Application of a Feed-Forward Artificial Neural Network as a Mapping Device[§]

Robert Kocjančič*,[†] and Jure Zupan[‡]

Jozef Stefan Institute, Jamova 39, 1001 Ljubljana, Slovenia, and National Institute of Chemistry, Hajdrihova 19, 1001 Ljubljana, Slovenia

Received March 7, 1997[⊗]

The article presents the ability to use a feed-forward neural network as a mapping tool. The objects are fed to the artificial neural network with two neurons in a hidden layer, and the result is compared to the targets which are in our case equal to the inputs themselves. After training one can plot the objects' labels to the map whose coordinates are the output values of the two neurons of the hidden layer. The clustering ability of such an arrangement is compared to those of Kohonen network feature maps and Principal Components Analysis using a well-known set of data, namely the set of 572 Italian olive oils. The results of the presented method seems to be better than those of the PCA. The power of the new tool lies in the possibility of coping with nonlinearities and in its continuous nature.

INTRODUCTION

In analytical chemistry as well as in chemical engineering, pharmacology, and other intensive data gathering research fields we often acquire and have to investigate large data sets. It is very important, especially in the preliminary phase of investigation, to have a good overview of the data which are usually multivariate within measurement space dimension of up to several thousand variables (for instance, a set of IR spectra, chromatograms, a time series of a characteristic pH curve in a chemical process, or the composition and properties of a certain sample). Merely examining the numbers and comparing pairs of objects (i.e., samples) gives poor results and does not provide a good overview of our data set. Usually, the data mapping takes place as a first step in the investigation of a data set in order to group data objects according to their intrinsic similarity with intention to reveal the hidden general information about the distribution of the data objects in the measurement space.

Over the years many powerful techniques for data visualization were developed in the chemometric fields of pattern recognition, data reduction, and clustering. The supervised methods include SIMCA,[1] UNEQ,[2] ALLOC,[3] rule building expert systems,[4] and linear discriminant analysis.[5] On the other hand, the unsupervised methods include the principal components analysis,[6] a variety of cluster analysis methods, projection pursuit,[7] and Kohonen neural networks.[8]

Probably, the most widely used method in data visualization is principal components analysis (PCA). It has the ability to extract information from multidimensional data sets and to put this information into new transformed "coordinates" called principal components (PCs) which are the same in number as the original variables. PCs are linear combinations of the old coordinates. The coefficients telling how much of the original variables are contained at each PC are called "loadings". These PCs are sorted by decreasing information content (i.e., decreasing variance) so that most of the information is preserved in the first few PCs. Another useful aspect of PCA is that the PCs are orthogonal and thus independent of each other—which is so often not the case in data which objects are represented by multivariate data like spectra or time series. Using PCA one can plot the projections of objects on the calculated PC axes, forming one-, two-, or three-dimensional plots of objects in principal components space.

The second technique to be mentioned here is a fast growing method developed by Teuvo Kohonen in the early 1980s.[8] It is called self-organizing feature maps, and it deals with a two-dimensional array of "neurons". A neuron consists of a set of weights (same in number as the number of variables) which are adjusted in order for this neuron to become, for instance, as similar as possible to the input object.

By presenting the objects of a training data set in an unsupervised way to the Kohonen neural network (KNN) several hundred epochs and each time adjusting weights according to a simple rule, the neurons excited by the objects form a recognizable pattern on a two-dimensional plane. Quite obviously, one can present new objects (unknown samples etc.) to the taught KNN and a well trained network can classify them in a region with most similar neurons (objects). There are several other techniques applicable for such preliminary studies, like the standard[9] or three distance clustering.[10]

The method we are dealing with in this paper, the Feed Forward Neural Networks (FFNN), is mainly used for the modeling of multivariate data. FFNN is very appropriate for handling highly nonlinear problems, and it can deal with multivariate input and multivariate responses. It has been 15 years since Werbos[11] followed by Rummelhart, Hinton, and Williams[12] published the error backpropagation learning algorithm for the multilayered FFNN. Since then the number of its applications increased exponentially. It is hard to find a field of research or exploration where this technique has not been successfully tested. The FFNN technique basically involves an optimization problem of adjusting the weights in neurons in order to achieve the lowest error, i.e., the least difference between the network

* Corresponding author: Department for inorganic chemistry and technology, Jozef Stefan Institute, Jamova 39, SLO-1001 Ljubljana, Slovenia. Phone: (386) 61-1773-291. Fax: (386) 61-1232-125. E-mail: Robert.Kocjancic@ijs.si.

