Linguistic neurocomputing

*Giovanni Bortolan and **Witold Pedrycz
* LADSEB - CNR
Corso Stati Uniti, 4, 35020 Padova, Italy

**Computational Intelligence Laboratory
Department of Electrical and Computer Engineering, University of Manitoba,
Winnipeg, Canada R3T 2N2

Abstract *A process of information granularization takes care of an enormous flood of numerical details that becomes summarized and hidden (encapsulated in the form of fuzzy sets) at the time of the design of a neural network. This substantially reduces the amount of training as the designed network needs to deal with a substantially reduced and highly compressed number of data that falls far below the size of the original training set. The same granularization mechanism delivers some highly advantageus regularization properties. The necessary effect of information granularization is accomplished in the framework of fuzzy sets, especially via context - sensitive (conditional) fuzzy clustering. Subsequently, the resulting neural network becomes an architecture with non-numeric connections. A thorough analysis of results of computing carried out in the setting of linguistic neurocomputing is also given.*

1. Introductory remarks

The issue of an efficient learning has been a focal point of a vast number of research endeavors in the area of neurocomputing. A toolbox of currently available training models is highly impressive. The main research agendas include various important tasks such as an efficiency of learning and generalization abilities of the designed networks. The always growing dimensionality of the problems tackled by neural networks along with an inevitable increase of the size of the training data accompanying these tasks make the learning more challenging. The technology of fuzzy sets has already contributed to the development, learning, and interpretation of neural networks. What becomes also a common denominator of all these approaches and enhancements is a strikingly dominant observation about the role of fuzzy sets in neurocomputing. By and large, fuzzy sets tend to look at data from a new and somewhat general standpoint; we would like to exploit these new characteristics of data in an attempt to increase robustness and augment learning capabilities. One should become aware that the ensuing optimization occurs at the numeric level (no matter in which way fuzzy sets enhance the original formulation of the problem).

The underlying objective of this study is to make the contribution of fuzzy sets more visible and radical in comparison to what has been already exercised in the literature. We depart from a purely numeric oriented style of the design of neural networks and revisit this very development problem at the non-numeric level. Here non-numeric information granules (especially fuzzy sets) are formed as a result of some sound summarization of the original numerical elements of the training set and made these granules directly available to neural networks for design purposes in lieu of the original numeric data set.

In what follows, we consider multivariable problems where neural networks have to deal with mappings from n - dimensional real space to the output being a subset of reals. In what follows, we denote the training set as $\{(\mathbf{x}(k), y(k))\}$, k=1, 2, ..., N, where $\mathbf{x}(k) \in \mathbf{R}^n$ and $y(k) \in \mathbf{R}$. The paper exploits a standard notation of fuzzy sets; for some brief introduction to the subject the reader can refer to one of the currently available texts (Klir and Folger, 1988; Pedrycz, 1995). The key conceptual notion used throughout the study concerns information granularity and granularity of fuzzy sets, in particular (Zadeh, 1979; 1994). The quantification of this notion could be done in many ways (including some associated notions such as set specificity).

2. Modes of utilization of training data in neural networks

As far as different modes of utilizing training data in the learning neural networks are concerned, we can distinguish between three main groups as portrayed in Fig. 1.

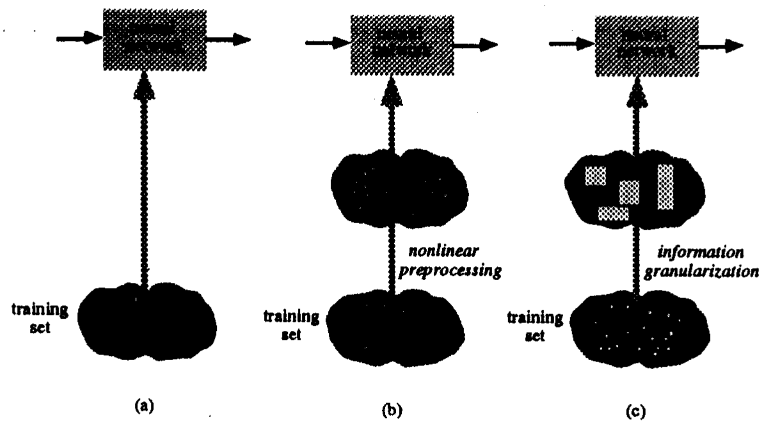


Fig. 1. Modes of utilization of training data in neurocomputing: direct use (a), nonlinear preprocessing (b), information granularization (c)

