Tuning Neural and Fuzzy-Neural Networks for Toxicity Modeling

P. Mazzatorta,*,† E. Benfenati,† C.-D. Neagu,‡ and G. Gini‡

Istituto Mario Negri, via Eritrea 62, Milan, Italy, and DEI, Politecnico di Milano, piazza L. da Vinci 32, Milan, Italy

Received August 19, 2002

The need for general reliable models for predicting toxicity has led to the use of artificial intelligence. We applied neural and fuzzy-neural networks with the QSAR approach. We underline how the networks have to be tuned on the data sets generally involved in modeling toxicity. This study was conducted on 562 organic compounds in order to establish models for predictive the acute toxicity in fish.

1. INTRODUCTION

More than 23 million chemical compounds are registered in the Chemical Abstract Service (CAS). In view of their abundance and wide use in all spheres of production we need a better understanding of their ecotoxicological impact on plant life, wildlife, and the environment in general. Apart from the ethical considerations associated with the use of animals, experimental determination of ecotoxicity and toxicity would require huge financial resources and much time to be done methodically on all the compounds of interest. Thus, new alternatives are needed.

The quantitative structure−activity relationship (QSAR) approach is based on the assumption that the structure of a molecule must contain the features responsible for its physical, chemical, and biological properties and on the possibility of representing a molecule by numerical descriptors.

The difficulty of predicting toxicity is due to the lack of knowledge of the toxic mechanisms for reactive chemicals and the complexity and heterogeneity of the data available. More powerful computational approaches have now opened new avenues to QSAR studies and several research papers have been published,[1−9] discussing the role that artificial intelligence (AI) could play in toxicity prediction and QSAR modeling. Despite their power these approaches have been criticized because of the number of parameters that need to be carefully tuned on the problem.

The paper is organized as follows. Section 2 presents the data preparation, the chemical descriptors, and describes the structure based on Fuzzy Neural Networks and Multi-Layer Perceptrons[10] of the hybrid intelligent system NIKE.[11] Section 3 shows the correlation between parameters and performances for different models developed using the same data set. A preliminary study was done on the optimum number of hidden neurons for the proposed neural structures. After fixing this parameter, we set out to interpret the influence of the fuzzification of data. Conclusions are summarized in the last section.

* Corresponding author phone: 39 023 901 4499; fax: 39 023 900 1916; e-mail: mazzatorta@marionegri.it.

† Istituto Mario Negri.

‡ DEI, Politecnico di Milano.

Table 1. Statistical Indices of the Data Set, Scaled between 0 and 1

	value (mg/L)
max	0.9756
min	0.0476
range	0.9280
SD	0.2560
variance	0.0655
geom mean	0.5131
average	0.5971

2. MATERIALS AND METHODS

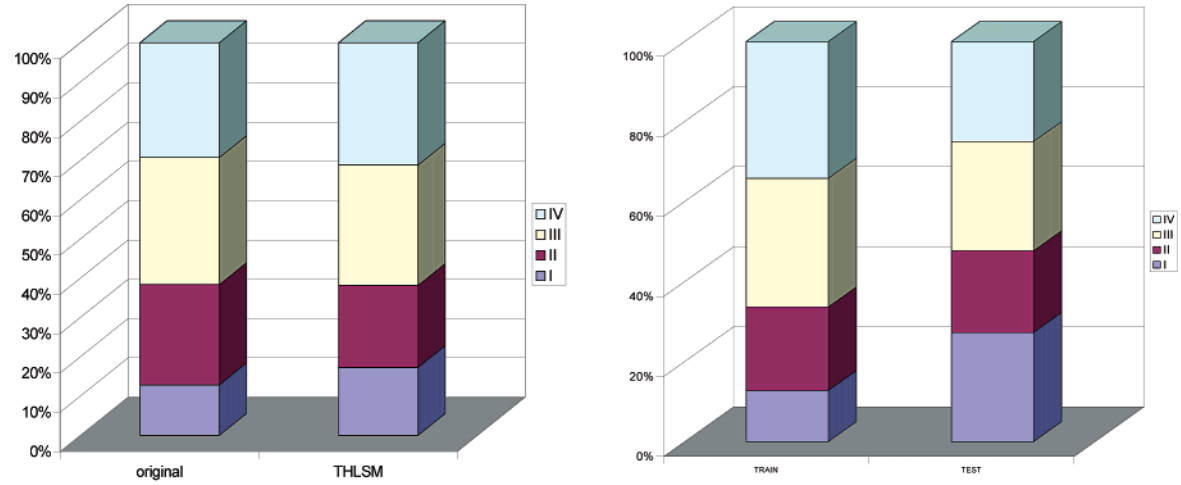
2.1. Data Set. We mined a data set of 562 organic compounds commonly used in industrial processes. The U.S. Environmental Protection Agency[12−15] built up this data set, starting from a review of experimental data in the literature, referring to acute toxicity 96-h LC_{50} (mg/L), for the fathead minnow (*Pimephales promelas*). The compounds were randomly partitioned 70−30% between 392 training cases, used to develop the models, and 170 testing cases, used to evaluate their ability in prediction. The toxicity was scaled between 0 and 1 and processed using a Tangent Hyperbolic − Logarithmic Scaling Modified procedure (THLSM).[16] Some statistical information about the data set is summarized in Table 1.