† Jozef Stefan Institute.

‡ National Institute of Chemistry.

§ Keywords: data reduction, data mapping, feed-forward neural networks.

⊗ Abstract published in *Advance ACS Abstracts,* November 1, 1997.

S0095-2338(97)00223-0 CCC: $14.00

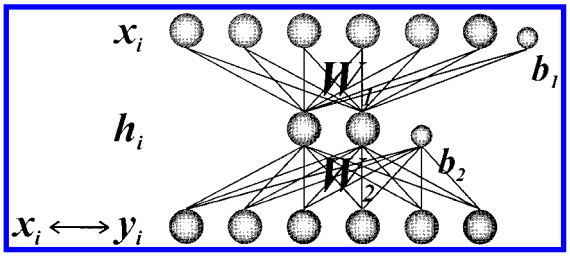

Figure 1. The topography of a FFNN for learning in an autoassociative manner. W_1 and W_2 are the weight matrices connecting input-to-hidden and hidden-to-output layers, respectively, and b_1 and b_2 are the corresponding bias vectors.

outputs and the desired targets (the scaled responses). We will not give details on the method's algorithm since it is well described elsewhere.[13] As we mentioned above, finding the appropriate weights is an optimization problem and as such other optimization methods apart from the backpropagation of error can be used. To mention only two of them, the genetic algorithms[14,15] or the Levenberg−Marquardt nonlinear parameters optimization technique[16,17] can be applied.

THE PROPOSED MAPPING METHOD

Let us first describe the notation to be used in this paper: matrices, vectors, and scalars are denoted as uppercase bold italic letters (e.g., $\boldsymbol{M}$), lowercase bold italic letters (e.g., $\boldsymbol{v}$) and lowercase italic letters (e.g., s), respectively.

The method of employing the FFNN as a mapping device involves a slightly different approach to the standard FFNN. We are trying to map m-dimensional objects in the study (patterns) into a space of lower dimensionality, for instance into a two-dimensional space (the map). Let us start with the data set presented as a matrix of p objects ($\boldsymbol{x}_i$) and m variables (size $p \times m$) and denote it $\boldsymbol{X}$:

$$\boldsymbol{X} = ||\boldsymbol{x}_i|| \quad 1 \le i \le p$$

$$\boldsymbol{x}_i = (x_{ij}) \quad 1 \le j \le m$$

Now we can construct a three layer (input, hidden, output) FFNN in an autoassociative manner. It has a small number k (usually two) of neurons in a hidden layer and an equal number of input and output nodes. An example with $m = 6$ and $k = 2$ is shown in Figure 1. We produce two matrices of weights, $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$, the first one connecting the input nodes to the hidden layer and the second one connecting the hidden layer to the output nodes. The dimensions of the weight matrices $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are $m \times k$ and $k \times m$, respectively. There is also an input node bias $\boldsymbol{b}_1$ (sized $1 \times k$) and a hidden node bias $\boldsymbol{b}_2$ (sized $1 \times m$). If $\boldsymbol{x}_i$ (which at the same time is treated as a target) is presented to the network, the output of the hidden neurons ($\boldsymbol{h}_i$) is defined by

$$\boldsymbol{h}_i = \mathrm{TF}(\boldsymbol{x}_i \cdot \boldsymbol{W}_1 + \boldsymbol{b}_1)$$

and the output of the network $\boldsymbol{y}_i$ is similarly defined by

$$\boldsymbol{y}_i = \mathrm{TF}(\boldsymbol{h}_i \cdot \boldsymbol{W}_2 + \boldsymbol{b}_2)$$

where TF stands for the transfer function, which can be logsigmoidal or some other nonlinear function.

