Neural Network Studies. 1. Comparison of Overfitting and Overtraining

Igor V. Tetko,[†] David J. Livingstone,[‡,§] and Alexander I. Luik*[,†]

Biomedical Department, Institute of Bioorganic and Petroleum Chemistry, Murmanskaya, 1,
Kiev-660, 253660, Ukraine, ChemQuest, Cheyney House, 19-21 Cheyney Street, Steeple Morden,
Herts, SG8 0LP, U.K., and Centre for Molecular Design, University of Portsmouth,
Portsmouth, Hants, PO1 2EG, U.K.

Received January 27, 1995[⊗]

The application of feed forward back propagation artificial neural networks with one hidden layer (ANN) to perform the equivalent of multiple linear regression (MLR) has been examined using artificial structured data sets and real literature data. The predictive ability of the networks has been estimated using a training/test set protocol. The results have shown advantages of ANN over MLR analysis. The ANNs do not require high order terms or indicator variables to establish complex structure–activity relationships. Overfitting does not have any influence on network prediction ability when overtraining is avoided by cross-validation. Application of ANN ensembles has allowed the avoidance of chance correlations and satisfactory predictions of new data have been obtained for a wide range of numbers of neurons in the hidden layer.

INTRODUCTION

Recent years have been characterized by great progress in neural networks. These methods have become more and more popular in various branches of science including chemistry.[1,2] A special interest in networks arises from their ability to perform nonlinear approximation. It is known that ANN with one hidden layer (and also higher layer networks) can interpolate any multidimensional function with given accuracy and can exactly implement an arbitrary finite training set (a global existence theorem was formulated by A. N. Kolmogorov[3] and a possible application to ANNs was suggested by R. Hecht-Nielsen[4]). Biological phenomena are considered nonlinear by nature. Therefore, the contribution of physicochemical and substructural parameters to biological activity can be nonlinear and the above mentioned property of network mapping is very important in structure–activity relationship (SAR) and quantitative SAR (QSAR) studies. However, the outstanding capabilities of the nonlinear approximation of ANN can easily result in bad generalization of trained ANN to new data, giving rise to chance effects. Since a network with a sufficient number of neurons in the hidden layer can exactly implement an arbitrary training set, it can learn both investigated dependencies and a noise that will lower the predictive ability of the network. Two conditions influence the problem:

- size of ANN
- time of ANN training

The overfitting problem refers to exceeding some optimal ANN size, while overtraining refers to the time of ANN training that may finally result in worse predictive ability of a network.

In the SAR literature great deal of time has been devoted to the analysis of overfitting, while overtraining problems have remained *terra incognito*. Usually, the next algorithm was used in such studies. ANN was trained with different numbers of neurons in the hidden layer on a learning set and was used to predict the activity of compounds from the control set. It was shown, that predictive ability of network has substantially lowered, if the number of neurons in the hidden layer increased. This tendency of networks to "memorize" data was well demonstrated by the pioneering work of Andrea and Kalayeh.[5] The network described in this report was characterized by a parameter, ϱ, that is the ratio of the number of data points in a learning set to the number of connections (*i.e.*, the number of ANN internal degrees of freedom). This parameter was analyzed in great detail in a series of articles.[6,7] The experiments with random numbers allowed the authors to propose criteria that can be useful to avoid overfitting and chance correlation.[8] However, their final investigation both with artificial structured data files and real examples resulted in rather pessimistic conclusions about the possible application of ANNs.[9]

In several other articles some attention was devoted to analysis of the process of ANN training. During the optimization procedure of the network, the mean square error (MSE) is used as a criterion of network training

$$\mathrm{MSE} = \frac{\sum (Y_i - O_1)^2}{(\text{no. of compds.} \times \text{no. of output units})} \qquad (1)$$

where O_i is a calculated and Y_i is a target value and summation is over all patterns in the analyzed data set. It was shown that ANN frequently exhibits behavior shown on Figure 1. While the MSE of a network for a learning set gradually decreases with time of training, predictive performance of the network has parabolic dependence. It is optimal to stop net training before complete convergence has occurred (so called "early stopping).[10,11] A probable description of overtraining is that the network learns the gross structure first and then the fine structure that is generated by noise. A point when a network training should be stopped is determined by testing of the network on an additional test set (cross-validation method).

[†] Institute of Bioorganic and Petroleum Chemistry. ibpc@bioorganic.kiev.ua.

[‡] ChemQuest. davel@chmqst.demon.co.uk.

[§] Centre for Molecular Design.

[⊗] Abstract published in *Advance ACS Abstracts*, July 15, 1995.

0095-2338/95/1635-0826$09.00/0

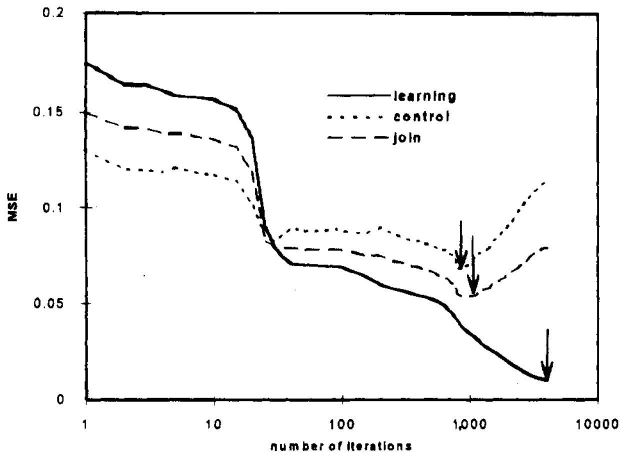

Figure 1. Mean square error (MSE) as a function of the number of training epochs for linear ASDS. The number of hidden units is five. Arrows depict stop points of ANN training and correspond to the MSE error minimum of network for control set (point S_1), control plus learning sets (point S_2), and learning set (point S_3). The last point (S_3) frequently coincides with the end of network training.

