Revue Roumaine de Chimie, **1998**, *43* (8), 775–783

ARTIFICIAL NEURAL NETWORKS APPLICATIONS. PART 5.[1]

PREDICTION OF THE SOLUBILITY OF C_{60} IN ORGANIC SOLVENTS

Ovidiu IVANCIUC

Department of Organic Chemistry, Faculty of Industrial Chemistry
University "Politehnica" of Bucharest, Splaiul Independenţei 313,
77206 Bucharest, Roumania
E-mail: o_ivanciuc@chim.upb.ro

Received January 21, 1997

The solubility of C_{60} in a variety of solvents was estimated with multi-layer feedforward neural networks using as independent variables four solvent parameters: n, the index of refraction, ε, the DC dielectric constant, V, the molar volume, and δ, the Hildebrand solubility parameter. The neural network model gives better results than the multiple linear regression one. The study presents the influence of three activation functions (hyperbolic tangent, linear and bell) on the network performances, both in calibration and prediction.

INTRODUCTION

The solubility of C_{60} in organic solvents plays an important role in its extraction, chromatographic separation, and reactions.[2–5] An understanding of the physical factors which influence the interaction of fullerenes with various solvents will allow a better design of their extraction and purification. The solubility of C_{60} in 47 solvents was recently reported,[5] and the substituted naphthalenes such as 1-methyl, 1-chloro, and 1-phenyl were found to exhibit the highest room temperature solubility for C_{60}. Despite some qualitative predictions concerning the dependence between C_{60} solubility and physical parameters of the solvents, no quantitative relationship was obtained.

In recent years, Artificial Neural Networks (ANN) were used as universal approximators that can model Quantitative Structure-Property Relationships (QSPR). A neural network is a simplified brain model that can detect patterns and correlations in data.[6–8] Unlike the usual Multi-Linear Regression (MLR) model used in QSPR, ANN can explore the nonlinear relationship between structural parameters and physico-chemical properties. The use of ANN has some important advantages, due to the fact that the mathematical form of the relationship between the input and output data does not need to be provided, and ANN are able to represent nonlinear relationships between input and output patterns, as well as interaction between the independent parameters.

Recent reports have demonstrated the utility of ANN in investigating chemical phenomena.[9,10] Among them, related to the present investigation are the estimation of the aqueous solubility of organic compounds [11,12] and the computation of the partition coefficient.[13]

The scope of the present paper is to model the solubility of the fullerene C_{60} in different solvents by using Multi-Layer Feedforward (MLF) neural networks.

RESULTS AND DISCUSSION

The solubility data of C_{60} were taken from literature:[5] [C_{60}], the concentration of C_{60} in mg/mL, and $X_i(C_{60})$, the mole fraction of C_{60}. Four solvent parameters were considered to influence the solubility of C_{60}: n, the index of refraction (Na D line); ε, the DC dielectric constant; V, the molar volume, which is equal to the molecular weight divided by the density at 298 K; and δ, which is the Hildebrand solubility parameter. The list of solvents, experimental solubility, and solvent parameters are presented in Table 1 for the 42 solvents with complete characterization of the parameters in ref. 5.

Table 1

Solvents, refraction index, n, dielectric constant, ε, molar volume in cm^3/mol, V, Hildebrand solubility parameter in $cal^{1/2}/cm^{-3/2}$, δ, solubility of C_{60} in mg/mL, [C_{60}], and mole fraction, $X_i(C_{60})\times10^4$