The consequence of constructing such FFNN (having only two neurons in the hidden layer) is that m-dimensional representation of every object $\boldsymbol{x}_i$ presented to the network will be (in the hidden layer) transformed into two numbers (i.e., two outputs of two hidden neurons). Once the ANN is trained, these two numbers (the outputs of hidden layer h_1 and h_2) are a unique representation of the object in a two-dimensional space. We hope to train the network, i.e., adjust the weights so that similar objects will form distinctive groups in this two-dimensional map.

During the training the output, $\boldsymbol{y}_i$ is compared to the target (and input) vector $\boldsymbol{x}_i$. After learning of patterns from the data set is completed, we can map the objects onto an h_1 vs h_2 plot. Variables h_1 and h_2 are the two outputs of the hidden layer and at the same time can be treated as coordinates of the object that was fed to the network. As will be illustrated in the case study, the pattern recognition capabilities of such an arrangement are comparable to those of KNN with some features that make this method even more attractive.

Table 1. The Nine Olive Growing Regions of Italy with a Number of Samples Analyzed and the Division of Data into a Training and Test Set

olive growing regions (classes)[a]		no. of objects in training set	no. of objects in test set
1	North Apulia	15	10
2	Calabria	35	21
3	South Apulia	40	166
4	Sicily	20	16
5	Inland Sardinia	30	35
6	Coastal Sardinia	20	13
7	East Liguria	30	20
8	West Liguria	30	20
9	Umbria	30	21
	total	250	322

[a] The numbers depict class numbers in plots.

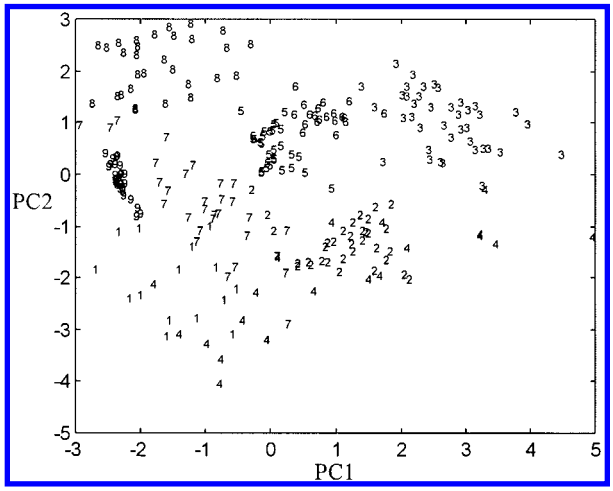

Figure 2. The PCA plot of scores on first two PCs.

THE CASE STUDY

In order to compare the proposed method to other visualization methods we have used the well-known data set of 572 Italian olive oils which has become a standard data set for testing various chemometric methods.[18] The data set was produced by Prof. M. Forina and his co-workers.[19] They determined the content of eight fatty acids in 572 olive oils from nine different olive growing regions in Italy. Table 1 lists these nine regions with the numbers of samples analyzed. The eight fatty acids analyzed were palmitic, palmitoleic, stearic, oleic, linoleic, eicosanoic, linolenic, and eicosenoic. It was shown[18] that by plotting oil representations transformed by PCA in the space of the first two PCs, the oils produced in some regions are separated from those produced in other regions and some overlap (Figure 2).

1				7				14	1
	9	9	9	7	7	7	7		
7	9	9		7				8	8
8	7	9		5	5	5		8	8
8				5	5	6	6		8
8	3	3	3			6	6		8
3	3	3	3	3			6		8
3	3	3	3	3	4	2	4	2	2
4	24	2	2	4	2	2	2	4	
1	4	2	2	2	2	2	2	4	14

Figure 3. The taught 10 × 10 Kohonen network showing samples' grouping. The numbers depict class numbers in Table 1. The size of the number is proportional to the number of objects that "fall" into that box. The neurons where regions overlap are shaded.

