# Neural Network Studies. 2. Variable Selection

Igor V. Tetko,*,†,‡ Alessandro E. P. Villa,‡ and David J. Livingstone§

Institute of Bioorganic and Petroleum Chemistry, Ukrainian Academy of Sciences, Murmanskaya, 1, Kiev-660, 253660, Ukraine, Laboratoire de Neuro-Heuristique, Institut de Physiologie, Faculté de Médicine, Université de Lausanne, Rue du Bugnon 7, Lausanne, CH-1005, Switzerland, ChemQuest, Cheyney House, 19-21 Cheyney Street, Steeple Morden, Herts, SG8 0LP, UK, and Centre for Molecular Design, University of Portsmouth, Portsmouth, Hants, PO1 2EG, UK

Received June 9, 1995⊗

Quantitative structure−activity relationship (QSAR) studies usually require an estimation of the relevance of a very large set of initial variables. Determination of the most important variables allows theoretically a better generalization by all pattern recognition methods. This study introduces and investigates five pruning algorithms designed to estimate the importance of input variables in feed-forward artificial neural network trained by back propagation algorithm (ANN) applications and to prune nonrelevant ones in a statistically reliable way. The analyzed algorithms performed similar variable estimations for simulated data sets, but differences were detected for real QSAR examples. Improvement of ANN prediction ability was shown after the pruning of redundant input variables. The statistical coefficients computed by ANNs for QSAR examples were better than those of multiple linear regression. Restrictions of the proposed algorithms and the potential use of ANNs are discussed.

## INTRODUCTION

Early QSAR investigations used a rather limited number of physicochemical descriptors (*e.g.*, lipophilicity, Taft's constants, *etc*.), while in recent years a great number of new descriptors (e.g., topological indexes, embedding frequencies, properties from computational chemistry,[1] *etc*.) have been defined. The basic question raised by using QSAR analysis is how to determine the relevant variables that satisfactorily represent dependencies between the activity and descriptors. Determination of the most important variables allows theoretically a better generalization by all pattern recognition methods.[2,3]

Traditional methods used in QSAR studies, such as multiple linear regression (MLR), provide to the researcher a tool to reduce nonrelevant variables.[2] This problem has been less investigated for ANN and will be analyzed here in detail. However, a fundamental question should be raised: "What is the meaning of the best set of variables for ANN regression?"

It is known from the literature that an increasing number of hidden layer neurons in ANN permits it to remember *an arbitrary pattern* with a given accuracy.[4−7] Our own calculations also suggest that statistical coefficients calculated for a learning set without cross-validation should not be used as a measure of ANN predictive ability and thus used to judge the quality of a test variable set.[8] Hence, in this study only the statistical coefficients calculated by the cross-validation leave-one-out method (LOO) are used to compare the performances of variable sets.

Let us consider some possible algorithms for determination of the most relevant variables (Figure 1). Suppose that we would like to determine the best subset of a set containing *n* variables. There are several potential ways: (i) a complete analysis of all subsets; (ii) a genetic algorithm,[9] evolutionary programming;[10−13] (iii) an heuristic stepwise regression analysis; and (iv) direct estimations (pruning methods). A special case of the (iii) algorithms is a consequent analysis, that is a stepwise estimation of all subsets with $(n - 1)$ variables and usage for the next step of the analysis only the most pertinent subset.

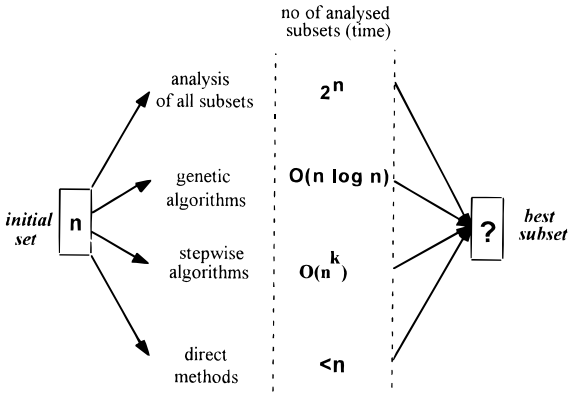


**Figure 1.** The possible approaches to find the best subset of parameters.

The first three methods are inappropriate for ANN which represents a rather slow computational algorithm that becomes even more time consuming whenever it is attempted to avoid possible chance correlation due to overtraining and overfitting.[8] Generally, only direct estimation methods are used by the ANN researcher, except in the case of very small networks and few analyzed patterns. An evaluation of a variable by such methods is done by introducing a sensitivity term for a variable. The sensitivities of all variables are calculated and the less sensitive variables are pruned. Selection of variables by such methods in QSAR studies was pioneered by Wikel and Dow.[14] In one of their models, a data set of 31 compounds and 53 descriptors was used to identify the most relevant properties of the dependent variable. The pertinence of input variables was estimated

† Ukrainian Academy of Sciences. e-mail: tetko@bioorganic.kiev.ua.
‡ Université de Lausanne.
§ ChemQuest and University of Portsmouth.
⊗ Abstract published in *Advance ACS Abstracts,* February 15, 1996.

S0095-2338(95)00204-6 CCC: $12.00 

by the weight values of the connection between input and hidden layers. Color maps were used to indicate the magnitude of the hidden weights, and the largest values were identified visually as the most important descriptors. MLR analysis was then applied to the data set using the descriptors identified by ANN. Two of these properties characterized by high weights were ultimately incorporated into the final equation. Despite the fact that the variables selected by ANN were not used to build an ANN classifier, this experiment indicates the possibility of selection of variables by an analysis of weight magnitudes. Overtraining a neural network may produce incorrect results. The authors limited this risk by stopping the network training after some fixed number of epochs, while poor generalization ability of the method after a large number of epochs was indicated.[15] The overtraining was characterized by large weight values of nearly all of the used variables, *i.e.*, the network started to "memorize" data with noise, and it was impossible to identify the most relevant variables. However, it was pointed out[16] that this approach did not provide a panacea to avoid chance effects; hence, it was possible that apparently important descriptors were chosen by chance.

Recently, new approaches that are based on statistical averaging over independent network predictions were proposed.[17−19] Criteria for the optimal stopping of network training were found, and a method for avoiding the overtraining of ANNs was elaborated.[8] This algorithm calculates statistically reliable predictions of ANNs and avoids chance correlation of ANNs for a wide range of a number of neurons in the hidden layers.

This study is intended to apply previously developed methodology[8] for input variable selection by ANNs in a statistically reliable way. Several pruning algorithms adapted for determination of the most relevant variables by ANNs are proposed and compared. Despite the fact that most of these algorithms were developed from previously elaborated weight pruning methods their use for input variable estimations is demonstrated here for the first time. It is shown, that the optimization of a set of input variables significantly improves the prediction ability of ANNs.

## ANN IMPLEMENTATION

The details of the ANN program can be found in refs 8 and 20. ANNs with 10 neurons in one hidden layer were used in the calculations (Figure 2). The bias neuron was presented on the input and on the hidden layer. At least $M = 100$ independent ANN were trained to analyze each set of variables. The predicted values of each analyzed case were averaged over all $M$ ANN predictions, and the means were used to calculate statistical coefficients with targets.

