Neural Network Studies. 4. Introduction to Associative Neural Networks

Igor V. Tetko†

Laboratoire de Neuro-Heuristique, Institut de Physiologie, Rue du Bugnon 7,
Lausanne, CH-1005, Switzerland, and Biomedical Department, IBPC, Ukrainian Academy of Sciences,
Murmanskaya 1, Kiev-660, 253660, Ukraine

Received August 29, 2001

Associative neural network (ASNN) represents a combination of an ensemble of feed-forward neural networks and the *k*-nearest neighbor technique. This method uses the correlation between ensemble responses as a measure of distance amid the analyzed cases for the nearest neighbor technique. This provides an improved prediction by the bias correction of the neural network ensemble. An associative neural network has a memory that can coincide with the training set. If new data becomes available, the network further improves its predictive ability and provides a reasonable approximation of the unknown function without a need to retrain the neural network ensemble. This feature of the method dramatically improves its predictive ability over traditional neural networks and *k*-nearest neighbor techniques, as demonstrated using several artificial data sets and a program to predict lipophilicity of chemical compounds. Another important feature of ASNN is the possibility to interpret neural network results by analysis of correlations between data cases in the space of models. It is shown that analysis of such correlations makes it possible to provide "property-targeted" clustering of data. The possible applications and importance of ASNN in drug design and medicinal and combinatorial chemistry are discussed. The method is available on-line at http://www.vcclab.org/lab/asnn.

INTRODUCTION

It was recently shown[1] that the prediction of lipophilicity of chemical compounds dramatically decreases for cross—series predictions, i.e., in cases when compounds in training and test data sets are coming from the different structural series of compounds. It is possible that a similar decrease in prediction performance can be also observed for prediction of other physical and, probably, biological properties of chemicals. These findings have provided doubt on our ability to develop "a universal" physical property prediction program for analysis of, at least, lipophilicity of chemicals and, at least, using the approach analyzed in that study.

There are also some other conclusions of this article. It forecasts that programs developed using the same training set of compounds would have similar or, probably, poorer predictive abilities for compounds that have different chemical diversity compared to the training set of compounds. The diversity of molecules in the training set determines some Optimal Prediction Space, i.e., space outside of which the model is not applicable.[2] The lipophilicity programs, such as CLOGP,[3] KOWWIN,[4] ACD/logP,[5] XLOGP,[6] ALOGPS,[1] ALOGP,[7] VLOGP,[2] and many others were developed using compounds either or in combination from the same public set of databases, Biobyte[8] Star and PHYSPROP.[9] These series of public compounds define some Public Optimal Prediction Space (POPS). The programs properly developed with such compounds will have small variation of their results and reliable prediction of compounds that are similar to those inside of POPS. Outside of the POPS a prediction of all these methods will be poor, and a larger variation of program results could be observed. This provides some pessimistic implications about the predictive abilities of these programs and their use, e.g. for prediction of chemical compounds in pharmaceutical companies. Indeed, since compounds used by companies are of great commercial value, their structures and experimental data usually remain unavailable before the drug is patented and released. Thus, it is hardly possible that any experimental values for such series will be publicly available during the development stage of the drug, i.e., when there is the highest need for the prediction of such series. Therefore, it is possible that programs developed "in house" by commercial firms will over perform and will be more useful for their own series than the best sophisticated algorithms developed using the public data sets.

One of the possibilities of improving predictive abilities of programs developed using public data sets is to make such programs smart enough to learn on-the-fly specific structural features of the user's data and to adjust the program predictions according to such specific features. For example, ACD/logP has a possibility of system training with the user-supplied data using so-called user's training feature of the program.[10] When new data are provided, an algorithm breaks the structure down into most important components and then rescales its parameter estimation based on newly identified fragments. Another lipophilicity program, CLOGP,[3] generates missed fragments, i.e., fragments that were never observed in the training set. The user-supplied experimental data can be used to identify contribution of such fragments and to improve the predictive ability of this program.

† Corresponding author phone: +49-89-3187-3575; fax: +49-89-3187-3585; e-mail: itetko@yahoo.com. Current address: Institute for Bioinformatics, GSF − Forschungszentrum für Umwelt und Gesundheit, GmbH, Ingolstädter Landstrasse 1, D-85764 Neuherberg, Germany.

10.1021/ci010379o CCC: $22.00
Published on Web 03/26/2002

The option to learn on-the-fly could provide significant advantages for a program. So far this option was only possible for methods based on linear regression equations, and no simple approach was available for nonlinear methods, such as neural networks. A traditional approach, to use old and new data and to redesign neural network from scratch, could be unfeasible due to time and resources limitations or privacy issues, as discussed further in the article.

Artificial neural network methods are increasingly being used in the field of medicinal chemistry and structure−activity relationship studies,[11] as they are able to elucidate structure−activity relationships and to take into account nonlinear characteristics of these relationships. Thus, the development of a method that could improve a neural networks predictive ability without the need to retrain the neural network weights could potentially have a large field of applications.

The recently proposed Associative Neural Network (ASNN)[12,13] offers an elegant approach to incorporate "on-the-fly" the users' data. The current article provides an introduction to this method and explains why this method has superior generalization ability in comparison to traditional back-propagation neural networks and *k*-nearest neighbor approaches. The advantages of the new method compared to traditional methods are illustrated using simulated artificial data sets and several real data sets looking at lipophilicity of chemical compounds.

DATA SETS

Several artificial data sets were used in the current study to demonstrate properties of the ASNN. The first set was represented by Gauss function

$$y = \exp(-\mathbf{x}^2) \tag{1}$$

with dimension of vector **x** equal to 1. In the second set a two-dimensional vector $\mathbf{x} = (x_1, x_2)$ was used in the same function as

$$y = \exp(-\mathbf{x}^2) = \exp(-(x_1 + x_2)^2) \tag{2}$$

The third example was represented by function that was not symmetric over the *y*-axis:

$$\begin{aligned} &y = \exp(-\mathbf{x}^2), \text{ for } \mathbf{x} = x_1 + x_2 < 0 \text{ and} \\ &y = 1 - (1-1/e)\cdot\mathbf{x}, \text{ for } \mathbf{x} = x_1 + x_2 > 0 \end{aligned} \tag{3}$$

The variables of the vector **x** were uniformly distributed over the same interval (−3, 3). The training and test sets included $N = 100$ and 1000 cases, respectively.

PHYSPROP Data Sets. The PHYSPROP database[9] used in our study included 12908 diverse compounds (May 2000) with experimental values for lipophilicity (logP), selected as described elsewhere.[1] The predictive ability of the ASNN was also evaluated using equal size subsets of 6454 molecules and several other sets derived from these database as described below.

The PHYSPROP database contained many compounds that were from the BioByte[8] Starlist. A run of CLOGP program over all 12 908 compounds provided experimental values for 9429 compounds, named as the "star" set. The remaining set of 3479 compounds was referred to as the "nova" set. XLOGP was developed using a subset of PHYSPROP database that included 1853 molecules.[6] It was shown that the nova and the XLOGP training sets contained less structural diversity compared to their complement sets.[1] These sets were used in our study to demonstrate the user-training features of the ASNN method.

The input indices for analysis of the PHYSPROP data set and its subsets included the number of hydrogen and non-hydrogen atoms and 73 E-state indices[14−16] (in total 75 input parameters), as described in our previous study.[1] Some E-state indices had zero occurrences in the subsets, resulting in only 70 and 61 input indices in the star and in the XLOGP sets, respectively.

Quinazolones. The set of 18 quinazolones with experimental lipophilicity values[17] was used to explain the user-training feature of the ACD/logP program,[10] and it was also selected to demonstrate a similar feature of the ASNN.

BASF Compounds. This data set included 6100 private compounds used in BASF. Due to privacy restrictions the structures of private molecules as well as their experimental values were not available for us, and only the results of the analysis were reported by BASF.[12] These results are summarized in the current article.