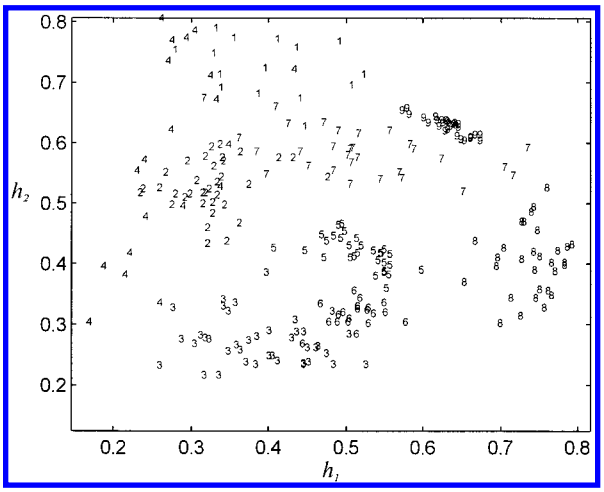

Figure 4. The mapping capability of a FFNN.

Comparably, the Kohonen feature mapping method, applied on the same data set, provides a better pattern recognition.[20] The result of such unsupervised learning is shown in Figure 3. It can be observed that the discrimination between olive growing regions is better than in the case of PCA and that only a few objects fall into "mixed" regions.

Now we can compare the results of the two methods above to those obtained by the proposed method using only two hidden neurons in the hidden layer of a FFNN trained by the error backpropagation algorithm. The map resulting from the training set of 250 objects is shown in Figure 4.

Most olive growing regions are well separated. One cluster (representing mainly Sicilian oils, labeled "4") exhibits slight mixing with other southern regions. This is the case for which other methods exhibit the same extent of mixing, and therefore it seems that these oil samples were not "purely Sicilian". Anyway, the preservation of topology is excellent and similar to that of Kohonen neural networks.[20]

However, the true power of the proposed method lies in the continuous nature of objects' coordinates comparing to discrete ones of Kohonen neural networks. It is possible to examine how a specific change in a certain variable (or more variables) moves any object over the map. In this way one can distinguish between influential and obsolete variables (and thus, for instance, reduce and/or optimize the required analytical procedures), or one can find the variables that make the difference between different regions.

A suggestion as to how to find influential variables is illustrated in Figure 5. For each region we can find an average object $\bar{\mathbf{x}}_i$

$$\bar{x}^i = \frac{1}{n_i}\sum_{j=1}^{n_i} x_{kj}^i \quad 1 \le i \le 9, 1 \le k \le 8$$

where i is the class (or region) and n_i is the number of objects in ith class. The variable of the object j belonging to class i is labeled as x_{kj}^i. Then, one can take one variable at a time and vary it from the minimal to the maximal value of its range. By propagating such objects through the first layer of a FFNN, one gains a pair of coordinates (h_1,h_2) for each object which determine the position of these objects in a map. We can plot the model set objects for this olive growing region together with the line produced by our "average" object in which one variable was actually varying. Particularly, for Figure 5 the South Apulia region was selected, and the concentration of oleic acid (variable number 4) was varied from 65% (the value 0.0 on the line) to 81% (the value 1.0 on the line).

In order to compare to the published example[18] only two regions are shown in Figure 6, namely North and South Apulia. The training set objects from these two regions are denoted by 1 and 3, respectively. Each plot shows sensitivity to change in one of the **variables** (the content of the eight fatty acids) for the two mentioned regions. The circle denotes the minimum value for that variable (as is the value 0.0 in Figure 5), while the line connects the minimum and