The distribution of compounds with respect on EU classification for the original and THLSM data set is shown in Figure 1a; the distribution for training and test set in the THLSM data set is shown in Figure 1b.

2.2. Descriptors. A large number of descriptors was calculated using different software: Hyperchem 5.0 (Hypercube Inc., Gainsville, Florida, U.S.A.), CODESSA 2.2.1 (SemiChem Inc., Shawnee, Kansas, U.S.A.), Pallas 2.1 (CompuDrug; Budapest, Hungary). The variables describing the molecules best must be selected to obtain a good model. There is the risk that some descriptors do not add information, increase noise, and make it more complicated to analyze the result. In addition a smaller number of variables reduces the risk of overfitting. The descriptors (Table 2) were selected by Principal Components Analysis (PCA), with the Principal Components Eigenvalue (Scree) Plot method, and Genetic Algorithms.

2.3. NIKE. NIKE (Neural explicit&Implicit Knowledge interference systEm) is a hybrid intelligent system[11] devel-

10.1021/ci025585q CCC: $25.00
Published on Web 01/23/2003



(a) distribution of the data sets **(b) distribution of training and test sets**

Figure 1. Distribution for the original and THLSM data set with respect on EU classification (adapted from Mazzatorta et al.[16]) (a). Distribution for the training and test sets in the THLSM data set with respect on EU classification (b).

Table 2. Descriptors Involved in the Models

descriptor	code
total energy (kcal/mol)	QM1
LUMO (eV)	QM6
molecular weight (amu)	C35
Kier & Hall index (order 0)	T6
molecular volume	G10
molecular surface area	G12
TMSA total molecular surface area [Zefirov's PC]	E13
FPSA-3 fractional PPSA (PPSA-3/TMSA) [Zefirov's PC]	E31
LogP	LogP

oped in Matlab 6 (MathWorks Inc.) for prediction, based on modular neural and neurofuzzy networks. The original NIKE automates the tasks involved in this process, from data representation for toxicity measurements to the prediction of a given new input. NIKE contains two modules that can be used individually and operate on the same inputs in order to model and predict toxicity. We defined *implicit knowledge* as the knowledge represented by neural/neurofuzzy networks, created and adapted by a learning algorithm. The representation of implicit knowledge is based on the numerical weights of the connections between neurons.

The first module, called IKM-CNN (Implicit Knowledge Module-based on Crisp Neural Networks), models the data set as a multilayer perceptron (MLP[10]). The second module, called IKM-FNN (Implicit Knowledge Module-based on Fuzzy Neural Networks), is implemented as a multilayered neural structure with an input layer, establishing the input for the membership degrees of the current values, a fully connected three-layered FNN2,[17] and a defuzzification layer[11] (Figure 2). The weights of the connections between layer 1 and layer 2 are set at one. The linguistic variable X_i is described by m_i fuzzy sets, A_{ij}, having the degrees of membership performed by functions $\mu_{ij}(x_i)$, $j = 1,2,...,m_i$, $i = 1,2,...,p$ (in our case, on the descriptors and the toxicity values). Since layers 1 and 5 are used in the fuzzification process in the training and prediction steps, layers 2−4 are organized as a feed-forward network to represent the implicit rules through FNN training.[17,18]

3. RESULTS AND DISCUSSION

The back-propagation algorithm with a learning rate of 0.7 and a momentum term of 0.9 was used for training.[10]

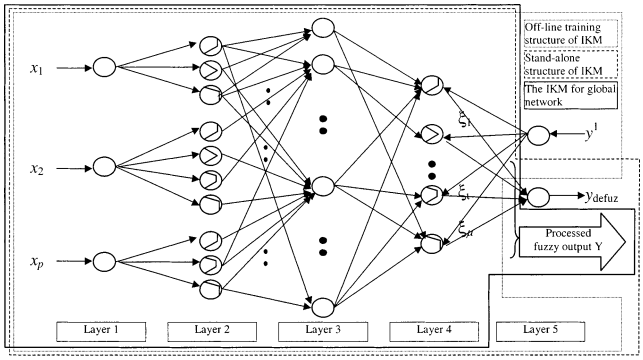


Figure 2. Implicit Knowledge Module implemented as FNN.

The networks were trained up to 5000 epochs. The parameters themselves are variables of the network analyzed, but these values, especially the very large number of training epochs, ensure a stable convergence, though the computing time is lengthy.

3.1. Number of Hidden Neurons. A first analysis was done to establish the best number of hidden neurons. For an artificial neural network (ANN) to be able to generate closed decision regions the minimum number of hidden units must be greater than the number of input units.[19] To obtain the maximum number of hidden units in the network, we used results based on Kolmogorov's theorem. Hecht-Neilson[20,21] established that the maximum number of hidden neurons needed to represent any function of n variables is less than twice the number of inputs $2 \times n\_input + 1$.

