A Combined Application of Two Different Neural Network Types for the Prediction of Chemical Reactivity

Vera Simon,[†] Johann Gasteiger,[*,†] and Jure Zupan[‡]

Contribution from the Organisch-chemisches Institut, Technische Universität München, Lichtenbergstrasse 4, D-85747 Garching, Germany, and National Institute of Chemistry, 61115 Ljubljana, Slovenia

Received February 5, 1993[®]

Abstract: A multilayer neural network trained by the back-propagation algorithm is developed that is able to predict which single bonds in aliphatic molecules will break preferentially. Each potential bond breaking is described by seven empirical physicochemical parameters that allow the treatment of large datasets of organic molecules. This makes the approach outlined in this paper attractive for an automatic learning from reaction databases. It is demonstrated that a Kohonen network can be used as a basis for the selection of a training set for a supervised learning method. For training a multilayer neural network this selection gives results that are superior to a random selection and also to an experimental design technique. A detailed analysis of the Kohonen mapping shows that the chemical similarity of bond breakings is perceived by the topology-conserving Kohonen mapping of a multidimensional space.

1. Introduction

One of the fundamental tasks in organic chemistry is the prediction of the course of chemical reactions. A chemical reaction consists of the breaking of one or more bonds in the reactants and then the making of one or more new bonds to give the products. Reaction mechanisms are sequences of such single reaction steps. In order to predict how a molecule will react, information about the most reactive bonds within the molecule is needed.

For several years databases on chemical reactions have been increasingly used by chemists to determine if a particular reaction can be achieved or not. Conclusions are mostly drawn by analogy. In such reaction databases, the bonds broken and made in a reaction are marked. It would be highly desirable to have automatic methods that can learn and then generalize on chemical reactivity from such instances of chemical reactions.

Chemists base their knowledge about the reactivity of bonds on a variety of effects that are of energetic, electronic, and steric nature. However, these effects are mostly used in a qualitative manner, and the prediction of reactivity is quite often an intuitive process. By looking for functional groups a chemist can pinpoint the bonds that are potentially breakable. However, if there are several functional groups present that interfere and compete with one another, bonds that are not part of functional groups might become reactive.

A solution to this problem can be found if quantitative values can be assigned to the various chemical effects influencing the reactivity of a bond. In the last decade, empirical methods have been developed for the calculation of the magnitude of electronic and energy effects. The values calculated by these methods have been used to correlate or calculate physical data and thus their physicochemical significance has been established.[1] On the basis of these results the use of such parameters for correlating and predicting chemical reactivity and for proposing detailed mechanisms of organic reactions was investigated.[2,3]

Statistical and pattern recognition methods allowed the extent of the contribution of the various chemical effects to reactivity to be deciphered.[4,5] Equations were derived for the calculation of quantitative values on chemical reactivity, mostly by multilinear regression analyses and by the partial least-squares (PLS) technique. Such an approach implies that the relationship between chemical reactivity and the values of the electronic and energy effects is of a linear nature or can easily be transformed into a linear problem. Such a simple linear model rarely describes the often complex problem satisfactorily.

Recently a new approach to the problem of prediction of chemical reactivity has been tried. Instead of modeling the relationship in an explicit function, an associative memory system has been developed that stores the relationship in an implicit manner in computer memory.[6] In this way, the power of an associative memory system[7] as a model of the information processing method in the cerebellum has been applied to chemical data.

In recent years, models of the information processing method in the human brain have gained much prominence. The development of such models, called artificial neural networks, or neural networks for short, picked up momentum with the publication of Hopfield's paper in 1982 that introduced the concept of nonlinearity.[8] The decisive impulse, however, came when Rumelhart, Hinton, and Williams introduced in 1986 the idea of learning by error back-propagation in a multilayer neural network.[9] Whatever the overwhelming success of this particular neural network model, it should not be forgotten that other neural network models have been developed that have important merits and deserve closer inspection and more widespread application. This is particularly true for the Kohonen network which is a "self-organized topological feature map" and bears close similarity to the information processing in the sensory, auditory, and visual cortex of the human brain. The Kohonen network was developed between 1979 and 1982[10] and is described in detail in a

[†] Technische Universität München.

[‡] National Institute of Chemistry.

[®] Abstract published in *Advance ACS Abstracts*, September 1, 1993.

[1] Gasteiger, J. In *Physical Property Prediction*; Jochum, C., Hicks, M. G., Sunkel, J., Eds.; Springer: Heidelberg, 1988; pp 119–138.

[2] Gasteiger, J.; Hutchings, M. G.; Christoph, B.; Gann, L.; Hiller, C.; Löw, P.; Marsili, M.; Saller, H.; Yuki, K. *Top. Curr. Chem.* **1987**, *137*, 19–73.

[3] Gasteiger, J.; Marsili, M.; Hutchings, M. G.; Saller, H.; Löw, P.; Röse, P. J.; Rafeiner, K. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 467–476.

[4] Gasteiger, J.; Saller, H.; Löw, P. *Anal. Chim. Acta* **1986**, *191*, 111–123.

[5] Gasteiger, J.; Schulz, K. P.; Kredler, C. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 385–394.

[6] Schulz, K. P.; Gasteiger, J. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 395–406.

[7] Albus, J. S. In *Brains, Behaviour, and Robotics*; Byte Books: Petersborough, NH, 1981; pp 139–180.

[8] Hopfield, J. J. *Proc. Natl. Acad. Sci. U.S.A.* **1982**, *79*, 2554–2558.

[9] Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*; MIT Press: Cambridge, MA, 1986; Vol. 1, pp 318–362.

[10] Kohonen, T. *Biological Cybernetics* **1982**, *43*, 59–69.

0002-7863/93/1515-9148$04.00/0

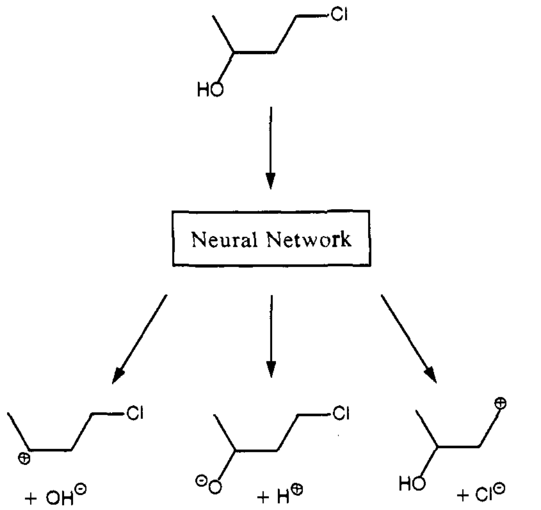


Figure 1. Objective of the study: prediction of polar reactivity by a neural network.

monograph.[11] Kohonen's work is related to the work of Willshaw and von der Malsburg.[12]

Neural networks have also been applied to a host of chemical problems,[13] mostly by resorting to the back-propagation algorithm. Several applications of neural networks to chemical reactions have been published.[14-16] In their first study, Elrod, Maggiora, and Trenary investigated the regioselectivity in the nitration of monosubstituted benzene derivatives.[14] Subsequently, they investigated a representation of chemical structures for the treatment of additions to alkenes, Diels–Alder and retro-Diels–Alder reactions, and Saytzeff eliminations.[15] Luce and Govind developed a neural network for retrosynthetic analysis.[16] Neural networks offer the possibility of following the reasoning of the chemist in deriving knowledge about chemical reactivity. The chemist makes observations on which parts (bonds) of a molecule react. Based on a series of such observations he or she makes generalizations and derives rules which allow him to draw inferences on which bonds of a given molecule will preferentially break.

In analogy to this, we wanted to develop a neural network that, when presented with a structural formula, would point out those bonds that break preferentially (Figure 1).

Two different neural network models are used in this study, a multilayer neural network trained by the back-propagation algorithm[9] and the Kohonen network trained by competitive learning.[11] This exploits the specific characteristics of these two different methods in analyzing and modeling data on chemical reactivity.

It is shown that a Kohonen network is a powerful tool for the selection of optimum datasets in comparison to a random selection and to a selection by experimental design techniques.[18] Furthermore, it is demonstrated that the topology conserving mapping by a Kohonen network maintains important chemical relationships and shows chemical similarities. A Kohonen network has already been used in a quantitative structure activity study.[17]

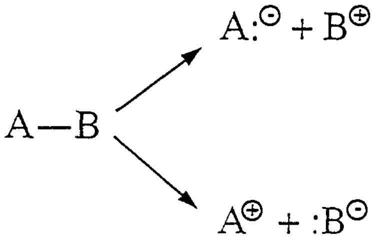


Figure 2. The two possibilities for heterolysis of a bond.