METHOD

General Considerations. The traditional artificial feed-forward neural network is a memoryless approach. This means that after training is complete all information about the input patterns is stored in the neural network weights and the input data is no longer needed, i.e., there is no explicit storage of any presented example in the system. Contrary to that, such methods as the *k*-nearest-neighbors (KNN),[18] the Parzen-window regression,[19] etc., represent the memory-based approaches. These approaches keep in memory the entire database of examples, and their predictions are based on some local approximation of the stored examples. The neural networks can be considered global models, while the other two approaches are usually regarded as local models.[20]

Consider a problem of multivariate function approximation, for example finding a mapping $R^m \Rightarrow R^n$ from a given set of sampling points. For simplicity, let us assume that $n = 1$. A global model provides a good approximation of the global metric of the input data space R^m. However, if the analyzed function, *f*, is too complicated, there is no guarantee that all details of *f*, i.e., its fine structure, will be represented. Thus, the global model can be inadequate because it does not describe equally well the entire state space with poor performance of the method being mainly due to a high bias of the global model in some particular regions of space. The neural network variance can also contribute to poor performance of this method.[21] However, the variance can be decreased by analysis of a large number of networks, i.e., using artificial neural network ensemble (ANNE), and taking, for example, a simple average of all networks as the final model. The problem of bias of neural networks cannot be so easily addressed, e.g. by using larger neural networks, since such networks can fall in a local minimum and thus can still have a considerable bias.

Associative Neural Network. ASNN represents a combination of memoryless and memory-based methods.[12,13] Let us consider an ensemble of M neural networks

$$[\mathbf{ANNE}]_M = \begin{bmatrix} ANN_1 \\ \vdots \\ ANN_j \\ \vdots \\ ANN_M \end{bmatrix} \tag{4}$$

The prediction of a case $\mathbf{x}_i$, i=1, ..., N can be represented by a vector of output values $\mathbf{z}_i = \{z^i_j\}^M_{j=1}$ where $j = 1, ..., M$ is the index of the network within the ensemble.

$$[\mathbf{x}_i]\cdot[\mathbf{ANNE}]_M = [\mathbf{z}_i] = \begin{bmatrix} z_1^{\ i} \\ \vdots \\ z_j^{\ i} \\ \vdots \\ z_M^{\ i} \end{bmatrix} \tag{5}$$

A simple average

$$\bar{z}_i = \frac{1}{M}\sum_{j=1,M} z_j^{\ i} \tag{6}$$

is usually used to predict the test cases with neural network ensemble. This average ignores the predictions (variations) of individual neural networks. These variations can be used to introduce a measure of similarity of data cases in the output space,[22,23] e.g. using Pearson's linear correlation coefficient, r_{ij}^2, between the vectors of predicted values $\mathbf{z}_i$ and $\mathbf{z}_j$. The introduced measure of similarity can be used to identify, for each new analyzed data case i, a number of nearest neighbors $N_k(\mathbf{x}_i)$ in the output space, i.e., in space of the neural network models. The ASNN corrects the ensemble value $\bar{z}_i$ according to the following formula

$$\bar{z}'_i = \bar{z}_i + \frac{1}{k}\sum_{j\in N_k(\mathbf{x}_i)} (y_j - \bar{z}_j) \tag{7}$$

where y_j are the experimental values and the summation is over the k-nearest neighbor cases determined using Pearson's r_{ij}, as described above. Since the variance of ensemble prediction $\bar{z}_i$ can become sufficiently small by analysis of a sufficient number of neural networks in the ensemble, the difference ($y_i-\bar{z}_i$) corresponds mainly to the bias of the ANNE for the case $\mathbf{x}_i$. Thus this formula explicitly corrects the bias of the analyzed case according to the observed biases calculated for the neighboring cases. The proposed method is named an "associative neural network",[12,13] since the final prediction of new data by this method is done according to cases, i.e., prototypes of the analyzed example or associations, found in the "memory" of the neural networks. The detected prototypes are used to correct the systematic error of neural network ensemble, i.e., its bias.

Notice, that a traditional implementation of KNN method for regression consists of the calculation of a predicted value of case i as

$$z(\mathbf{x}) = \frac{1}{k}\sum_{j\in N_k(\mathbf{x})} y(\mathbf{x}_j) \tag{8}$$

where $z(\mathbf{x})$ is a predicted value for case $\mathbf{x}$ and $N_k(\mathbf{x})$ is the collection of the k nearest neighbors of $\mathbf{x}$ among the input vectors in the training set $\{\mathbf{x}_\mathrm{i}\}^N_{i=1}$ using the Euclidean metric $||\mathbf{x}, \mathbf{x}_\mathrm{i}||$.

The memory of both KNN and ASNN is represented by the training set $\{\mathbf{x}_\mathrm{i}\}^N_{i=1}$, and the number of nearest neighbors, k, is selected to provide lowest leave-one-out (LOO) error for this set. The performance of both these methods for the training set is estimated using the LOO error.

Instead of the Pearson's linear correlation coefficient, different measures of distance, such as Euclidean distance, Spearman or Kendall rank correlations[24] can be used. The Parzen-window regression[19] or weighted KNN method can be used instead of the simple KNN method. For example, in the ALOGPS 2.0 program a weighted average was used in eq 7 as

$$\bar{z}'_i = \bar{z}_i + \sum_{j\in N_k(\mathbf{x}_i)} (y_\mathrm{j} - \bar{z}_j)\cdot r_{ij}^{\ 2} / \sum_{j\in N_k(\mathbf{x}_i)} r_{ij}^{\ 2} \tag{9}$$

A weighted average provides greater impact for the data entries that are more closely correlated with the analyzed molecule. A detailed analysis of aforementioned parameters of ASNN program using an example of lipophilicity program is provided elsewhere.[25] For the purpose of the current analysis, only a simple average given by eq 7 will be used to illustrate the main properties of the method.

Neural Network Training. The neural networks used in the current study were trained using Early Stopping over Ensemble (ESE) method.[22,26,27] In this method the initial training set was randomly divided into equal size learning and validation sets for each neural network in the ensemble. Thus, each neural network had its own learning and validation sets. The training was stopped when a minimum error for the validation set was calculated ("early stopping" point).[27] The updating of neural network weights was performed using Levenberg−Marquardt algorithm.[24] The number of hidden neurons was fixed to be 5, and a 50 network ensemble was used for the lipophilicity analysis. The artificial data sets were analyzed using a 100 network ensemble.

Implementation. The developed algorithm was programmed in ANSI C++ language and is currently running on Linux, Windows, Sun, and Mac Os operation systems. The online demo version of the algorithm is available at *http://www.vcclab.org/lab/asnn*. The stand-alone version of the program is available on request from the author, and it is free for private users and universities.

RESULTS

Analysis of Simulated Data Sets. Gauss Function Interpolation. The first analysis was done with input vector $\mathbf{x}$ of dimension one. The number of nearest neighbors $k = 1$ provided minimum leave-one-out error (LOO) for the training set using the KNN method. Using this number of the nearest neighbors, the KNN calculated the root-mean-squared error, $RMSE$ = 0.02, for the test set. A worse result, $RMSE$ = 0.075, was calculated by an ANNE with two hidden neurons. In this specific example the KNN was based on the metric that corresponded to the learned task (e.g., cases that had minimum distance in the input space also had minimum distance in the output space), and its performance was quite good.

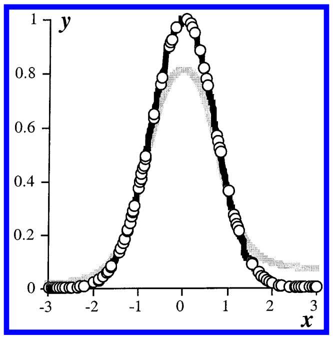


Figure 1. Gauss function $\mathbf{y} = \exp(-(\mathbf{x} = x_1+x_2)^2)$ approximation by an ensemble of artificial neural networks (ANNE) with two hidden neurons (gray line) trained using 100 cases (circles). Associative neural network (ASNN) calculates significantly improved results (black line). The response of ANNE and ASNN are calculated using 1000 test data cases.

The performance of KNN method for the same example considerably decreased if the input metric was changed. Indeed, if a two dimension input vector $\mathbf{x} = \{x_1, x_2\}$ was used as input data (see eq 2), the best KNN performance for these data, *RMSE* = 0.13, $k = 2$, was about an order of magnitude worse than in the first example. Thus, the Euclidean metric was not optimal for the KNN method, and this method was unable to correctly determine the nearest neighbors in the space of new variables.

Contrary to that the ANNE with two hidden neurons calculated prediction error, *RMSE* = 0.084, that was similar to the one-dimensional case (Figure 1). Thus, neural networks correctly learned the internal metric of the example, i.e., $\mathbf{x} = x_1 + x_2$. However, ANNE tended to provide poor predictions near the central part and tails of the function; i.e., its predictions in these regions have a systematic error (bias). This behavior could be also observed for different numbers of hidden neurons however it was more pronounced for networks with two hidden neurons that, therefore, were selected for further analyses with artificial data sets.