## ANN OVERTRAINING

Wikel and Dow[14] pointed out that the overtraining of neural networks resulted in rather poor discrimination between the most relevant and irrelevant variables, and it should be avoided in order to achieve correct pruning of unnecessary inputs. The overtraining/overfitting of ANNs was analyzed in detail recently. We have shown that overfitting does not have any influence on network prediction ability when overtraining is avoided by cross-validation. Application of ANN ensembles (ANNE) allowed the avoidance of chance correlation, and satisfactory predictions of

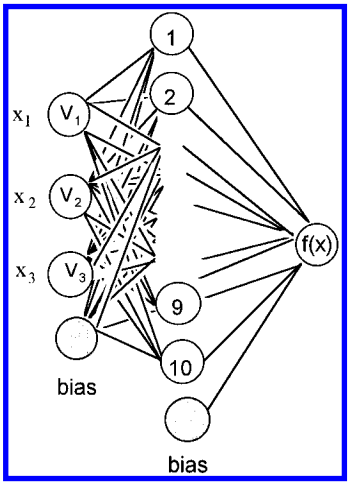


**Figure 2.** Topology of an artificial neural network used to learn the examples of Artificial Structured Data Sets (ASDS).

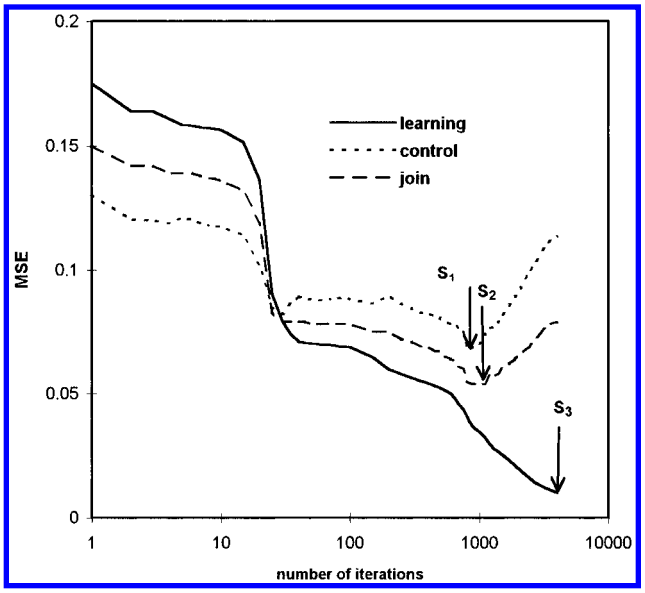


**Figure 3.** Mean square error (MSE) as a function of the number of training epochs for linear ASDS. The number of hidden units is five. Arrows depict stop points of ANN training and correspond to the MSE error minimum of network for control set (point $S_1$), control plus learning sets (point $S_2$), and learning set (point $S_3$). The last point $S_3$ frequently coincides with the end of network training.

new data were obtained for a wide range of numbers of neurons in the hidden layer.[8] The same approach will be used in this work too.

Let us outline some general principles of the elaborated method. We used a subdivision of the initial training set into two equal learning/control subsets.[8] The first set was used by the ANN to learn while the second one was used to monitor the training. The mean square error (MSE)[21] was used as a criterion of network training. Figure 3 illustrates how the termination of ANN learning between point $S_1$, corresponding to the best fit of a network according to the control set, and $S_2$, corresponding to a best fit of the network to the initial training set, could avoid the overtraining problem and improve prediction ability of the ANNE. This technique is referred to as an "early stopping" method.[22,23] The statistical coefficients calculated by the leave-one-out (LOO) method for point $S_1$ were used as a measure of the predictive ability of the networks. An additional point $S_3$ was analyzed in our study. It corresponds to an MSE

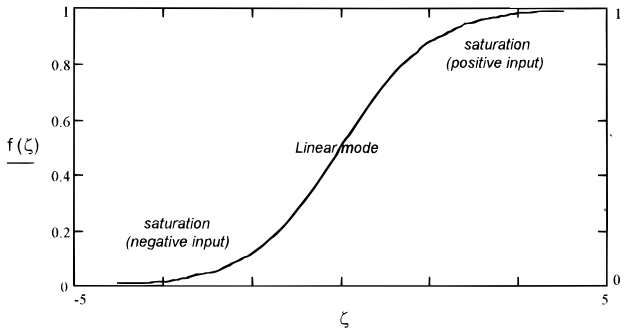


**Figure 4.** Modes of an activation function $f(\zeta) = 1/1 + e^{-\xi}$ in dependence from the input $\zeta = \sum_{iwif} \cdot a_i$ of lower layer nodes. Here $a_i$ is value of a neuron $i$ on the previous layer.

minimum of ANN on the learning set and as a rule matches to the end of a network training, where ANN overtraining takes place. The $S_3$ point was used to investigate the influence of overtraining on variable selection and to prevent the network falling into "local minima". The termination of network training consisted in limiting the network run to $N_{all} = 10\,000$ epochs or to stop its learning after $N_{stop} = 2000$ epochs following the last improvement of MSE in either of the $S_1$ or $S_2$ points (see details in ref 8).

## ANALYZED PRUNING METHODS

Pruning methods take an important place in the ANN literature. Two main groups of methods can be identified: (1) sensitivity methods and (2) penalty term methods.

The sensitivity methods introduce some measures of importance of weights by so called "sensitivities". The sensitivities of all weights are calculated, and the elements with the smallest sensitivities are deleted.[24−26] The second group modify the error function by introducing penalty terms.[27,28] These terms drive some weights to zero during training. A good overview of the literature of these and some other types of pruning methods can be found in ref 29.

Usually, pruning methods were used to eliminate redundant weights or to optimize internal architecture—number of hidden layers and number of neurons on the hidden layers[30,31]—of ANN. Occasionally, some conclusions about the relevance of input variables were made, if all weights from some input neurons were deleted. Authors did not try to focus their analysis on the determination of some relevant set of variables, because, it was impossible to do when pruning was concentrated at the level of a single weight. However, the weight sensitivities from the first group of methods can be easily converted to neuron sensitivities.

We investigate here five pruning methods **A**−**E**. A sensitivity $S_i$ of input variable $i$ is introduced by

$$S_i = \sum_{k \epsilon \Omega_i} s_k \equiv \sum_{j=1}^{n_j} s_{ji} \tag{1}$$

where $s_k$ is a sensitivity of a weight $w_k$, and summation is over a set $\Omega_\kappa$ of outgoing weights of the neuron $i$ or, using another order of weight numeration, $s_{ji}$ is a sensitivity of a weight $w_{ji}$ connecting the $i$th neuron to the $j$th neuron in the next layer. The method **B** was developed by us previously,[32,33] while in the other methods an estimation of weight sensitivity was updated from the appropriate references.

The sensitivity of a neuron of ANN in the first two methods is based on direct analysis of the magnitudes of its outgoing weights ("magnitude based" methods). Conversely, in the last three methods, the sensitivity is approximated by a change of the network error due to the elimination of some neuron weights ("error-based" methods).

(A) The first method was inspired by ref 14. The generation of color maps is useful for visual determination of the most important input variables, but it is rather subjective. This technique cannot be used, however, for automatic processing of large amounts of data. Therefore, the absolute magnitude of a weight $w_k$

$$s_k = |w_k| \tag{2}$$

was used as its sensitivity in eq 1. In such a way the most appropriate relation between ref 14, and our model is achieved. Sensitivities of neurons calculated accordingly were successfully used to prune unimportant neurons responsible for four basic tastes (see ref 34 for details).

(B) The preliminary version of the second approach was used to predict activity of anti AIDS substances[33] according to

$$S_i = \sum_{j=1}^{n_j} \left( \frac{w_{ji}}{\max_a |w_{ja}|} \right)^2 \tag{3}$$

and $\max_a$ is taken over all weights ending at neuron $j$. The rationale of this equation is that different neurons in an upper layer can work in different modes (Figure 4). Thus, a neuron working near saturation has bigger input values in comparison with one working in linear mode. An input neuron connected principally to the linear mode neurons on the hidden layer is always characterized by smaller sensitivities when compared with a neuron primarily connected to the saturated ones. The sensitivity calculated by (3) eliminates this drawback. This method was improved to take into account the sensitivity of neurons in the upper layers

$$S_i = \sum_{j=1}^{n_j} \left( \frac{w_{ji}}{\max_a |w_{ja}|} \right)^2 \cdot S_j \tag{4}$$

where, $S_j$ is a sensitivity of the $j$th neuron in the upper layer. This is a recurrent formula calculating the sensitivity of a unit on layer $s$ via the neuron sensitivities on layer $s + 1$. The sensitivities of output layer neurons are set to 1. All sensitivities in a layer are usually normalized to a maximum value of 1. This type of sensitivity calculation was especially useful in the case of simultaneous analysis of several output patterns.