We developed numerous models with different numbers of hidden neurons in order to find the optimum. The networks analyzed are CNN and FNN with different membership functions. Table 3

Figure 3 shows how R^2 behaves in relation to the number of hidden neurons for each network. The maximum performances are clear, using 13 neurons for the FNN and 11 for CNN.

3.2. Membership Functions. A membership function is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. There are too many possible membership function shapes to systematically analyze every combination.

Table 3. Performances of the Networks

	mean	SD	max	min	R^2
CNN8H	0.1111	0.0926	0.4996	3.42E-04	0.6697
CNN9H	0.1116	0.0930	0.5072	6.14E-05	0.6670
CNN10H	0.1092	0.0945	0.5279	2.80E-04	0.6710
CNN11H	0.1125	0.0924	0.5135	2.28E-04	0.6654
CNN12H	0.1167	0.0900	0.4793	1.78E-04	0.6569
CNN13H	0.1129	0.0920	0.4889	2.18E-04	0.6650
CNN14H	0.1101	0.0938	0.5313	9.90E-05	0.6700
CNN15H	0.1132	0.0924	0.5417	2.61E-04	0.6630
CNN20H	0.1136	0.0922	0.5281	1.54E-03	0.6619
CNN25H	0.1181	0.0912	0.5171	1.27E-04	0.6484
CNN30H	0.1148	0.0918	0.5457	1.24E-04	0.6588
CNN35H	0.1117	0.0940	0.5878	9.15E-04	0.6634
CNN40H	0.1145	0.0927	0.4969	8.53E-05	0.6571
FNN11H	0.1117	0.0920	0.5176	1.70E-04	0.6695
FNN12H	0.1098	0.0905	0.4495	1.91E-04	0.6804
FNN13H	0.1078	0.0890	0.5015	4.73E-04	0.6917
FNN14H	0.1107	0.0865	0.4904	1.62E-03	0.6884
FNN15H	0.1091	0.0906	0.4664	2.34E-04	0.6827
FNN20H	0.1099	0.0914	0.4865	2.23E-04	0.6775
FNN25H	0.1138	0.0926	0.5166	9.19E-04	0.6601
FNN30H	0.1145	0.0906	0.5015	1.39E-04	0.6636
FNN35H	0.1108	0.0871	0.5114	3.94E-04	0.6867
FNN40H	0.1181	0.0928	0.5461	2.98E-04	0.6439
FNN11H−I	0.1202	0.0897	0.4965	5.10E-04	0.6449
FNN12H−I	0.1181	0.0903	0.5159	1.27E-03	0.6509
FNN13H−I	0.1189	0.0905	0.4934	5.28E-04	0.6472
FNN14H−I	0.1209	0.0889	0.5044	1.06E-04	0.6444
FNN15H−I	0.1200	0.0906	0.5019	6.67E-04	0.6431
FNN20H−I	0.1191	0.0904	0.5276	4.44E-04	0.6468
FNN25H−I	0.1265	0.0868	0.4594	5.00E-03	0.6283
FNN30H−I	0.1235	0.0898	0.4839	1.38E-04	0.6319
FNN11H−II	0.1180	0.0891	0.4735	7.57E-04	0.6549
FNN12H−II	0.1146	0.0914	0.5116	1.81E-04	0.6606
FNN13H−II	0.1131	0.0925	0.4883	7.09E-04	0.6632
FNN14H−II	0.1166	0.0903	0.4797	9.13E-04	0.6565
FNN15H−II	0.1158	0.0901	0.4971	4.67E-04	0.6600
FNN20H−II	0.1157	0.0902	0.4763	1.61E-03	0.6600
FNN25H−II	0.1151	0.0901	0.4787	4.87E-04	0.6626
FNN30H−II	0.1164	0.0900	0.4748	1.93E-03	0.6584

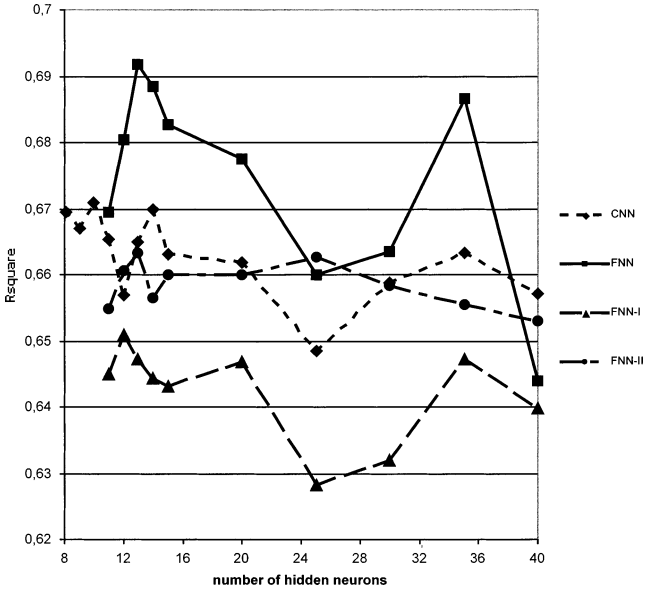


Figure 3. Performances of the networks developed.