The ASNN method improved the predictive ability of ANNE at the regions with significant bias, and a *RMSE* = 0.011 was calculated. The correction according to eq 7 provided better results than the use of the KNN method (*RMSE* = 0.02) in one-dimensional space. Such a result was possible as the ASNN used the neighborhood relations to improve the already good model. For example, if neural networks were ignored and only similarity in the space of models was used to select the nearest neighbors for the KNN method, a lower result, *RMSE* = 0.04, was calculated. Nevertheless this result was still improved compared to the KNN analysis in the original space of parameters.

This example demonstrates that ASNN can improve performance of neural networks by correcting for their bias. Neural networks additionally provided a significant improvement over the KNN approach.

Interpretation of the ASNN Results. Let us consider an example of the prediction of a data point with coordinates $\mathbf{x} = (0, 0)$. The KNN method applied in the original data space selected the nearest neighbors that were quite distant from the analyzed case in the output space (Figure 2). A much greater density of samples would be required for the KNN method in two-dimensional input space to get the same performance as in one-dimensional space. The use of the correlation measure, on the ASNN, made it possible to correctly determine the neighborhood relations and to detect all cases that have similar *y*-values and properties in the output space. Thus explaining the improved predictive ability of ASNN compared to the KNN method.

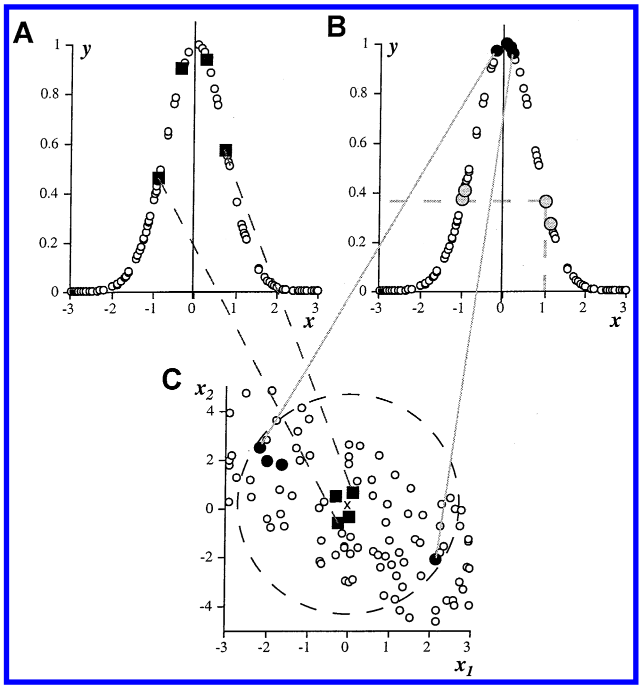


Figure 2. Training data cases (open circles) and the nearest neighbors (filled circles and rectangles) calculated in the space of input parameters (A) and in the space of neural network models (B) for Gauss function $\mathbf{y} = \exp(-(\mathbf{x} = x_1+x_2)^2)$ interpolation. The distribution of cases in the original space of parameters, x_1 and x_2, is also shown (C). The nearest neighbors of data point with coordinates $(x_1, x_2) = (0, 0)$ in the initial space of parameters (rectangles) are sparse. On the contrary, the nearest neighbors detected in the space of models (black circles) are sparse in the original input space of parameters. Notice, that the nearest neighbors of data point $\mathbf{x} = (0, 1)$ in the space of neural network models are symmetric along vertical axis.

The neural networks learned both $\mathbf{x} = x_1 + x_2$ mapping and symmetry of the Gauss function. For example, two out of four nearest neighbors detected for the data case with coordinate $\mathbf{x} = (0, 1)$ were from the region symmetrical along *y*-axis. Indeed, a computationally simpler model could be calculated if one considers the symmetry of Gauss function. The ASNN used the symmetry of the function to increase by a factor of 2 the number of nearest neighbors for bias correction in eq 7. Of course, the neural networks always detected these relations. However, the use of correlation-based similarity measures in the space of models makes it possible to visualize these relationships and to better understand the properties of neural networks. If instead of a symmetric Gauss function the nonsymmetrical function generated according to eq 3 was used to perform the same analysis (Figure 3), the neural networks did not learn the symmetry of the function. For example, all four nearest neighbors detected for data case with coordinate $\mathbf{x} = (0, 1)$ were on the same side of the function near to the analyzed point. Notice, that the *y*-values of the nonsymmetric function were identical for $\mathbf{x} = (0, 1)$ and $\mathbf{x} = (0, -1)$, but the local properties of the function at both points (e.g., derivatives) were completely different.

These results demonstrate that the use of correlations in the space of the models makes it possible to provide an

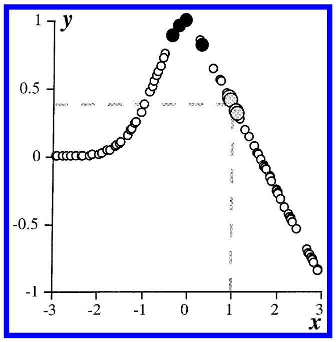


Figure 3. Training data case (open circles) and the nearest neighbors of data points (filled circles) in the space of neural network models for nonsymmetric function example given by eq 3. The nearest neighbors of data point with coordinates (x_1, x_2) = (0, 0) and (0, 1) are shown as black and gray circles, respectively. The symmetric relations for case (0, 1) are not observed (ca. Figure 2B).

interpretation of neural network results and to better understand their properties. The similarities in this space make it possible to identify clusters of data cases that have similar functional properties and not only similar target values.

Gauss Function Extrapolation. The problem to extrapolate neural network predictions presents a difficult and interesting task. Let us consider a subset of the training set with data points such as $x_1 + x_2 < 0$, i.e., only the right part of the Gauss function. The region with $x_1 + x_2 < 0$ provides some kind of OPS for data analysis. An extrapolation of the function beyond OPS can be done in many ways, and it is impossible "*a priory*" to select the correct extension without some assumptions. For example, extension of the function can be either symmetric or nonsymmetric, as it was in the first two examples. In case of multidimensional data, such dependencies cannot be easily visualized.

Incorporation of new information into traditional neural networks is impossible without retraining. ASNN makes it possible to include new information and to extend its OPS without these problems. Indeed, since KNN correction is applied following ensemble analysis, one can easily expand the ASNN memory without the need to retrain the whole neural network ensemble.

The ASNN trained with 50 data cases, such as $x_1 + x_2 < 0$, provided a "flat" response for the region $x_1 + x_2 > 0$ (Figure 4A). Thus, a prediction performance of this method for data with $x_1 + x_2 > 0$ was low both for Gauss and nonsymmetric function approximation. However, if the remaining 50 cases from the training sets of the Gauss function were added to the memory of ASNN to extend its OPS, the neural networks response was dramatically improved and provided a good approximation of the Gauss function in the whole region (Figure 4B). On the contrary, if 50 cases with positive x-values from the nonsymmetric function were added to the ASNN memory, the neural network response became similar to the nonsymmetric function (Figure 4C). The extension of the OPS provided a lower predictive ability of ASNN compared to training with the whole set, e.g. *RMSE* = 0.011 and *RMSE* = 0.042 calculated for the Gauss function approximation using the whole set (Figure 1) and the extension of OPS (Figure 4B), respectively. KNN (Figure 4D) calculated a much worse result, *RMSE* = 0.13, for the same set of 100 points.

This example demonstrates that the extension of the OPS of an ASNN with new data can provide a significant improvement to the predictive ability of a neural network without the need to retrain neural network weights. The ASNN results were also significantly improved compared to the traditional KNN method. This feature of ASNNs has important practical applications that will be demonstrated below.