However, criteria for optimal stopping of network training in some points do not solve the problem with chance effects that were analyzed in refs 6–8. Using ANN ensembles (*i.e.*, averaging neural network predictions over several independent networks) allows the avoidance of this problem. A good theoretical discussion about the advantages of using multiple ANN predictions over single ANN prediction can be found in refs 12 and 13. Our previous analyses on SAR studies have also shown benefits of statistical averaging of network prognosis and helped us to avoid chance correlation on analyzed data.[14]

We decided to investigate the relationship between overtraining and overfitting of networks and their influence on ANN generalization performance. The explorations with artificial and real QSAR data sets conducted in ref 9 seem to be a very fruitful model, and thus this approach has been adopted here.

EXPERIMENTS WITH ARTIFICIAL DATA SETS

Data Set Generation. The first two artificial structured data sets (ASDS) were generated analogously to ref 9. Numbers generated by the random function of Borland C ++ 3.0 were scaled between 0.1 and 0.9 and were used as input variables for ASDS. One hundred target values were generated for each of three analyzed functions as shown below. The generated values were also scaled between 0.1 and 0.9. The first 50 cases were used for network training. The extra cases never participated in the learning procedure and were used as a test set. The authors from ref 9, used only 15 points for network testing. The greater number of points in our test series allowed us to avoid chance correlation when determining statistical parameters for the test series.

The ASDS had a general form

$$\text{target} = f(\theta) = g(\theta) + \epsilon \qquad (2)$$

where

$$\theta = \sum_{i=1}^{4} V_i x_i \qquad (3)$$

$g(\theta)$ is an analyzed function, x_i are four independent variables, V_i are some coefficients, and ϵ is a "noise" generated according to a Gaussian distribution with a mean equal to zero.

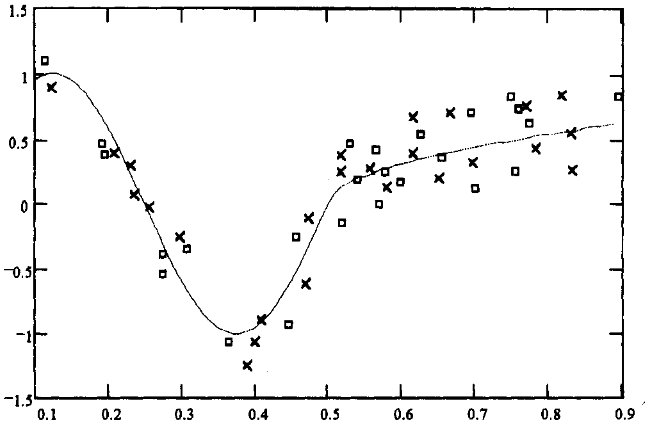

Figure 2. A function used for the nonlinear example of ASDS. Plot shows example of training set partition into learning (×) and control (◇) sets.

Linear. The function $g(\theta)$ in eq 2 was a linear one

$$g(\theta) = \theta \qquad (4)$$

The coefficients $V_i = \{-0.067, 0.67, -0.198, 0.33\}$ were generated by chance. Noise ϵ was added to keep the relationship between the independent variables and the target to a correlation coefficient R of about 0.87. A cross validation MLR by "leave-one-out" (LOO) was done for the data set. A correlation between for LOO estimation made up $R_{cv} = 0.84$. The additional coefficient, so-called cross-validated r^2 calculated by

$$\text{cross validated } r^2 = \frac{\text{SD} - \text{press}}{\text{SD}} \qquad (5)$$

was used.[15] This coefficient is similar to R; however, it was simpler to estimate its confidence interval, as shown below. Here SD is the variance of target value relative to its mean, and *press* is the average squared errors. The cross-validated r^2 calculated for MLR and LOO MLR made up $r^2 = 0.75$ and $r^2_{cv} = 0.72$ accordingly.

Quadratic. The target was generated using three variables that included the square of the first variable. The quadratic term was not used for training and testing of the network, and only two input parameters were provided for network calculations. The correlation coefficient for MLR (with squared term) was $R = 0.93$ ($r^2 = 0.87$ and $r^2_{cv} = 0.86$), while it made up only $R = 0.86$ ($r^2 = 0.74$ and $r^2_{cv} = 0.72$) when the squared term was not provided for regression analysis.

$$\text{target} = V_1 x_1 + V_2 x_1^2 + V_3 x_2 + \text{noise} \qquad (6)$$

Here $V_i = \{0.3, 2.0, 0.5\}$ were chosen to represent a quadratic nature of the data.

Nonlinear. This data set was not used in ref 9. It is an example of a rather complex relationship between input variables and the target function $g(\theta)$ generated by

$$g(\theta) = \begin{cases} \sqrt{\theta - 0.5}, & \text{if } \theta > 0.5 \\ \sin(4\pi\theta), & \text{if } \theta < 0.5 \end{cases} \qquad (7)$$

where $V_i = \{-0.175, 0.202, -0.048, 0.903\}$ (see Figure 2). Correlation coefficient between $f(x)$ and $g(x)$ was $R = 0.94$ ($r^2 = 0.79$). The correlation for the MLR was $R = 0.38$.

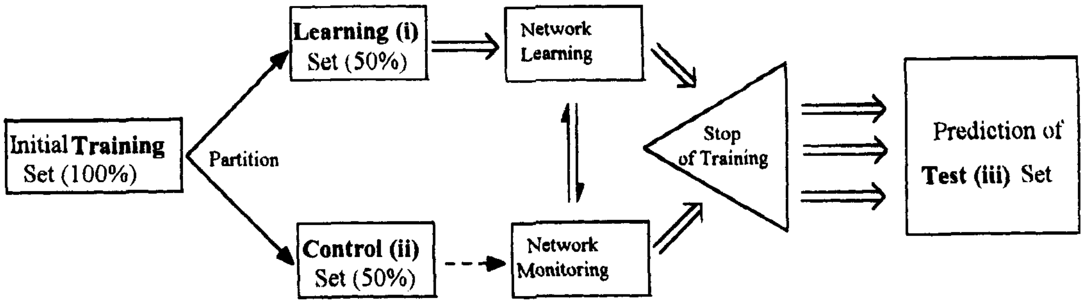

Figure 3. Partition procedure and process of neural network learning which was used in the work.