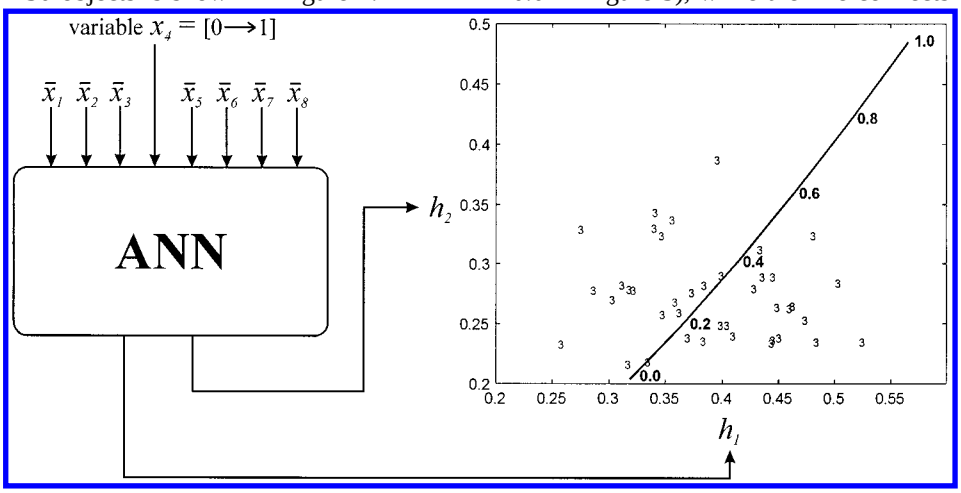

Figure 5. An example of a sensitivity study procedure. All variables are held at their average values except for the concentration of the oleic acid which is varied from its lower to its upper range value (65−81%). Since all the variables are scaled, the value is changed between 0.0 and 1.0. One can observe the movement of this object on a map.