Solvent	n	ε	V	δ	[C_{60}]	$X_i(C_{60})\times10^4$
n-pentane	1.36	1.84	115	7.0	0.005	0.008
cyclopentane	1.41	1.97	93	8.6	0.002	0.003
n-hexane	1.38	1.89	131	7.3	0.043	0.073
cyclohexane	1.43	2.02	108	8.2	0.036	0.059
n-decane	1.41	1.99	195	8.0	0.071	0.19
decalins	1.48	2.20	154	8.8	4.6	9.8
dichloromethane	1.42	9.08	60	9.7	0.26	0.27
chloroform	1.45	4.81	86	9.3	0.16	0.22
carbon tetrachloride	1.46	2.24	80	8.6	0.32	0.4
1,2-dibromoethane	1.54	4.79	72	10.4	0.5	0.6
trichloroethylene	1.48	3.40	89	9.2	1.4	1.7
tetrachloroethylene	1.51	2.46	102	9.3	1.2	1.7
1,1,2,2-tetrachloroethane	1.49	8.20	64	9.7	5.3	7.7
methanol	1.33	33.62	41	14.5	0	0
ethanol	1.36	24.3	59	12.7	0.001	0.001
nitromethane	1.38	35.9	81	12.7	0	0
nitroethane	1.39	28.0	105	11.1	0.002	0.002
acetone	1.36	20.7	90	9.8	0.001	0.001
acetonitrile	1.34	37.5	52	11.8	0	0
benzene	1.50	2.28	89	9.2	1.7	2.1
toluene	1.50	2.44	106	8.9	2.8	4
xylenes	1.50	2.40	123	8.8	5.2	8.9
1,3,5-trimethylbenzene	1.50	2.28	139	8.8	1.5	3.1
tetralin	1.54	2.76	136	9.0	16	31
o-cresol	1.54	11.50	103	10.7	0.014	0.029
benzonitrile	1.53	25.60	97	8.4	0.41	0.71
fluorobenzene	1.47	5.42	94	9.0	0.59	0.78
nitrobenzene	1.56	35.74	103	10.0	0.8	1.1
bromobenzene	1.56	5.40	105	9.5	3.3	4.8
anisole	1.52	4.33	109	9.5	5.6	8.4
chlorobenzene	1.52	5.71	102	9.2	7	9.9
1,2-dichlorobenzene	1.55	9.93	113	10.0	27	53
1,2,4-trichlorobenzene	1.57	3.95	125	9.3	8.5	15
1-methylnaphthalene	1.62	2.92	142	9.9	33	68
dimethylnaphthalenes	1.61	2.90	156	9.9	36	78
1-phenylnaphthalene	1.67	2.50	155	10.0	50	131
1-chloronaphthalene	1.63	5.00	136	9.8	51	97
carbon disulfide	1.63	2.64	54	10.0	7.9	6.6
tetrahydrofuran	1.41	7.60	81	9.1	0	0
tetrahydrothiophene	1.50	2.28	88	9.5	0.03	0.036
2-methylthiophene	1.52	2.26	96	9.6	6.8	9.1
pyridine	1.51	12.30	80	10.7	0.89	0.99

The estimations of the ANN model will be compared with the results provided by a MLR model. For [C_{60}], the following equation was obtained:

$$[C_{60}] = -169.77(\pm 167.48) + 80.38(\pm 79.30)n - 0.142(\pm 0.140)\varepsilon +$$

$$+ 0.184(\pm 0.182)V + 4.078(\pm 4.023)\delta \qquad (1)$$

$$N = 42 \quad r = 0.784 \quad s = 8.521$$

where N is the number of compounds used in the correlation, r is the correlation coefficient, and s is the standard deviation. The standard error of estimation of each coefficient at the 95% confidence level is given in parentheses. The partial correlation coefficients are $r(n) = 0.675$, $r(\varepsilon) = -0.237$, $r(V) = 0.483$, and $r(\delta) = 0.030$, and their low values lead to the conclusion that no single parameter can estimate the solubility data. The correlation coefficient of eq. (1) is very low, and this relationship has only a qualitative value, indicating that n and V are the most important solvent parameters which influence the solubility of C_{60}. The collinearity between the four variables is quite low as indicated by the intercorrelation matrix:

	n	ε	V	δ
n	1.000	–0.420	0.377	–0.166
ε		1.000	–0.449	0.708
V			1.000	–0.511

A similar MLR model was computed for the dependence of the mole fraction of C_{60}, $X_i(C_{60})$, on the solvent parameters:

$$X_i(C_{60}) \times 10^4 = -365.67(\pm 373.42) + 165.18(\pm 168.67)n -$$

$$- 0.285(\pm 0.291)\varepsilon + 0.447(\pm 0.457)V + 9.332(\pm 9.530)\delta \qquad (2)$$

$$N = 42 \quad r = 0.773 \quad s = 19.354$$

The partial correlation coefficients are also very low, with the largest values exhibited by the correlation with n and V: $r(n) = 0.644$, $r(\varepsilon) = -0.219$, $r(V) = 0.498$, and $r(\delta) = 0.036$. The low value of r and high value of s in eq. (2) indicates that this type of equation has a small modeling power.

Recently it was proposed to express the solubility of the fullerenes as a linear correlation between log(solubility) and $n' = (n^2 - 1)/(n^2 + 2)$.[14] Because the MLR model in eq. (1,2) was of low predictive value, we will present the MLR model which expresses the log(solubility) as a linear function of n', $\varepsilon' = (\varepsilon - 1)/(\varepsilon + 2)$ (as proposed in ref. 2), V, and δ. Because C_{60} was insoluble in four solvents (methanol, nitromethane, acetonitrile, and tetrahydrofuran) their data were not considered in developing the log(solubility) MLR model. For log[C_{60}] the following equation was computed:

$$\log[C_{60}] = -6.940(\pm 5.464) + 29.906(\pm 23.545)n' - 0.225(\pm 0.177)\varepsilon' +$$

$$+ 0.002(\pm 0.0017)V - 0.223(\pm 0.175)\delta \qquad (3)$$

$$N = 38 \; r = 0.862 \; s = 0.731$$

The correlation coefficient is higher, indicating that this equation can be used as a qualitative model of the solubility of C_{60}, but the large uncertainty in the values of the coefficients prevents its use as a quantitative model for predicting the solubility in new, untested solvents. The partial correlation

coefficients for eq. (3) show that n' is the most important parameter in determining the solubility of C_{60} : $r(n') = 0.836$, $r(\varepsilon') = -0.199$, $r(V) = 0.303$, and $r(\delta) = -0.016$. The low values of the intercorrelation coefficients is an indication that all four parameters can be used in a MLR model:

	n'	ε'	V	δ
n'	1.000	–0.043	0.211	–0.222
ε'		1.000	–0.409	0.640
V			1.000	–0.375

The MLR relationship between log $[X_i(C_{60})\times 10^4]$ and the solubility parameters gives a slightly better model, as indicated by the increase of the correlation coefficient:

$$\log [X_i(C_{60})\times 10^4] = -7.127(\pm 5.330) + 30.557(\pm 22.850)n' - $$
$$- 0.176(\pm 0.131)\varepsilon' + 0.005(\pm 0.004)V - 0.243(\pm 0.182)\delta \qquad (4)$$
$$N = 38 \quad r = 0.873 \quad s = 0.730$$

Again, the largest part of the variation of log $[X_i(C_{60})\times 10^4]$ is expressed by n', as indicated by the partial correlation coefficients for eq. (4): $r(n') = 0.834$, $r(\varepsilon') = -0.221$, $r(V) = 0.366$, and $r(\delta) = -0.046$.

The statistical results of the MLR models presented in eq. (1–4) indicate that only qualitative conclusions can be obtained for the dependence of the C_{60} solubility on the solvent parameters. The error in determining the values of the coefficients is as large as the values of the coefficients, which indicates a very low confidence level for the MLR models.

The low value for the correlation coefficient in the MLR model has two possible explanations: (a) the dependence between the solvent parameters and C_{60} solubility is strongly nonlinear; (b) not all important solvent parameters were identified and used in the MLR model. If the low performance of the MLR model is due to the factor (a), the ANN model will give better results, otherwise, in the case (b), the neural model will fail to give a good fit for the solubility of C_{60}.