Here we followed a reasoning based on the complexity of the data set mined and on the results obtained in the course of the work.

The input and the output are fuzzified using the membership functions shown in Figure 4. The use of a linear membership function (Figure 4a) is equivalent to use a continuous value, which brings consistent and precise information. For the inputs this choice comes from the idea that they are descriptors computed by well-defined algorithms and so not afflicted with experimental errors.[22] In the case of the output, it does not suffer any fuzzy transformation, in other word we used for the output the original value and not linguistic values.

In Figure 4f,g four trapezoidal membership functions were chosen for the output in order to follow the EC classification for acute toxicity for fish.[23] The particular shape of the functions in Figure 4g lies in a conservative approach to the problem. Because the more toxic classes are the lower ones these shapes ensure a degree of memberships to the lower class, taking into account a possible experimental undervalue of the toxicity.

Other memberships functions were analyzed in order to understand the effect that the fuzzyfication of the data has on the performance of the model.

The results of the models developed using the above-described membership functions are summarized in Table 5.

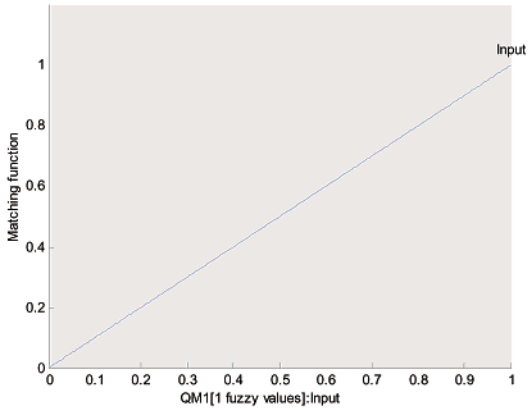
Figure 5 illustrates some examples of the defuzzification method for the output involved in this study. The centroid calculation, used in this study, is the most popular defuzzification method and it returns the center of area under the curve.

4. DISCUSSION AND CONCLUSIONS

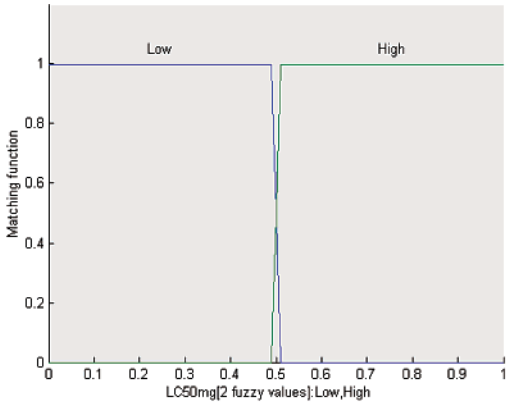
One of the problems arising during neural network training is called overfitting. The error on the training set is driven to a very small value, but when new data are presented to the network the error is large. The network has memorized the training examples but has not learned to generalize to new situations. One method for improving this generalization is to use a network that is just large enough to provide an adequate fit. The larger the network you use, the more complex the functions it can create. A small enough network will not have enough power to overfit the data. Unfortunately, it is difficult to know beforehand how large a network should be for a specific application. The study on the number of hidden neurons showed that in this situation small networks are preferable to large ones (Table 3 and Figure 3).

The use of fuzzy logic for this application helped improve the performances (Table 3 and Figure 3). The reason lies in the peculiarity of this approach. Fuzzy logic is tolerant of imprecise data, because fuzzy sets describe vague concepts (e.g. low, medium, high toxicity, etc.) and admits the possibility of partial membership in these categories. The degree to which an object belongs to a fuzzy set is indicated by a membership value between 0 and 1 (see Figure 5). The performances (Table 5) of FNN13H−D, FNN13H−II, and FNN13H−VI or FNN13H−III and FNN13H−IV, which have the same membership functions for the output and differ for the fuzzification of the inputs, are very close. This is because the input, i.e., the descriptors computed, are relatively precise and homogeneous compared to the output.[22] In fact even if the inputs are not fuzzified, i.e., using a linear membership function (FNN13H−II and FNN13H−IV), the performances are comparable.

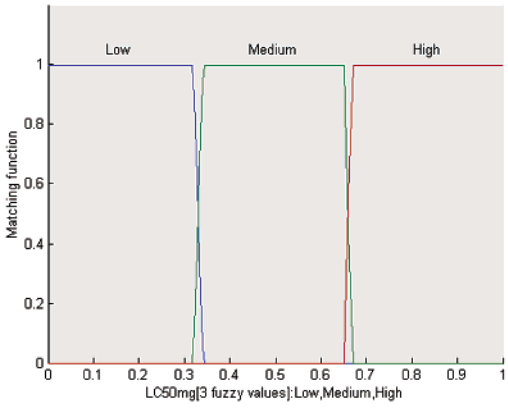
The opposite reasoning holds for the output. FNN13H−I, FNN13H−II, FNN13H−IV, or FNN13H−D, FNN13H−III,