ANN Implementation. ANNs were programmed using the Borland C ++ 3.0 language. Only single hidden layer ANNs (with bias neurons) were used in the calculations. The networks were trained by the batch back propagation method using the SuperSAB algorithm.[16] This algorithm has been shown to converge faster than the simple descent technique. We used the recommended value for updating constants with upper limits of learning rate $\epsilon_{\max} = 10$.[17] The range of initial weights of the network was restricted to $[-3/\sqrt{N};3/\sqrt{N}]$, where N denotes the number of weights leading to a particular node. This initialization avoids weights being either highly active or inactive, allows the network to escape local minima, and speeds up the learning procedure.[18] The other details of ANN implementation can be found in ref 19.

Partitioning of ASDS for ANN Training. The following training/testing procedure was used. An initial training set was divided into two equal size subsets (see Figure 3):

(**i**) a subset actually used for network training or **learning** set;

(**ii**) a subset used to control the learning process or **control** set. The cases (representing molecules in a QSAR set) from

(**iii**) the **test** set was used to estimate the performance of a network and never participated in network training as a learning or control set.

One of the most important questions is how the initial training set should be subdivided into subsets (**i**) and (**ii**). The most simple way is to take samples into each set randomly. However, this procedure may be ineffectual, because of disproportional distribution of cases, if their number is small. The more logical partition of training set was used as described below.

Each case (molecule) represents a point in a space of input $\bar{x}$ and output $\bar{y}$ parameters. By partition of the training set and consequent cross-validation training we would like to determine the time when the network begins to remember noise. Suppose, that we train a network to interpolate the function shown in Figure 2. The best partition is achieved by taking from each two adjacent points one into the control and one into the learning set. The proximity of cases should be calculated in the space of both input and output variables. The dimensions of the input and output variables are different. We used as a measure of proximity the mean Euclidean distance calculated in the space of input and output parameters according to

$$d_{i,j} = \sqrt{|\bar{x}_i - \bar{x}_j|| + |\bar{y}_i - \bar{y}_j||} \qquad (8)$$

where, ||•| denotes Euclidean distance.

We found two cases from the training set with maximum $d_{i,j}$, took randomly one case into (**i**) and the other into (**ii**) and repeated the search amongst the remaining cases. The cases from the control set used in learning once and as control for the next training. A new partition was done before odd network training.

Training Protocol. Three points S_1, S_2, and S_3 were used to test ANN performance (see Figure 1). In all of these points ANN weights were saved in memory and then used to determine ANN performance.

•First point S_1 determines a best fit of network to the control set (**ii**) and corresponds to the minimum shown on Figure 1.

•Second point S_2 determines a best fit of network to union of sets (**i**) and (**ii**).

•The last one S_3 corresponds to error minimum of ANN on learning set (**i**) and as a rule matches to the end of a network training.

We used batch-training and recalculated all points after each reduction of the MSE of a network. Two criteria were used to stop network training. A network was not allowed to run more than $N_{\text{all}} = 10\,000$ epochs, or its learning was stopped after $N_{\text{stop}} = 2000$ epochs following the last improvement of any of the first two points.

One of the major problems which has a great influence on training of ANN is the matter of "local minima". This problem becomes more difficult when the number of hidden neurons in a network decreases. However, it is not known *a priori* if a network failed in a local minimum or if it reached its lowest state for this network architecture. This condition can be clarified only after the analysis of several network trainings using different initial weights. We used statistical averaging of network predictions as shown below and trained at least 200 networks having the same architecture for the same training and test sets but with different initial weights. The mean value $\text{M}\hat{\text{S}}\text{E}_3$ and standard deviation $\Delta\text{M}\hat{\text{S}}\text{E}_3$ were calculated for the S_3 stop points. The ith network was retained if

$$\text{M}\hat{\text{S}}\text{E}_3^i \geq \text{M}\hat{\text{S}}\text{E}_3 + 3\Delta\text{M}\hat{\text{S}}\text{E}_3 \qquad (9)$$

In such a way outliers (*i.e*, networks which have fallen into a local minimum) were excluded from further analysis. Actually, for the analyzed data sets outliers were absent for networks with numbers of hidden layer neurons greater than five.

The predicted values of cases from set (**i**), (**ii**), and (**iii**) were separately saved for three analyzed points $S_1 - S_3$ and were analyzed as shown below. Of course, both the learning and control sets were composed from the same cases, but the value that we kept for a case was calculated from the appropriate partition of the data.

Neural Network Ensembles. As was pointed out above there have been several reports indicating advantages of using ANN ensembles over single ANN prediction.[12,13] A rather large number of averaging methods is currently used to build the final network classifier from independent predictions of networks.[20] The simplest averaging over network predictions was used in this work

$$\bar{O}_i = \frac{1}{v}\sum_{l=1}^{v} O_{i,l} \qquad (10)$$

where $\bar{O}_i$ is the predicted value that was averaged over all $l = l,...,v$ prognostications of v analyzed ANNs.

It was very important in any calculations to determine the level of probability of computed results. As was mentioned before, we experienced difficulty in finding an analytical equation for estimation of the confidence interval for R, while for r^2 the following formula was derived

$$r_0^2(1 - \Delta r) < r^2 < r_0^2(1 + \Delta r) \qquad (11)$$

$$r_0^2 = \frac{\sum(Y_i - \bar{Y})^2 - \sum(Y_i - \bar{O}_i)^2}{\sum(Y_i - \bar{Y})^2} \qquad (12)$$

$$\Delta r - \frac{\sum\{2\Delta_i|Y_i - \bar{O}_i + \Delta_i^2\}}{\sum(Y_i - \bar{Y})^2 - \sum(Y_i - \bar{O}_i)^2} \qquad (13)$$

where $\bar{Y} = \frac{1}{K}\sum_{i=1}^{K} Y_i$. The summation in the eqs 11−13 is over all $i = l,...,K$ cases in the analyzed data set; Δ_i are the limits of the confidence interval for $\bar{O}_i$ values

$$\Delta_i = t_{\nu-1,\alpha/2}S_i/\sqrt{\nu}, \qquad (14)$$

where

$$S_i = \sqrt{\frac{1}{\nu-1}\sum_{l=1}^{\nu}(\bar{O}_i - O_{i,l})^2} \qquad (15)$$