The neural networks used in the present study are three-layer MLF networks, with four input units representing the four solvent parameters (n, ε, V, and δ), and one output unit representing the values of $[C_{60}]$ and $X_i(C_{60}) \times 10^4$ from Table 1. The training was done with the backpropagation algorithm,[6,8] the input and output scalings were set between –0.9 and 0.9, the initial weights scaling between –0.1 and 0.1, the learning rate for the hidden layer 0.01, and the momentum 0.8. The learning rate for the output layer depends on the output activation function, as indicated in Tables 2 and 3. The usual activation function has a sigmoid shape, but in previous investigations we have obtained good results with a bell-shape activation function[15–18] for the neurons from the hidden layer. The bell-shape function, with the formula Act $(z) = 1/(1 + z^2)$, can be used in neural models with good calibration and prediction results, being a good alternative to the sigmoidal or linear activation functions. The use of the bell-shape function has also a theoretical basis: Kreinovich[19] demonstrated that an arbitrary nonlinear activation function in the hidden layer is sufficient to represent all functions by neural networks. Other activation functions used in this study were the hyperbolic tangent, tanh, and the linear activation function (only for the output layer).

In order to compare the performance of the ANN model with the statistical results of the MLR equations, we have used the standard deviation s and the correlation coefficient r of the linear correlation between Y_{exp} (the experimental C_{60} solubility, either $[C_{60}]$ or mole fraction) and Y_{ANN} (the solubility estimated by the ANN model):

$$Y_{exp} = A + B \cdot Y_{ANN} \qquad (5)$$

Table 2

Calibration results for the estimation of the C_{60} solubility in organic solvents.
The table reports the network specifications, number of hidden neurons (H), number of training epochs (E), linear regression coefficients (A and B), correlation coefficient (r), and standard deviation (s).

Network Specifications	H	E	A	B	r	s
Y = [C60]	1	1200	0.892	1.019	0.9366	4.6260
Act(Hid) = bell	2	7100	0.15	1.003	0.9477	4.2151
Act(Out) = tanh	3	7900	0.912	0.985	0.9783	2.7342
η(Out) = 0.01	4	5300	0.249	0.997	0.9899	1.8721
	5	5400	0.368	0.989	0.9882	2.0206
Y = [C60]	1	300	0.078	1.002	0.9405	4.4876
Act(Hid) = tanh	2	7900	0.286	0.986	0.9774	2.7906
Act(Out) = tanh	3	11700	0.293	0.991	0.9887	1.9794
η(Out) = 0.005	4	10500	0.535	0.986	0.982	2.4871
	5	14800	0.119	0.976	0.9947	1.3528
Y = [C60]	1	500	0.26	0.994	0.9408	4.4762
Act(Hid) = tanh	2	16100	0.195	1.009	0.9801	2.6151
Act(Out) = linear	3	16300	0.259	0.998	0.9801	2.6179
η(Out) = 0.005	4	12200	0.498	1.004	0.9802	2.6118
	5	16500	0.077	1.008	0.9903	1.8389
$Y = X_i(C60)\times10^4$	1	1600	0.852	1.006	0.9496	9.2044
Act(Hid) = bell	2	5300	0.323	0.998	0.9612	8.1031
Act(Out) = tanh	3	4800	0.532	1.005	0.9841	5.2078
η(Out) = 0.01	4	6900	0.275	1.005	0.9907	3.9939
	5	6300	0.239	1.002	0.9948	2.9925
$Y = X_i(C60)\times10^4$	1	1400	-0.239	1.013	0.9561	8.6088
Act(Hid) = tanh	2	9300	-0.031	1.010	0.9750	6.5231
Act(Out) = tanh	3	10900	0.143	1.006	0.9904	4.0574
η(Out) = 0.005	4	10600	0.411	0.994	0.9875	4.6343
	5	9300	0.307	0.997	0.991	3.8845
$Y = X_i(C60)\times10^4$	1	600	0.455	0.994	0.9577	8.4507
Act(Hid) = tanh	2	1800	0.409	0.995	0.9589	8.3368
Act(Out) = linear	3	9800	0.446	0.999	0.9753	6.4809
η(Out) = 0.005	4	10700	0.506	1.001	0.9762	6.3728
	5	8900	0.529	1.001	0.9762	6.3669