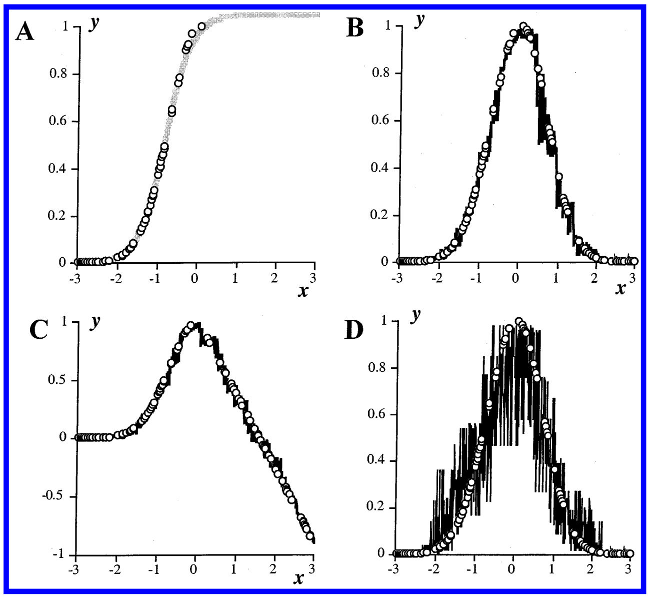


Figure 4. (A) The gray line corresponds to the response of ASNN trained with 50 cases (circles) that had only negative values, **x** = $x_1 + x_2 < 0$ for Gauss function approximation. The neural network provided good interpolation of function for the negative **x**-values and flat response for the positive **x**-values, **x** = $x_1 + x_2 > 0$. The responses of ASNN (black line) were changed if an additional 50 cases (circles) with the positive **x**-values generated according to eq 2 (B) or eq 3 (C) were added to the ASNN memory. (D) The interpolation of the Gauss function using KNN method.

Analysis of Real Data Sets. Bias Correction of Neural Networks. The first study was performed to demonstrate an improved prediction ability of ASNN compared to traditional ANNE using an example of lipophilicity prediction. Half of the molecules were selected by chance from the total PHYSPROP data set and were used as the training set. The other half was used to test the neural network performance. Each neural network had five hidden neurons in one hidden layer. The training of ANNE using Levenberg−Marquardt algorithm[24] required 2.5 days of cpu-time using a 800 MHz Athlon and provided a predictive ability with a *RMSE* = 0.53 and mean average error *MAE* = 0.38 (*RMSE* = 0.46, *MAE* = 0.36 without 97 outliers, identified as molecules with absolute prediction error greater than 1.5 log units). ASNN method used $k = 9$ nearest neighbors and provided additional improvement of results, *RMSE* = 0.49 *MAE* = 0.35 (*RMSE* = 0.43, *MAE* = 0.33 without 82 outliers). The KNN method applied in the Euclidean space of the input variables provided much poorer results for the test set, *RMSE* = 0.88, *MAE* = 0.62 (*RMSE* = 0.64, *MAE* = 0.50 without 514 outliers) using $k = 3$ nearest neighbors.

It was investigated to see whether an increase of the neural network capacity could further improve ANNE performance for this task. The training of ANNE with 10 hidden neurons was completed in 52 days (calculations were done in parallel using a LINUX cluster), giving marginally improved results, *RMSE* = 0.51, *MAE* = 0.37 and *RMSE* = 0.48, *MAE* = 0.34 calculated by ANNE and ASNN ($k = 13$), respectively.

Table 1. Prediction of Star and Nova Sets

method	star set, 9429 molecules			nova set, 3479 molecules		
	RMSE	MAE	outliers[a]	RMSE	MAE	outliers[a]
CLOGP	0.36	0.25	74	0.62	0.50	558
KNN memory: star set, $k = 2$	0.54	0.43	507	0.62	0.60	878
KNN memory: PHYSPROP, $k = 2$	0.54	043	457	0.57	0.46	269
Neural Networks Trained with PHYSPROP (Star + Nova) Set, 12 908 Molecules						
ANNE1	0.42	0.33	61	0.49	0.38	68
ASNN1 memory: PHYSPROP set, $k = 4$	0.37	0.27	29	0.43	0.32	50
Neural Networks Trained with Star Set, 9429 Molecules						
ANNE2	0.42	0.33	78	0.59	0.47	480
ASNN2 memory: star set, $k = 3$	0.37	0.27	41	0.57	0.45	461
ASNN2 memory: PHYSPROP set, $k = 5$	0.37	0.27	49	0.46	0.35	98
Neural Networks Trained with XLOGP Set, 1853 Molecules						
ANNE3[b]	0.53	0.41	416	0.65	0.52	647
ASNN3 memory: XLOGP set, $k = 3$	0.51	0.39	395	0.63	0.51	619
ASNN3 memory: PHYSPROP set, $k = 3$	0.40	0.30	163	0.47	0.36	141

[a] Molecules predicted outside ±1.5 log units were considered as outliers. [b] Memory of ASNN method corresponds to the molecules used to correct the ANNE predictions in eq 7. The number of k nearest neighbors used in ASNN is also indicated.

For both types of networks the ASNN demonstrated a significant improvement in prediction ability compared to traditional method. This increase could not be easily achieved by the use of neural networks with a larger number of hidden neurons. While it is possible that a larger or more sophisticated ANNE could probably provide similar results to an ASNN trained with five hidden neurons, it would require much more computational resources than that of the ASNN. The ASNN1 developed using the whole PHYSPROP data set also showed improvement over ANNE1 results (Table 1).

This example demonstrated that ASNN improves prediction of traditional neural networks by correction of their biases. The neural network results were also significantly better in comparison to traditional KNN method.

Extension of Optimal Prediction Space of Neural Networks. The user training feature of ASNNs introduces a fundamental and very important property of this method. It will be illustrated using several examples.

Analysis of PHYSPROP Data Sets. In agreement with our previous results, the predictive performances of neural networks trained using the star set (ANNE2) were low for the nova set (Table 1). This result can be explained by the significant chemical diversity contained within both sets of molecules as discussed elsewhere.[1] ASNN2 selected $k = 3$ nearest neighbors and provided improved results compared to the ANNE2 for both star and nova sets although the performance of ASNN2 was still not greatly improved for the nova set. The molecules from the nova set were then added to the memory of ASNN2 method in order to extend its OPS, and the LOO results for both star and nova sets were again calculated. The predictive ability of ASNN2 for the nova set increased and became similar to the LOO results calculated for this set using PHYSPROP data set (ASNN1). The number of outliers for the nova set also decreased by about 5 times. Thus, the extension of OPS provided a significant improvement of results for the nova set. The KNN method applied in the Euclidean space of input variables calculated poorer predictive models compared to neural networks developed with the same number of molecules (Table 1).

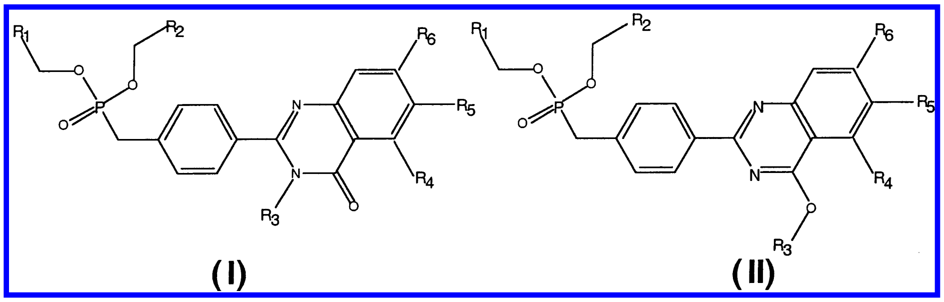
It was shown that the set of 1853 molecules used to develop XLOGP program[6] was not representative of both star and nova sets.[1] The ASNN3 predictive results were poorer for both star and nova sets (Table 1); however, they dramatically improved when both nova and star sets were added to the memory of this network. The performance of the ASNN3 and the ASNN2 methods with extended OPS was quite similar to that of the ASNN1 for the nova set. The main difference was in the numbers of outliers that increased inversely proportional to the number of E-state indices used to develop ASNN2 and ASNN3 models.

Analysis of Quinazolones. Nine of the 18 analyzed quinazolones had a general structure designated as (I) and nine had the general structure (II) as shown in Table 2. The ASNN1 and KNN provided significantly better prediction of quinazolones compounds with general structure (I) compared to compounds with general structure (II). The poorer performance of both these methods for compounds with general structure (II) suggested that this set of compounds did not have analogues in the PHYSPROP data set (Figure 5). Indeed, the nearest analogues detected in the space of neural network models, e.g. for compound no. 9, were significantly different from the target structure (Figure 5B), and only one molecule, oxodiazinon, contained phosphorus. The nearest neighbors of compound 1 (Figure 5A) were more similar to the basic structure (I), and thus the OPS for this set of molecules was better covered with the training set of compounds. This explains the higher performance of neural networks and KNN for this subset of compounds.

The ACD/logP program can provide user training, even using only one compound. The results for such training with compound numbers 1 and 9, representing the simplest versions of (I) and (II), are shown in Table 2.[10] The training dramatically improved the performance of the ACD/logP program for both subsets of quinazolones.