The back-propagation algorithm is used to model the relationships between electronic and energy parameters of bonds in an organic molecule and their propensity for heterolysis. This implicit nonlinear modeling is used for a classification of bonds into reactive and non-reactive and thus for the prediction of the ease of polar breaking of a chemical bond.

2. Polar Bond Breaking

The aim of this study is to develop a neural network that is able to predict the polar reactivity of single bonds in aliphatic compounds. In particular, the neural network should predict which bonds in an aliphatic molecule are preferentially broken to give positively and negatively charged species.

A dataset of 29 aliphatic molecules was selected that covers the range of polar reactivity in aliphatic compounds as broadly as possible. This dataset contains 385 single and double bonds which correspond to 770 polar bond breakings as each bond can break in two directions (Figure 2).

Interest was only focused onto the 724 breakings of *single* bonds. As quite a few of the bonds are equivalent (e.g., the three C–H bonds of a methyl group) the dataset can be reduced to 373 chemically different breakings of single bonds. From these bond breakings 116 were selected and classified by chemists as either reactive or non-reactive. Altogether 42 bond breakings were considered reactive and 74 non-reactive. In a later stage of the study it was found necessary (see section 8) to classify an additional 33 bond breakings as reactive (1) or non-reactive (32). Thus, altogether 149 bond breakings were classified as 43 reactive and 106 non-reactive. All 149 bond breakings are indicated in Scheme I.

The dataset of 116 bond breakings has already been studied with various statistical and pattern recognition methods and with an associative memory system (AMS).[4-6]

3. Neural Networks

A variety of different types of artificial neural networks have been developed to model different activities of the human brain. These computerized neural networks can be applied to various tasks in information processing: classification, modeling, association, and mapping.[13,19] The different neural network models are applicable to these types of problems to different extents.

In this paper we will demonstrate the capability of a multilayer neural network trained with the back-propagation algorithm for *classifying* chemical data. In particular, we will show how the susceptibility of a bond in an aliphatic organic molecule for polar breaking can be predicted from a series of electronic and energy variables calculated for this bond breaking.

In addition, we will show the merits of a Kohonen network for *mapping* multidimensional data. Thus, both supervised and unsupervised learning techniques are employed in analyzing data on chemical reactivity.

Multilayer Neural Network. The purpose of a multilayer neural network is to convert a series of input data **x** (x_1, x_2, ..., x_i, ..., x_m) into one or more output data y (y_1, y_2, ..., y_j, ..., y_n). In this sense one can conceive such a neural network as an alternative

[11] Kohonen, T. *Self-Organization and Associative Memory*, 2nd ed.; Springer: New York, 1988.

[12] Willshaw, D. J.; von der Malsburg, C. *Proc. R. Soc. London B* **1976**, *194*, 431.

[13] Zupan, J.; Gasteiger, J. *Anal. Chim. Acta* **1991**, *248*, 1–30.

[14] Elrod, D. W.; Maggiora, G. M.; Trenary, R. G. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 477–484.

[15] Elrod, D. W.; Maggiora, G. M.; Trenary, R. G. *Tetrahedron Comput. Meth.* **1990**, *3*, 163–194.

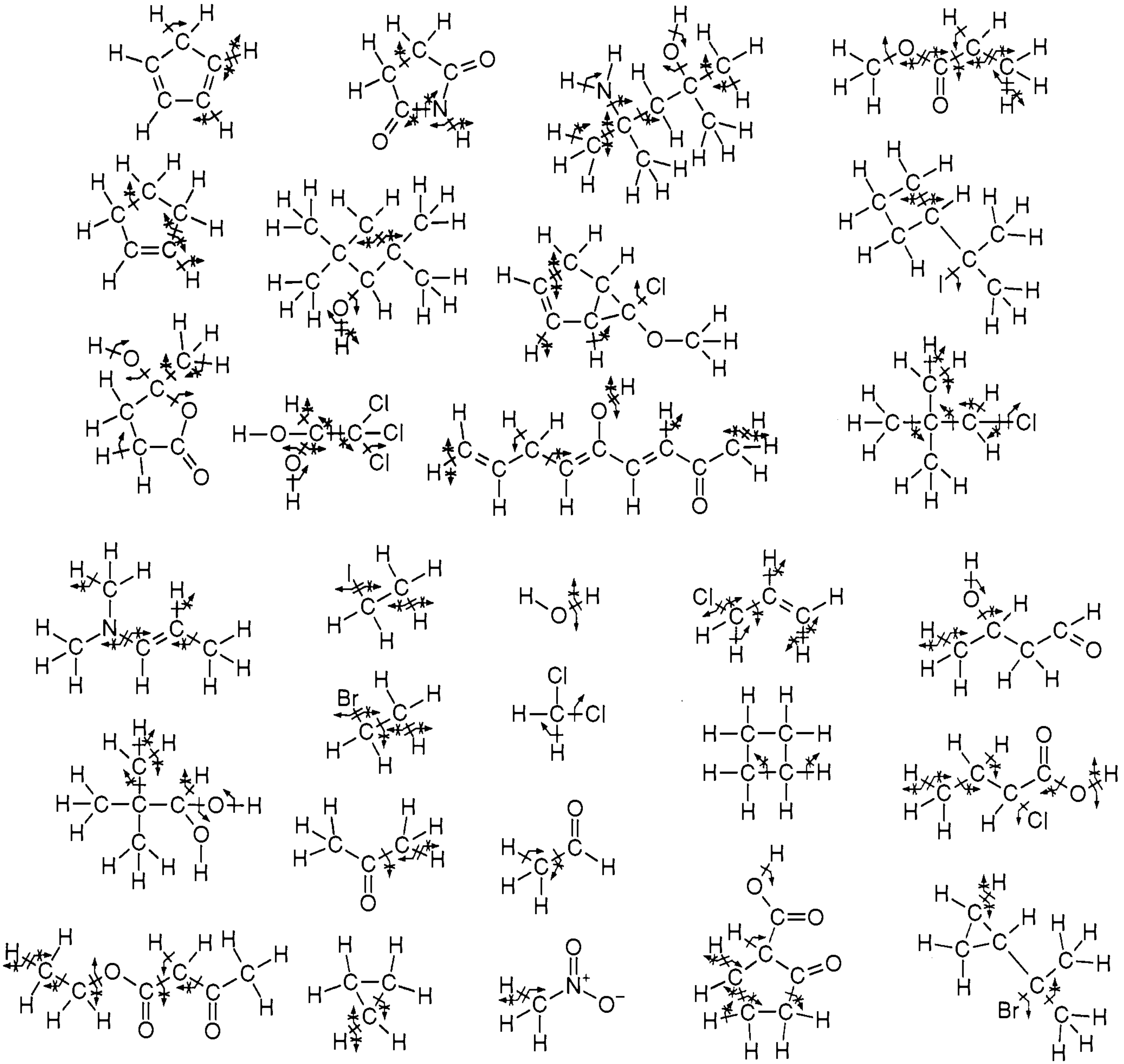
[16] Luce, H. H.; Govind, R. *Tetrahedron Comput. Meth.* **1990**, *3*, 143–161.

[17] Rose, V. S.; Crooll, I. F.; MacFic, H. J. H. *Quant. Struct. Act. Relat.* **1991**, *10*, 6–15.

[18] Deming, S. N.; Morgan, S. L. *Experimental Design: A Chemometric Approach*; Elsevier: Amsterdam, 1987.

[19] Gasteiger, J.; Zupan, J. *Angew. Chem.* **1993**, *105*, 510–536; *Angew. Chem., Int. Ed. Engl.* **1993**, *32*, 503–527. Zupan, J.; Gasteiger, J. *Neural Networks for Chemists—An Introduction*; VCH: Weinheim, 1993.

Scheme I. Dataset of 29 Molecules with 149 Classified Bond Breakings[a]



[a] Arrows indicate the shift in the electron pair on polar breaking of a bond. Bond breakings classified non-reactive are marked by a crossed arrow.

to a set of explicit mathematical functions, f_j, relating $\mathbf{y}_j$ to $\mathbf{x}$ (eq 1).

$$\mathbf{y}_j = \mathrm{f}_j(\mathbf{x}) \qquad j = 1, ..., n \qquad (1)$$

However, a neural network does not express such a relationship in an explicit manner but contains it internally in an implicit manner (in the form of weights as will be shown next).