Table 3

Leave-one-out cross-validation results for the prediction of the solubility of fullerene C_{60}. The notations are explained in Table 2

Network Specifications	H	A	B	*r*	*s*
Y = [C60]	1	0.862	0.926	0.8588	6.7635
Act(Hid) = bell	2	2.159	0.624	0.5974	10.5874
Act(Out) = tanh	3	1.959	0.552	0.6419	10.1236
η(Out) = 0.01	4	1.259	0.666	0.6978	9.4569
	5	1.152	0.69	0.7617	8.5548
Y = [C60]	1	0.421	0.869	0.8403	7.1568
Act(Hid) = tanh	2	0.753	0.881	0.8982	5.8035
Act(Out) = tanh	3	1.394	0.682	0.7457	8.7969
η(Out) = 0.005	4	1.648	0.661	0.7208	9.1517
	5	1.249	0.673	0.7804	8.2558
Y = [C60]	1	0.511	0.879	0.8551	6.845
Act(Hid) = tanh	2	0.523	0.867	0.8408	7.1477
Act(Out) = linear	3	0.525	0.867	0.8428	7.1066
η(Out) = 0.005	4	0.521	0.875	0.8454	7.0529
	5	0.579	0.87	0.8416	7.1309
Y = X_i(C60)×104	1	0.533	0.929	0.8771	14.1068
Act(Hid) = bell	2	0.473	0.875	0.8097	17.2358
Act(Out) = tanh	3	2.225	0.595	0.6828	21.4563
η(Out) = 0.01	4	1.954	0.591	0.7429	19.6573
	5	0.734	0.713	0.8012	17.5725
Y = X_i(C60)×104	1	-0.009	0.926	0.8496	15.4887
Act(Hid) = tanh	2	0.575	0.884	0.7870	18.1175
Act(Out) = tanh	3	2.045	0.717	0.7376	19.8291
η(Out) = 0.005	4	3.014	0.607	0.6337	22.7173
	5	2.599	0.626	0.7097	20.6896
Y = X_i(C60)×104	1	0.544	0.961	0.8613	14.9203
Act(Hid) = tanh	2	0.675	0.943	0.8458	15.6696
Act(Out) = linear	3	0.633	0.939	0.8315	16.3163
η(Out) = 0.005	4	0.524	0.955	0.8583	15.072
	5	0.678	0.921	0.8389	15.9858

While the number of neurons in the input and output layers of the network are predetermined by the number of structural descriptors used in the model, the number of neurons in the hidden layer was selected on the basis of empirical trials, in which ANN with different number of hidden neurons are trained to predict the Y_{exp} values. The training was done by randomly presenting to the network the values of the parameters for the 42 solvents from Table 1, until the correlation coefficient in eq. (5) improved with less than 10^{-4} in 100 epochs.

Selected results obtained for the calibration of the neural model are presented in Table 2. The number of hidden neurons was set between one and five, because increasing the number of hidden neurons beyond five did not improve the model. Moreover, a network with a too low ratio between the number of data in the learning set and the number of connections has the tendency to memorize the data and presents the danger of chance correlation.

To be able to select the best ANN model, in Table 2 we present the network specifications (the solubility data Y estimated by the model, the activation function for the hidden and output layers, and the learning rate for the output layer), the number of hidden neurons (H), the number of training epochs (E), linear regression coefficients (A and B) for eq. (5), correlation coefficient (r), and standard deviation (s).