(C) Let us consider an approximation of the error function $E$ of an ANN by a Taylor series. When the weight vector **W** is perturbed, the change in the error is approximately

$$\delta E = \sum_i g_i \delta w_i + \frac{1}{2} \sum_i h_{ii} \delta w_i^2 + \frac{1}{2} \sum_{i \neq j} h_{ij} \delta w_i \delta w_j + O(||\delta \mathbf{W}||^3) \tag{5}$$

where the $\delta w_i$ are the components of $\delta \mathbf{W}$, $g_i$ are the components of the gradient of $E$ with respect to $W$, and the $h_{ij}$ are elements of the Hessian matrix **H**

$$g_i = \frac{\partial E}{\partial w_i} \tag{6}$$

$$h_{ij} = \frac{\partial^2 E}{\partial w_i \partial w_j} \tag{7}$$

Mozer and Smolensky[35] proposed to estimate relative importance of a weight $s_k$ by perturbation of the error function of the network following the weight elmination $s_k = E(w_k = 0) - E(w_k = w_k^f)$ by the first term in the Taylor series as

$$\hat{s}_k = -g_k \delta w_k = - -\frac{\partial E}{\partial w_k} \cdot \delta w_k \Big|_{\delta w_k = w_k f} \approx - \frac{\partial E}{\partial w_k}\Big|_{w f_k} \cdot w_k^f \tag{8}$$

and disregarded by other terms in eq 5. Here, $w_k^f$ is the final value of the weight $w_k$. Some problems were indicated for this estimation, arising partly from the nature of the gradient descent technique and type of the error function used in the back propagation algorithm. Since pruning is done on a well-trained function (e.g., at its minimum) the first term in eq 5 as well as the sensitivities in eq 8 are decreasing to zero. The drawback of this algorithm was partially removed by a simple "shadow procedure" of E. D. Karnin.[36] He approximated eq 8 by summation over all discrete steps that the network passes during learning

$$\hat{s}_k \approx \sum_{t=0} \frac{\partial E}{\partial w_k}(t) \Delta w_k(t) \frac{w_k}{w_k^i - w_k^f} \tag{9}$$

where $N$ is the number of epochs and $w_k^i$ is the initial value of weight $w_k$. The application of this method was successfully demonstrated.[36] The square of a single weight sensitivity from eq 9

$$s_k = (\hat{s}_k)^2 \tag{10}$$

was used in eq 1.

(D) LeChun et al.[24] measured the sensitivity ("saliency", it is an increase in error of a network that results when the analyzed weight is eliminated) of a weight by estimating the second derivative of the error with respect to the weight. Thus this technique, also known as Optimal Brain Damage (OBD),[24] is different from algorithm C, which was based on the first derivatives. Assuming that pruning is done in a minimum of function $E$ the authors disregarded the first order terms in eq 5. Two additional assumptions—the perturbation terms $\delta w_i$ are small, and the Hessian matrix **H** is very large—allowed them to ignore the third- and higher-order terms as well as off-diagonal elements of the Hessian matrix. This leaves

$$\delta E \approx \frac{1}{2} \sum_i h_{ii} \delta w_i^2 \tag{11}$$

The second derivatives $h_{kk}$ can be calculated by a modified back-propagation rule. The sensitivity ("saliency") of a weight $w_k$ is then

$$s_k = h_{kk} w_k^2 / 2 \tag{12}$$

This weight sensitivity is used in eq 1.

(E) The last investigated method, so called Optimal Brain Surgeon (OBS), was developed by Hassibi and Stork.[25] This technique is very similar to that of OBD, with the exception that the authors did not ignore the diagonal terms of the Hessian matrix **H** in eq 5. The "saliency" of weight $w_k$ is determined as

$$s_k = \frac{1}{2} \frac{w_k^2}{[\mathbf{H}^{-1}]_{kk}} \tag{13}$$

This saliency was used as a sensitivity of a weight $w_k$ in eq 1. Here, $[\mathbf{H}^{-1}]_{kk}$ represents the diagonal element of inverse matrix $\mathbf{H}^{-1}$. OBS produces the same results as OBD in the special case of diagonal **H**. Some improvement of OBS over OBD was demonstrated. However, the drawback of this algorithm consists in time expensive calculations of inverse matrix $\mathbf{H}^{-1}$.

Input neurons with the lowest sensitivities were determined for each ANN at the stop points $S_1 - S_3$ and the numbers of ANNs when the input had lowest sensitivity were counted.

## STATISTICAL COEFFICIENTS

The MSE error $E$ was computed as a criterion of network learning to determine the stop points of a training procedure. Two generally accepted criteria were used for comparison of variable set qualities: the correlation coefficient $R$, defined in the usual way,[37] and the cross-validated $q^2$ value, namely

$$\text{cross validated } q^2 = \frac{\text{SD} - \textit{press}}{\text{SD}} \tag{14}$$

introduced by Cramer et al.[38] Here SD represents the variance of a target value to its mean, and "press" is the average squared errors of predicted values. Use of the cross-validated coefficient $q^2$ makes redundant the analysis of residuals by means of standard deviation, because both coefficients are interrelated and can be derived one from another.

## EXPERIMENTS WITH ARTIFICIAL STRUCTURED DATA SETS

Artificial Structured Data Sets (ASDS) were used to compare the pruning methods. ASDS unlike real data sets have virtually no relationships amid the input variables. This property allows us to compare with a better confidence different pruning procedures and to avoid chance effects due to internal relationships amid input variables.

Three types of ASDS were used. Generated by a random function (Borland C++ 3.0) the numbers were scaled between 0.1 and 0.9 and used as input variables. Target values were calculated as shown below and were also scaled between 0.1 and 0.9. The training sets contained 50 cases.

The ASDS had a general form

$$\text{target} \equiv f(\theta) = g(\theta) + \epsilon \tag{15}$$

where

$$\theta = \sum_{i=1}^{3} V_i \cdot x_i \tag{16}$$

$g(\theta)$ is an analyzed function, $x_i$ are three independent variables, $V_i$ are some coefficients, and $\epsilon$ is a "noise"

**Table 1.** Artificial Neural Network (ANN) Sensitivities Calculated for Linear Example by Analyzed Pruning Methods

| | point $S_1$[a] | | | | | point $S_2$ | | | | | point $S_3$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| input variable | A[b] | B | C | D | E | A | B | C | D | E | A | B | C | D | E |
| $x_1$ | 6[c] | 6 | 30 | 4 | 6 | 13 | 9 | 20 | 11 | 19 | 14 | 14 | 7 | 17 | 14 |
| $x_2$ | 5 | 4 | 9 | 9 | 4 | 22 | 25 | 2 | 32 | 16 | 40 | 36 | 0 | 43 | 35 |
| $x_3$ | 89 | 90 | 61 | 87 | 90 | 65 | 66 | 78 | 57 | 65 | 46 | 50 | 93 | 40 | 51 |

[a] The point, in which learning of the network was terminated (see Figure 3 and text for details). [b] Sensitivity calculation method. [c] The number of ANNs (100 networks were used in an ensemble), when the input variable shown in the first row had the lowest sensitivity; magnitudes of constants used to generate the data set in eq 16 were $V_1 = 0.2$, $V_2 = 0.4$, and $V_3 = 0.05$.

generated according to a Gaussian distribution with a mean equal to zero.