To make a similar comparison with the ACD/logP program, the ASNN1 was applied using compound numbers 1 and 9 as the memory of the neural networks. The predicted results for the test set compounds improved significantly and were similar to those of the ACD/logP program. The KNN results (Table 2) also improved when compound numbers 1 and 9 were added to the PHYSPROP data set; however, they were the poorest of the three analyzed methods.

Table 2. Experimental and Calculated LogP Values for Quinazolones



								ACD/logP[a]		ASNN1		KNN	
N	-R_1	-R_2	-R_3	-R_4	-R_5	-R_6	logP_{exp}[b]	logP_{calc}	logP_{calc} (after training)	logP_{calc}	logP_{calc} (after training)	logP_{calc}	logP_{calc} (after training)
							Substituted Analogues of Structure (I)						
1	-Et	-Et	-Me	-H	-H	-H	3.29	1.99	*in training*	2.88	*in memory*	1.22	*in memory*
2	-Et	-Et	-Me	-H	-Br	-H	4.09	2.96	4.26	3.98	3.91	4.63	3.98
3	-Et	-Et	-Me	-F	-H	-H	3.18	1.62	2.92	3.64	3.47	1.22	2.04
4	-Me	-Me	-Me	-H	-OMe	-OMe	2.65	0.92	2.21	2.28	2.48	1.4	1.4
5	-Et	-Et	-Me	-H	-OMe	-OMe	3.41	1.98	3.28	3.39	3.44	2.53	2.04
6	-Et	-Et	-CH_2Ph	-H	-Br	-H	5.17	4.74	6.04	4.91	5.11	4.63	4.28
7	-*i*-Pr	-*i*-Pr	-Me	-H	-Br	-H	4.88	3.66	4.96	4.77	4.35	4.23	4.20
8	-*i*-Pr	-*i*-Pr	-Me	-H	-OMe	-OMe	4.20	2.67	3.97	3.36	3.90	4.23	4.20
9	-Et	-Et	-CH_2Ph	-H	-OMe	-OMe	4.49	3.76	5.05	4.97	4.65	5.00	4.28
							MAE	*0.73*	*0.34*	*0.33*	*0.22*	*0.94*	*0.71*
							Substituted Analogues of Structure (II)						
10	-Et	-Et	-Me	-H	-H	-H	5.26	3.01	*in training*	3.67	*in memory*	1.86	*in memory*
11	-Et	-Et	-Et	-H	-Br	-H	7.01	4.31	6.56	4.54	6.21	3.38	4.97
12	-Me	-Me	-Me	-H	-Br	-H	5.61	2.72	4.97	3.72	4.89	2.76	4.92
13	-Et	-Et	-Me	-F	-H	-H	5.13	3.06	5.31	4.14	5.18	1.86	3.66
14	-*i*-Pr	-*i*-Pr	-Me	-H	-OMe	-Ome	5.53	3.44	5.69	4.23	5.81	2.68	3.66
15	-Me	-Me	-Me	-H	-OMe	-Ome	3.75	1.68	3.93	3.54	4.29	2.68	4.28
16	-Et	-Et	-Me	-H	-OMe	-Ome	4.51	2.75	5.00	4.02	5.27	3.17	3.66
17	-Et	-Et	-Me	-H	-Br	-H	6.37	3.78	6.03	4.08	5.74	3.38	4.97
18	-Et	-Et	-Ph	-H	-Br	-H	7.67	5.86	8.11	5.32	7.10	4.63	4.97
							MAE	*2.25*	*0.32*	*1.28*	*0.54*	*2.72*	*1.44*
						total	*MAE*	*1.49*	*0.33*	*0.81*	*0.38*	*1.83*	*1.08*

[a] ACD/logP values are from ref 10. [b] Experimental data are from ref 17. KNN were trained using PHYSPROP data set of 12 908 molecules.

Testing of ALOGPS Program by BASF. The lipophilicity program ALOGPS 2.0[1] provides a method to include proprietary compounds to the memory of ASNN using the so-called LIBRARY mode. This program was tested at BASF AG (Germany) with 6100 in-house compounds with known experimental *logP* values.[12] Two types of results were reported by BASF (Table 3). In the first set of results the program was applied to the investigated compounds without adding any of the BASF compounds to the memory of the ASNN. The calculated results were quite poor (Table 3) and were similar to the results for cross-series predictions using the star and nova sets.

In the second set of results, the BASF in-house compounds were added to the memory of the ASNN using the LIBRARY mode. The predictive ability of the program was estimated using the built-in leave-one-out method. The calculated results improved considerably and were similar to those calculated for the training set of molecules (Table 3). Thus the poor performance of the original ALOGPS was due to its systematic bias for a training set that did not contain BASF-type compounds. Contrary to that, the program with compounds added to its memory provided satisfactory prediction of compounds coming from the series used by this company, without the need to retrain the neural network ensemble.

These three examples clearly demonstrate that extension of the OPS of an ASNN dramatically improves the predictive ability of this method for the test set molecules without a need to retrain neural networks. The neural network results were superior to the KNN method that was applied using the same training sets in the space of input variables.

Analysis of the Nearest Neighbors in the Space of Neural Network Models. As it was shown for the analysis of artificial data sets and prediction of quinazolones, the nearest neighbors detected using ASNN could be used to interpret neural network results. The next example demonstrates a deeper analysis of this feature of ASNNs.

Figure 6A shows examples of the nearest neighbors detected for benzene using the Pearson's correlation coefficient in the output space of ASNN1. The nearest neighbors of benzene were its derivatives, i.e., ASNN correctly identified the nearest neighbors of this molecule from the chemical point of view. Use of the KNN method provided acceptable accuracy of benzene lipophilicity prediction using the detected nearest neighbors. For example, $logP_{KNN} = 2.73$, 2.42, 2.56, and 2.74 were predicted for benzene using 1, 2, 3, and 4 nearest neighbors, respectively.

Different nearest neighbor relations were observed if the same data set of 12908 molecules was used to predict another physical property of the compounds, namely their molecular

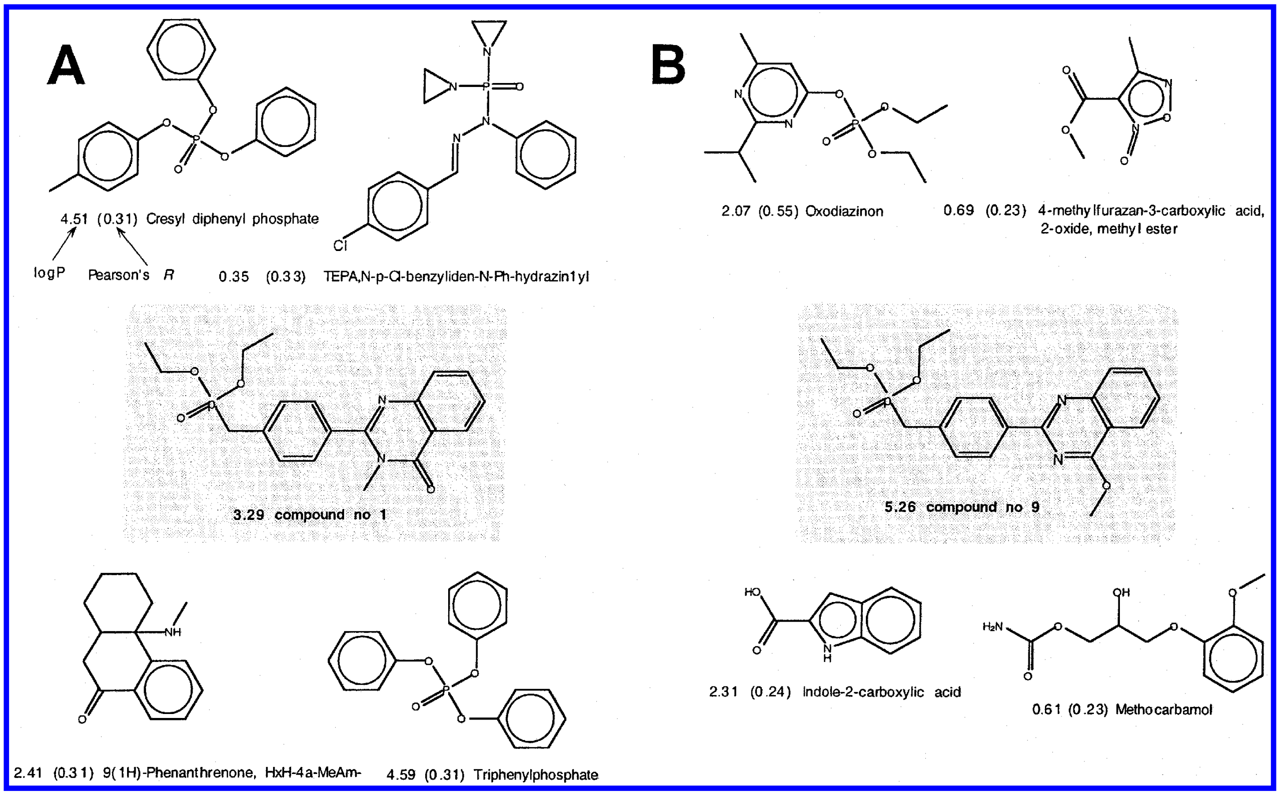

A
4.51 (0.31) Cresyl diphenyl phosphate
logP
Pearson's R
0.35 (0.33) TEPA,N-p-Cl-benzyliden-N-Ph-hydrazin1yl
3.29 compound no 1
2.41 (0.31) 9(1H)-Phenanthrenone, HxH-4a-MeAm-
4.59 (0.31) Triphenylphosphate
B
2.07 (0.55) Oxodiazinon
0.69 (0.23) 4-methylfurazan-3-carboxylic acid, 2-oxide, methyl ester
5.26 compound no 9
2.31 (0.24) Indole-2-carboxylic acid
0.61 (0.23) Methocarbamol