1. The direct use of raw data forming the training set, Fig.1(a). This is one of the most frequent approaches one can encounter in the area of neurocomputing. Essentially, all available data are fed into the network that, through its supervised learning, develops a relationship (mapping) between the inputs (input variables) and the output that is consistent with the training data (in other words, the mapping approximates the data set). The only very limited preprocessing of the data that may eventually take place arises in a form of a straightforward linear normalization of the input variables. There is an abundance of literature on this way of utilization of the training data (Zurada, 1992; Hassoun,1995).

2. The use of non linearly preprocessed training data, Fig. 1(b). The crux of this methodological avenue pursued in neural networks is to carry out some preprocessing of the training data with an intent of enhancing the learning abilities of the original neural network and accelerating its pace of learning. Radial Basis Function (RBF) neural networks are an excellent example of such structure equipped with preprocessing activities. What happens there is a process of a well-defined nonlinear normalization and a suitable transformation of the input data realized in order to simplify both the architecture of the network and make its learning more efficient. The granularity and distribution of RBFs are the two key components predetermining a success (or failure) of the RBF neural network. In fact, the existing literature (Zurada, 1992; Hassoun, 1995) reports on faster learning of the RBF neural networks and underlines some features of the learning such as an ability to avoid local minima.

3. The use of summarized (fused) learning data, Fig. 1(c). This approach is radically different from the two already discussed and as such departs from the concepts of learning that is completed at the numeric level. Essentially, the training of the neural networks is completed in presence of non numeric information. Thus the admitted elements of the training set could be fuzzy sets, sets, numeric intervals, or alike. The highly reduced training set contributes a lot to the faster training and results in a more compact neural network. One point to be made clear concerns the use of this type of the designed networks. In general, such networks are not compatible with the constructs developed throughout the first two types of learning. They will not compete on a basis of numeric accuracy. In fact, it would be absolutely unfair to expect that the network being trained on data of lower granularity will perform equally well on numeric data.

These three groups of learning become somewhat complementary as far as the underlying information granularity and the ensuing computational effort are concerned. The first two are the highest in terms of the computational effort required and the highest (numeric) information granularity associated with that. For the third mode, the computational effort becomes reduced while, simultaneously, the granularity concerned stays lower. The original numeric data are prudently summarized thus giving rise to non numeric elements (fuzzy sets). It is important to note that the cardinality of this new data set is far lower than the original one including only numeric pairs.

3. Context-sensitive fuzzy clustering

In what follows, we concentrate on the contextual Fuzzy C - Means (FCM) (Pedrycz, 1996). The method was also discussed in the setting of knowledge discovery and data mining (Pedrycz, 1997). The conditional aspect (context sensitivity) of the clustering mechanism is injected into the algorithm by taking into consideration an auxiliary context (conditioning) variable defined in Y. For this context variable, we define a fuzzy set of context

$$A: Y \rightarrow [0,1]$$

The original data x_k are augmented by this auxiliary variable, say y_k, producing a new vector denoted now as $[x_k\ y_k]$. We also assume that the values of context for the given data (patterns) are available and equal to $f_1, f_2, ..., f_N$. The value $f_k = A(y_k)$ connotes a level of involvement of the k-th data element in the assumed context. The way in which f_k can be associated with or allocated among the computed membership values of x_k, say $u_{1k}, u_{2k}, ..., u_{ck}$, is not unique. Here, we admit f_k to be distributed additively across the entries of the k-th column of the partition matrix meaning that

$$\sum_{i=1}^{c} u_{ik} = f_k \tag{1}$$

k=1, 2, ..., N. Bearing this in mind, let us modify the requirements to be met by the original partition matrices and define the new family of matrices

$$\mathcal{U}(\mathbf{f}) = \{\ u_{ik} \in [0, 1] \mid \sum_{i=1}^{c} u_{ik} = f_k\ \forall_k \text{ and } 0 < \sum_{k=1}^{N} u_{ik} < N\ \forall_i \}$$