**Linear.** The function $g(\theta)$ in eq 15 was a linear one

$$g(\theta) = \theta \tag{17}$$

$V_1$ and $V_2$ were constant coefficients $V_i = \{-0.2,0.4\}$, while several values were investigated for $V_3 = \{0.01,0.02,0.05,-0.1,0.2,0.4\}$. Noise $\epsilon$ was added to keep the relationship between the independent variables and the target to a correlation coefficient $R$ of about 0.89.

**Nonlinear.** This is an example of a rather complex relationship between input variables and the target function $g(\theta)$ generated by

$$g(\theta) = \begin{cases} \sqrt{\theta - 0.5}, & \text{if } \theta > 0.5 \\ \sin(4\pi\theta), & \text{if } \theta \le 0.5 \end{cases} \tag{18}$$

$V_i$ and $x_i$ were the same as in the linear example. Correlation coefficient between $f(\theta)$ and $g(\theta)$ was $R = 0.95$ (see Figure 3 of ref 8).

**Quadratic.** The target was generated using three variables that included the square of the first variable. Correlation coefficient for MLR (with squared term) was $R = 0.88$

$$\text{target} = V_1 \cdot x_1 + V_2 \cdot (x_1)^2 + V_3 \cdot x_2 + \epsilon \tag{19}$$

Coefficients were $V_i = \{0.2,-0.6,0.15\}$. This example investigates whether or not pruning methods allow the recognition of a nonrelevant term, instead of selecting a variable with smaller coefficients.

## CALCULATION RESULTS FOR ASDS

**Linear.** The calculated results for the analyzed methods and $V_3 = 0.05$ are shown in Table 1. All analyzed ANN pruning methods found input variable $x_3$ (it corresponds to the input neuron $V_3$, see Figure 2) as redundant. The difference of sensitivities between important and irrelevant inputs was more clear at point $S_1$ than at point $S_2$ for all methods except method C. The sensitivites of inputs became nondistinguishable for these methods at point $S_3$ where the overtraining of networks took place. This finding is in agreement with the similar observations of Wikel and Dow.[14] However, the network overtraining did not have any influence on correct determination of neural sensitivities by method C, which calculated the sharpest sensitivities at point $S_3$.

An increase of the magnitude of constant $V_3$ in eq 16 decreases the number of times, when the corresponding input neuron is determined as redundant. This is demonstrated in Table 2 where, for the sake of simplicity, only sensitivities calculated by method B at point $S_1$ are shown. Increase of the magnitude of constant $V_3$ up to $V_3 = 0.4$ makes the corresponding input neuron more significant than input neuron $V_1$. Indeed, for this value of constant $V_3$ variable $x_3$ becomes more significant than variable $x_1$ in eq 16, in as much as the magnitude of constant $V_3 = 0.4$ is larger than that of constant $V_1 = 0.2$.

**Table 2.** Artificial Neural Network Sensitivities Calculated by Method B for Linear Data Sets Generated with Different Values of Constant $V_3$

| | magnitude of the constant $V_3$ | | | | | |
|---|---|---|---|---|---|---|
| input variable | 0.01 | 0.02 | 0.05 | 0.1 | 0.2 | 0.4 |
| $x_1$ | 1[a] | 0 | 6 | 7 | 22 | 69 |
| $x_2$ | 2 | 1 | 4 | 4 | 3 | 6 |
| $x_3$ | 97 | 99 | 90 | 89 | 75 | 25 |

[a] Here and in all subsequent tables, if it is not stated otherwise, sensitivities and statistical parameters of ANNs were calculated at point $S_1$ by statistical analysis of 100 networks.

**Nonlinear.** Similar sensitivities were calculated by all methods. Sensitivities calculated by method A in point $S_1$ are shown in Table 3. MLR failed to find any relationship amongst the analyzed data, while ANN detected it (Table 4). The predictive ability of ANN for a data set of 50 cases decreased proportionally to the increase of the magnitude of constant $V_3$. These findings can be easily understood. The influence of variable $x_3$ is negligible, if the magnitude of $V_3$ is small ($V_3 = 0.01, 0.05$]), and ANN deals with the function of only two variables $f(\theta) \approx f(x_1,x_2)$. This influence increases as a function of the magnitude of the constant $V_3$, and this variable cannot be disregarded in the range $V_3 =$ [0.1, 0.4]. ANN works with a function of three variables in this range of $V_3$. The number of cases (50), *i.e.*, the amount of information, that is sufficient to represent the function of two variables becomes inadequate to characterize the three dimensional function.[39] The increase of the importance of $V_3$ is perfectly controlled by the analyzed sensitivity calculation methods. The sensitivities of all neurons became approximately the same for $V_3 =$ [0.1, 0.4].

We supplied more cases for the network. The results for large data sets with 100 and 200 cases are shown in Tables 3 and 4. ANNs calculated higher correlation coefficients for these data and were able to determine relationships for data sets with $V_3 =$ [0.1, 0.4]. Increase of the size of data sets resulted in a more distinct determination of the $V_3$ input as unimportant for $V_3 =$ [0.01, 0.05].

Pruning of the $V_3$ neuron resulted in some improvement of predictive ability of neural networks for $V_3 =$ [0.01, 0.02] (Table 4), while the correlation coefficient decreased when pruning the $V_3$ neuron with magnitude [0.1, 0.4], where the variable $x_3$ becomes significant. These results could be anticipated by the analysis of the sensitivities of the neurons. The sensitivities of the $V_3$ input were statistically lower for $V_3 =$ [0.01, 0.05], *i.e.*, so far the neuron was unimportant

**Table 3.** ANN Sensitivities Calculated by Method A for the Nonlinear Example[a]

| | number of cases in the training data set | | |
|---|---|---|---|
| input variable | 50 | 100 | 200 |
| | $V_3 = 0.01$ | | |
| $x_1$ | 20 | 4 | 0 |
| $x_2$ | 4 | 0 | 1 |
| $x_3$ | 76 | 96 | 99 |
| | $V_3 = 0.02$ | | |
| $x_1$ | 20 | 14 | 8 |
| $x_2$ | 4 | 0 | 0 |
| $x_3$ | 76 | 86 | 92 |
| | $V_3 = 0.05$ | | |
| $x_1$ | 24 | 12 | 20 |
| $x_2$ | 7 | 1 | 2 |
| $x_3$ | 69 | 87 | 78 |
| | $V_3 = 0.1$ | | |
| $x_1$ | 25 | 30 | 39 |
| $x_2$ | 32 | 12 | 1 |
| $x_3$ | 43 | 58 | 60 |
| | $V_3 = 0.2$ | | |
| $x_1$ | 44 | 30 | 38 |
| $x_2$ | 25 | 28 | 13 |
| $x_3$ | 31 | 42 | 49 |
| | $V_3 = 0.4$ | | |
| $x_1$ | 54 | 48 | 42 |
| $x_2$ | 13 | 24 | 32 |
| $x_3$ | 33 | 28 | 26 |

[a] The input variable $x_3$ is determined with higher confidence as redundant (for constant $V_3 = [0.01, 0.1]$), when the number of cases in the training data set increases.