Excellent results are obtained in all cases investigated, and even for networks with only two hidden neurons the correlation coefficient is higher than 0.950. The results in Table 2 show that if the hidden layer contains three neurons, the performance of the ANN model increases, and remains almost constant for hidden layers containing four or five neurons. As expected, the best estimations are provided by a network with five hidden neurons, but as we have already pointed, they represent only marginal improvements when compared with the network with three hidden neurons. If we compare the estimations of the MLR model represented by eq. (1–4) with the statistical indices of the ANN models from Table 2, it is clear that the neural network model outperforms the MLR model and provides superior mapping of the solvent parameters to the solubility of C_{60}. Also, we can conclude that there is a nonlinear dependence between the solvent parameters and the solubility of C_{60}, and the ANN model used those parameters that describe the influence of the solvent on the solubility of C_{60}.

In QSPR studies one must consider that feedforward neural networks are universal approximators, which are capable of arbitrarily accurate approximation to arbitrary functions, when the network has sufficiently many hidden units.[20–22] Due to the fact that MLF networks are universal approximators, a network with too many connections (adjustable parameters) can offer excellent calibration results for the patterns in the training set, but will have poor performances for predicting the properties for new patterns, which were not present in the learning set.

In order to investigate the prediction of the C_{60} solubility with the neural model we have used the Leave-One-Out (LOO) cross-validation method. In the LOO technique, an untrained network is first created, then one pattern is taken out of the learning set of patterns and the network is trained with the remaining patterns. When the learning process is finished, the network predicts the output value for the pattern which was eliminated from the learning set. The pattern is then put back in the set and the next one is taken out to repeat the process, starting with the untrained network. In the LOO cross-validation method each pattern serves as an unknown once and as a training pattern all the other times. The LOO cross-validation results are presented in Table 3.

A general conclusion of the cross-validation results is that the linear output function offers constant good predictions of the C_{60} solubility for a size of the hidden layer between one and five neurons, with a cross-validation correlation coefficient (r_{cv}) higher than 0.84. The networks provided with a tanh output function are more sensitive to the number of hidden neurons: the cross-validation correlation coefficient decreases when the number of hidden neurons increases. The networks with one hidden neuron have the highest r_{cv}, with the exception of the prediction of [C_{60}] with tanh-tanh network, where the network with two hidden neurons has the highest r_{cv}. The conclusion of the LOO cross-validation is that for the prediction of the solubility of C_{60} with a neural network it is safe to use networks with one

hidden neuron for the bell-tanh and tanh-linear networks, and with two hidden neurons for the tanh-tanh network.

The best prediction of the solubility of $[C_{60}]$ is provided by the tanh-tanh network (denoted NN_1) with two hidden neurons, with $r_{cv} = 0.898$, while $X_i(C_{60})$ is best predicted by the bell-tanh network (denoted NN_2) with one hidden neuron, with $r_{cv} = 0.877$. Table 4 presents the prediction of the

Table 4

The predicted values of the C_{60} solubility in LOO cross-validation with neural networks NN_1 and NN_2

Solvent	$[C_{60}]$	NN_1[a]	res_1[b]	$X_i(C60) \times 10^4$	NN_2[c]	res_2[d]
tetralin	16	10	6	31	20	11
1,2-dichlorobenzene	27	3	24	53	12	40
1-methylnaphthalene	33	43	–10	68	87	–19
dimethylnaphthalenes	36	44	–8	78	95	–17
1-phenylnaphthalene	50	46	4	131	100	31
1-chloronaphthalene	51	46	5	97	78	19

[a] $[C_{60}]$ values predicted in the LOO cross-validation by the network NN_1, with tanh-tanh activation functions and two hidden neurons

[b] $res_1 = [C_{60}]_{exp} - [C_{60}]_{NN_1}$

[c] $X_i(C_{60})$ values predicted in the LOO cross-validation by the network NN_2, with bell-tanh activation functions and one hidden neuron

[d] $res_2 = X_i(C_{60})_{exp} - X_i(C_{60})_{NN_2}$.

solubility of C_{60} for the six solvents in Table 1 with the highest solubility. It is a well-known fact that the ANN model has a low performance with extremal data (patterns with extreme input or output values), and the good prediction for the data with high solubility is a remarkable fact. Only the solubility of 1,2-dichlorobenzene is highly underestimated, and this can be a result of the errors in the solubility parameters or a lack of suitable parameters. We are presently investigating the use of other solubility parameters for the prediction of the fullerene solubility.