Here α is a level of significance for the t-Student distribution with $(\nu - 1)$ degrees of freedom. The Student distribution for $\nu > 30$ is approximated by the U-distribution (normal)

$$\Delta_i \approx U_{\alpha/2}S_i/\sqrt{\nu} \qquad (16)$$

In our calculations about $\nu = 50$ estimations were carried out for each point, and, therefore, the normal approximation 16 was used to determine Δ_i. The confidence intervals for r^2 are proportional to the square root of the number ν of estimations $\Delta r \sim O(1/\sqrt{\nu})$, and r^2 can be calculated with required precision by increasing the number of calculations.

Calculation Results for ASDF. The ANNs used for study of linear, quadratic, and nonlinear data have the architecture of 4 × P × 1, 2 × P × 1, and 4 × P × 1 neurons, respectively, where the parameter P—the number of hidden layer neurones—is in a range from 1 to 40. These numbers do not include the extra bias neurone presented on the input and hidden layers. At least 100 calculations with initial random weights were done for each data file. Computed values were averaged and used to determine statistical coefficients both with training and test sets.

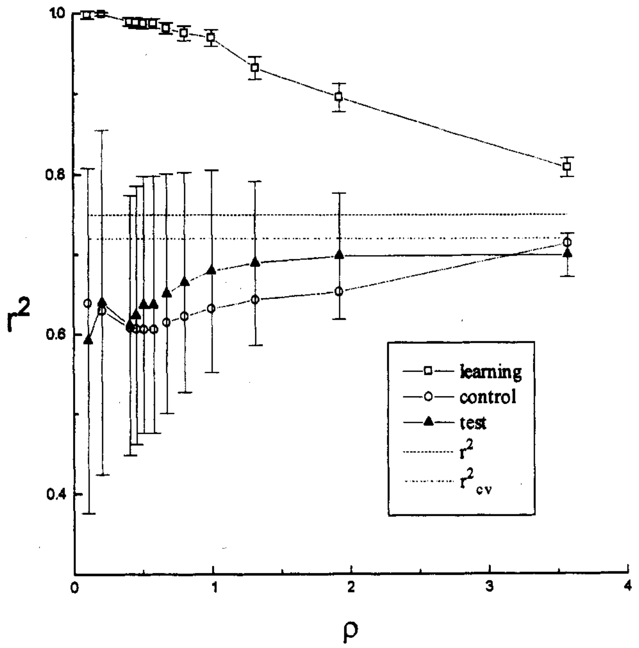

Figure 4. Plot showing the results of multiple linear regression using traditional statistics and neural networks for the linear ASDS. The cross-validated coefficient has been plotted against ϱ. ANNs training was stopped at point S_3 (see Figure 1), where overtraining of networks took place. The horizontal lines represent the results obtained using MLR. Upper line (r^2) shows cross-validated coefficient for recognition, while lower (r^2_{cv}) depicts it for LOO procedure. Overtrained ANNs shown perfect fit for the learning set (**i**) but failed to correctly predict data points from control (**ii**) and test (**iii**) sets. Error bars were calculated by eq 13. They show limits of r^2 for learning and test sets at the level of significance $\alpha < 0.05$.

Linear. Figure 4 illustrates results for point S_3 that correspond to complete learning of network. These results are similar to those of ref 9. As hidden layer units were added (i.e., ϱ decreases) the cross-validated r^2 gradually increased for the learning subset (**i**), while predictive performance for subsets (**ii**) and (**iii**), on the other hand, decreased. This is a distinct example of overtraining and overfitting influence on the predictive ability of a network. These investigations led to conclusions about the importance of using networks with lower numbers of neurons in the hidden layer.[9]

Rather different behavior is depicted in Figures 5 and 6 where the optimal stop point of network training was used (points S_1 and S_2). Increasing the number of neurons in the hidden layer does not have any influence on the predictive performance of ANN. The predictive performance of neural networks coincides with that calculated by MLR LOO method.

Quadratic. Figure 7 illustrates results calculated for quadratic data. Stopping of ANN training in points S_1 and S_2 allowed significant increase of ANN predictive performance for test data sets in comparison to that of overtrained networks from point S_3. The cross validated coefficient r^2 calculated by ANNs coincides with that of MLR using a quadratic term. This indicates that the neural network discovered the quadratic nature of the investigated data points.

Nonlinear. At first, our attempts to achieve any suitable prediction for nonlinear data points failed. ANNs showed very bad results for all three stop points, while they showed perfect r^2 coefficients for recognition (see Table 1). An

Table 1. Neural Network Results for Nonlinear Example[a]

data	R^G (learning set)	R			r^2 (test set)		
set size	S_1[c]	S_1	S_2	S_3	S_1	S_2	S_3
100	0.991	−0.03	−0.05	−0.05	-0.4 ± 0.2[d]	-0.8 ± 0.3	-1.2 ± 0.8
200	0.973	0.87	0.85	0.79	0.61 ± 0.1	0.62 ± 0.1	0.60 ± 0.2
400	0.932	0.88	0.88	0.85	0.71 ± 0.06	0.72 ± 0.07	0.69 ± 0.1

[a] Networks had 10 neurons in the hidden layer, extra bias neuron was presented in the input and hidden layer; 100 calculations with random initial weights were made to compute final network predictions. [b] R correlation coefficient, r^2-cross-validated coefficient (see eq 5). [c] S_1−S_3 depict points used to stop network learning (see Figure 1). [d] Limits were calculated at the level of significance $\alpha < 0.05$ by eq 13.