**Table 4.** LOO Correlation Coefficients Calculated by ANN and MLR for the Nonlinear Example with and without (in Parentheses) Input Variable $x_3$[c]

| magnitude of constant $V_3$ | ANN, $R$ | | | MLR, $R$ |
|---|---|---|---|---|
| | 50[a] | 100 | 200 | 200 |
| 0.01 | 0.79 (**0.84**)[b] | 0.77 (**0.87**) | 0.89 (**0.92**) | 0.24 (0.20) |
| 0.02 | 0.70 (**0.77**) | 0.78 (**0.85**) | 0.89 (**0.91**) | 0.20 (0.15) |
| 0.05 | 0.67 (0.65) | 0.82 (0.81) | 0.88 (0.84) | 0.21 (0.18) |
| 0.1 | 0.20 (0.44) | 0.72 (0.53) | 0.88 (0.63) | 0.23 (0.17) |
| 0.2 | 0.24 (−0.15) | 0.78 (0.32) | 0.87 (0.40) | 0.14 (0.08) |
| 0.4 | 0.08 (−0.03) | 0.68 (−0.27) | 0.86 (0.47) | 0.20 (0.20) |

[a] Training data set size. [b] In bold are shown data sets, where an improvement of ANN prediction ability has been observed after pruning of the input variable $x_3$. [c] MLR = multiple linear regression; LOO = leave-one-out. There is an improvement of prediction ability of ANN with increase of the data set size. MLR is not able to determine any relationship amongst analyzed cases even for the largest data set of 200 cases.

for learning and could be deleted. However, all inputs had similar sensitivities for $V_3 = [0.2, 0.4]$, *i.e.*, they were approximately equally important for learning, and no one should be deleted. This result demonstrates that an improvement of modeling by pruning of unnecessary inputs should be expected only if these variables have statistically low sensitivities when compared with those of the other inputs.

The aforementioned examples indicate the potential advantages of the sensitivity calculation methods. We would like to point out some restrictions of these approaches too. The next artificial task shows an example of a data set, where all the analyzed pruning methods failed to determine correctly an irrelevant input.

**Table 5.** Sensitivites and Correlation Coefficients $R$ Calculated for Quadratic Example[b]

| input variable | sensitivity methods | | | | | correlation coefficient $R$, LOO | |
|---|---|---|---|---|---|---|---|
| | A | B | C | D | E | ANN | MLR |
| $x_1$ | 18 | 12 | 28 | 12 | 14 | 0.87[a] | 0.85 |
| $(x_1)^2$ | 13 | 5 | 26 | 18 | 19 | 0.88 | 0.77 |
| $x_2$ | 69 | 83 | 46 | 70 | 67 | 0.82 | 0.82 |

[a] Correlation coefficients computed without the input variable shown in the first column (*e.g.*, in this row without $x_1$). [b] The same correlation coefficient $R = 0.88$ was calculated for both MLR and ANN methods when using all three input variables $x_1$, $(x_1)^2$, and $x_2$.

**Quadratic.** It was shown previously[8] that ANNs are able to discover the quadratic nature of analyzed data even without a square term. The results shown in Table 5 confirm this assertion. Correlation coefficients calculated without quadratic $(x_1)^2$ or linear $x_1$ input variables were higher than those computed by MLR LOO. This means that the ANNs found and modulated both square (when input variable $(x_1)^2$ was not supplied for training and testing) and square root terms (without input variable $x_1$). The third variable $x_2$ was undoubtedly an important one for network training. Both ANNs and MLR were characterized by the low correlation coefficient, when this variable was removed from the learning data set. Nonetheless, all the analyzed sensitivity calculation methods found just this variable $x_2$ as unimportant, which is incorrect.

**Experiments with Real QSAR Data Set.** Three QSAR examples were taken from the literature. The first two data sets, 51 benzodiazepine derivatives with *anti*-pentylenetetrazole activity (data set **1**, Table 6a) and 37 2,5-bis(1-aziridinyl)-*p*-benzoquinones with antileukemic activity (data set **2**, Table 6b), were previously investigated using MLR,[40] Functional-Link Net (FUNCLINK), and ANNs.[41] Authors from ref 41 used back propagation ANN trained by simplest Generalized Delta Rule (GDR) algorithm with momentum and referred to their network as a GDR net. The descriptors from ref 41 were used in this study. The last data studied (data set **3**) were 74 2,4-diamino-5-(substituted benzyl)-pyrimidines. Their biological activities were measured by inhibition constants (log $K_i$) to dihydrofolate reductase (DHFR) from MB1428 *E. coli*.[42] These data were extensively studied by several QSAR methods including ANNs and logic programming (see ref 43 and references in it). It was shown that the difference in calculated results is not statistically significant when using two different sets of variables: Hansch descriptors and 27 physicochemical molecular attributes. We decided to examine, if all 27 reported physicochemical attributes were relevant for ANN training.

It was shown earlier, that the first two data sets contained some irrelevant variables.[41] Two variables ***PI-3*** and ***R-4*** were excluded in a previous analysis by the MLR and FUNCLINK algorithms for the benzodiazepines set. ANN sensitivity methods A, B, and E excluded the same two variables for this data set (Table 6a). For the benzoquinones MLR analysis excluded $\boldsymbol{MR}_{1,2}$ and $\boldsymbol{PI}_{1,2}$, while FUNCLINK excluded $\boldsymbol{MR}_{1,2}$ and $\boldsymbol{PI}_2$ variables (Table 6b). ANN sensitivity methods A, B, D, and E excluded two variables for this data set similarly to FUNCLINK method. Further removal of variables resulted in lowering of prediction ability of ANNs.

**Table 6.** Calculated Results for Real QSAR Examples

(a) 51 Benzodiazepine Derivatives (Data Set **1**)

| | analyzed input variables | | | | | | | correlation coefficient, *R* | | cross-validation $q^2$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| method[a] | ***MR-3*** | ***PI-3*** | ***MR-7*** | $\sigma_m$***-3*** | ***F-4*** | ***R-4*** | ***I-7*** | learn. | LOO | learn. | LOO |
| MLR | | | | | | | | 0.86 | 0.77 | 0.74[b] | 0.57 |
| MLR | | X[c] | | | | X | | 0.86 | 0.80 | 0.73 | 0.63 |
| FUNCLINK | | X | | | | X | | 0.88 | 0.84 | 0.77 | 0.71 |
| GDR | | | | | | | | 0.865 | 0.566 | | |
| A[d] | 10 | **158** | 23 | 17 | 18 | 96 | 78 | | | | |
| B | 8 | **160** | 15 | 10 | 25 | 112 | 70 | 0.99 | 0.80 | 0.98 ± 0.002[e] | 0.64 ± **0.03** |
| ANN C | 69 | **137** | 63 | 1 | 14 | 40 | 76 | | | | |
| D | 71 | **122** | 3 | 1 | 96 | 41 | 35 | | | | |
| E | 13 | **181** | 18 | 6 | 19 | 109 | 54 | | | | |
| A | 17 | | 45 | 26 | 55 | **137** | 120 | | | | |
| B | 19 | | 40 | 19 | 56 | **154** | 112 | 0.99 | 0.81 | 0.98 ± 0.002 | 0.66 ± **0.03** |
| ANN C | 84 | X | 91 | 0 | 22 | 49 | **154** | | | | |
| D | **138** | | 54 | 1 | 85 | 57 | 65 | | | | |
| E | 24 | | 32 | 16 | 54 | **162** | 112 | | | | |
| ANN | | X | | | | X | | 0.98 | 0.82 | 0.97 ± 0.005 | 0.67 ± **0.02** |