Figure 3 shows a feed-forward two-layer neural network. Squares indicate input units and circles represent neurons. Each neuron of one layer is connected with each neuron of the layers directly above and below. These connections are associated with weights that represent the strength of a connection. The top layer is called the input layer and the bottom layer is called the output layer. There exists one additional unit, called bias, which is connected with each neuron of all the non-input layers. All neurons of each level perform the same basic operations: they obtain input signals (data), convert these signals to a net input signal (Net), and then transform this signal into an output signal (out) (Figure 4).

In calculating the net input of a neuron, j, the inputs $\mathbf{x}$ (x_1, x_2, ..., x_i, ..., x_m) from all m neurons of the preceding layer are taken into account. The values of x_i are usually scaled into the range from 0 to 1 or from −1 to +1. The net value of neuron j, Net$_j$, is calculated from all m input data, x_i.

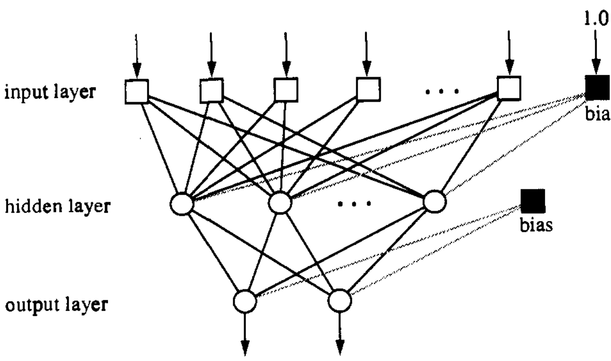


Figure 3. Two-layer neural network. Squares indicate input units and circles neurons; only the layers of the neurons are counted in specifying the number of layers. The information flows from top (input) to bottom (output). The black square indicates the bias.

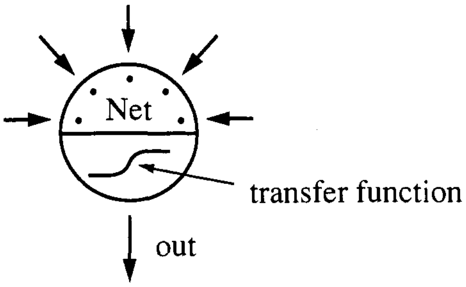


Figure 4. Basic operations of a neuron. The dots indicate weights and the arrows the incoming signals (data).

$$Net_j = \sum_{i=1}^{m} w_{ji}x_i \qquad (2)$$

An element w_{ji} of the weight matrix expresses the weight value between neurons i and j and can take either a positive or a negative value. The output, out_j, of a neuron j is obtained from the net input, Net_j, by application of eq 3.

$$out_j = f(Net_j) = 1/[1 + \exp(-\alpha Net_j + \theta_j)] \qquad (3)$$

The function f is called a transfer function and is here a sigmoid function, but any differentiable function can be used instead. θ_j is a threshold value for neuron j. This neuron only fires when this threshold has been reached. All output values are calculated synchronously.

Given m units at the input layer, a set of input data is represented by an m-dimensional vector, called "input pattern". Likewise the output data can be regarded as a vector, called "output pattern". During the training process the input patterns are fed to the network to give a result at the output side, out_j. This result is compared with the expected target output pattern, t_j, and the network's weights are changed to minimize the error. The training is therefore called a supervised process.

The training is carried out according to the back-propagation algorithm[7] until the error ϵ (eq 4) becomes acceptably small.

$$\epsilon = {}^1/_2\sum_j (out_j - t_j)^2 \qquad (4)$$

The next step is the recursive adaptation of all weights in the network so that the error between the target and the obtained output vector becomes as small as possible. The weights w_{ji} are adjusted by an amount that is proportional to

$$\Delta w_{ji} = \eta\delta_j x_i + \mu\Delta w_{ji}^{(\text{previous cycle})} \qquad (5)$$

Here, η is a parameter which determines the shift for correction in recursive cycles and is called the learning rate. The parameter μ, the momentum term, filters out high-frequency variations of the error surface in the weight space and thus tries to avoid the convergence being trapped in local minima. Furthermore, the momentum term μ allows the use of larger values of the learning rate, which leads to faster learning. The error δ_j can be obtained for the last (output) layer from the error ϵ. Then, the error δ_j is back-propagated through the network. For a detailed description of the back-propagation algorithm the reader is referred to the original publication[9] or to the cited review papers.[13,19]

One learning cycle (epoch) is defined by the input of the data and the comparison and correction of weights in all layers for all training vectors. To achieve convergence, usually many hundreds or even thousands of cycles are necessary.

Kohonen Network. The perception of similarity in objects is a fundamental and frequent activity. In a self-organizing neural network the neurons are arranged in a two-dimensional array to generate a two-dimensional feature map such that similarity in the data is preserved. In other words, if two input data vectors are *similar*, they will be mapped to neurons that are *close* together in the two-dimensional layer.

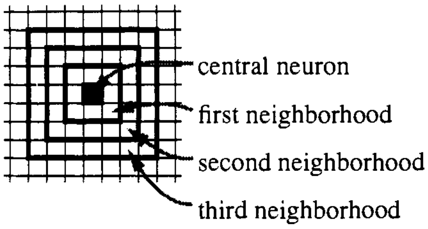


Figure 5. Neighborhood of the first, second, and third sphere of a central neuron in a Kohonen network.

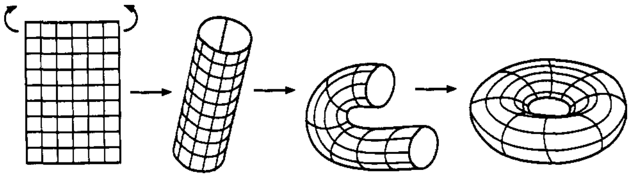


Figure 6. The conversion of a rectangular network layer into a toroid.

A key difference between the self-organizing maps, often called Kohonen networks because of the pioneering role of Kohonen in their development, and many other networks is that the self-organizing maps learn without supervision, hence the word self-organizing. The input data are fed into the network and after a certain number of cycles the network reaches a steady or stable state that represents topologies and structures of the multidimensional input spaces.

The two-dimensional setup of neurons, the Kohonen layer, the heart of such a network, is thought to act similarly to biological systems in the sence that it preserves order, compacts the representation of sparse data, and spreads out dense data.

In a self-organizing map the input units are fully connected to the two-dimensional Kohonen layer. Each neuron within the Kohonen layer has a well-defined topology, which means a defined number of neurons in its neighborhood. Both a quadratic and a hexagonal neighborhood can be used. The quadratic neighborhood offers four neighbors with the shortest distance (top, bottom, left, and right) and another four neighbors at the diagonals. Most implementations of a Kohonen network use all 8 neurons for defining the first sphere of neighbors, 16 neurons for the second, and so on (Figure 5).

In our application, each neuron must have the same number of neighbors in each neighborhood sphere. Thus, borders in the network layer had to be avoided. By connecting the left end of the plane with the right end and then the upper with the lower end a torus is obtained that has a plane without beginning and without end. Thus, each neuron has the same number of neighbors in each sphere (Figure 6).

The geometric figure of a torus can easily be modeled by an algorithm using the modulo function each time an access to a neuron within the Kohonen layer is made. The importance of having the same neighborhood for all neurons is tied to the competitive learning applied in a Kohonen network. An incoming input pattern is presented to each neuron of the Kohonen layer. In the end, only *one* neuron is selected to represent this pattern. This learning rule is therefore often called "the winner takes it all". That neuron is selected as winner that has the smallest Euclidian distance between the presented m-dimensional input pattern vector $\mathbf{x}$ (x_1, x_2, ..., x_i, ..., x_m) and the m-dimensional weight vector $\mathbf{w}_j$ (w_{j1}, w_{j2}, ..., w_{ji} ..., w_{jm}) of the j neurons within the Kohonen layer (eq 6).

$$out_c = \min[\sum_{i=1}^{m}(x_i - w_{ji})^2] \qquad j = 1, 2, ..., m \qquad (6)$$

Learning within a Kohonen layer consists of the adjustment of the weights, w_{ji}, in such a manner that the weights of the