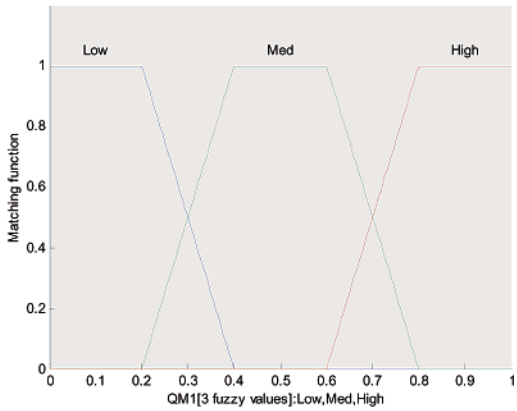
(a) Linear function



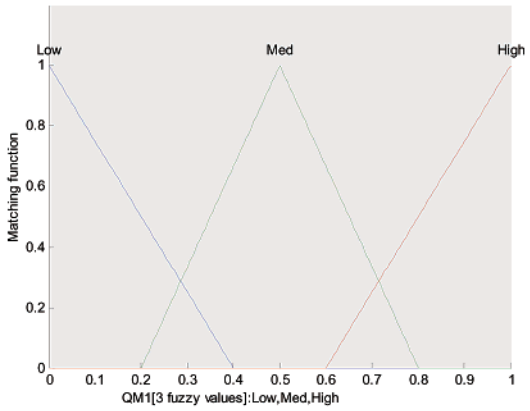
(b) Two trapezoidal functions



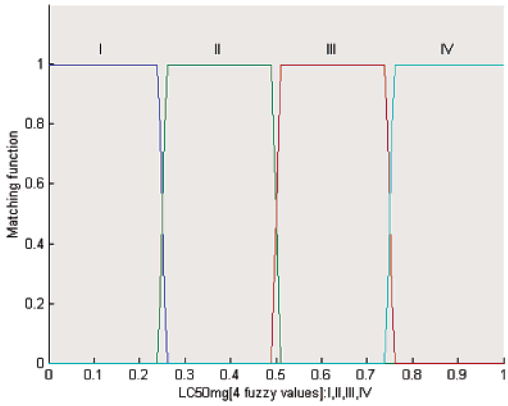
(c) Three trapezoidal functions



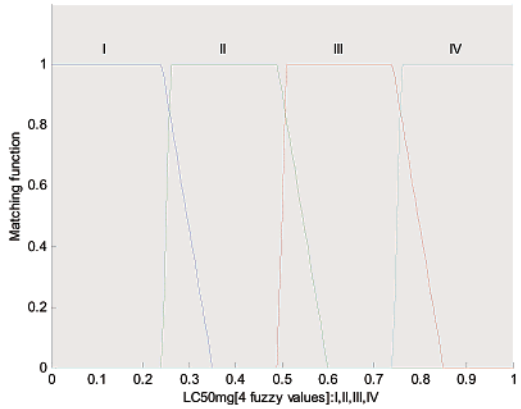
(d) Three trapezoidal functions



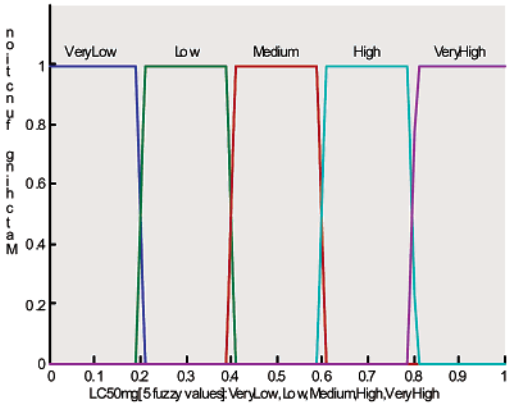
(e) Three triangular functions



(f) Four trapezoidal functions



(g) Four trapezoidal asymmetric functions



(h) Five trapezoidal functions

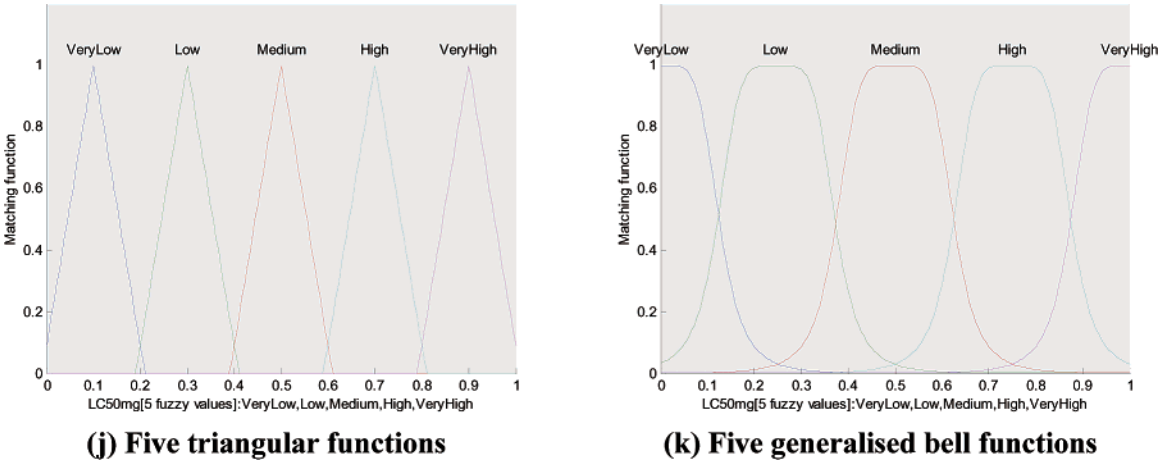


Figure 4. Representation of the membership functions used in this study for the fuzzification of the input and the output.