(b) 37 Carboquinone Derivatives (Data Set **2**)

| | analyzed input variables | | | | | | correlation coefficient, R | | cross-validation $q^2$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| method | $MR_{1,2}$ | $PI_{1,2}$ | $PI_2$ | $MR_1$ | ***F*** | ***R*** | learn. | LOO | learn. | LOO |
| MLR | | | | | | | 0.91 | 0.80 | 0.82 | 0.60 |
| MLR | X | X | | | | | 0.89 | 0.84 | 0.80 | 0.71 |
| FUNCLINK | X | | X | | | | 0.95 | 0.94 | 0.92 | 0.87 |
| GDR | | | | | | | 0.94 | 0.86 | | |
| A | **158** | 8 | 131 | 33 | 16 | 54 | | | | |
| B | **167** | 16 | 135 | 50 | 8 | 24 | 0.99 | 0.89 | 0.99 ± 0.001 | 0.79 ± **0.03** |
| ANN C | 63 | 2 | 17 | 53 | **246** | 19 | | | | |
| D | **130** | 16 | 125 | 43 | 86 | 0 | | | | |
| E | **152** | 10 | 138 | 34 | 21 | 45 | | | | |
| A | | 24 | **239** | 49 | 21 | 67 | | | | |
| B | | 17 | **270** | 64 | 7 | 42 | 0.98 | 0.92 | 0.97 ± 0.003 | 0.84 ± **0.02** |
| ANN C | X | 3 | 33 | 23 | **313** | 28 | | | | |
| D | | 25 | **209** | 106 | 60 | 0 | | | | |
| E | | 22 | **234** | 52 | 17 | 75 | | | | |
| ANN | X | | X | | | | 0.98 | 0.93 | 0.97 ± 0.003 | 0.85 ± **0.02** |

[a] 500 ANNs with random initial weights were calculated to compute a final prediction for each set of variables. [b] $q^2$ coefficients for MLR and FUNCLINK were estimated using calculation results from ref 41. [c] X denotes the variable that was not used in the analysis. [d] The method of sensitivity estimation by ANN. [e] The limits at confidence level $\alpha < 0.05$ are indicated (see ref 8 for details of calculations). [f] FUNCLINK = functional link net, GDR = generalized delta rule net with 10 hidden neurons from ref 41; see text for details.

A detailed and more time consuming examination of the data sets by consequent pruning showed that only those sets of variables found by ANNs produced the most significant statistical parameters. It should also be noted, that all the best sets of variables containing 3−6 variables found by methods A, B, and E for data set **2** as well as those containing 3−5 variables found by methods A, B, D, and E for data set **1**, conincided with those found by consequent pruning. For sets of only two variables, several combinations of variables calculated similar statistical coefficients, and such a comparison was rather ambiguous.

Method D failed to find a best set of variables for the benzodiazepine set, while method C was not able to determine the correct order of variable pruning for either data set.

It should be noted, that the use of another training algorithm (SuperSAB instead of GDR with momentum) and avoidance of overtraining in data sets **1** and **2** have produced improved ANN prediction ability according to the LOO method in comparison to the similar results reported previously[41] (see Table 6).

For the third data set of pyrimidines it was very time consuming to do a consequent pruning of parameters, and we restricted our calculations only to complete pruning of the initial set of parameters by all five analyzed methods. The calculated result shows that reduction of the number of analyzed variables by methods B, E, and D substantially improved the predictive ability of ANNs (Table 7). The methods determined very similar sets. The variables found by method B represent a kernel set that is included in the

**Table 7.** The Best Sets of Variables Found by Pruning Methods for 74 Pyrimidines (Data Set **3**)[a]

| method of parameter pruning | best set of variables | correlation coefficient *R*, LOO | cross-validation $q^2$, LOO |
|---|---|---|---|
| without pruning | 1−27 | 0.66 | 0.42 ± **0.05** |
| B | **2−4, 6, 20, 23, 26** | 0.76 | 0.56 ± **0.04** |
| D | **2−4, 6, 20, 23, 19,** 8, 15 | 0.77 | 0.60 ± **0.04** |
| E | **2−4, 6, 20, 23, 26,** 12, 17 | 0.78 | 0.61 ± **0.04** |

[a] The numbers of variables correspond to those (the numbers of columns) in data files from ref 43. The data for analysis were taken at the UCI Machine Learning Repository (anonymous ftp:ics.uci,edu). The variables in bold are common for all three found sets. Note, that the vaiables 19 and 26 are highly correlated with $R = 0.937$, and they can be considered as the same variable. 400 ANNs with random initial weights were calculated to compute a final prediction for each set of variables.

best sets of methods E and D. The variable #26 is absent in the best set of method E. However, this set contains an analog of variable #26—variable #19. Both variables are highly correlated with $R = 0.937$. Two additional variables #8 and #15 or #12 and #17 that are present in the best sets for methods E and D, respectively, slightly improve the predictive ability of ANNs for these sets in comparison to that of method B.

Methods A and C failed to determine sets of parameters with cross-validated coefficient $q^2$ statistically higher than that of the initial set with all 27 parameters.

## DISCUSSION

A main advantage of the direct pruning methods investigated here in comparison with other variable selection procedures for ANNs is their relative speed. Some of these methods were able to discover the same best sets of parameters that can only be found by a more time consuming consequent pruning algorithm.

Tables 1−3 and 5 and 6 show that any one of the analyzed algorithms can provide reliable pruning of the input variables based on the results of only a few ANN trainings. Only statistical analysis of an ensemble of ANNs allows the discovery of the best set of variables in a statistically reliable way and avoids chance correlation when using the pruning methods.

The elaborated methods can be used to prune not just a single but a group of variables with lowest sensitivities simultaneously; this is very important for fast analysis of large data sets. Such a possibility is shown by the analysis of Table 6. Usually, the variable with the second largest number of counts becomes the least sensitive for the next step of pruning, and, therefore, at least a group of a few variables can be deleted simultaneously.

The sensitivities, estimated by the analyzed methods can give some hints about the importance of the used variables for nonlinear regression by ANNs. This information can be useful for interpretation of calculated results. An analysis of the ANN sensitivities in 3D QSAR programs (like CoMFA,[37] GRID,[44] etc.) can allow plotting of 3D contour maps of the importance of molecular regions and visualization of the calculated results. This visualization can be extremely useful for new drug design and makes possible a wide application of ANNs in 3D QSAR studies.

Sometimes the proposed methods failed to determine an irrelevant variable, as demonstrated by the quadratic artificial example or application of the methods to real QSAR analysis. Could we correct such a drawback of the algorithms? Removal of the important neuron usually notably decreased the LOO correlation coefficients. A decrease of correlation coefficients should be a stop point for following network pruning by the used method. We should either terminate a further reduction of the analyzed variables at this step or use a more detailed and time consuming consequent pruning of each of the remaining variables or use another pruning algorithm. This situation hardly appears in the data sets with a large number of variables, where the proposed algorithms are expected to be the most useful.

The ANNs have some major advantages over standard statistical methods of modeling data. They permit the recognition of complex relationships in the data without providing in the model any *a priori* explicit description, as shown by the nonlinear example. A comparison of linear and nonlinear examples shows that the internal structure of data sets a key role in neural network learning. The ANNs are more easily trained to learn simple, linear dependencies while more cases (*i.e.*, information) are required for the training of complex nonlinear functions. Maybe, this property of networks can be used to measure the complexity of analyzed data.

Direct variable selection by ANN is time consuming. Our calculations of only one data set of the linear example took approximately 1.5 h on an IBM PC 486 (DX33). The alternative approach consists in selection of variables by fast methods such as MLR, potential functions, cluster analysis, etc. The chosen variables would be used for the final ANN analysis. Such an algorithm was used, *e.g.*, in the work of Bodor *et. al*[45] who used the most pertinent molecular properties discovered by MLR to train a network. Data preprocessing by the nearest neighbors method has also been reported.[46] However, the potential use of such an approach depends on the internal structure of the analyzed data set. MLR and ANN have shown similar variable estimations for the linear data set, and the very same variables were excluded for the benzodiazepine derivatives. It would be possible to use MLR analysis to exclude an irrelevant variable for these data sets. However, the variables chosen by MLR would not be effective for the nonlinear example (where MLR failed to find any dependency amongst the analyzed data) as well as for the carboquinone derivatives (see Table 6b).