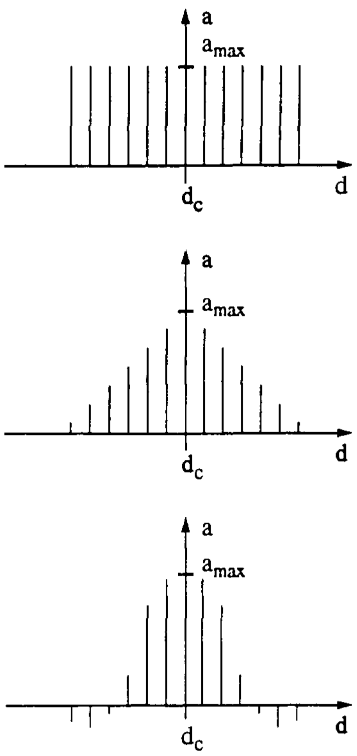


Figure 7. Typical neighborhood functions: constant, triangular, and "Mexican hat".

winning neuron, c, are shifted closer to the values of the input data. However, not only the weights of the winning neuron are adjusted but also those of the neighboring neurons. Equation 7 gives the correction formula for the weights.

$$\Delta w_{ji} = w_{ji} + f(t,a)(x_i - w_{ji}) \qquad (7)$$

The correction factor, $f(t,a)$, has the largest value for the weights in the winning neuron, c, and decreases with increasing distance, $d_c - d_j$ (eq 8). This distance is defined as the number of spheres of neighbors between the winning neuron, c, and the neuron j being considered (see Figure 5).

$$f(t,a) = \eta(t)a(d_c - d_j) \qquad (8)$$

here, t means the number of objects entered into the training process since the beginning of the training. During training, $\eta(t)$ becomes smaller and smaller and at the end only the winning neutron is adjusted. The value of the neighborhood function, a, depends on the distance between the winning neuron, c, and the actual neuron, j. Neighborhood functions often used are the constant, the triangular, or the so-called "mexican hat" (Figure 7).

When presenting an input pattern to the trained network one neuron with the largest response is obtained. By creating a map of the neurons together with markers categorizing the input data that have them excited, the topology of the input data from the multidimensional space is reflected in the topology in the two-dimensional Kohonen map. As an example of such a map see Section 9 and Figure 10.

4. Basic Approach

The aim of the study was to develop a neural network that, on input of a chemical structure, should predict which bond breakings are reactive in this molecule and which are not.

The process for solving this problem was broken down into several steps (Figure 8): (1) conversion of the structure input into a connection table describing each atom and bond of a molecule; (2) calculation of seven electronic and energy variables for each bond breaking of a molecule; (3) construction of a two-layer neural network consisting of seven input units, three neurons in a hidden layer, and one output neuron that classifies a bond breaking into reactive or not; (4) input of these seven variables for each bond breaking into a multilayer neural network; and (5) supervised training of this multilayer neural network by the back-propagation algorithm with datasets of bond breakings selected by three different procedures (random, experimental design, and the Kohonen network). These steps will be given in detail below.

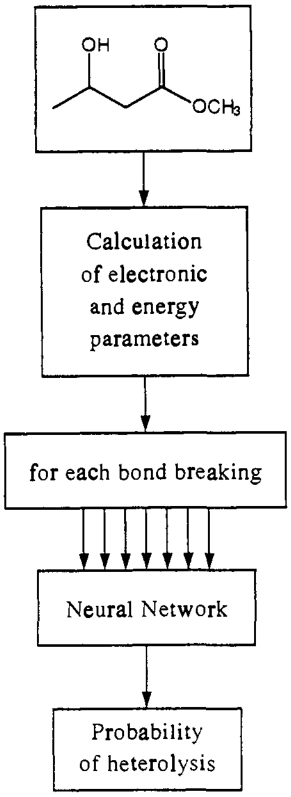

OH
O
OCH3
Calculation of electronic and energy parameters
for each bond breaking
Neural Network
Probability of heterolysis

Figure 8. The basic steps in classifying a bond in an organic structure into reactive or non-reactive.

5. Chemical Effects

The polar breaking of a bond is influenced and facilitated by a variety of physicochemical effects. To model these effects and their relative importance, parameters calculated for several electronic and energy effects by previously published empirical methods[20-24] were used. A common feature of these methods is that they derive their results directly from a connection table, from the constitution of a molecule. However, in contrast to most topological indices, they also take into account of the identity of the atoms, describing them by their valence state ionization potentials and electron affinities[20,21] or by additivity increments derived from mean molecular polarizabilities or from heats of formation.

In principle, theoretical methods like *ab initio* quantum mechanical procedures can be used to derive variables for the description of electronic and energy effects in the various bonds of a molecule. However, we wanted to use the trained neural network in the reaction prediction system EROS (Elaboration of Reactions for Organic Synthesis)[2] where many molecules with a sizable number of atoms have to be evaluated. This led us to resort to the less computer-time demanding empirical methods.

[20] Gasteiger, J.; Marsili, M. *Tetrahedron* **1980**, *36*, 3219–3228.

[21] Gasteiger, J.; Saller, H. *Angew. Chem.* **1985**, *97*, 699–701; *Angew. Chem., Int. Ed. Engl.* **1985**, *24*, 687–689.

[22] Hutchings, M. G.; Gasteiger, J. *Tetrahedron Lett.* **1983**, *24*, 2541–2544.

[23] Gasteiger, J.; Hutchings, M. G. *J. Chem. Soc., Perkin Trans. 2* **1984**, 559–564.

[24] Gasteiger, J. *Tetrahedron* **1979**, *35*, 1419–1426.

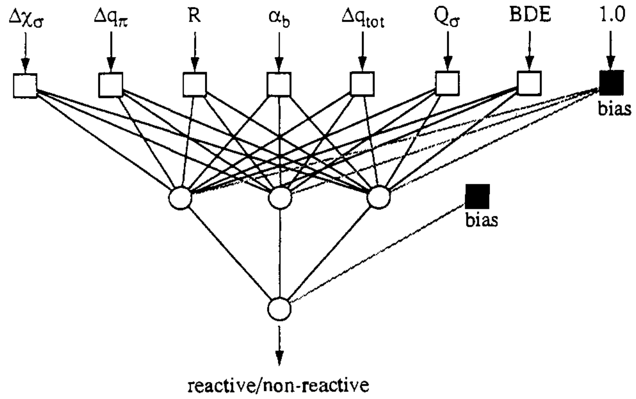


Figure 9. The architecture of the network used in this study.

The significance of the parameters obtained by the empirical method has been established in a series of calculations and correlations of physical[1] and chemical[3] data. Thus, they seemed appropriate for the description of chemical bond breakings.

The variables used include values for the charge distribution,[20,21] for the inductive,[22] the resonance,[4,21] and the polarizability effect,[23] and for bond dissociation energies.[3,24] For each bond breaking the following seven parameters were calculated: the difference in total charge (Δq_{tot}), the difference in π-charge (Δq_π), the difference in σ-electronegativity ($\Delta\chi_\sigma$), the amount of charge, shifted in the PEOE method across a bond as a measure of bond polarity (Q_σ), the resonance stabilization of charges generated by heterolysis (R), the bond polarizability (α_b), and the bond dissociation energy (BDE). The signs of the first four variables depend on the direction of the polar bond breaking; these variables change their sign on reversing the polarity of the bond breaking. The resonance stabilization is normally quite different not only in sign but also in its magnitude for the two possible heterolyses, because the mechanisms for the stabilization of a positive or a negative charge on the same atom differ. The last two variables are independent of the direction of bond breaking.

6. Back-Propagation Network Architecture

With seven variables for describing a bond breaking, the chosen neural network has seven input units, one for each variable. On the output side there are two classes, reactive and non-reactive bond breakings. This information can be coded with one or two neurons. Tests showed that when two output neurons are used the sum of the two output values was always exactly 1.0. Coding a simple yes–no decision on two neurons leads to the same results as with one output neuron. The weights between the hidden layer and the output layer of the second neuron have the same values but are of opposite sign to that of the other neuron. Therefore, only one neuron was chosen at the output side.

The next problem to solve was the number of hidden layers and the number of neurons contained in them. The selection of the number of hidden neurons is usually done by trial and error. The goal is to find a network with a minimum number of neurons and connections that is able to solve the problem. Networks with many connections can learn a problem perfectly, but their predictive ability may be very low. A network with one hidden layer of three neurons was found to be sufficient to solve our problem of bond reactivity (Figure 9). We also tried two hidden layers and varied the number of hidden neurons from 2 to 7; however these networks have lower prediction ability, cannot learn the dataset completely, or the training time increases without leading to better results. Altogether there are $7*3 + 3*1 + 3 + 1 = 28$ adjustable weights, including the connections to the bias. If the network has more than three hidden neurons the network can learn the problem correctly, but the prediction becomes worse.