Note that the standard normalization condition standing in (1') is replaced by the involvement (conditioning) constraint. The optimization problem is now reformulated accordingly (Pedrycz, 1996)

$$\min_{U,\, v_1, v_2, ..., v_c} Q \quad \text{subject to} \quad U \in U(f) \tag{2}$$

Again the minimization of the objective function is carried out iteratively where

$$u_{ik} = \frac{f_k}{\sum_{j=1}^{c} \left(\frac{\|x_k - v_i\|}{\|x_k - v_j\|} \right)^2}$$

i=1, 2, ..., c, k=1, 2, ..., N. We arrive at the above formula by transforming (4) to a standard unconstrained optimization by making use of Lagrange multipliers and determining a critical point of the resulting function. The computations of the prototypes are the same as for the original FCM method. Moreover, the convergence conditions for the method are the same as thoroughly discussed for the original FCM algorithm (Bezdek, 1981).

4. Fuzzy neural networks

Our main thrust in the design of a fuzzy neural network is to support processing of linguistic (non numeric) information. Bearing this in mind, we admit the individual inputs and outputs to be fuzzy sets rather than numeric quantities, Fig. 1. More specifically, we confine ourselves to the class of fuzzy sets with trapezoidal membership functions. This choice is fully justified considering the generality of such constructs regarded as reasonable models of uncertain quantities along with the ensuing straightforward computations. In particular, trapezoidal or triangular fuzzy sets have already been recognized as versatile models in the technology of fuzzy sets.

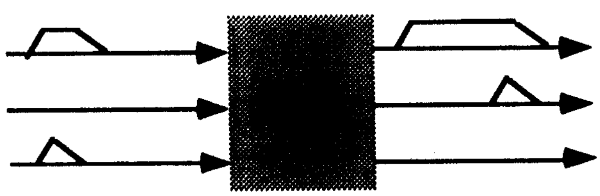


Fig. 1. Neural network with its inputs and outputs regarded as trapezoidal and triangular fuzzy sets

In the discussed model of the network, we use a slightly different description of trapezoidal fuzzy sets. Instead of the commonly utilized notation, T(x; a, b, c, d), we introduce the notation T" (x; b, c, a, b) where a and b are two explicitly articulated spreads of the fuzzy set, Fig. 2.

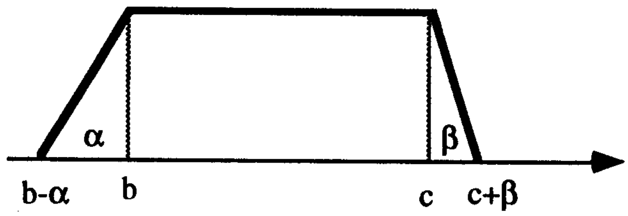


Fig.2. Fuzzy set T" (x; b, c, a, b) - a basic notation

As in the previous notation "b" and "c" describe the core part of the fuzzy set. By a fuzzy neuron, we mean a processing unit with fuzzy inputs and fuzzy outputs, all being modeled as trapezoidal fuzzy sets. The weights that characterize the connections between the nodes are represented by four coordinates (b_W, c_W, a_W, b_W), one for every defining point of the trapezoidal fuzzy sets (to simplify notation, we have dropped the indexes indicating the individual nodes of the network).