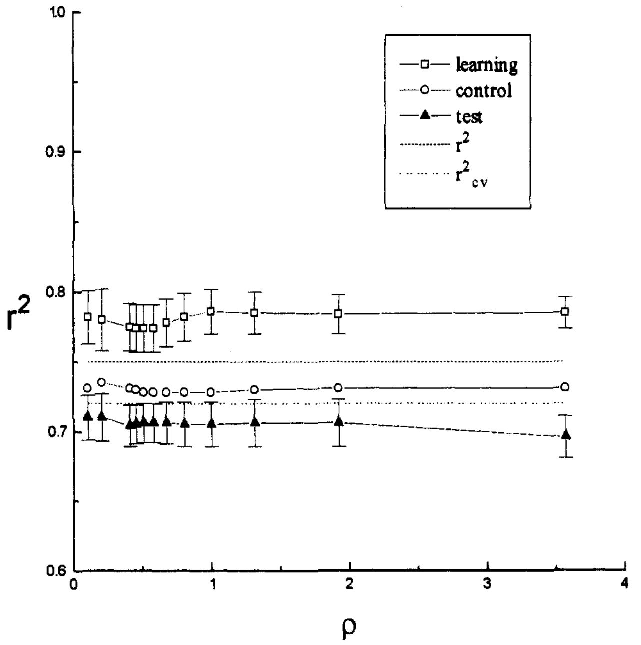

Figure 5. Plot showing the results of multiple linear regression using traditional statistics and neural networks for the linear ASDS. The cross-validated coefficient has been plotted against ϱ. ANNs training was stopped at point S_1 (see Figure 1). The horizontal lines represent the results obtained using MLR. Upper line (r^2) shows cross-validated coefficient for recognition, while lower (r^2_{cv}) depicts it for LOO procedure. ANNs show both perfect fits for learning set (**i**) and rather good prediction of data points from control (**ii**) and test (**iii**) sets. Error bars were calculated by eq 13. The shown limits of r^2 for learning and test sets at the level of significance $\alpha < 0.05$.

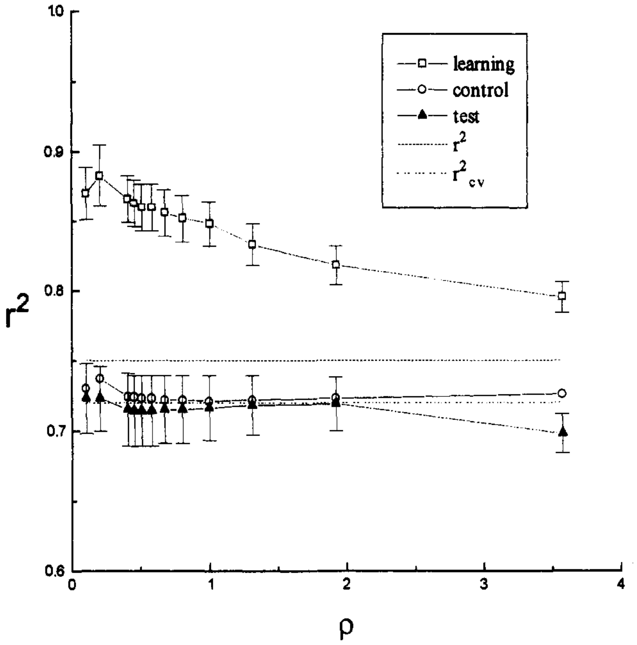

Figure 6. Plot showing the results of multiple linear regression using traditional statistics and neural network for the linear ASDS. The cross-validated coefficient has been plotted against ϱ. ANNs training was stopped at point S_2 (see Figure 1). The horizontal lines represent the results obtained using MLR. Upper line (r^2) shows correlation coefficient for recognition, while lower (r^2_{cv}) depicts it for LOO procedure. ANNs show both perfect fits for learning set (**i**) and rather good prediction of data points from control (**ii**) and test (**iii**) sets. Error bars were calculated by eq 13. The shown limits of r^2 for learning and test sets at the level of significance $\alpha < 0.05$.

average correlation coefficient was above 0 for the test data set (**iii**). We assumed that the number of cases (*i.e.*, the amount of information provided for network learning) was not adequate to train ANN. The whole number of analyzed cases was gradually increased to 200 and 400. Only ANNs with 10 neurons in the hidden layer were used in this study. The results shown in Table 1 illustrate that the predictive ability of neural networks gradually increased proportionally to the number of points used. The cross-validated coefficient calculated for 400 data point training sets is rather good if we take into account the complexity of the analyzed relationship.

EXPERIMENTS WITH REAL QSAR DATA

QSAR Data Sets. The same real QSAR data sets as in ref 9 were used. These four examples contain variously structured data sets:

1. linear[21]
2. linear with indicator variable[22]
3. quadratic[23]
4. quadratic with indicator variable[22]

The terms linear and quadratic imply the form of the observed relationship between the parameters of the molecules and their activity. Indicator means that a binary variable, generally coding discrete changes in structure, was included in the calculated relationship. The linear data set involved the description of octanol/water log P values in terms of calculated polarizability, dipole moment, and energy of the highest occupied molecular orbital. The linear with indicator set and the quadratic with indicator set concerned the inhibition of *L. casei* dehydrofolate reductase by triazines, while the quadratic set was made up of antitumor 4′-(9-acridinylamino)methanesulfonanilide analogues.