The input variables have different ranges, e.g., one varies from -0.2 to $+0.2$ and another from 200 to 500. The input units expect values between 0 and 1, therefore the input values have to be scaled. Each input value is separately scaled between its minimum and its maximum value in the entire dataset. The classification is coded as a 0 for a non-reactive bond breaking and a 1 for a reactive bond breaking. During the training phase a stringent quality criterion is applied. An output is considered as correct if the error is ≤ 0.1.

Reproduction Ability. In the first test, the network was presented with the entire dataset of bond breakings. After 400 learning cycles only one breaking could not be learned correctly. This is the breaking of the C–C bond in acetone to give CH_3^- and CH_3CO^+ which is learned as reactive though it is classified as non-reactive. Comparison of this result with earlier investigations of this dataset[5] offers an explanation: in plots by a principal component analysis this bond breaking lies as a separated point in a region of reactive bond breakings. In such cases, with only one single point in a space of inversely classified bond breakings, neural networks which base their decisions on the similarity of vectors have difficulties in learning results correctly (see also the KNN results[5]). The error was exactly 1.0. Trying to reduce this error in additional cycles was not successful. Analyzing the equations used for correction offers an explanation. The error signal, δ_j, is given by

$$\delta_j = (t_j - out_j)out_j(1 - out_j)$$

The error signal δ_j becomes zero, even if the difference between the target and the obtained output is not zero, if either the second or the third term becomes zero.

This problem was solved by adding a small constant to δ_j and in the next 600 cycles the error signal slowly disappeared. After 1000 cycles the entire dataset had been learned correctly with a maximum error of 0.05.

Prediction Ability. The intention is to develop a network that can *predict* the reactivity of each single bond breaking in a given molecule if the seven bond variables are presented as input. The dataset must therefore be divided into two sets: one for training and the other one for testing the prediction ability of the network.

7. Random Selection

One way of selecting training sets is to take a random selection from the entire dataset. A reasonable approach is to split a dataset in half and take half of the data for training and the other half for testing. This would have amounted to 58 bond breakings in the dataset.

The investigation reported in Section 8 shows that 64 bond breakings are necessary to cover the information space evenly. In order to make a fair comparison 64 bond breakings were also chosen in the random selection.

Ten different datasets with 64 data each were selected and presented to the network. The results were all comparable: the network learned eight datasets correctly in 500–2100 cycles. In two cases the network is not able to learn one bond breaking even after 4000 training cycles. During the training phase the classification of a bond breaking was considered in error as long as the output value differed from the expected value (target) of 0.0 or 1.0 by 0.1. For the test cases, a bond breaking is considered as wrongly classified when a reactive bond breaking has an output value smaller than 0.5 and a non-reactive bond breaking has an output value larger than 0.5. From among the 85 bond breakings in the test set the number of wrong classifications ranges from 3 to 12 with 7.5 on average. Table I shows the results in detail.

An average error of about 9% in the number of predictions on bond breakings may be considered acceptable. However, it does leave something to be desired. This led us to strive for additional improvements. The selection of the dataset for training the network is an obvious starting point.

A random selection from such a small dataset usually does not lead to a homogeneously distributed training set. Therefore, other

Table I. Predictions by Two-Layer Networks Trained with 10 Randomly Selected Datasets with 64 Bond Breakings for Training and 85 for Testing

	1	2	3	4	5	6	7	8	9	10
training										
no. of epochs	>4000	1100	700	500	1000	800	>4000	800	1000	2100
no. of errors	1	0	0	0	0	0	1	0	0	0
rms[a]	0.12	0.02	0.02	0.02	0.02	0.02	0.12	0.02	0.02	0.02
prediction										
no. of errors	7	3	12	10	6	7	5	9	7	5
rms[a]	0.28	0.15	0.36	0.23	0.23	0.23	0.23	0.28	0.28	0.21

[a] Root mean square.

Table II. Correlation Matrix of the Parameters for All 149 Classified Bond Breakings

parameter	$\Delta\chi_\sigma$	Δq_π	R	α_b	Δq_{tot}	Q_σ	BDE
$\Delta\chi_\sigma$		0.19	0.07	0.00	–0.11	–0.74	–0.01
Δq_π			–0.14	0.03	–0.31	–0.41	0.01
R				0.30	0.12	0.06	–0.28
α_b					–0.03	–0.02	–0.72
Δq_{tot}						0.84	0.02
Q_σ							0.04
BDE							

methods for selecting the training set to cover the problem space better were sought.

8. Training Set by Experimental Design Techniques

Experimental design techniques[18] allow one to select the minimal number of data that cover a problem space as homogeneously as possible. By such techniques one can *reduce* the number of required experiments and draw conclusions which are valid in the entire problem space.

In an *m*-level experimental design, each of the coordinates (variables) of the space is divided into *m* intervals and from each interval one point is selected. For example, when three intervals are chosen, objects should be selected so that each variable is represented by a low, a middle, and a large value. The case of a three level design space with only two variables asks for a minimum of $3^2 = 9$ data points.

Dataset. The seven variables characterizing a bond breaking span a seven-dimensional hyperspace. If *m* is chosen as 3 (low, middle, large), $3^7 = 2187$ subspaces are required. As we have only 373 bond breakings in the entire dataset most of the subspaces cannot be occupied by bond breakings. Therefore, the problem space has to be reduced to a lower dimension and/or *m* has to be taken as 2. We decided to take only four of the seven variables for experimental design and to divide a variable into three regions. Thus, the problem space has $3^4 = 81$ subspaces.

The selection of the variables for the experimental design can be based on the correlation matrix of the variables (see Table II).

Variables that are highly correlated with each other were substituted with only one of them in experimental design. On the basis of chemical expectation and an analysis of the correlation matrix the following variables were selected: the resonance effect (R), the bond polarizability (α_b), the measure of the polarity (Q_σ), and the difference in σ-electronegativity ($\Delta\chi_\sigma$).

The Δq_π variable was deleted because for quite a few of the bond breakings considered none of the two atoms bears a π-charge. Thus, as this variable is zero for many bond breakings, it is, although important, not statistically balanced over the entire dataset.

The difference in total charge, Δq_{tot}, is highly correlated with the polarity measure, Q_σ, and therefore only one, Q_σ, was taken. BDE is correlated to a rather large degree with the bond polarizability α_b. Therefore, also here only one, α_b, was included.

Experimental design investigations showed the following data distribution for all 373 bond breakings: of the $3^4 = 81$ possible subspaces only 40 were occupied by the initially classified 116 bond breakings, 21 subspaces were empty, and 20 subspaces contained bond breakings that were not yet classified. On analysis of the 20 subspaces containing bond breakings that were not yet classified, it was found that an additional 33 (20 for training and 13 for testing) of them can be classified with a certain reliability for correctness. Altogether the new dataset of 149 bond breakings now occupies 60 subspaces. However, there are four subspaces with six bond breakings of opposite classifications.

Back-Propagation Network. Ten different training sets were tested with the multilayer neural network indicated in Figure 9. Each of the test sets received one bond breaking from each occupied subspace (60) and additionally four bond breakings out of those subspaces that contained both reactive and non-reactive bond breakings. From those subspaces containing more than one bond breaking of the same class one bond breaking was arbitrarily chosen as representing this group. The network was trained with 64 bond breakings and the predictions were tested with the remaining 85 of the 149 classified bond breakings. A few clusters contained only one classified bond breaking, while others contained up to 11 bond breakings. The highly occupied clusters are in those regions of the hyperspace corresponding to low or high reactivity. It seems to be easier for an organic chemist to classify a bond breaking in these cases than in cases where bond breakings are in regions of transition from reactive to non-reactive.

The training was started with a value of 0.5 for both the learning rate (η) and the momentum parameter (μ). After 200 cycles the momentum parameter was reduced to 0.05 and after 300 cycles the learning rate parameter (η) was reduced to 0.1. After 400 cycles the network had either learned all bond breakings correctly or there were a few error signals with a value of 1.0. As mentioned above (Section 6) a constant (0.05) was added to the error signal, δ_j, and in the following cycles the network could learn these bond breakings, too.

In comparison with the results from randomly selected training sets (Section 7; Table I) the predictions for the bond breakings of the test set were better. The number of wrongly classified bond breakings ranged between one and six with an average of 3.4 misclassifications per set. This is a remarkable improvement over the results obtained with randomly selected training sets that gave three to twelve errors and, on average, 7.5 misclassifications in a set. Table III shows the results in detail.