CONCLUSIONS

The present study demonstrated that neural networks represent an easy and effective way of exploring nonlinearity in data, with good results in predicting the solubility of the C_{60} fullerene.

The networks with a bell activation function in the hidden layer offer good results both in the calibration phase and in the LOO cross-validation of the neural model, and we can consider this new type of activation function as a good alternative to the usual sigmoidal function. This new type of activation function has some interesting properties, and deserves further study.

ACKNOWLEDGEMENT. We thank the Ministry of Research and Technology for financial support of this research under Grants 381 TA10.

REFERENCES

1. Part 4: O. Ivanciuc, *Rev. Roum. Chim.*, **1998**, *43*, 255–260.
2. N. Sivaraman, R. Dhamodaran, I. Kaliappan, T. G. Srinivasan, P. R. Vasudeva Rao and C. K. Mathews, *J. Org. Chem.*, **1992**, *57*, 6077–6079.
3. W. A. Scrivens and J. M. Tour, *J. Chem. Soc. Chem. Commun.*, **1993**, 1207–1209.
4. C. G. Screttas and G.A. Heropoulos, *J. Org. Chem.*, **1993**, *58*, 3654–3659.
5. R. S. Ruoff, D. S. Tse, R. Malhotra and D. C. Lorents, *J. Phys. Chem.*, **1993**, *97*, 3379–3383.
6. D. E. Rumelhart and J. L. McClelland, "Parallel Distributed Processing", MIT Press, Cambridge, MA, 1986.
7. P. D. Wasserman, "Neural Computing", Van Nostrand Reinhold, New York, 1989.
8. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Nature*, **1986**, *323*, 533–536.
9. J. Gasteiger and J. Zupan, *Angew. Chem. Int. Ed. Engl.*, **1993**, *32*, 503–527.
10. J. R. M. Smits, W. J. Melssen, L. M. C. Buydens and G. Kateman, *Chemom. Intell. Lab. Syst.*, **1994**, *22*, 165–189.
11. N. Bodor, A. Harget and M.-J. Huang, *J. Am. Chem. Soc.*, **1991**, *113*, 9480–9483.
12. N. Bodor, M.-J. Huang and A. Harget, *Int. J. Quantum Chem.: Quantum Chem. Symp.*, **1992**, *26*, 853–867.
13. N. Bodor, M.-J. Huang, and A. Harget, *J. Mol. Struct. (Theochem)*, **1994**, *309*, 259–266.
14. N. Sivaraman, R. Dhamodaran, I. Kaliappan, T. G. Srinivasan, P. R. Vasudeva Rao and C. K. Mathews, *Fullerene Sci. Technol.*, **1994**, *2*, 233–246.
15. O. Ivanciuc, *Rev. Roum. Chim.*, **1995**, *40*, 1093–1101.
16. O. Ivanciuc, J.-P. Rabine, D. Cabrol-Bass, A. Panaye and J. P. Doucet, *J. Chem. Inf. Comput Sci.*, **1996**, *36*, 644–653.
17. O. Ivanciuc, J.-P. Rabine, D. Cabrol-Bass, A. Panaye and J.P. Doucet, *J. Chem. Inf. Comput Sci.*, **1997**, *37*, 587–598.
18. O. Ivanciuc, J.-P. Rabine and D. Cabrol-Bass, *Comput. Chem.*, **1997**, *21*, 437–443.
19. V. Kreinovich, *Neural Networks*, **1991**, *4*, 381–384.
20. G. Cybenko, *Math. Control Signals Syst.*, **1989**, *2*, 303–314.
21. K. Funahashi, *Neural Networks*, **1989**, *2*, 183–192.
22. K. Hornik, M. Stinchcombe, and H. White, *Neural Networks*, **1989**, *2*, 359–366.