The variables selected by ANNs for data sets **1** and **2** coincided with those selected by the FUNCLINK approach. It should be noted that FUNCLINK equations included highly nonlinear terms of input variables according to

$$\log(1/C) = (1 + e^{-I_q})^{-1} \tag{20}$$

where the analyzed activity $\log(1/C)$ is normalized to fall in the range [0.0, 1.0] and

$$\boldsymbol{I_q} = -3.057\boldsymbol{MR} - 3 + 0.519\cos(\pi\boldsymbol{MR} - 7) + 1.036\sigma_m - 3^2 - 0.552\cos(\pi\boldsymbol{F} - 4) - 0.63\,\textbf{I}\text{-}7 - 0.033 \tag{21}$$

for data set **1** and

$$I_q = 1.894\cos(\pi\boldsymbol{PI}_{1,2}) + 0.757\cos(\pi\boldsymbol{MR}_1) + 2.438\cos(\pi\boldsymbol{F}) - 3.938\boldsymbol{R} \tag{22}$$

for data set **2**.

Since ANNs calculated significantly better statistical coefficients than MLR, they restored some of these (or similar) nonlinear dependencies. This result demonstrates the power of the ANN method.

It is not the purpose of this paper to compare the FUNCLINK and ANN methods. This work has been done to show the possibility of direct pruning of input variables in ANNs when using sensitivity methods and to introduce several such methods. However, some disadvantages of the FUNCLINK models can be pointed out. FUNCLINK models interpret calculated dependencies, and this is a very big advantage of this method. But, the LOO method is part of a variable selection procedure of FUNCLINK, and LOO statistical coefficients do not estimate appropriately the predictive ability of FUNCLINK. It is possible that FUNC-

LINK results were calculated by chance, as it was in early applications of back-propagation ANNs,[6,19] because the training algorithm of FUNCLINK is very similar to that of ANN (sensitivity to initial random initialization, stepwise training algorithm) and, perhaps, all of the problems observed in ANNs are relevant to FUNCLINK nets too. Note, that the problem of chance correlation is even more important for FUNCLINK, since it uses an enhanced number of base functions, *i.e.*, functions used to set a relation between the analyzed independent and dependent variables, such as $x_i^2$, cross-terms $x_ix_j$, $\sin(\pi x_i)$, $\cos(\pi x_i)$, $\log(\pi x_i)$, and $1/(1 + \exp(-x_i))$ while only one such function $1/(1 + \exp(-x))$ is used in ANNs. Use of a large set of base functions also increases the chance correlation problem when analyzing a data set.[3] In any case all these issues should be carefully investigated before systematic use of FUNCLINK. A detailed discussion of the chance correlation problem in FUNCLINK models can be also found in ref 47.

In most of the experiments described here, the differences between sensitivities of important and irrelevant neurons for methods A, B, D, and E were most evident at point $S_1$. Therefore, this point should be recommended for variable estimation by these methods.

The overtraining of ANN did not have any significant influence on sensitivity calculation by method C. The possible interpretation of this effect is explained by eq 8. During the overtraining of ANN both the error $E$ of the net and $\partial E/\partial w_k(n)$ terms are small (the gross structure of data was already learned and function $E$ is near a minimum). Indeed, the sufficiently large changes in $\Delta w_k$ weight magnitude do not influence considerably the sensitivity of the weights and, accordingly, the sensitivity of the node and makes it insensitive to overtraining. However, this method usually failed to correctly find the best set of variables and can hardly be recommended for further use.

The "magnitude-based" sensitivity methods A and B provided almost the same estimations of input variables, as the more sophisticated "error-based" methods D and E. This result is rather unexpected, taking into account that the second group of methods rests upon a better theoretical basis. However, let us remember that the main assumption of the "error-based" methods—pruning is done in a minimum of function $E$ of ANN—is not valid, if an "early stopping" technique is used. The stopping point $S_1$ (in which pruning is done) does not correspond to the true minimum of the error function of ANN for the training data set and the presuppositions about significance of terms in eq 5 are generally not valid. That is why "error-based" methods can fail to find the best sets of parameters. No advantage of "error-based" over "magnitude-based" methods have been detected for the analyzed data sets. In addition, the most precise "error-based" method E becomes rather slow for a large data set or a large neural network analysis. Its speed decreases as $z^{-2}$, where z is the number of weights in a neural network. Therefore this method can hardly be recommended for analysis of data sets with hundreds or thousands of input variables.

In summary, the best estimations of input variables for the analyzed examples have been made by pruning methods B and E. A final answer for the question "Which is the best pruning method?" can be provided only after systematic application of the analyzed methods.

## ACKNOWLEDGMENT

This study was partially supported by Swiss National Science Foundation grants FNRS 31-37723.93 and 7-UKPJ-041507. The authors would like to thank Vasiliy V. Kovalishin for the programming of OBS and OBD algorithms and the referees, who suggested we include in the analysis some of the "error-based" methods.

## REFERENCES AND NOTES

(1) Hyde, R. M.; Livingstone, D. J. Perspectives in QSAR: Computational Chemistry and Pattern Recognition *J. Comp.-Aided Mol. Design* **1988**, *2*, 145−155.
(2) Aivazyan, S. A.; Buchstaber, V. M.; Yenyukov, I. S.; Meshalkin, L. D. *Applied Statistics. Classification and Reduction of Dimensionality*. Finansy i statistika: Moscow, 1989.
(3) Vapnik, V. Estimation of Dependencies Based on Empirical Data. Springer-Verlag: New York, 1982.
(4) Kolmogorov, A. N. On the Representations of Continuous Functions of Many Variables by Superposition of Continuous Functions of One Variable and Addition. *Dokl. Adad. Nauk USSR* **1957**, *114*, 953−956.
(5) 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.
(6) Manallack, D. T.; Livingstone, D. J. Artificial Neural Networks: Application and Chance Effects for QSAR Data Analysis. *Med. Chem. Res.* **1992**, *2*, 181−190.
(7) Livingstone, D. J. and Manallack, D. T. Statistics Using Neural Networks: Chance Effects. *J. Med. Chem.* **1993**, *36*, 1295−1297.
(8) Tetko, I. V.; Livingstone, D. J.; Luik, A. I. Neural Network Studies. 1. Comparison of Overfitting and Overtraining *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 826−833.
(9) Rogers, D.; Hopfinger, A. J. Application of Genetic Function Approximation to Quantitative Structure-Activity Relationships and Quantitative Structure-Property Relationships. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 854−856.
(10) Ambati, B. K.; Ambati, J.; Mokhtar, M. M. Heuristic Combinatorial Optimisation by Simulated Darwinian Evolution: A Polynomial Time Algorithm for the Travelling Salesman Problem. *Biological Cybernetics* **1991**, *65*, 31−35.
(11) Fogel, D. B. Empirical Estimation of the Computation Required to Discover Approximate Solutions to the Travelling Salesman Problem Using Evolutionary Programming. *Proc. of 2nd Ann. Conf. on Evolutionary Programming*, Fogel, D. B., Atmar, W. Eds.; Evol. Prog. Soc.: La Jolla, CA, 1993; pp 56−61.
(12) Kubinyi, H. Variable Selection in QSAR Studies. I. An Evolutionary Algorithm. *Quant. Struct.-Act. Relat.*, **1994**, *13*, 285−294.
(13) Tetko, I. V.; Tanchuk, V. Yu.; Luik, A. I. Application of an Evolutionary Algorithm to the Structure-Activity Relationship. *Proc. of 3rd Ann. Conf. on Evol. Program.* Sebald, A. V., Fogel, L. J., Eds.; World Scientific: River Edge, NJ, 1994; pp 109−119.
(14) Wikel, J. H.; Dow, E. R. The Use of Neural Networks for Variable Selection in QSAR. *Bioorg. Med. Chem. Lett.* **1993**, *3*, 645−651.
(15) One epoch corresponds to the presentation of all input patterns to the ANN followed by updating of the net weights (batch training).
(16) Manallack, D. T.; Livingstone, D. J. Neural Networks—A Tool for Drug Design. In: *Advanced Computer-Assisted techniques in Drug Design*; Van de Waterbeemd, H., Ed.; VCH Weinheim, 1994; pp 293−319.
(17) Hansen, L. K.; Salamon, P. Neural Networks Ensembles. *IEEE Trans. Pattern Anal. Machine Intell.* **1990**, *12*, 993−1001.
(18) Perrone, M. P. General Averaging Results for Convex Optimisation. In *Proceedings of the 1993 Connectionist Models Summer School*; Erlbaum Associates: Hillsdale, NJ, 1994; pp 364−371.
(19) Tetko, I. V.; Luik, A. I.; Poda, G. I. Applications of Neural Networks in Structure−Activity Relationships of a Small Number of Molecules. *J. Med. Chem.* **1993**, *36*, 811−814.
(20) Tetko, I. V. Application of Neural Networks in Structure−Activity Relationship Studies. PhD Dissertation. Institute of Bioorganic & Petroleum Chemistry: Kiev; 1994.
(21) MSE error is defined as