9. Kohonen Mapping

The technique of experimental design as described above has two disadvantages: First, the selection of variables on which the experimental design is based is difficult and leads to a loss of information. Second, the choice of the number of intervals and the positioning of the boundaries between them to reduce the dimensionality of the space is somehow arbitrary and introduces a bias from the side of the user.

We were therefore looking for a method that does not require a reduction of the dimensionality of the problem space and does not need a tedious optimization of the boundaries between the intervals. We show here that a Kohonen network can provide this functionality. A Kohonen map is a two-dimensional representation of a multidimensional parameter space and this is exactly what is needed.

Table III. Predictions by Two-Layer Networks Trained with 10 Different Datasets Selected by Experimental Design with 64 Bond Breakings for Training and 85 for Testing

	1	2	3	4	5	6	7	8	9	10
training										
no. of epochs	1000	900	1200	2000	1200	1300	1400	1100	1100	1000
no. of errors	0	0	0	0	0	0	0	0	0	0
rms[a]	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
prediction										
no. of errors	1	3	4	3	4	2	3	5	6	3
rms[a]	0.10	0.16	0.23	0.16	0.19	0.16	0.15	0.24	0.23	0.22

[a] Root mean square.

Entire Dataset. Learning in a Kohonen network is an unsupervised process and therefore the network can be trained with all 373 single bond breakings—both classified and unclassified—of the dataset.

The dimension of the Kohonen layer, i.e., the number of neurons, determines the maximum number of different categories of the input data and therefore plays an important role. If the number of neurons is too small, many data will be mapped into the same neuron and the probability for conflicts between reactive and non-reactive bonds ending up in the same neuron will increase. On the other hand, if the dimension of the Kohonen net is too large, the bond breakings will be spread over the neurons in such a way that hardly any neuron obtains more than one bond breaking. However, we want to have several bond breakings to end up in the same neuron in order to be able to select one for the training set and the other for testing. For if several bond breakings end up in the same neuron, this is an indication that these bond breakings carry similar information as expressed by the electronic and energy variables characterizing a bond breaking. Then it suffices to take only one of such bond breakings being projected into the same neuron into the training set when maximum coverage of the entire information space is desired.

Having 149 classified bond breakings we decided to take a 9 × 9 Kohonen layer, thus forcing several bond heterolyses into the same neuron. In addition, with 81 neurons we have the same number of subspaces as in the study by experimental design. This allows a fair comparison of the use of a Kohonen network as an alternative to traditional experimental design techniques. Stabilization was reached after 30 training cycles. Figure 10 shows the resulting Kohonen map, where each neuron of the map is represented by a box. The different shadings indicate the classes assigned to the bond breaking that have excited this neuron. Black is for bond breakings classified as reactive, light gray those classified as non-reactive, and dark gray those for unclassified bond breakings. Neurons that obtain both types of bond breakings, reactive and non-reactive (conflict neurons), are indicated by a cross. Neurons that do not receive any bond breaking at all are in white. Remember, however, that in the learning phase class assignment plays no role. These assignments are only used for evaluating the results of the Kohonen mapping. This map shows that the classified bond breakings are distributed over 56 neurons, i.e., 56 categories. Six neurons are empty and 19 neurons are occupied by non-classified bond breakings. Twelve neurons have only bond breakings classified as reactive and 42 neurons only bond breakings classified as non-reactive. Among the 56 neurons occupied by classified bond breakings, two contained conflicts. Conflicts had also been observed in the experimental design investigations. Interestingly enough the same bond breakings are concerned. Also a 9 × 10 and a 10 × 10 Kohonen layer were used but the conflicts remained. If a smaller network was used the number of conflicts increased.

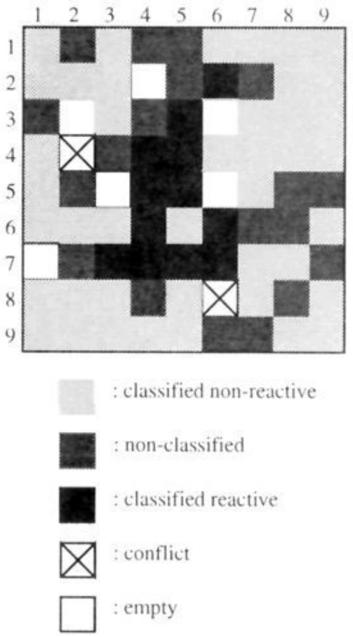


Figure 10. The Kohonen map obtained for 373 bond breakings. The shading of the neurons indicates which types of bond breakings end up in that particular neuron. Nonclassified bond breakings are indicated only for those neurons that have bond breakings classified as neither reactive nor non-reactive.

Training Set by a Kohonen Network. Ten different training sets for the previously designed back-propagation network (Figure 9) were chosen. For each of the 56 neurons with bond breakings of only one classification type arbitrarily one bond breaking was chosen as representing this group of bond breakings. In the case of those two neurons containing bond breakings of opposite classification one representative for each class was selected. The resulting training sets thus have 58 data, six bond breakings less than those used in the experimental design study. With use of the same training conditions as in the experimental design study, the behavior of the network in the training phase was quite similar in both cases. With 700–2800 cycles the training sets can be learned completely. For the test sets with 91 bond breakings the number of wrongly predicted bond breakings ranges from 0 to 2. The average number of misclassifications has now dropped to only 0.9. Five of the ten neural networks predict all test data correctly. Table IV shows the results in detail. These results demonstrate that a Kohonen mapping provides a basis for the selection of an optimum dataset for the training of a multilayer neural network by the back-propagation algorithm. This combination of two neural networks, of an unsupervised and a supervised learning method, has here be proven to be superior to any other explored approach.

Prediction Result for Bonds in an Additional Molecule. In this section we discuss the prediction ability for bond breakings that are in neither the training nor the test set. Two additional molecules not included in the dataset before were selected to test the prediction ability of the trained network. The network with the 7-3-1 architecture and trained for 1600 epoches with 58 bond breakings selected from the Kohonen mapping was chosen.

The molecule in Figure 11 has three functional groups: an aldehyde, a secondary alcohol, and a secondary amine group. These groups are fairly separated from one another and therefore

Table IV. Predictions by Two-Layer Networks Trained with 10 Different Datasets Selected on the Basis of the Kohonen Network with 58 Bond Breakings for Training and 91 for Testing

	1	2	3	4	5	6	7	8	9	10
training										
no. of epochs	900	700	1200	1600	900	1500	1400	2800	1500	800
no. of errors	0	0	0	0	0	0	0	0	0	0
rms[a]	0.03	0.04	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.04
prediction										
no. of errors	2	2	1	0	2	0	0	0	0	2
rms[a]	0.15	0.15	0.07	0.02	0.15	0.06	0.05	0.05	0.02	0.16

[a] Root mean square.

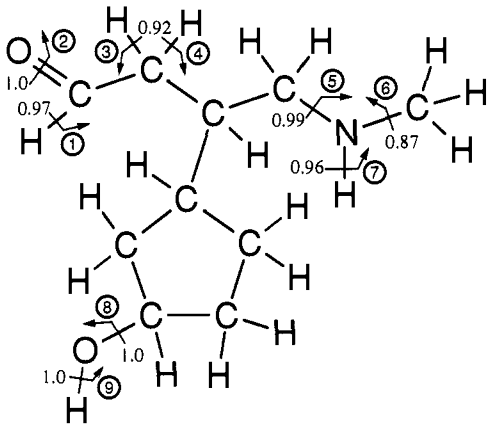


Figure 11. Predictions of bond breakings made by a two-layer neural network. Only reactive bond breakings are indicated; all other bond breakings have output values smaller than 0.5.

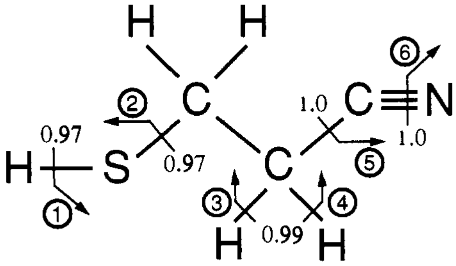


Figure 12. Predictions of bond breakings made by a two-layer neural network. Only reactive bond breakings are indicated; all other bond breakings have output values smaller than 0.5.

they should not interfere much with each other. Only those bond breakings considered reactive are indicated.

The network correctly predicts a high reactivity for the deprotonations of the hydroxyl group (bond breaking no. 9), the NH group (no. 7), and the methylene group in the α-position (nos. 3 and 4) to the aldehyde group. The loss of a hydroxyl anion (no. 8) is also correctly considered as occurring easily.