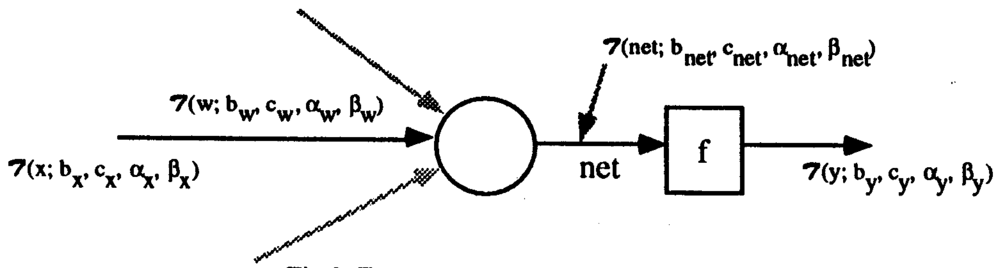


Fig.3. Fuzzy neuron - notational details

The output of the neuron can be approximated as a trapezoidal fuzzy set; we enumerate two phases of this construct:

- first a weighted sum is formed as T" (net; b_{net}, c_{net}, a_{net}, b_{net}) where

$$b_{net} = \sum_{i}^{n} b_w(i) b_x(i) \qquad c_{net} = \sum_{i}^{n} c_w(i) c_x(i)$$

$$\alpha_{net} = \sum_{i}^{n} \alpha_w(i) \alpha_x(i) \qquad \beta_{net} = \sum_{i}^{n} \beta_w(i) \beta_x(i)$$

and "n" denotes the number of the inputs; the index (i) pertains to the number of the connection and the associated input. This trapezoidal fuzzy set should satisfy the obvious constraints

$b_{net} \leq c_{net}$,

$a_{net} \geq 0$, $b_{net} \geq 0$

Assuming that the inputs are given, the constraints translate into a number of admissible strategies regarding the changes of the connections of the network. In Bortolan (1997) discussed were a number of strategies governing the changes of the connections that retain this consistency condition. Here we proceed with a two-stage mode starting from all coordinates being equal (that is, $b_W = c_W = a_W = b_W$) and allowing them to become different afterwards (that occurs over the progress of the training).

-secondly, the trapezoidal fuzzy set of the weighted sum becomes transformed nonlinearly via an activation function (f), Fig.3. To retain a homogeneous computing environment, we approximate the result as a trapezoidal fuzzy set T" (y; b_y, c_y, a_y, b_y) with the parameters computed as

$T''(y; b_y, c_y, a_y, b_y) =$
$= (y; f(b_{net}), f(c_{net}), f(b_{net}) - f(b_{net} - a_{net}), f(c_{net} + b_{net}) - f(c_{net}))$

The neural architecture to be used is feedforward and formed by the neurons with the fuzzy connections; we will be referring to this topology as a fuzzy neural network.

5. The design of the fuzzy neural network

The contexts and the resulting prototypes in the input space are directly used towards the construction of the fuzzy neural networks. Note that the contexts are already non numeric quantities. We would like to summarize the input data in the same way. The simplest option to be pursued deals with trapezoidal (triangular) fuzzy sets formed around the numeric prototypes of the clusters. As, in fact, all membership values are available through the partition matrices, it is quite straightforward to approximate them as trapezoidal membership functions. The idea of this approximation is illustrated in Fig.14. We construct the segments of the membership function through the experimental membership values by optimizing the knots of the trapezoidal function. The optimization is completed separately for the knots situated at the left and right hand side of the prototype, Fig.4.

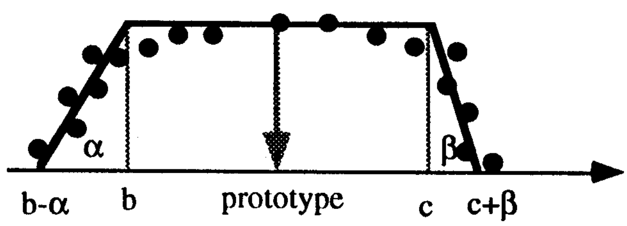


Fig. 4. Approximation of the membership functions with the use of experimental data - a, b, c, b are the parameters to be determined

All of these membership functions form the training family of input - output fuzzy sets. Remarkable is the reduction of its size in comparison to the original training set composed of numerical values. The obtained size of the new training set is equal to the product of the number of the clusters for the individual contexts, namely $c_1c_2...c_p$. Considering that the number of contexts is usually limited to around 5 and assuming the same number of clusters (c) for each context, we get c^5 elements in the new condensed training set.