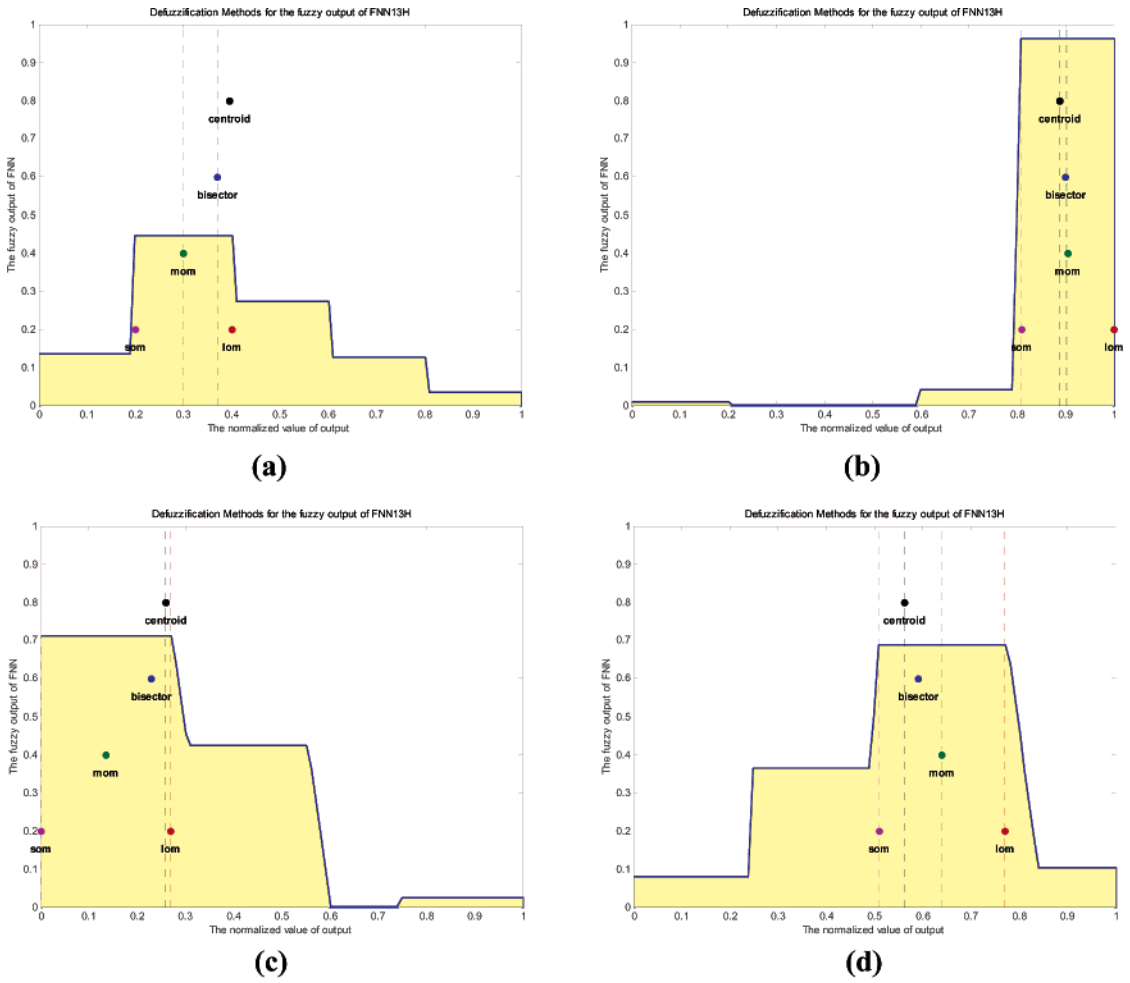


Figure 5. Defuzzification of the fuzzy output: (a) good fuzzy inference prediction for *p*-phenoxybenzaldehyde (FNN13H−D prediction: 0.39605; experimental: 0.38999); (b) bad fuzzy inference prediction for 1,1-dimethylhydrazine (FNN13H−D prediction: 0.88874; experimental: 0.46858); (c) good fuzzy inference prediction for *p*-chlorophenyl-*o*-nitrophenyl ether (FNN13H−I prediction: 0.26025; experimental: 0.26745); and (d) bad fuzzy inference prediction for 2-methyl-1,4-naphthoquinone (FNN13H−I prediction: 0.56213; experimental: 0.068761).