Calculation Results for Real QSAR Data. ANNs with 10 (plus extra bias) neurons in the hidden layer were used for these calculations. It was found that usually less than 200 epochs were enough to find S_1 and S_2 points for all QSAR data sets. That is why a lower number of epochs $N_{all} = 2000$ ($N_{stop} = 400$) was used. The indicator variables and squared terms were not provided as inputs to network training for data sets 2−4. The LOO cross-validation procedure was used to supervise the predictive performance

Table 2. Comparison of MLR and ANN methods on Real QSAR Data Sets[a]

	MLR, r^2		ANN, r^2, LOO			R, LOO	
data set	learning	LOO	S_1[b]	S_2	S_3	MLR	ANN S_2
linear (37)[21]	0.847	0.781	0.763 ± 0.02[c]	0.775 ± 0.02	0.709 ± 0.04	0.885	0.881
linear plus indicator (38)[22]	0.931 (0.835)[d]	0.916 (0.805)	0.903 ± 0.005	0.904 ± 0.005	0.894 ± 0.01	0.957 (0.898)	0.948
quadratic (50)[23]	0.821 (0.709)	0.788 (0.661)	0.747 ± 0.02	0.751 ± 0.03	0.740 ± 0.04	0.888 (0.814)	0.872
quadratic plus indicator (34)[21]	0.837 (0.430)	0.779 (0.246)	0.715 ± 0.03	0.724 ± 0.02	0.691 ± 0.04	0.884 (0.529)	0.856

[a] R = correlation coefficient, r^2 = cross-validated coefficient (eq 5), MLR = multiple linear regression, ANN = artificial neural network, LOO = "leave-one-out". Quadratic and indicator terms were not provided to train networks. ANN had 10 neurons in the hidden layer. The extra bias neuron was presented on the input and the hidden layer. Two hundred calculations with random initial weights were made to compute final network predictions. [b] S_1-S_3 depict points used to stop network learning (see Figure 1). [c] Limits were calculated at the level of significance $\alpha <$ 0.05 by eq 13. [d] Regression results without indicator variable or/and quadratic term are indicated in parentheses.

of ANN. The molecules were randomly partitioned for learning (i) and control sets (ii) each time before network learning was indicated for ASDS. Each molecule from control set (ii) was virtually removed in test set and stop points S_1-S_3 were determined especially for it. Such a procedure considerably increased the speed of LOO. Actually, we need only about $2n$ network trainings to calculate n predictions for each molecule from the data set. Two hundred independent ANN trainings were done. The cross-validated coefficients between averaged values and experimental activities of the molecules are shown in Table 2. ANN predictions are similar to LOO MLR results, despite indicator variables and squared terms being left out of the input. Rather high coefficients were calculated for the S_3 point. The possible explanation of this is that ANN did not reach the overtraining region because of the restricted number of calculations $N_{\text{all}} = 2000$.

DISCUSSION

The calculated results for the ASDS examples indicate that statistical coefficients calculated without cross-validation should not be used to estimate the predictive ability of ANNs.

The real examples indicate the importance of avoiding ANN overtraining and suggest a rather wide tolerance of ANN to the number of the hidden layer neurons (in other words to the overfitting problem). It is easy to determine from several initial trainings an approximate range of numbers of neurons in the hidden layer and to use this number of neurons for further calculations. Such determination of the number of hidden layer neurons is very important to avoid the problem of "local minima". The frequency of networks falling into these minima is inversely proportional to the network degrees of freedom. This problem does not occur for "sufficiently" large networks which particular size for a given task can be easily determined using several probe neural network trainings.

Some remarks should be made on the problem of global minima. We do not need a global minimum for the learning subset, *i.e.*, a "recognition minimum", and did not try to locate it in our calculations. Finding it will not improve network prediction (*i.e.*, on the test set), because of the overtraining problem. All our efforts were made to find "prediction minima" that were discovered by analysis of the control (point S_1) or joint (point S_2) sets. However, it is not convenient to keep a number of coefficients for control, joint, and learning sets (*i.e.*, the r^2 coefficients for the joint set at S_1, S_2 and S_3 points) for comparison of network performances in the analyzed points. The more convenient way is to use the coefficients calculated by LOO, at it was done for real QSAR examples. Besides that, for LOO estimations there is no danger that an analyzed case from control or joint sets can influence the stop point determinations, *i.e.*, supplies a network in such a way some information about itself.[24] Would we check, if the utilized network parameters (*i.e.*, architecture of ANN, number of epochs for the training algorithm) were adequate to locate the global minima at S_1 and S_2 points? This question can be answered by comparison of LOO coefficients at these points with those at S_3. The used network parameters ***should be adequate to permit ANN to overtrain*** at the S_3 point, *i.e.*, the calculated LOO coefficients at the S_3 point should be ***statistically lower*** than those at S_1 and S_2 points. We would like to illustrate this reasoning by the next examples.

Example 1: (a) The restricted number of epochs $N_{\text{all}} = 50$ ($N_{\text{stop}} = 10$) was used for the linear example (networks had 10 neurons in the hidden layer). The LOO r^2 coefficients were the same $r^2 = 0.66 \pm 0.03$ at all points S_1-S_3, *i.e.*, the used number of epochs was not adequate to overtrain ANN at S_3 point and, probably, to find global minima for S_1 and S_2 points. Indeed, the LOO coefficients at S_1 and S_2 points were statistically lower than those found with $N_{\text{all}} = 2000$, *i.e.*, ANN did not locate proper global minima for $N_{\text{all}} = 50$ epochs.

Example 2: (b) The ANN with a single hidden neuron calculated the same $r^2 = 0.76 \pm 0.02$ by LOO at points S_1-S_3 (see Figure 7). This network architecture was insufficient to overtrain ANN, and, probably, we have not calculated the best prediction at point S_1-S_2. Indeed, the coefficients at points S_1 and S_2 were higher for networks with a greater number of hidden layer neurons. The r^2 calculated for networks with numbers of hidden neurons > 3 at S_1 and S_2 points were statistically indistinguishable and superior to those at the S_3 point. This means, that ANNs with these numbers of hidden layer neurons were adequate to learn correctly the quadratic example for the used algorithm and used number of epochs. Maybe, if another learning algorithm was used, the other numbers of hidden layers would be required.

Therefore, it is very important to overtrain a network to point S_3 in order to be sure that the used network parameters were adequate to solve the analyzed task properly.

The prediction performance of ANNs from points S_1 and S_2 are rather similar (with some preference to S_2). This means that actually all networks from point S_1 to S_2 (100–200 epochs) may be taken to avoid overtraining of networks.

