
One class difference	46 (23.96%)

(a)

Training epochs	100
Error after training	0.1540
No difference	151 (78.65%)
One class difference	41 (21.35%)

(b)

It can be seen that the performance was found to be better with the Chi2 method than that with the previous one.

4.2 Golf Course Problem

We assume that a suitable location for a golf course in the South Island can be determined from the observed data of mean summer temperature, mean annual rainfall, mean altitude, and distance from the nearest of four principle urban centers on the South Island. The output of golf course suitability is taken to have five possible values, ranging from 0 (very unsuitable) to 4 (very suitable) [Purvis *et al*, 1997].

MFs based on fixed center-based approach

All input attributes: altitude, rainfall, temperature, and distance are represented as five fuzzy values. The membership functions for rainfall are illustrated in Figure 3(a) [Purvis *et al*, 1997]. A fuzzy neural network, with 20 input nodes, 20 hidden nodes, and 5 output nodes was trained, using 1000 samples of the 153,036 total data examples. After evaluation over the entire data set, the FuNN was found to classify 85.6% correctly and another 14.2% were off by one membership class.

MFs based on Chi2 approach

MFs for golf course problem based on Chi2 approach are shown in Table 4, which also lists the number of membership function for each attribute based on the center-based approach. The membership functions for rainfall are shown in Figure 3(b) A three layer fuzzy neural network was created with 18 input nodes, 20 hidden nodes, and 5 output nodes to calculate the output membership degrees. The fuzzy neural network was trained with the 10,000 fuzzified samples. After the FuNN was tested again over the entire data set, 92.0% correct values over the full test set and 7.9% were off by one membership class.

Table 4. Number of MF based on two methods

	Center-based method	Chi2-based method
Altitude	5	5
Rainfall	5	4
Temperature	5	5
Distance	5	4

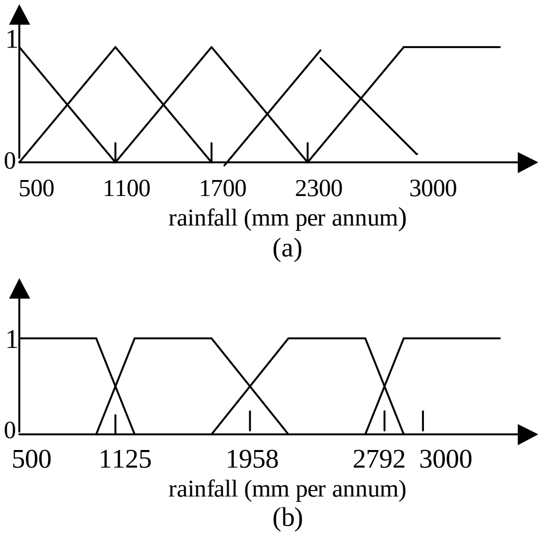

Figure 3. MFs for the rainfall by (a) center-based approach and (b) Chi2-based approach.

5 Conclusion

The fixed center-based approach to membership function selection is for the designer to make a subjective, often arbitrary, determination of the number of fuzzy set values, each with the same function width. The connectionist approach, such as using GAs, to select membership functions is based on training a neural network. The Chi2-based membership offers an alternative approach to let the data determine the nature of the membership functions. Our results show that this approach can lead to satisfactory performance for fuzzy-neural networks. Further research is planned towards on-line adaptation of MFs in a FuNN structure.

Acknowledgements

This work is funded by the PGSF of the New Zealand FRST projects U00606 and U00509.

References

[Purvis *et al*, 1997] Purvis, M., Kasabov, N., Benwell, G., Zhou, Q. and Zhang, F. (1997) Neuro-fuzzy Methods for Environmental Modeling, To appear in Proceedings of the Second International Symposium on Environmental Software Systems, Chapman and Hall, London.

[Kasabov, 1996] Kasabov, N., (1996) Foundation of Neural Networks, Fuzzy Systems and Knowledge Engineering. MIT Press, Cambrige, MA.

[Kasabov, 1993] Kasabov, N., (1993) Learning Fuzzy Rules and Membership Functions in Fuzzy Neural Networks, Proceeding of ANNES'93, Dunedin, New Zealand.

[Hauptmann *et al*, 1995] Hauptmann, W., Heesche, K. (1995) A neural Net Topology for Bidirectional Fuzzy-Neuro Transformation. Proceedings of the International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and the Second International Fuzzy Engineering Symposium, Yokohama, Japan, Vol. 3, 1071 – 1718.

[Horikawa *et al*, 1992] Horikawa, A., Furuhashi, T. and Uchikawa, Y. (1992) On Fuzzy Modeling Using Fuzzy Neural Networks with the Back-Propagation Algorithm. IEEE Transactions On Neural Networks, Vol. 3, No. 5, 801 – 806.

[Kasabov *et al*, 1997] Kasabov, N., Kim, J., Watts, M. and Gray, A. (1997) FuNN/2–A Fuzzy Neural Network Architecture for Adaptive Learning and Knowledge Acquisition. To appear in Information Science Application.

[Mang *et al*, 1995] Mang, G., Lan, H. and Zhang, L. (1995) A Genetic-based method of Generating Fuzzy Rules and Membership Functions by Learning from Examples. Proceedings of ICONIP'95, China, Vol. 1, 335 – 338.

[Setiono, 1997] Setiono, R. (1997) Extracting Rules from Neural Networks by Pruning and Hidden-unit Splitting. Neural Computation 9, No. 1, 321 – 341.