There are some implementation issues worth highlighting:

- the choice of the fuzzy sets of contexts. These fuzzy sets are the starting points of the entire summarization process and, as such, should be prudently selected to become meaningful in the learning process. Observe several obvious facts:

(i) if $A \subset A'$ (that is $A(y) < A'(y)$) for all y in **Y**, then the cardinality of the elements to be clustered under the first context is lower than in the second scenario. In limit, if $A(y) = 1$ for all y in **Y**, we end up with a context - free clustering. If, however, A gets very specific, that is $A(y) = 1$ for $y \in [y_0, y_0+c]$ with c being a small positive constant, then the process of clustering could be meaningless as tagging only a very few data points to be summarized.

(ii) for a two-class classification problem, where $A(y) \in \{0, 1\}$, this particular context implies the clustering of data belonging to the same category. The clustering can be regarded as a prerequisite to the design of two-valued neural classifiers.

-as we have stressed, the choice of A_is should reflect the needs of the neural network. If we would like to focus on some subregions of the output variable, there should be some fuzzy sets of context capturing this intent. They could be eventually of higher granularity in comparison to some other less interesting regions of the output variable. In any case, we should be aware that defining a vacuous context (that is such for which there are no or very few data in the training set) defeats the purpose of having this context at all. Evidently, the neural network will not be able to do any training as there is no input for this particular context. Thus it is worthwhile to compute the sum

$$\sum_{l=1}^{N} A(y_l)$$

that reflects how much the given context is supported (activated) by the current training set. To make the context A meaningful we should require that this sum becomes different from zero or exceeds a certain minimal threshold value,

- the choice of the number of clusters c(1), c(2), ..., c(p) is a general problem coming with any clustering method. The use of some cluster validity indexes (Bezdek, 1981) can offer some help. Again, the number of clusters could vary depending upon the size of the information granule of the context triggering this clustering.

The produced linguistic data are directly used to train the fuzzy neural network. Even though the network looks simple, it is inherently nonlinear. The nonlinearity resides within the receptive fields themselves as well as the summation effect completed at the level of the fuzzy sets of the connections is nonlinear.

6. Numerical studies

The experimental study deals with a real-world problem of Boston housing data. The data set is available from the machine learning site at the University of California at Irvine (http://www.ics.uci.edu/~mlearn/MLRepository.html). The data set deals with housing conditions encountered at several suburban areas of Boston. There are 13 input variables; these include factors such as crime rate (x_1), nitric oxides concentration (x_5), student - teacher ratio (x_{11}), price of real estate (x_{13}). For more details consult the description of the dataset at the above site. The context we are interested in are the prices of houses that fall under three naturally acceptable categories, all in $1,000:

- *low* real estate price described by the T(y, 0, 5, 15, 20)
- *middle* range real estate price characterized by T(y, 15, 20, 25, 30)
- *high* range of real estate prices defined by T(y,25, 30, 50, 55)

These naturally reflect our perception of the linguistic categories (contexts) in this particular real estate problem. More importantly, by selecting some other linguistic terms, we can model a certain requirements about the problem or a way in which we would like to explore the data and focus the ensuing training of the neural networks. For instance, one would be interested in a particular price range that should be emphasized and attempt to make it more restricted, say *around* 15K $, *around* 45K $, etc. This will call for some specific contexts introduced into the model. Some other eventual modeling request would be to focus the training of the network on the broad range of low prices and eventually make the range of high prices relatively narrow. All of these settings of the linguistic contexts underline a proactive role of the network's designer in the utilization of the data and the customization of the neural network.