Figure 5. Nearest neighbors detected by ASNN1 for compound nos. 1 (A) and 9 (B) in the quinazolones set.[17]

Table 3. Distribution of Residuals (Absolute Values in %) between the Experimental and the Predicted logP Values for the ALOGPS[a] Program

		prediction of 6100 BASF compounds	
$\lvert \mathrm{logP}_{\mathrm{pred}}\text{-}\mathrm{logP}_{\mathrm{exp}} \rvert$	leave-one-out results for 12 908 training set molecules	"as is" results[b]	leave-one-out results[c]
0−0.3	63	30	60
0−0.5	81	49	80
0−1.0	96	82	96
0−2.0	99	98	99

[a] ALOGPS is an artificial neural network program for calculation of logP and logS.[1] This program is available on-line at http://www.lnh.unil.ch/~itetko/logp. [b] ALOGPS results for the test set compounds. [c] The same results after adding of the BASF compounds to the memory of the ASNN.

weight. The ASNN calculated significant correlation between the indices and molecular weight, $r^2 = 0.99$, $RMSE = 3.3$, i.e., modeling of molecular weight was more accurate than that of lipophilicity. However, different molecules were found as the nearest neighbors of benzene (Figure 6B). Pyridine, pyrimidine, and pyrazine had very similar properties to benzene from the point of view of molecular weight. The difference between these compounds corresponds to a small change in the molecular weight between carbon and nitrogen.

An attempt to predict logP value using the KNN method was not successful in the space of molecular weight models. For example, $logP_{KNN} = 0.65$, 1.69, 0.99, and 0.68 was predicted for benzene using 1, 2, 3, and 4 nearest neighbors, respectively. This result was much worse than the similar analysis in the lipophilicity space, i.e., the detected relationships in the molecular weight space were invalid from the point of lipophilicity estimation. On the contrary in the molecular weight space one could get good results for the molecular weight prediction, e.g. $MW_{KNN} = 79$, 86, 84, 83 were predicted for benzene using 1, 2, 3, and 4 nearest neighbors. Thus, it is easily evident, the clustering in space of models is taking into account the target property.

The nearest neighbor relations detected in the output space, of course, critically depend on the choice of input parameters. If the parameters are not adequate, it is difficult to expect good interpretable clustering in output space and good performance of ASNN. For example, let us develop lipophilicity and molecular weight prediction programs with some kind of "atom-type" based approach, e.g. using counts of H, C, N, O, P, S, Br, Cl, F, I atoms available in the PHYSPROP set as input indices. To avoid linear fitting for the molecular weight prediction, a square of this parameter was used as the target property for neural network training. The ASNN provided very good results, $r^2 = 1$ and $RMSE = 0.005$, for the square of molecular weight prediction. The lipophilicity of compounds was not well predicted using the counts of atoms; $RMSE = 0.87$, MAE = 0.61 ($RMSE = 0.60$, MAE = 0.48 without 999 outliers) calculated using $k = 45$. This result was much worse than that developed using E-state indices. The detected nearest neighbors of benzene for molecular weight and lipophilicity were quite different from the analyses using E-state indices. The atom counts were not able to distinguish aromatic and nonaromatic compounds, and this was immediately reflected in the detected neighborhood relations for the investigated compounds (Figure 6). However, similarly to the previous example there are no nitrogen containing compounds of benzene in the lipophilicity space (Figure 6C), while such compounds are among the neighbors of this compound in the molecular weight space (Figure 6D). Again, much better prediction of the lipophilicity of benzene is calculated using

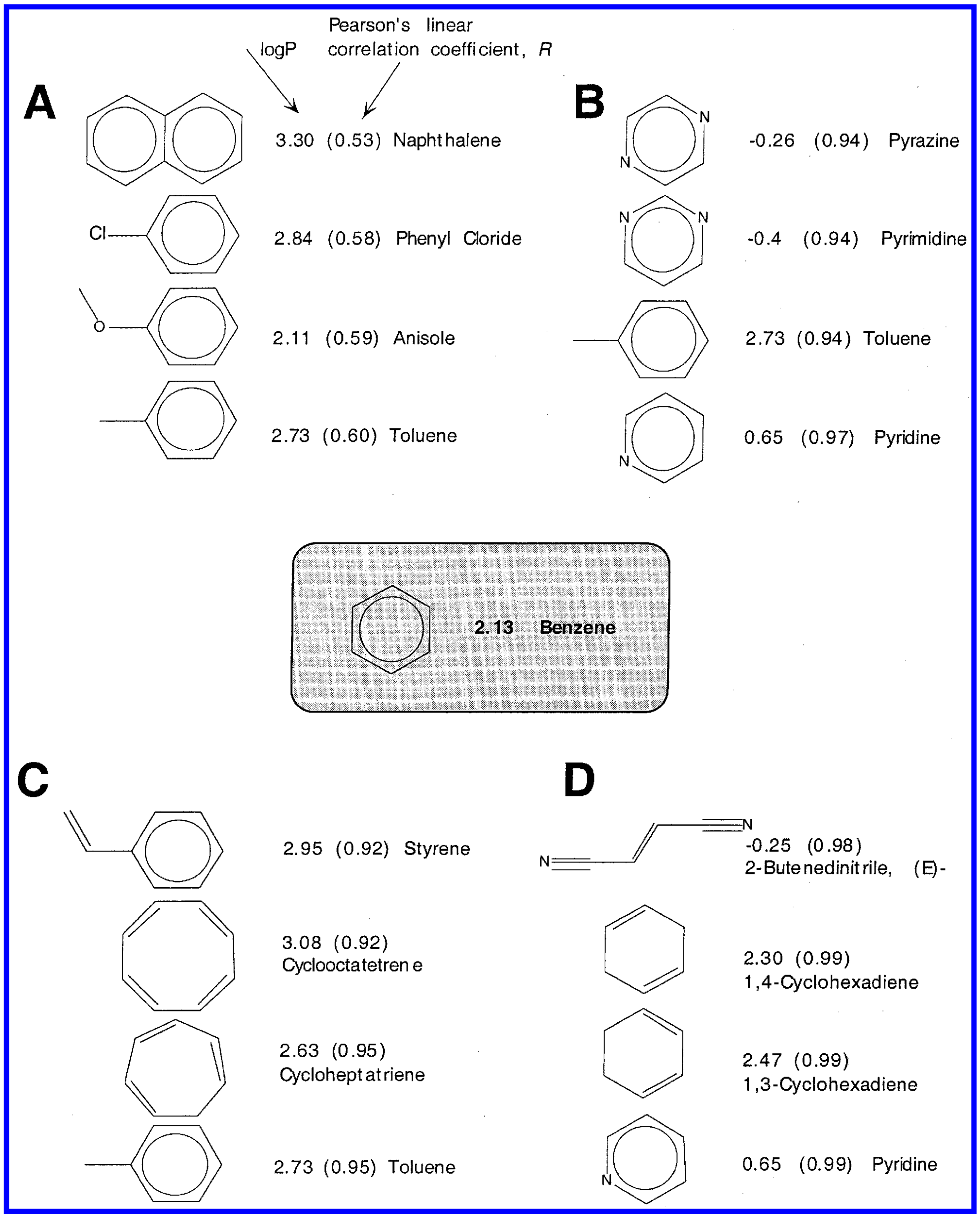


Figure 6. The nearest neighbors of benzene calculated in the space of neural network models. Neural networks trained to predict lipophilicity (A) and molecular weight (B) using 75 E-state indices detected only aromatic compounds as the nearest neighbors of benzene. On the contrary, both aromatic and nonaromatic compounds were detected as the nearest neighbors for lipophilicity (C) and molecular weight models (D) developed using atom counts.