The correctness of the partition procedure, that was used in our calculation, depends on the quality of the used input parameters. It can incorrectly portion points from the

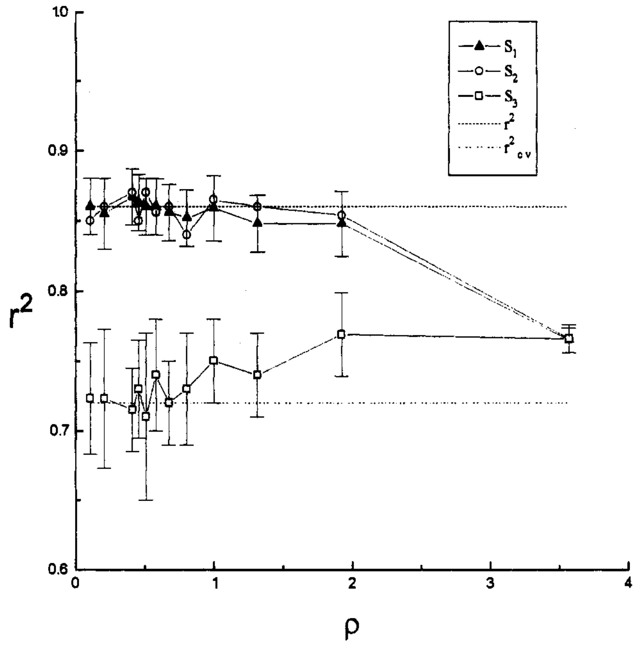

Figure 7. Plot showing the results of multiple linear regression using traditional statistics and neural networks fro the quadratic ASDS. The correlation coefficient for the test data set (iii) has been plotted against ϱ. The quadratic term was not included for network training. The horizontal lines represent the results obtained using LOO MLR with (upper line) and without the quadratic (lower line) term. ANNs stopped at points S_1 and S_2 (see Figure 1) show perfect prediction, while that of the overtrained network (point S_3) was significantly lower. Error bars were calculated by eq 13. The shown limits of r^2 for stop points S_1 and S_3 at the level of significance $\alpha < 0.05$.

training set, if there are many noisy input parameters. However, there are several ANN pruning algorithms that can be used for determining the most important parameters during network learning.[25-28]

A very promising method for data partition is the Kohonen self-organizing map (SOM).[29] This kind of ANN allows one to create a mapping of multidimensional information onto a layer of neurons that preserves the essential content of the information (relationships). An application of a Kohonen network to the analysis of chemical reactivity allowed the authors to form correctly a learning set and to achieve results superior to other investigated techniques.[30] SOM may also allow one to sufficiently decrease the size of a learning set. It is especially useful for training networks with a large number of data points, when use of a 50/50 partition for learning/control set may result in very slow network training. However, Kohonen networks so far were used for classification on classes and some work should be done to update this encouraging method for regression analysis.

Previous results have indicated that the number of neurons in a hidden layer should be kept as small as possible.[5-9] Indeed, if we look at Figures 4−6 for the linear example, it may be seen that networks with only one neuron in the hidden layer show rather good predictions of test data for all points S_1-S_3. The networks with only one neuron in the hidden layer appear incapable of overtraining.

The quadratic data example illustrates that a network with a single hidden layer neuron was also incapable of overtraining (see Figure 7). However, predictions from this network were worse in comparison with those of nonovertrained networks having greater numbers of hidden layer neurons. This pattern illustrates that the strategy of using ANNs with the smallest number of hidden layer neurons sometimes does not provide the best solutions. Nevertheless, even such a scheme sometimes allowed the calculation of results superior or similar to those of traditional methods.[31]

A very interesting interpretation can be made for the nonlinear ASDS results. Usually, when examining new nonlinear phenomena we do not know exactly if we have only a random number of points or really some complex relationship is presented in the data. ANNs seem to be a suitable tool to find such relationship or give a competent conclusion that we work with noise. The example clearly shows that increasing the presented information can dramatically change the "opinion" of neural networks about investigated data. At first, networks failed to find any relationship (see Table 1), but after an increase of the number of points in the data set, they showed a rather good prediction of the test data set. This property of networks is unique and should be investigated more carefully in the future.

The last notion concerns eq 13, calculation of the confidence limits Δr of the cross-validated r^2 coefficient. We have used a rather crude estimation of Δr. Actually, the Δ_i terms are not correlated. They compensate one another for real word examples and diminish the confidence limits Δr. More correct estimations should be derived for Δr.

CONCLUSIONS

The calculated results have shown the advantage of ANN over MLR analysis. ANNs do not require high order terms or indicator variables to establish complex structure−activity relationships. The overfitting problems do not have any influence on ANN predictive ability when overtraining is avoided by cross-validation.

Supporting Information Available: The derivation of eq 13 (the confidence interval of r^2) (2 pages). Ordering information is given on any current masthead page.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their helpful remarks.

REFERENCES AND NOTES

(1) Gasteiger, J.; Zupan, J. Neural Networks in Chemistry. *Angew. Chem., Int. Ed. Engl.* **1993**, *105*, 503−527.

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

(3) Kolmogorov, A. N. On the Representations of Continuous Functions of Many Variables by Superposition of Continuous Functions of One Variable and Addition. *Dokl. Akad. Nauk USSR* **1957**, *114*, 953−956.

(4) Hecht-Nielsen, R. Kolmogorov's Mapping Neural Network Existence Theorem. *Proceedings of the International Conference on Neural Networks*; IEEE Press: New York, 1987; pp 11−14.

(5) Andrea, T. A.; Kalayeh, H. Applications of Neural Networks in Quantitative Structure-Activity Relationships of Dihydrofolate Reductase Inhibitors. *J. Med. Chem.* **1990**, *33*, 2583−2590.

(6) Manallack, D.; Livingstone, D. J. Artifical Neural Networks: Application and Chance Effects for QSAR Data Analysis. *Med. Chem. Res.* **1992**, *2*, 181−190.

(7) Livingstone, D. J.; Salt, D. W. Regression Analysis for QSAR Using Neural Networks. *Bioorg. Med. Chem. Let.* **1992**, *2*, 213−218.