Discussing the already defined three context (*low*, *medium*, *high* prices), the conditional clustering is performed with four clusters allocated per class. The distance function used in the clustering is computed using normalized inputs. After the linguistic summarization at the input level, we end up with 12 elements of the training set to be used by the fuzzy neural network - a substantial 50-fold reduction in comparison with the original numerical data set. It would be a nuisance to report on the entire linguistic data set; here we restrict ourselves to the four arbitrarily selected inputs (input variables) such as

x_1 - crime rate
x_5 - nitric oxides concentration (parts per million)
x_8 - weighted distances to five Boston employment centers (in miles)
x_{11} - student - teacher ratio

We feel that these variables could be of particular interest as meaningful factors distinguishing between the formulated categories. The prototypes of the clusters are summarized in the tabular form below

crime rate	*low* price	*middle* price	*high* price
$cluster_1$	11.685	0.558	0.987
$cluster_2$	1.7434	0.183	0.727
$cluster_3$	18.466	5.413	0.990
$cluster_4$	10.185	0.333	0.123

nitric oxides conc.	*low* price	*middle* price	*high* price
$cluster_1$	0.687	0.542	0.526

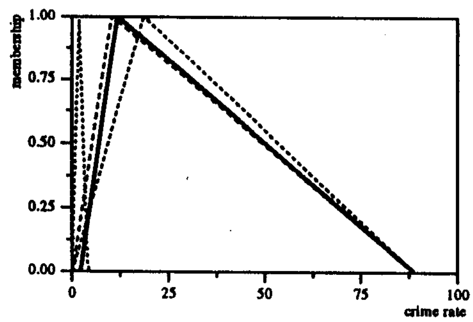
$cluster_2$	0.556	0.440	0.502
$cluster_3$	0.689	0.641	0.527
$cluster_4$	0.693	0.493	0.421

weighted distance	*low* price	*middle* price	*high* price
$cluster_1$	2.089	3.288	3.411
$cluster_2$	3.908	6.720	3.823
$cluster_3$	1.672	2.765	3.406
$cluster_4$	1.908	4.505	6.547

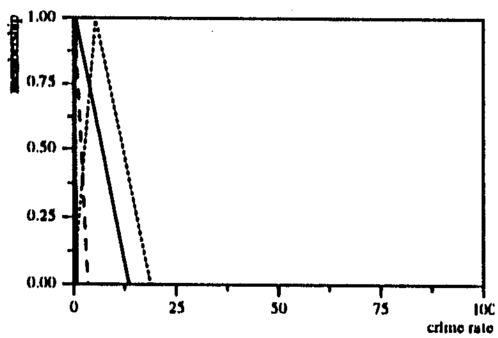
student -teacher ratio	*low* price	*middle* price	*high* price
$cluster_1$	20.04	18.55	16.60
$cluster_2$	20.00	17.57	16.98
$cluster_3$	19.81	19.88	16.59
$cluster_4$	20.00	18.67	15.82

The results for this reduced subset of inputs are also given in the form of the trapezoidal membership functions, Fig. 5 (*low* price- class#1, *middle* price - class#2, *high* price - class#3). In fact, these fuzzy sets are far more illuminating than the single numerical values of the prototypes themselves.