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

where $O_i$ is a calculated and $Y_i$ is a target value, and summation is over all patterns in the analyzed data set.

(22) Thodberg, H. H. A Review of Bayesian Neural Networks with an Application to Near Infrared Spectroscopy. *IEEE Trans. Neural Networks*, in press.
(23) Borggaard, C.; Thodberg, H. H. Optimal Minimal Neural Interpretation of Spectra. *Anal. Chem.* **1992**, *64*, 545−551.
(24) LeChun, Y.; Denker, J. S.; Solla, S. A. Optimal Brain Damage. In: *Advances in Neural Processing Systems 2 (NIPS*2)*; Touretzky, D. S., Ed.; Morgan-Kaufmann: 1990, pp 598−605.
(25) Hassibi, B.; Stork, D. Second Order Derivatives for Network Pruning: Optimal Brain Surgeon. In *Advances in Neural Processing Systems 5 (NIPS*5)*; Hanson, S. Cowan, J., Giles, C., Eds.; Morgan-Kaufmann Publishers: San Mateo, CA, 1993; pp 164−171.
(26) Hansen, L. K.; Rasmussen, C. E. Pruning from Adaptive Regularization. *Neural. Comp.* **1994**, *6*, 1223−1233.
(27) Chauvin, Y. A. Back-Propagation Algorithm with Optimal Use of Hidden Units. In *Advances In Neural Processing Systems 1 (NIPS*1)*; Touretzky, D. S., Ed.; Morgan-Kaufmann: 1989; pp 519−526.
(28) Weigen, S. A.; Rumelhart, D. E.; Huberman, B. A. Generalization by Weight-Elimination with Application to forecasting. In *Advances in Neural Processing Systems 3 (NIPS*3)*; Lippann, R., Moody, J., Touretzek, D. S., Eds.; Morgan-Kaufmann: 1991; pp 875−882.
(29) Reed, R. Pruning Algorithms−a Survey. *IEEE Trans. Neural Networks*. **1993**, *4*, 740−747.
(30) Sietsma, J.; Dow, R. Creating Artifical Neural Networks that Generalize. *Neural Networks* **1991**, *4*, 67−79.
(31) Kruschke, J. K.; Movellan, J. R. Benefits of Gain: Speeded Learning and Minimal Hidden Layers in Back-Propagation Networks. *IEEE Trans. Syst. Man Cybern.* **1991**, *21*, 273−280.
(32) 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.
(33) 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 Artificial Neural Networks. *J. Med. Chem.* **1994**, *37*, 2520−2526.
(34) Nagai, T.; Yamamoto, Y.; Katayama, H.; Adachi, M.; Aihara, K. A Nove; Method to Analyse Response Patterns of Taste Neurons by Artifical Neural Networks. *NeuroReport* **1992**, *3*, 745−748.
(35) Mozer, M.; Smolensky, P. Skeletonization: a Technique for Trimming the Fat from a Network via Relevance Assessment. In *Advances in Neural Processing Systems 1 (NIPS*1)*; Touretzky, D. S., Ed.; Morgan-kaufmann Publishers: San Mateo, CA, 1989; pp 107−115.
(36) Karnin, E. D. A simple Procedure for Pruning Back-Propagation Trained Neutral Networks. *IEEE Trans. Neural Networks* **1990**, *1*, 239−242.
(37) For example: Jonhos, N. N.; Leone, F. G. *Statistic and Experimental Design in Engineering and the Physical Sciences*; John Wiley & Sons: New York, 1977.
(38) This coefficient was introduced as $r^2$: Cramer III, R. D.; 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. However, in order to avoid confusion with the analogous conventional $r^2 = R*R$ value the new designation $q^2$ was recommended: Cramer III, R. D.; DePriest, S. A.; Patterson, D. E.; Hecht, P. The Developing Practice of Comparative Field Analysis, In *3D QSAR in Drug Design: Theory Methods and Applications*; Kubinyi, H., Ed.; ESCOM: The Netherlands, 1993; pp 443−486.
(39) The training procedure works with only half the number of cases of the learning data set (25 in this example) for network training, while the other cases are used only to monitor network training. See Figure 3 and more details in ref 8.
(40) Nakao, H.; Arakawa, M.; Nakamura, T.; Fukushima, M. Antileukemic Agents. II. New 2,5-bis(1-aziridinyl)-*p*-benzoquinone Derivatives. *Chem. Pharm. Bull.* **1972**, *20*, 1968−1979.
(41) Liu, Q.; Hirono, S.; Moriguchi, I. Comparison of Functional-Link Net and the Generalised Delta Rule Net in Quantitative Structure−Activity Relationship Studies. *Chem. Pharm. Bull.* **1992**, *40*, 2962−2969.
(42) Li, R. L.; Hansch, C.; Kaufman, B. T. A Comparison of the Inhibitory Action of 5-(substituted-benzyl)-2,4-diaminopyrimidines on Dihydrofolate Reductase from Chicken Liver with That from Bovine Liver. *J. Med. Chem.* **1982**, *25*, 435−440.
(43) Hirst, J. D.; King, R. D.; Sternberg, M. J. E. Quantative Structure−Activity Relationships by Neural Networks and Inductive Logic Programming. 1. The Inhibition of Dihydrofolate Reductase by Pyrimidines. *J. Comput.-Aided Mol. Design*. **1994**, *8*, 405−420.
(44) Goodford, P. J. A. Computational Procedure for Determining Energetically Favorale Binding Sites on Biologically Important Macromolecules. *J. Med. Chem.* **1985**, *28*, 849−857.
(45) Bodor, N.; Harget, A.; Huang, M.-J. Neural Network Studies. 1. Estimation of the Aqueous Solubility of Organic Compounds *J. Am. Chem. Soc.* **1991**, *113*, 9480−9483.
(46) Brill, F. Z.; Brown, D. E.; Martin, W. V. Fast Genetic Selection of Features for Neural Network Classfier. *IEEE Trans Neural Networks* **1992**, *3*, 324−328.
(47) Manallack, D. T.; Livingstone, D. J. Limitations of Functional-Link Nets as Applied to QSAR Data Analysis. *Quant. Struct.-Act. Relat.* **1994**, *13*, 18−21.

CI950204C