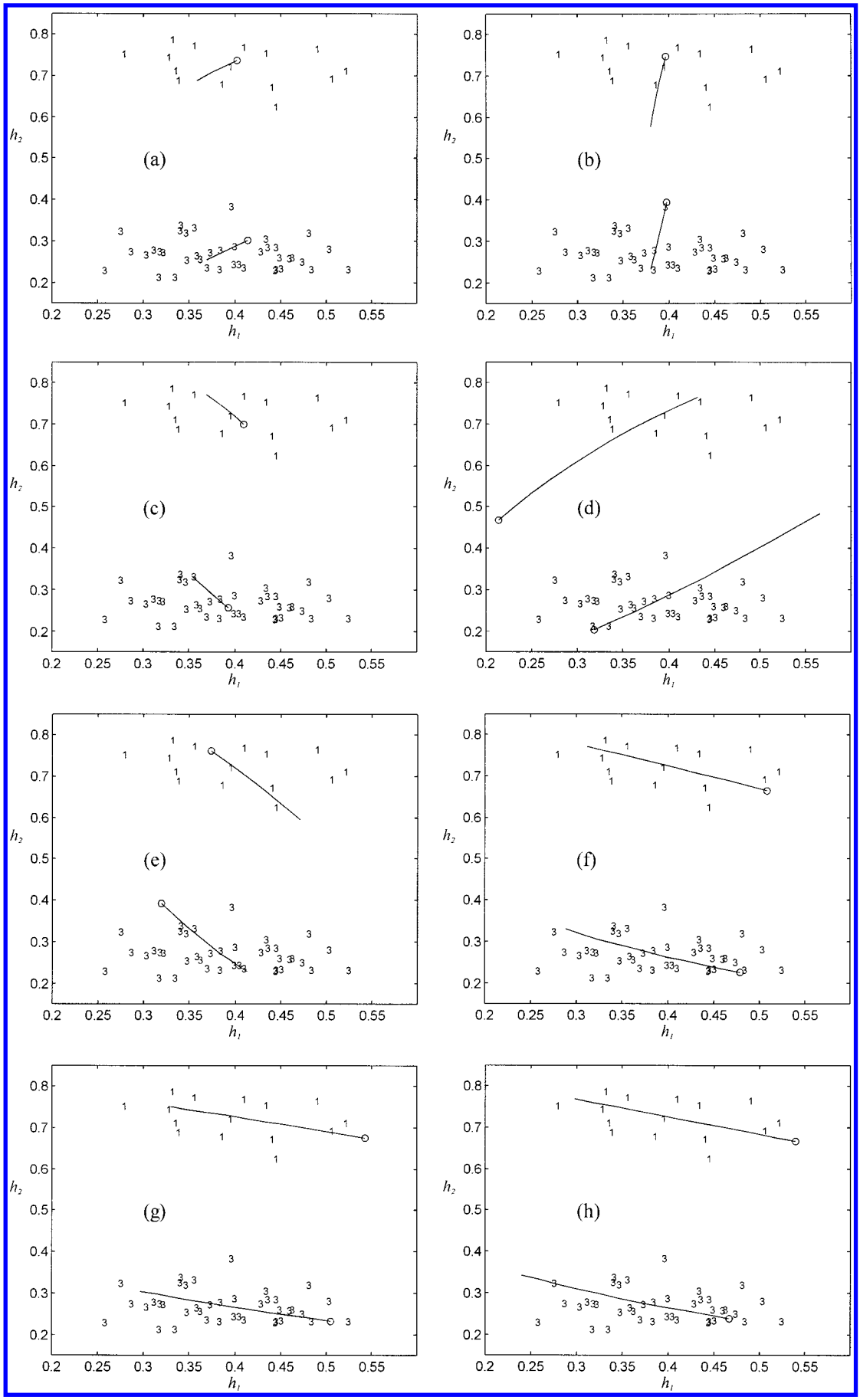

Figure 6. The sensitivity analysis for only two regions (1, North Apulia; 3, South Apulia) and all eight fatty acids (a, palmitic acid; b, palmitoleic acid; c, stearic acid; d, oleic acid; e, linoleic acid; f, eicosanoic acid; g, linolenic acid; h, eicosenoic acid). The lines were drawn as shown in Figure 5. In order to preserve space, two regions are drawn in one plot.

the maximum value. The min−max range was taken from all objects in the model data set. If we examine the plot (d) in Figure 6, we observe that high oleic acid concentration classifies the sample in the North Apulia region, whereas low oleic acid concentration classifies it in the South Apulia region. Many other variables, like stearic in plot (c) in Figure 6, do not exhibit such discriminating capabilities and can thus be considered noninfluential.

It is also possible to classify a new, unknown object. This can be done by letting the new object $\boldsymbol{x}_u$ through the neural net, acquiring the hidden node outputs $\boldsymbol{h}_u$, which can be compared to the centroids of all known classes. The centroids for nine classes are defined as

$$\boldsymbol{c}_i = \left(\frac{1}{n_{ij}}\sum_{j=1}^{n_i}\boldsymbol{h}_{1j}, \frac{1}{n_{ij}}\sum_{j=1}^{n_i}\boldsymbol{h}_{2j}\right) \quad 1 \le i \le 9$$

where $\boldsymbol{c}_i$ is the centroid of class i, n_i is the number of objects in class i and, h_{1j} and h_{2j} are the outputs of the hidden nodes 1 and 2, respectively, for the jth member of the class i. Having these nine centroids, one can calculate the distances from the unknown sample's $\boldsymbol{h}_u = (h_{u1},h_{u2})$ to the nine centroids using, e.g.,

$$d_i = \sqrt{\frac{1}{2}\sum_{j=1}^{2}(c_{ij} - h_{uj})^2} \quad 1 \le i \le 9$$

where d_i is the euclidean distance from the unknown sample's hidden node outputs ($\boldsymbol{h}_u$) to the ith centroid. We can classify the unknown sample into class l, if

$$d_l = \min(\boldsymbol{d}) \quad \boldsymbol{d} = (d_1,d_2,d_3,...,d_9)^T$$

CONCLUSION

It was shown that the simple two-layered FFNN can provide good mapping capabilities by a special autoassociative arrangement of the data and training requirements. Its continuous nonlinear nature and the freedom to choose between one-, two-, or more dimensional representations make this method suitable for the initial inspection of the data, the data clustering, the classification of unknown objects, the sensitivity studies, and above all for the selection of an arbitrary dimensional representation of complex data.

In the case study above, it was shown that the data clustering can be done by teaching the FFNN in an autoassociative manner, i.e., to reproduce the input object at the net's output. Usually, objects are mapped according to their similarity into a one- or two-dimensional map. The objects' coordinates are the outputs of two neurons in the hidden layer by presenting these objects to a neural net. It is possible to distinguish between different classes of objects, and the quality of the separation is comparable to those of PCA and KNN.

When used as a classification tool, the FFNN can map new, unknown objects on a previously established map. The groups in the map are labeled according to the labels of the data in the training set. By doing so, the new object acquires its coordinates in the map, and it can be declared as a member of one of the existing groups based on some criterion, like distance from groups' centroids as shown in the case study above.

The potential of the usage in the sensitivity studies seems promising. As shown in the case study, one can examine variable after variable (and also combinations of variables) the impact that a change in a variable's value has on the object's position in the map. It is easy to distinguish between "important" and "nonimportant" variables. This is often crucial in a chemical laboratory where an analysis of a certain sample comprises of different analytical techniques, some simple, some laborious, some long lasting, etc. When obtaining clear evidence that some measurement is not informative enough, it can simply be discarded resulting in a faster and more economical analysis of a sample.