class #1



class #2



class#3

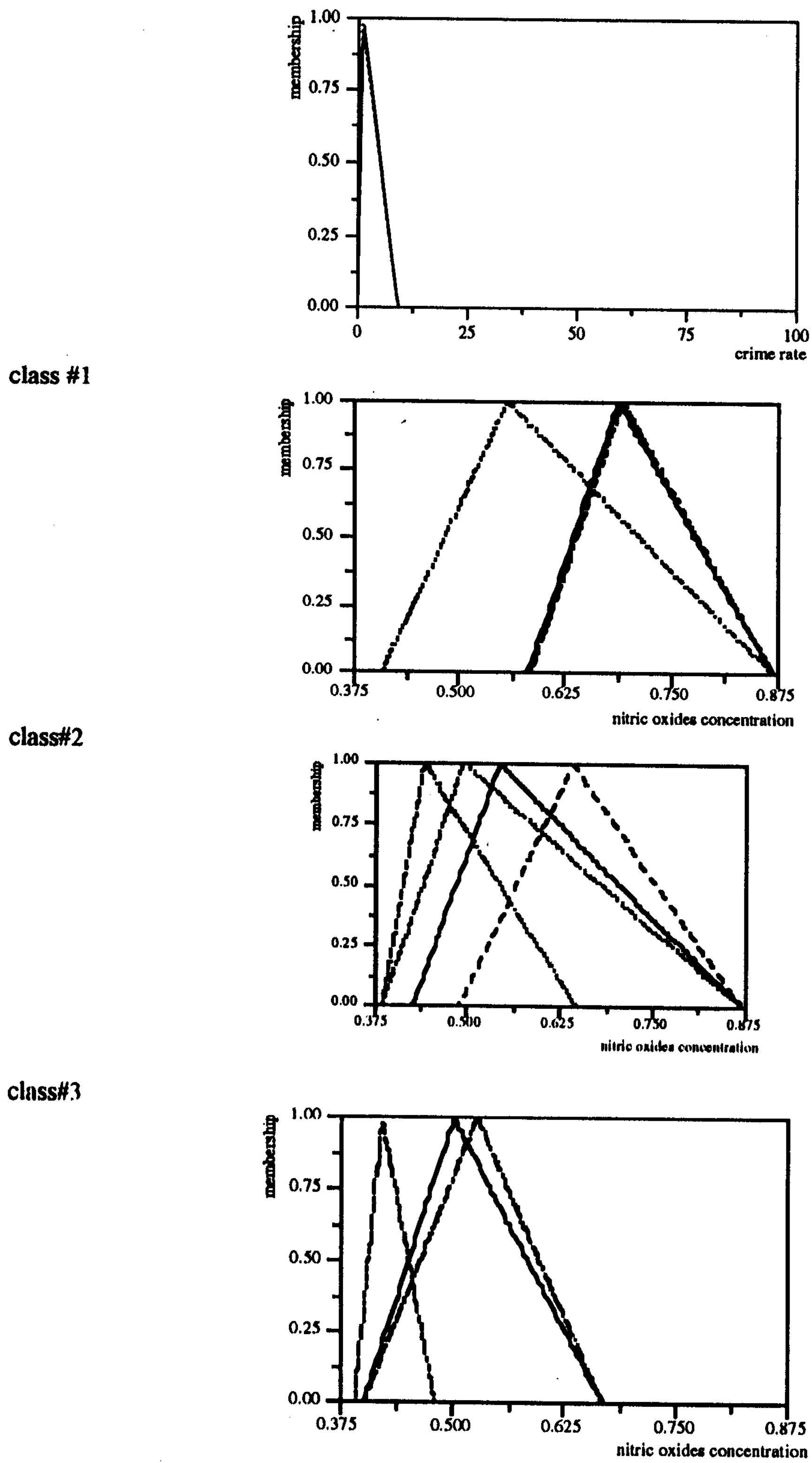


Fig. 5. Linguistic representation of prototypes of selected input variables

The findings one can easily summarize from these fuzzy sets are intuitively appealing. For the class of real estate staying in the range of the lowest prices, we get a relatively high crime rate. Note that even

though the modal values of the crime rate are not that high, the long tails of the membership functions stretch to much higher values when compared with some other categories. The crime rates for the two remaining classes are lower and the corresponding membership functions are quite narrow and concentrated around the modal values. The environmental conditions (expressed via a nitric oxide concentration) also worsen for the first class. For the third class of real estate, we get fuzzy sets of far lower spread than those occurring in the second category. The dwellings of the first category are close to the employment centers. The distribution of the distances for the second category is more spread while all the clusters in the third category of the most expensive houses are located at a similar distance from the employment centers yet this distance is almost two times higher as in the first class. This explicitly reflects the preferences of this community underlying that the people do not prefer living too close to their workplace. Finally, the student - teacher ratio happens to be an important indicator of the standard of living. For the third category this ratio is far lower than in the first category. The second category locates somewhere in-between with several clusters characterized by different values of this ratio.
The same data set was processed using a standard feedforward neural network with a single hidden layer with h=13 nodes in this layer; each neuron was equipped with a standard sigmoidal element. The training took 100,000 learning epochs. The comparison of the learning effort expressed in terms of the training epochs is misleading, though, because of the different sizes of the training sets involved. Instead, it is far more legitimate to compare the CPU time used for the training: on a VAX 4000/600 it took 50'12" to train the fuzzy neural network versus $6^{h}42' 42"$ to train the network with the original data set. Thus the achieved speedup is at the range of 8.02 times.