The loss of a proton at the aldehyde group (no. 1) is also predicted as feasible. Such a reaction is usually not observed as most bases are also strong nucleophiles and rather undergo a nucleophilic attack at the carbonyl group. However, deprotonation of formic esters, compounds that have an H—C═O group, have been observed. Although the network has been trained only for single bonds, it is able to further generalize and to also assign the correct reactivity to the C═O (no. 2) double bond. The breaking of the C—N bonds (nos. 5 and 6) is considered as feasible. In principle, such C—N bond breakings corresponding to a nucleophilic aliphatic substitution can be observed but usually only after some activation.

Figure 12 shows those bond breakings predicted to be reactive for a second molecule. Only those bond breakings considered reactive are indicated. This molecule was selected to test the ability of the multilayer neural network for generalizing what it has learnt on the reactivity of functional groups. The dataset used for training (Scheme I) contained neither a nitrile nor a mercapto group; in fact it did not contain any sulfur atom at all. Nevertheless, the network correctly perceives the acidity of the thiol group (bond breaking no. 1) and the fact that it can be exchanged by nucleophilic aliphatic substitution (bond breaking no. 2). The latter reaction will, however, need some activation. Although the network has not seen any thiol group or sulfur atom, it can make predictions on the influences of a sulfur atom on chemical reactivity. This is a result of the coding of atoms and bonds chosen in this study. Rather than coding atoms, bonds, or functional groups explicitly, a more general scheme of coding them by electronic and energy variables has been selected. Thus, we are not presenting the network with a S–H or a C–S bond, bonds that it has never seen and therefore would be unable to make predictions. Rather, the network is presented with bonds that have a certain bond dissociation energy, a certain electronegativity difference, a certain bond polarity, etc., variables that the network knows how to deal with and thus is able to utilize for predictions.

By the same token the network can make predictions on the influence of a cyano group on chemical reactivity. The increase in acidity of a CH bond (bond breaking nos. 3 and 4) in the α-position to a cyano group is correctly perceived. Furthermore, the propensity of the cyano group for nucleophilic displacement (bond breaking no. 5) is indicated. And finally, although the network has been trained on the reactivity of aliphatic single bonds, it can generalize to make predictions on multiple bonds, in this case showing that the triple bond of the cyano group will react under nucleophilic attack (bond breaking no. 6).

10. Comparison of the Dataset Selection

The experimental design technique is tedious and time consuming. The selection of the variables for the experimental design is the most difficult task. For a given problem one might have a host of properties available to characterize the problem. However, one might not be able to predict which parameters are the most influential ones. Thus, the selection of the parameters for the experimental design introduces some arbitrariness and leads to a loss of information. Next, the number of ranges for the division of each parameter and the boundaries between these ranges has to be determined. A series of careful studies of the dataset must be performed to find an acceptable solution.

With a Kohonen neural network all these steps are unnecessary. The dimension of the Kohonen layer has to be chosen and the entire dataset is run through the network with all available variables. The network is rapidly trained and the resulting Kohonen map is built. In the worst case the map will show many neurons with conflicts and a larger Kohonen layer has to be chosen.

11. Analysis of the Kohonen Map

In Section 9 and Figure 10 we presented the results of the Kohonen mapping of the entire dataset of 373 bond breakings. This Kohonen map was taken as a basis for the selection of a dataset for training a back-propagation network that covers the information space as broadly as possible.

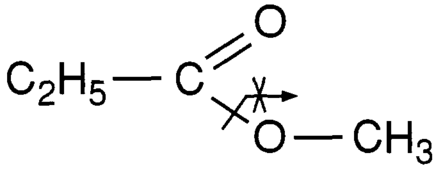


Figure 13. A bond breaking classified as non-reactive that ends up in the domain of reactive bond breakings.

We now analyze the Kohonen map in more detail and show that it lends itself to further chemical interpretation. First, one can see from Figure 10 that all bond breakings classified as reactive end up in neurons that form one coherent region (black space) in the center of the figure. This shows that the self-organization in the Kohonen learning is able to perceive the similarity of all reactive bonds in the dataset and puts them into topologically adjacent neurons, forming the observed cluster of (black) neurons.

There is one intruder into this domain of reactive bond breakings, a neuron with a bond breaking classified as non-reactive. Figure 13 shows this bond breaking.

The polar breaking of this bond is not directly observed in this molecule. It will occur only after a nucleophile (base) has been added to the carbonyl group forming a tetrahedral intermediate. Thus its classification as non-reactive is justified for this molecule. However, in compounds having bulkier groups instead of the C_2H_5 group, i.e., the mesityl group, the breaking of the analogous bond is observed. Thus, the bond breaking shown in Figure 13 can occur when it is not superseded by other types of reactions. In effect, it has to be considered as a *potentially* reactive bond. It is, therefore, quite acceptable that this bond breaking ends up in the cluster of reactive bonds.

Neurons activated by non-reactive bond breakings and neurons activated by reactive bond breakings touch each other only in a few places. This is a further indication of the ability of the Kohonen learning to differentiate and separate reactive from non-reactive bond breakings. On line with this is the observation that the cluster of reactive bonds is surrounded by neurons with conflicts or with non-classified bond breakings. Neurons with conflicts directly indicate the transition from reactive to non-reactive. Neurons with non-classified bond breakings indicate the cautiousness of the chemists in making a commitment to classify a bond breaking as reactive or non-reactive when reactivity is in doubt, when it is in between reactive and non-reactive.

After this discussion of the global features of the Kohonen map of the bond breakings a closer look at the bond breakings ending up in the individual neurons is taken. First, the cluster of neurons with reactive bond breakings is investigated. Figure 14 shows the types of bond breakings that are mapped into the individual neurons.

As can be seen, similar types of bond breakings end up in the same neuron. Thus, the similarity of the individual instances of a certain type of bond breaking is perceived. All carbon-heteroatom bond breakings are at the right-hand side of this part of the map, starting on top with carbon–iodine bonds and then passing through carbon–bromine, carbon–chlorine, and carbon–oxygen bonds to carbon–nitrogen bonds. In this sequence there is a clear tendency of decreasing polar reactivity from top to bottom.

The left-hand side shows bond breakings that correspond to the dissociation of a proton. Two such bond breakings are in neurons with conflicts, i.e., in neurons that also contain non-reactive bond breakings. One conflict occurs because the loss of a proton from dichloromethane is classified as reactive whereas the loss of a proton from the carbon atom of neopentyl chloride that bears the chloro atom is classified as non-reactive. Apparently, the increase in acidity by a second chloro atom is not high enough to separate these two bond breakings. The more acidic O–H and N–H bond breakings are more to the center of the cluster of reactive bonds. As O–H bonds are more acidic than N–H bonds and these much more than C–H bonds, there is apparently again a decrease in reactivity from top to bottom. The less acidic C–H bonds are more at the outskirts. In row 7, neuron no. 3 contains C–H bonds activated by either a carbonyl or a nitro group, neuron no. 4 contains three different C–H bonds activated by either a carbonyl or a C═C bond, and neuron no. 5 contains three C–H bonds that are activated by two carbonyl or two C═C bonds. Thus, there is an increase in reactivity from left to right.