FNN13H−V, and FNN13H−VII have the same membership functions for the inputs but differ for the fuzzification of the output. The worsening of the model is clear with a linear membership function for the output, but generally speaking the models get worse as the output gets more vague. The best results are with five very crisp membership functions (FNN13H−D and FNN13H−VI), and the models get worse and worse as the membership functions overlap more and more. This suggests that the starting data set, even though it is one of the most reliable used, is affected by unavoidable experimental errors. As shown in FNN13H−VIII, FNN13H−IX, FNN13H−X, and FNN13H−D fuzzy logic can deal with imprecise data and too few membership function are not enough to describe correctly the data set. In this case it helps us establish crisp linguistic classes for the toxicity. Many studies have set out to predict toxicity as a continuous value

Table 4. Membership Functions Involved in the Models Developed

	membership functions for the input	membership functions for the output
FNN13H−D	(d)	(h)
FNN13H−I	(a)	(g)
FNN13H−II	(a)	(h)
FNN13H−III	(d)	(k)
FNN13H−IV	(a)	(k)
FNN13H−V	(d)	(a)
FNN13H−VI	(e)	(h)
FNN13H−VII	(d)	(j)
FNN13H−VIII	(d)	(b)
FNN13H−IX	(d)	(c)
FNN13H−X	(d)	(f)

Table 5. Performances of the Networks

	mean	SD	max	min	R^2 training	R^2 test
FNN13H−D	0.1302	0.1099	0.5886	0.0005	0.6917	0.3475
FNN13H−I	0.1312	0.1012	0.5560	0.0005	0.6472	0.4979
FNN13H−II	0.1254	0.1044	0.5612	0.0004	0.6632	0.5019
FNN13H−III	0.1492	0.1040	0.5385	0.0021	0.6086	0.3314
FNN13H−IV	0.1455	0.1027	0.5642	0.0005	0.5960	0.4124
FNN13H−V	0.1890	0.1234	0.5355	0.0016	0.1953	0.0000
FNN13H−VI	0.1303	0.1086	0.5993	0.0008	0.6912	0.3593
FNN13H−VII	0.1367	0.1104	0.5790	0.0006	0.6441	0.3582
FNN13H−VIII	0.1462	0.1111	0.6067	0.0002	0.6046	0.3118
FNN13H−IX	0.1378	0.1075	0.5875	0.0004	0.6738	0.3160
FNN13H−X	0.1294	0.1134	0.5906	0.0001	0.6992	0.3088

or in classes.[24−28] Advantages of toxicity classes are that they are less sensitive to variability of the experimental value and that they can be related to regulatory toxicity classes. However, even a small difference in the algorithm can cause a change in the predicted class, resulting in a change in the output which is bigger than in the case of continuous values. As a result, it must be carefully studied how to shape the classes, and NIKE proves particularly useful, flexible, and speedy to cope with this problem.

This paper illustrates the difficulties of modeling the toxicity of chemicals. The data set mined is fundamental and, of course, strongly effects the resulting models. In this study the width and reliability of the Duluth data set make it a strong starting point, but its heterogeneity makes it extremely complex to model (see Table 1). Networks trained on this data set must therefore be as general as possible.

ACKNOWLEDGMENT

This work is partially funded by the EU under contract HPRN-CT-1999-00015. We thank Prof. A. R. Katritzky (Gainesville, Florida) and Prof. M. Karelson (Tartu, Estonia) for the use of CODESSA.

REFERENCES AND NOTES

(1) Benfenati, E.; Gini, G. Computational predictive programs (expert systems) in toxicology. *Toxicology* **1997**, *119*, 213−225.

(2) Benfenati, E. Multiple Toxicity Prediction Using Different Molecular Descriptors. V Girona Seminar on Molecular Similarity, 12−20 July 2001, University of Girona, Girona, Spain.

(3) Benigni, R.; Richard, A. M. QSARS of Mutagens and Carcinogens: Two Case Studies Illustrating Problems in the Construction of Models for Noncongeneric Chemicals. *Mutation Res.* **1996**, *371*, 29−46.

(4) Chemical Abstract Service, ACS, 2540 Olentangy River Road, P.O. Box 3012, Columbus, OH 43210.

(5) Dearden, J. C.; Barratt, M. D.; Benigni, R.; Bristol, D. W.; Combes, R. D.; Cronin, M. T. D.; Judson, P. N.; Payne, M P.; Richard, A. M.; Tichy, M.; Worth, A. P.; Yourick, J. J. The Development and Validation of Expert Systems for Predicting Toxicity. *ATLA* **1997**, *25*, 223−252.

(6) Gini, G.; Katritzky, A. R. *Predictive Toxicology of Chemicals: Experiences and Impact of AI Tools;* AAAI Press: Menlo Park, CA, 1999; pp 40−43.