7. Conclusions

We have proposed a new architecture of fuzzy neural networks, delivered its complete learning scheme and offered a new type of linguistic computing in the setting of neural networks. The study supports the commonly (yet not vigorously experimentally justified) claim that fuzzy sets do indeed contribute to the reduction of otherwise immense computational effort. The study shows that the training of fuzzy neural networks with summarized data rather than working with the individual elements of the input - output data cuts the required training time. Obviously, the granularity of the processed information is lower than the one originally residing within the training data. This, in turn, implies that the results of such linguistic processing are genuine fuzzy sets. There is no much sense analyzing the performance of the network at the purely numeric level as the structure has never been geared towards this level of processing. On the other hand, the processing at the linguistic level makes the network transparent and the patterns describing the linguistic data become readily available. Further impact of this methodology on data mining and interpretation of neural networks is inevitable and highly promising.

8. References

1. J. C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, N. York, 1981.
2. G. Bortolan, Inference with fuzzy logic, in: P. Ciarlini, M. G. Cox, R. Monaco, F. Pavese, Eds., Proc. of the Workshop Advanced Mathematical Tools in Metrology World Scientific, Singapore , 1994, pp. 47-56.
3. G. Bortolan, Neural networks for the processing of fuzzy sets, in: M. Marinaro, P. G.Morasso, Eds., Proc. of the International Conference on Artificial Neural Networks, Springer-Verlag, London , 1994, pp. 181-184.
4. G. Bortolan, An architecture of fuzzy neural networks for linguistic processing, Fuzzy Sets and Systems, to appear (1977).
5. G.E.Box, G. M.Jenkins, Time Series Analysis: Forecasting and Control, Holden Day, San Francisco, 1970.
6. D. Dubois, H. Prade, Possibility Theory - An Approach to Computerized Processing of Uncertainty, Plenum Press, New York, 1988.
7. J.Jang, C. Sun, Functional equivalence between radial basis function networks and fuzzy inference systems, IEEE Trans. on Neural Networks, 4, 1993, 156 - 158.
8. M. H. Hassoun, Fundamentals of Artificial Neural Networks, MIT Press, Cambridge, Mass, 1995.
9. G. J. Klir, T. A. Folger, Fuzzy Sets, Uncertainty, and Information, Prentice Hall, Englewood Cliffs, New Jersey, 1988.
10. W. Pedrycz, Fuzzy Sets Engineering, CRC Press, Boca Raton, 1995.
11. W. Pedrycz, Conditional Fuzzy C - Means, Pattern Recognition Letters, 17, 1996, 625 - 632.
12. W. Pedrycz, Fuzzy set technology in data mining and knowledge discovery, Fuzzy Sets and Systems, to appear.
13. T. Poggio, F. Girosi, Networks for approximation and learning, Proc. of the IEEE, 9, 1990, 1481 - 1497.
14. L. A. Zadeh, Fuzzy sets and information granularity, In: Advances in Fuzzy Set Theory and Applications, (M. M. Gupta, R.K. Ragade, R. R. Yager, eds.), North Holland, Amsterdam, 1979, pp. 3-18.
15. L. A. Zadeh, Fuzzy logic, neural networks, and soft computing, Communications of ACM , 37 , 3, 1994, 77-84.
16. J. Zurada, Introduction to Artificial Neural Systems, West Publishing Company, Minneapolis, 1992.