KNN method in the lipophilicity space compared to the molecular weight space.

These examples demonstrate that analysis in the space of neural networks provided a clustering of the molecules that is dependent on the predicted property, i.e., “property-targeted” clustering. Thus, despite the same input parameters being used to predict lipophilicity and molecular weight, the nearest neighbors detected for analyzed molecules in the space of models were completely different. This feature of ASNNs is very important in the interpretation of the calculated results, since one can have immediately see if detected nearest neighbors are representative of the analyzed molecule.

DISCUSSION

When predicting a property of an object, a person uses some global knowledge as well as particular properties of other similar objects (local knowledge). The final prediction represents the weighted sum of both these contributions. Equation 7 provides a simple model of such a process and proposes how memoryless and memory-based methods can be combined to improve their performances. The correction scheme used in an ASNN is easy to understand. The method searches the nearest neighbors of the analyzed case, calculates the average error for such cases, and corrects prediction of the new case according to this error. The major benefits of ASNNs are that the nearest neighbors are detected in the space of output models (given by trained neural networks) and not in the space of input parameters. The proposed method is simple and can be easily incorporated into existing neural network programs.

ASNNs can have a significant impact for practical applications in different fields of science and industry. One main application of this method, an improvement of generalization ability of the method by incorporation of new data, was clearly demonstrated in this article using a simulated data set and several examples of lipophilicity prediction. Indeed, new data made it possible to extend the OPS of the neural networks and to ultimately give a more predictive model when looking at the BASF compounds. An alternative approach would have been to retrain the artificial neural networks using all available data. Such retraining would probably provide a better model, as was demonstrated for the artificial data set analysis (ca. Figures 1, 2, and 4), although this may not be logistically possible. As it is not clear if the user would be proficient or willing enough to do such work (“the user’s restrictions”) and as well there may be privacy restrictions in providing the initial data that was used to develop the neural network model (“the developer’s restrictions”). In addition, there is a multitude of other problems that may be encountered in developing a predictive neural network model, i.e., chance effects, local minima, and appropriate parameter selection.

The memory-based user training of ASNNs has significant advantages over the user training feature of fragmental based approaches, such as the one used in the ACD/logP. The main conceptual difference is that no additional indices are required to provide user training in the ASNN algorithm. On the contrary fragmental based approaches may provide lower predictive capabilities due to fundamental problems related to the fragmental method, e.g. missing fragments. Indeed, the number of molecules initially supplied by the user can be too small to cover all new fragments. The performance of fragmental methods may also be poorer due to incorrect fragmental increments derived from unreliable experimental data. All these factors could be accumulated during the program development.[10]

The ASNN method searches analogues in its memory. This search is done in the space of neural network predictions, and, thus, there is no requirement to keep any input parameters, such as molecular indices, derived from structures of the molecules. The only information required for a search are the target values predicted by each network in the ensemble and experimental values of the molecules. This provides an easy way to improve predictive abilities of the ASNN method by incorporating private data from pharmaceutical firms as firms could calculate a vector of responses for their molecules using ANNE and provide such information as a public library for other users. This information could then be included in the memory of an ASNN improving its predictive abilities. Note that in this case any input indices, even explicitly coded specific structural fragments, can be used as input parameters. Such indices will be processed by nonlinear transformations on the hidden neurons of neural network and thus not disclose the underlining chemical structure. To this extent, a vector of responses of neural networks provides one more step ahead in molecular coding security compared to the use of the E-state indices.

Another important application of ASNN concerns development of the data clustering algorithms. As it was shown in the article, the neighboring relations detected amid the molecules in the space of neural network predictions depend on the analyzed property. The example with artificial data set has also indicated that not only local but also global features of the analyzed function such as symmetry or simple additive rules could be taken into account in such analyses. Analysis in the output space could provide much better clustering compared to that of the input space of parameters, in relation to the target property. This “property-targeted” diversity analysis can be an extremely valuable tool for drug design and screening of virtual libraries.[28−30] A properly trained ASNN could significantly improve selection of lead compounds and dramatically improve this field compared to a traditional neural network approach.[31,32]

The correlation coefficient calculated for the nearest neighbors could be used to determine reliability of ASNN predictions. For example, the nearest analogues calculated for the quinazolones with general structure (II) had on average smaller correlation coefficients compared to those with general structure (I) and benzene (Figure 5). This may explain the lower predictive ability of the method for this subset of compounds. Such comparison, however, should always be done inside of one model and cannot be used to judge the quality of different models. For example, much higher correlation coefficients were calculated for nearest neighbors of benzene using atom counts, compared to the use of E-state indices for lipophilicity predictions (Figure 6). Despite this the lipophilicity model developed with atom counts was much less accurate. Analysis of the nearest neighbor errors can also estimate the prediction errors. The simplest way to estimate the prediction error of an analyzed case would be to use the actual errors from the k nearest neighbors.

Efficient neural network training algorithms, such as Levenberg−Marquardt algorithm, Newton’s or conjugate gradient algorithms have $O(N^2)$ computational costs.[33] Thus, such methods could not be used for large data sets with hundreds of thousands of data cases. At the same time the use of the first-order training algorithms can be inefficient due to problem of local minima. ASNNs can provide an elegant solution to this problem. First, the powerful second-order methods could be trained using a limited subset of data. Second, all data could be used in the memory of ASNN. Thus, if unused data are lost for traditional neural networks, ASNN makes it possible to utilize each sample of such data for the nearest neighbor correction and thus can provide a gradual increase in the accuracy of the method as new data will become available.

The artificial neural network is not the sole method that can be used in ASNNs. For example, other nonlinear global minimization methods, i.e., a pool of models calculated with evolutionary programming, genetic algorithms,[34,35] or with the group method of data handling algorithms,[36,37] could also be used instead of the feed-forward neural network models. In such studies each model could have its own set of input variables. There is also no restriction to use more traditional approaches, such as principal component analysis, partial least squares, multiple linear regression models, or even human experts. For example, an ensemble of models for lipophilicity prediction could include all models developed so far in this field!

Another possibility to further develop ASNNs is to investigate whether other local regression techniques, e.g. multiple linear regression analysis or even ASNN, could be used instead of the KNN method. The initial input parameters could also be provided in addition to or instead of the correlation coefficients. To this extent the clustering in space of models will provide some kind of data filtering. Such analysis could be repeated several times using a successfully smaller amount of data. The result of this modeling could be a neural network consisting of several layers of ASNN. Each layer will provide a more and more accurate model using decreasing numbers of data cases preselected by lower-order layers. A similar idea to improve neural network predictions has been implemented in a twice-multilayered neural network.[38] The important conceptual difference is that the twice-multilayered neural network tries to calculate a global model, while the purpose of multilayered ASNN is to provide only the best possible local model, i.e., to predict just one case but with the highest possible confidence and accuracy. Of course, by calculating the best models for each particular case one will have the global model too.

The ASNN method is not the sole method that provides analysis in the space of output models. A very similar idea, clustering in the space of neural network weights, represents the basis of a recently proposed Volume Learning Algorithm (VLA).[39,40] The main feature of the VLA is to decrease the dimensionality of input data space that can be up to tens of thousands of inputs, by clustering the input-to-hidden neuron weights. While both methods have completely different targets, their main ideas are very similar: analysis of data preprocessed according to some target activity provides better clustering than the use of the raw parameters.

The idea to look for similarity of data cases in the space of models was first used to develop an efficient partition algorithm for neural network training.[22,23] The concept to estimate systematic error (bias) of nearest neighbors in the output space was developed in January−March 2001,[12] and the first description of the method was published on the web in April 2001.[13] Thus, this method is still in its relative infancy.

The main beneficial properties of this method are as follows: enhanced generalization ability compared to traditional back-propagation and KNN methods, easily extendable OPS without the need for retraining weights, simple visualization, and improved clustering of analyzed data cases in the property-targeted space. All these benefits make ASNNs an attractive tool for a wide range of applications in the fields of chemoinformatics, medicinal, and physical chemistry.