(7) Sztandera, L. M.; Trachtman, M,; Bock, C.; Velga, J.; Garg, A. Soft Computing and Density Functional Theory in the Design of Safe Chemicals. *Soft Comput. Chem.* **2002**, 67−93.

(8) Sztandera, L. M.; Trachtman, M,; Bock, C.; Velga, J.; Garg, A. Soft Computing in the Design of Non Toxic Textile Chemicals. *Soft Comput. Textile Sci*. **2002**, 51−74.

(9) Vracko, M. A study of Structure-Carcinogenic Potency Relationship with Artificial Neural Networks. The Using of Descriptors Related to Geometrical and Electronic Structures. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 1037−1043.

(10) Rumelhart, D. E.; McClelland, J. L. *Parallel Distributed Processing, Explanations in the Microstructure of Cognition;* MIT Press: 1986.

(11) Neagu, C.-D.; Avouris, N. M.; Kalapanidas, E.; Palade, V. Neural and Neuro-fuzzy Integration in a Knowledge-based System for Air Quality Prediction. *Appl. Intell. J.* **2002**, *17*, 141−168.

(12) ECOTOX. *ECOTOXicology Database System, Code List*. Prepared for U.S. Environmental Protection Agency, Office of Research, Laboratory Mid-Continent Division (MED), Duluth, Minnesota. By OAO Corporation Duluth, Minnesota, February 2000.

(13) ECOTOX. *ECOTOXicology Database System, Data Field Definition.* Prepared for U.S. Environmental Protection Agency, Office of Research, Laboratory Mid-Continent Division (MED), Duluth, Minnesota. By OAO Corporation Duluth, Minnesota, February 2000.

(14) ECOTOX. *ECOTOXicology Database System, User Guide*; Prepared for U.S. Environmental Protection Agency, Office of Research, Laboratory Mid-Continent Division (MED), Duluth, Minnesota. By OAO Corporation Duluth, Minnesota, February 2000.

(15) Russom, C. L.; Bradbury, S. P.; Broderius, S. J.; Hammermeister, D. E.; Drummond, S. J. Predicting modes of toxic action from chemical structure: acute toxicity in the fathead minnow (*pimephales promelas*). *Environ. Toxicol. Chem.* **1997**, *16*, 948−967.

(16) Mazzatorta, P.; Benfenati, E.; Neagu, C.-D.; Gini, G. The importance of scaling in data mining for toxicity prediction, *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 1250−1255.

(17) Fuller, R. Introduction to Neuro-Fuzzy Systems. *Advances in Soft Computing Series*; Springer-Verlag: Berlin, 1999.

(18) Neagu, C.-D.; Palade, V. An interactive fuzzy operator used in rule extraction from neural networks. *Neural Network World* **2000**, *10/4*, 675−684.

(19) Gori, M.; Scarselli, F. Are multilayer perceptrons adequate for pattern recognition and verification? *IEEE Trans. Pattern Anal. Mach. Intell* **1998**, *20/11*, 1121−1131.

(20) Hecht-Neilson, R. *Neurocomputing*; Addison-Wesley Pub Co: 1990.

(21) Kurkovà, V. Kolmogorov's theorem is relevant. *Neural Comput.* **1991**, *2/4*, 617−622.

(22) Benfenati, E.; Piclin, N.; Roncaglioni, A.; Varì, M. R. Factors Influencing Predictive Models For Toxicology. *SAR QSAR Environ. Res.* **2001**, *12*, 593−603.

(23) Directive 92/32/ECC the seventh amendment to Directive 67/548/ECC, OJL 154 of 5.VI.92, 1992; p 1. www.eurunion.org/legislat/chemical.htm.

(24) Benfenati, E.; Mazzatorta, P.; Neagu, C.-D.; Gini, G. Combining classifiers of pesticides toxicity through a neuro-fuzzy approach. In *Multiple Classifier Systems*; Roli, F., Kittler, J., Eds.; 2002; pp 293−303.

(25) Gini, G.; Lorenzini, M.; Benfenati, E.; Grasso, P.; Bruschi, M. Predictive Carcinogenicity: A Model for Aromatic Compounds, with Nitrogen-Containing Substituents, Based on Molecular Descriptors Using an Artificial Neural Network. *J. Chem. Inf. Comput. Sci*. **1999**, *39*, 1076−1080.

(26) Hermens, J. L. M. Prediction of environmental toxicity based on structure−activity relationship using mechanistic information. *Sci. Total Environ.* **1995**, *171*, 235−242.

(27) Neagu, C.-D.; Aptula, A. O.; Gini, G. Neural and Neuro-Fuzzy Models of Toxic Action of Phenols, IEEE Int'l Symposium 'Intelligent Systems' Methodology, Models, Applications in Emerging Technologies IS2002, Sept 10−12, 2002, Varna, Bulgaria.

(28) Neagu C.-D.; Benfenati E.; Gini G.; Mazzatorta P.; Roncaglioni A. Neuro-Fuzzy Knowledge Representation for Toxicity Prediction of Organic Compounds. *Procs. of the 15th European Conference on Artificial Intelligence*, July 21−26 2002, Lyon, France.

CI025585Q