ACKNOWLEDGMENT

The authors would like to thank the Slovene Ministry of Science and Technology for funding.

REFERENCES AND NOTES

(1) Wold, S.; Sjöström, M. SIMCA, a method for analyzing chemical data in terms of similarity. In *Chemometrics: Theory and Application*; Kowalski, B. R., Ed.; American Chemical Society: Washington, DC, 1977; pp 243−281.

(2) Derde, M. P.; Massart, D. L. UNEQ: a disjoint modeling technique for pattern recognition based on normal distribution. *Anal. Chim. Acta* **1986**, *184*, 33−51.

(3) Coomans, D.; Massart, D. L.; Broeckaert, I.; Tassin, A. Potential methods in pattern recognition. Part 1. Classification aspects of the supervised method ALLOC. *Anal. Chim.* Acta **1981**, *133*, 215−224.

(4) Derde, M. P.; Buydens, L.; Massart, D. L.; Hopke, P. K. Comparison of rule-building expert systems with pattern recognition for the classification of analytical data. *Anal. Chem.* **1987**, *59*, 1868−1871.

(5) Massart, D. L.; Kaufman, L. *The interpretation of analytical chemical data by the use of cluster analysis*; Wiley: New York, 1983.

(6) Wold, S.; Esbensen, K.; Geladi, P. Principal Components Analysis. *Chemom. Intell. Lab. Syst.* **1987**, *2*, 37−52.

(7) Glover, D. M.; Hopke, P. K. Exploration of multivariate chemical data by projection pursuit. *Chemom. Intell. Lab. Syst.* **1992**, *16*, 45−59.

(8) Kohonen, T. Self Organizing Maps; Springer Verlag: Berlin, 1995.

(9) Massart, D. L.; Kaufman, L. *The Interpretation of Analytical Chemical Data by the Use of Cluster Analysis*; Wiley: New York, 1983.

(10) Zupan, J. *Clustering of large data sets*; Wiley: Chichester, 1982.

(11) Werbos, P. Applications of Advances in Nonlinear Sensitivity Analysis. In Drenick, R., Kozin, F., Eds.; *System Modeling and Optimization: Proceedings of the International Federation for Information Processes*; Springer Verlag: New York, 1982; pp 762−770.

(12) Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. Learning Internal Representations by Error Propagation. In Rumelhart, D. E., McClelland, J., Eds.; *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*; MIT Press: Cambridge, 1986; Vol. 1, pp 318−362.

(13) Zupan, J.; Gasteiger, J. *Neural Networks for Chemists, An Introduction*; VCH: Weinheim, 1993.

(14) Goldberg, D. E. *Genetic algorithms in search, optimization, and machine learning*; Addison-Wesley Publishing Company, Inc.: U.S.A., 1989.

(15) Wienholt, W. Minimizing the system error in feedforward neural networks with evolutionary strategy. In *ICANN94, Proceedings of the International Conference on Artificial Neural Networks*; Springer Verlag: Berlin, 1994; pp 490−493.

(16) Levenberg, K. A Method for the Solution of Certain Nonlinear Problems in Least Squares. *Quart. Appl. Math.* **1944**, 164−168.

(17) Marquardt, D. W. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *J. Soc. Indust. Appl. Math.* **1963**, *11*, 431−441.

(18) Hopke, P. K.; Massart, D. L. Reference Data sets for Chemometrical Methods Testing. *Chemom. Intell. Lab. Syst.* **1993**, *19*, 35−41.

(19) Forina, M.; Tiscornia, E. Pattern Recognition Methods in the Prediction of Italian Olive Oil Origin by their Fatty Acid Content. *Ann. Chim. (Rome)* **1982**, *72*, 143−155.

(20) Zupan, J.; Novič, M.; Li, X.; Gasteiger, J. Classification of Multicomponent Analytical Data of Olive Oils Using Different Neural Networks. *Anal. Chim. Acta* **1994**, *292*, 219−234.

CI970223H