(8) Livingstone, D. J.; Manallack, D. T. Statistics Using Neural Networks: Chance Effects. *J. Med. Chem.* **1993**, *36*, 1295−1297.

(9) Manallack, D. T.; Ellis, D. D.; Livingstone, D. J. Analysis of Linear and Non-Linear QSAR Data Using Neural Networks. *J. Med. Chem.* **1994**, *37*, 3758–3767.

(10) Thodberg, H. H. A Review of Bayesian Neural Networks with an Application to Near Infrared Spectroscopy. *IEEE Trans. Neural Networks*, in press.

(11) Borggaard, C.; Thodberg, H. H. Optimal Minimal Neural Interpretation of Spectra. *Anal. Chem.* **1992**, *64*, 545–551.

(12) Hansen, L. K.; Salamon, P. Neural Networks Ensembles. *IEEE Trans. Pattern Anal. Machine Intell.* **1990**, *12*, 993–1001.

(13) Perrone, M. P. General Averaging Results for Convex Optimization. In *Proceedings of the 1993 Connectionist Models Summer School*; Erlbaum Associates: Hillsdale, NJ, 1994; pp 364–371.

(14) Tetko, I. V.; Luik, A. I., Poda, G. I. Application of Neural Networks in Structure-Activity Relationships of a Small Number of Molecules. *J. Med. Chem.* **1993**, *36*, 811–814.

(15) Cramer, R. D. III; Patterson, D. E.; Bunce, J. D. Comparative Molecular Field Analysis (CoMFA). 1. Effect of Shape on Binding of Steroids to Carrier Proteins. *J. Am. Chem. Soc.* **1988**, *110*, 5959–5967.

(16) Tollenaere, T. SuperSAB: Fast Adaptive Back Propagation with Good Scaling Properties. *Neural Networks* **1990**, *3*, 561–573.

(17) Schiffmann, W.; Joost, M.; Werner, R. Comparison of Optimized Backpropagation Algorithm. *Proceedings of the European Symposium on Artificial Neural Networks*; Verleysen, M., Ed.; Brussels, 1993; pp 97–104.

(18) Wessels, L. F. A.; Barnard, E. Avoiding False Local Minima by Proper Initialization of Connections. *IEEE Trans. Neural Networks* **1992**, *3*, 899–905.

(19) Tetko, I. V. Ph.D. Disseration, Application of Neural Networks in Structure-Activity Relationships Studies. Institute of Bioorganic & Petroleum Chemistry: Kiev, 1994; p 154.

(20) Hashem, S.; Schmeiser, B.; Yih, Y. Optimal Linear Combinations of Neural Networks: An Overview. *Proceedings of the 1994 IEEE (WCCI) International Conference on Neural Networks.* IEEE Press: 1994; Vol. 3, pp 1507–1512.

(21) Lewis, D. F. V. The Calculation of Molar Polarizabilities by the CNDO/2 Method: Correlation with the Hydrophobic Parameter, LogP. *J. Comput. Chem.* **1989**, *10*, 145–151.

(22) Coats, E. A.; Genther, C. S.; Dietrich, S. W.; Guo, Z.-r.; Hansch, C. Comparison of the Inhibition of Methotrexate-Sensitive and -Resistant *Lactobacillus casei* Cell Cultures with Purified *Lactobacillus casei* Dihydrofolate Reductase by 4,6-Diamino-1,2-dihydro-2,2-dimethyl-1-(3-substituted-phenyl)-s-triazines. Use of Quantitative Structure-Activity Relationships in Making Inferences about the Mechanism of Resistance and the Structure of the Enzyme in Situ Compared with the Enzyme in Vitro. *J. Med. Chem.* **1981**, *24*, 1422–1429.

(23) Denny, W. A.; Atwell, G. J.; Cain, B. F. Potential Antitumor Agents. 32. Role of Agent Base Strength in the Quantitative Structure-Antitumor Relationships for 4′-(9-Acridinylamino) Methanesulfonanilide Analogues. *J. Med. Chem.* **1979**, *22*, 1453–1460.

(24) Such danger does exist. For example, the r^2 coefficients calculated for control set in point S_1 were always slightly higher than LOO estimations. The difference was caused by aforementioned influences of molecules from the control set on determination of S_1 stop point.

(25) Hassibi, B.; Stork, D. Second Order Derivatives for Network Pruning: Optimal Brain Surgeon. *Advances in Neural Information Processing Systems 5*; Hanson, S., Cowan, J., Giles, C., Eds.; Morgan Kaufmann Publishers: San Mateo, CA, 1993; pp 164–171.

(26) Cun, Y. L.; Denker, J.; Solla, S. Optimal Brain Damage. *Advances in Neural Information Processing Systems 2*; Touretzky, D., Eds.; Morgan Kaufmann Publishers: San Mateo, CA, 1990; pp 598-605.

(27) Tetko, I. V.; Tanchuk, V. Yu.; Luik, A. I. Simple Heuristic Methods for Input Parameters' Estimation in Neural Networks. *Proceedings of the 1994 IEEE (WCCI) International Conference on Neural Networks*; IEEE Press: 1994; Vol. 1, pp 376–381.

(28) Wikel, J. H.; Dow, E. R. The Use of Neural Networks for Variable Selection in QSAR. *BioMed. Chem. Lett.* **1993**, *3*, 645–651.

(29) Kohonen, T. Self-Organization and Associate Memory. Springer-Verlag: Berlin, Heidelberg, New York, Tokyo, 1988.

(30) Simon, V.; Gasteiger, J.; Zupan, J. A. Combined Application of Two Different Neural Network Types for the Prediction of Chemical Reactivity. *J. Am. Chem. Soc.* **1993**, *115*, 9148–9159.

(31) Tetko, I. V.; Tanchuk, V. Yu.; Chentsova, N. P.; Antonenko, S. V.; Poda, G. I.; Kukhar, V. P.; Luik, A. I. HIV-1 Reverse Transcriptase Inhibitor Design Using Artifical Neural Networks. *J. Med. Chem.* **1994**, *37*, 2520–2516.

CI9502008