ACKNOWLEDGMENT

This study was partially supported by Virtual Computational Chemistry Laboratory, INTAS-OPEN 00-0363 grant. I thank Hugo Kubinyi (BASF AG, Germany) for the testing of the ALOGPS program and Javier Iglesias (Computer Science Institute, University of Lausanne) for providing me access to the Linux cluster used in this study. I also thank David Livingstone and Martin Ford (University of Portsmouth, U.K.) for useful discussions and Ben Tehan (Monash University, Australia) for careful reading of the manuscript. The calculation of E-state indices were done using program developed by Vsevolod Yu. Tanchuk (Institute of Bioorganic & Petroleum Chemistry, Kiev, Ukraine).

REFERENCES AND NOTES

(1) Tetko, I. V.; Tanchuk, V. Y.; Villa, A. E. P. Prediction of *n*-Octanol/Water Partition Coefficients from PHYSPROP Database Using Artificial Neural Networks and E-state Indices. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 1407−1421.

(2) Gombar, V. K. Reliable assessment of logP of compounds of pharmaceutical relevance. *SAR QSAR Environ. Res.* **1999**, *10*, 371−380.

(3) Leo, A. J.; Hoekman, D. Calculating log P(oct) with no Missing Fragments; The problem of Estimating New Interaction Parameters. *Persp. Drug Discov. and Design* **2000**, *18*, 19−38.

(4) Meylan, W. M.; Howard, P. H. Atom/fragment contribution method for estimating octanol−water partition coefficients. *J. Pharm. Sci.* **1995**, *84*, 83−92.

(5) Petrauskas, A. A.; Kolovanov, E. A. ACD/LogP Method Description. *Persp. Drug Discov. Design* **2000**, *19*, 1−19.

(6) Wang, R.; Gao, Y.; Lai, L. Calculating Partition Coefficient by Atom-Additive Method. *Persp. Drug Discov. Design* **2000**, *19*, 47−66.

(7) Ghose, A. K.; Viswanadhan, V. N.; Wendoloski, J. J. Prediction of hydrophobic (lipophilic) properties of small organic molecules using fragmental methods − an analysis of ALOGP and CLOGP methods. *J. Phys. Chem.* **1998**, *102*, 3762−3772.

(8) BioByte Corp., 201 W. Fourth Street, Claremont, CA.

(9) Syracuse Research Corporation. Physical/Chemical Property Database (PHYSPROP); SRC Environmental Science Center: Syracuse, NY.

(10) ACD/LogP Prediction Software. An Overview and Comparison; ACD, Inc.: 133 Richmond St. W., Suite 605, Toronto ON M5H 2L3, Canada, 2001.

(11) Zupan, J.; Gasteiger, J. *Neural Networks for Chemistry and Drug Design: An Introduction*; 2nd ed.; VCH: Weinheim, 1999.

(12) Tetko, I. V. Associative Neural Network. *Neur. Proc. Lett.* **2001**, submitted for publication.

(13) Tetko, I. V. Associative Neural Network, CogPrints Archive, cog00001441, available as http://cogprints.soton.ac.uk/documents/disk0/00/00/14/41/index.html, 2001.

(14) Kier, L. B.; Hall, L. H. An electrotopological-state index for atoms in molecules. *Pharm. Res.* **1990**, *7*, 801−807.

(15) Hall, L. H.; Kier, L. B. Electrotopological state indices for atom types − a novel combination of electronic, topological, and valence state information. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 1039−1045.

(16) Kier, L. B.; Hall, L. H. *Molecular Structure Description: The Electrotopological State*; Academic Press: London, 1999.

(17) Kurogi, Y.; Inoue, Y.; Tsutsumi, K.; Nakamura, S.; Nagao, K.; Yoshitsugu, H.; Tsuda, Y. Synthesis and hypolipidemic activities of novel 2-[4-[diethoxyphosphoryl)methyl]phenyl]quinazolines and 4(3H)-quinazolinones. *J. Med. Chem.* **1996**, *39*, 1433−1437.

(18) Dasarthy, B. *Nearest Neighbor (NN) Norms*; IEEE Computer Society Press: Washington, DC, 1991.

(19) Härdle, W. *Smoothing Techniques with Implementation in S*; Springer-Verlag: New York, 1990.

(20) Lawrence, S.; Tsoi, A. C.; Back, A. D. Function Approximation with Neural Networks and Local Methods: Bias, Variance and Smoothness. *Australian Conference on Neural Networks*; Australian National University: Australian National University, 1996; pp 16−21.

(21) Geman, S.; Bienenstock, E.; Doursat, R. Neural Networks and the Bias/Variance Dilemma. *Neural Computation* **1992**, *4*, 1−58.

(22) Tetko, I. V.; Villa, A. E. P. Efficient Partition of Learning Data Sets for Neural Network Training. *Neural Networks* **1997**, *10*, 1361−1374.

(23) Tetko, I. V.; Villa, A. E. P. Unsupervised and Supervised Learning: Cooperation toward a Common Goal. *ICANN'95, International Conference on Artificial Neural Networks NEURONIMES'95*; EC2 & Cie: Paris, France, 1995; pp 105−110.

(24) Press: W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P. *Numerical Recipes in C*, 2nd ed.; Cambridge University Press: New York, 1994; p 998.

(25) Tetko, I. V.; Tanchuk, V. Y. Application of Associative Neural Networks for Prediction of Lipophilicity in ALOGPS 2.1 program. *J. Chem. Inf. Comput. Sci.* **2002**, submitted for publication.

(26) 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.

(27) Bishop, M. *Neural Networks for Pattern Recognition*; Oxford University Press: Oxford, 1995.

(28) Xue, L.; Bajorath, J. Molecular descriptors in chemoinformatics, computational combinatorial chemistry, and virtual screening. *Comb. Chem. High. Throughput Screen.* **2000**, *3*, 363−372.

(29) Keseru, G. M.; Molnar, L.; Greiner, I. A neural network based virtual high throughput screening test for the prediction of CNS activity. *Comb. Chem. High. Throughput Screen.* **2000**, *3*, 535−540.

(30) Oprea, T. I.; Davis, A. M.; Teague, S. J.; Leeson, P. D. Is There a Difference between Leads and Drugs? A Historical Perspective. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 1308−1315.

(31) Sadowski, J.; Kubinyi, H. A scoring scheme for discriminating between drugs and nondrugs. *J. Med. Chem.* **1998**, *41*, 3325−3329.

(32) Ajay, A.; Walters, W. P.; Murcko, M. A. Can we learn to distinguish between "drug-like" and "nondrug-like" molecules? *J. Med. Chem.* **1998**, *41*, 3314−3324.

(33) Shepherd, A. J. *Second-Order Methods for Neural Networks*; Springer-Verlag: London, 1997; p 145.

(34) Mitchell, M. *An Introduction to Genetic Algorithms*; The MIT Press: Cambridge, MA, 1996.

(35) Tetko, I. V.; Tanchuk, V. Y.; Vasil'ev, S. A.; Khil'ya, V. P.; Poda, G. I.; Luik, A. I. Evolutionary computation for revealing structure−activity relationships in 3-phenoxychromone and 3-phenoxy-4-hydroxycoumarin derivatives. *Bioorg. Khim.* **1995**, *21*, 809−815.

(36) Ivakhnenko, A. G.; Ivakhnenko, G. A. Problems of Further Development of the Group Method of Data Handling Algorithms. Part I. *Pat. Recog. Imag. Anal.* **2000**, *10*, 187−194.

(37) Tetko, I. V.; Aksenova, T. I.; Volkovich, V. V.; Kasheva, T. N.; Filipov, D. V.; Welsh, W. J.; Livingstone, D. J.; Villa, A. E. P. Polynomial neural network for linear and nonlinear model selection in quantitative-structure activity relationship studies on the Internet. *SAR QSAR Environ. Res.* **2000**, *11*, 263−280.

(38) Ivakhnenko, A. G.; Ivakhnenko, G. A. Self-Organization of Neural Networks with Active Neurons. *Pat. Recog. Imag. Anal.* **1994**, *4*, 185−196.

(39) Tetko, I. V.; Kovalishyn, V. V.; Livingstone, D. J. Volume learning algorithm artificial neural networks for 3D QSAR studies. *J. Med. Chem.* **2001**, *44*, 2411−2420.

(40) Kovalishyn, V. V.; Tetko, I. V.; Luik, A. I.; Chretien, J. R.; Livingstone, D. J. Application of Neural Networks Using the Volume Learning Algorithm for Quantitative Study of the Three-Dimensional Structure−Activity Relationships of Chemical Compounds. *Bioorg. Khim.* **2001**, *27*, 267−277.

CI010379O