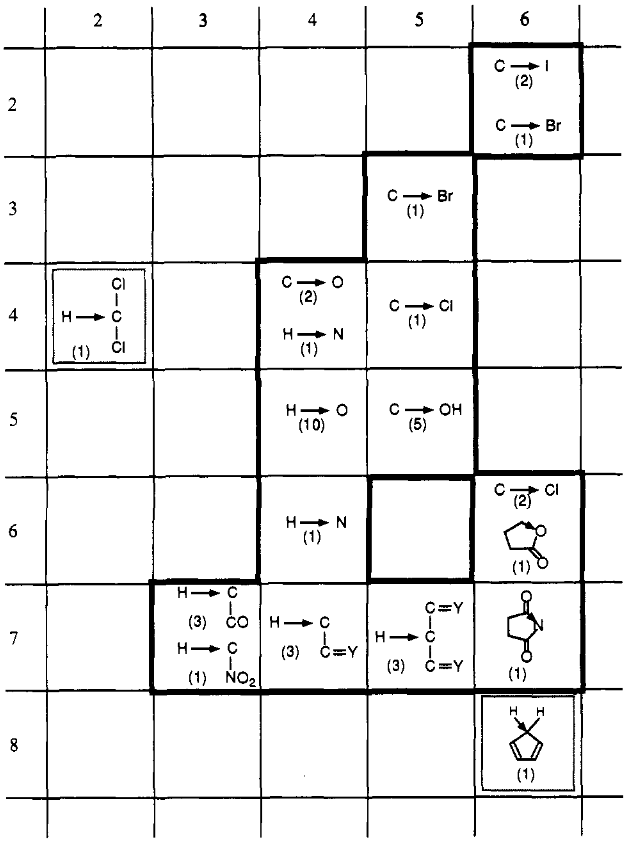
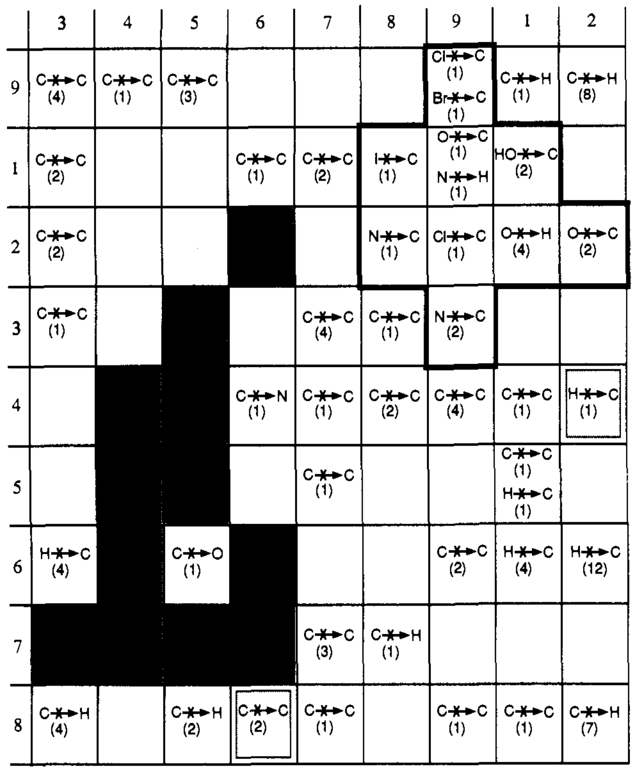


Figure 14. Distribution of bond breakings classified as reactive onto the individual neurons. This figure shows a cutting of Figure 10; the row and column indices refer to those of Figure 10. Neurons with reactive bond breakings are framed in bold; the two neurons with conflicts have extra boxes. The arrow indicates the shift of the electron pair on polar breaking. The numbers in parentheses give the number of bond breakings of the corresponding type contained in the indicated neuron. Y = O, CR_2

For a more detailed analysis of the bond breakings classified as non-reactive the Kohonen map of Figure 10 is shifted by two columns to the right and one row to the bottom. Remember, the neurons actually cover the surface of a torus and it is quite arbitrary where one makes the two cuts to convert the surface of the torus into a plane (cf. Figure 6). As will soon become clear, this allows one to show an important feature more distinctly. Figure 15 shows the map thus obtained together with an indication of the type and number of bond breakings that end up in the various neurons.

The polar breakings of the C—H and C—C bonds are distributed over a wide area of the Kohonen map. A closer inspection of the first and second sphere of neighboring atoms and bonds indicates the reasons for the various C—H and C═C bonds to end up in different neurons. A discussion of these small variations goes beyond the scope of this work.

However, one main feature of this map should be pointed out: the reactive bond breakings are now more to the lower left corner of the map (Figure 15). In the opposite direction, versus the upper right corner, are those bond breakings where a polar bond is broken *against* its inherent polarity. Thus, there is a strong reason why such bond breakings are not occurring; they would



	3	4	5	6	7	8	9	1	2
9	C✱→C (4)	C✱→C (1)	C✱→C (3)				Cl✱→C (1) Br✱→C (1)	C✱→H (1)	C✱→H (8)
1	C✱→C (2)			C✱→C (1)	C✱→C (2)	I✱→C (1)	O✱→C (1) N✱→H (1)	HO✱→C (2)	
2	C✱→C (2)			■		N✱→C (1)	Cl✱→C (1)	O✱→H (4)	O✱→C (2)
3	C✱→C (1)		■		C✱→C (4)	C✱→C (1)	N✱→C (2)		
4		■	■	C✱→N (1)	C✱→C (1)	C✱→C (2)	C✱→C (4)	C✱→C (1)	H✱→C (1)
5		■	■		C✱→C (1)			C✱→C (1) H✱→C (1)	
6	H✱→C (4)	■	C✱→O (1)	■			C✱→C (2)	H✱→C (4)	H✱→C (12)
7	■	■	■	■	C✱→C (3)	C✱→H (1)			
8	C✱→H (4)		C✱→H (2)	C✱→C (2)	C✱→C (1)		C✱→C (1)	C✱→C (1)	C✱→H (7)

Figure 15. Distribution of bond breakings classified as non-reactive onto the individual neurons. This figure is a shifted version of Figure 10 (see text); the row and column indices refer to the original Figure 10. The area in black indicates the cluster of reactive bond breakings (cf. Figures 10 and 14). The two conflict neurons have extra boxes. The bond breakings that would occur *against* a particularly strong polarization are framed in black. The arrow indicates the shift of the electron pair on polar bond breaking. The numbers in parentheses give the number of bond breakings of the corresponding type contained in the indicated neuron.

be particularly difficult to achieve. These distinctly non-reactive bond breakings can be separated quite clearly from the reactive ones.

This discussion should have shown that the self-organizing of the Kohonen learning is able, when presented with a proper representation of bond properties by fundamental physicochemical variables, to indicate in a two-dimensional map important features of the reactivity of bonds in organic molecules.

12. Comparison with Statistical and Pattern Recognition Methods and with an Associative Memory System

The dataset of 29 aliphatic molecules treated here having bonds classified for their ease of polar breaking was also investigated by statistical and pattern recognition methods including principal component analysis (PCA), linear discriminant analysis (LDA), partial least squares analysis (PLS), cluster analysis, *k*-nearest neighbor analysis (KNN), and logistic regression analysis (LoRA).[5] These methods could successfully be used to model the relationship between structure and reactivity. Details of the investigations can be found in ref 5. Some of the more prominent results are mentioned here for comparison purposes. The studies were performed with the entire initial dataset of 116 polar bond breakings (42 reactive, 74 non-reactive) (see Section 2).

A plot of the first component of a PCA against the second or third component leads to five wrong classifications of bond heterolysis. An LDA study also classified five polar bond breakings into the wrong reactivity class. A KNN investigation led, in the best result from among a series of different variable combinations and values of *k*, to one misclassified bond breaking. A PLS study either gave eight misclassified bond breakings or was unable to classify ten bond breakings. And finally, a logistic regression analysis gave an equation leading to one wrongly classified heterolysis.

Thus, only the *k*-nearest neighbor classification and the logistic regression analysis gave results that were as good as the results obtained from the classification of the best back-propagation network.

However, these optimum results of the KNN and LoRA studies were obtained only after exploration of a large variety of variable combinations. Furthermore, the number of wrong classifications were taken from the dataset of 116 bond breakings used for training the statistical and pattern recognition classifiers. Thus, all datapoints had already been "seen" by the KNN and LoRA classifier (and also the PCA, LDA, and PLS classifiers).

The misclassifications given in Table IV were obtained from a dataset of 91 bond breakings the back-propagation network *never had "seen"* before because it had been trained on another 58 bond breakings. Thus, the results of the neural network are genuine predictions whereas the results of the statistical and pattern recognition classifiers have been obtained on *training* data.

There are some additional disadvantages in using statistical and pattern recognition methods for such classification problems as compared with neural networks: Many require a reduction in the number of independent variables leading to some loss of information; most methods are quite sensitive to correlations between independent variables; and finally, in many cases assumptions on the mathematical form of the relationship between independent (physicochemical) variables and dependent (reactive or non-reactive bond) variable (mostly linear) have to be made.

The combination of a Kohonen network and a multilayer neural network trained by the back-propagation algorithm can work with all independent variables presented, even if they are highly correlated. Furthermore, it develops an implicit, and also nonlinear if necessary, relationship between input and output data. Clearly, a Kohonen network by itself can already be taken as a classifier. After a Kohonen network has been trained, new bonds can be input and each bond breaking will end up in a particular neuron. By inspection of the reactivity of the other bond breakings that are contained in that neuron a prediction of the reactivity of the newly mapped bond breaking can be made. However, Figure 10 shows that the situation is not that simple. Clearly, if a bond breaking ends up in a neuron having only reactive or only non-reactive bonds the prediction is unequivocal. But what will be the prediction when the neuron is empty, when it carries conflicts, or when it has only non-classified bonds? From all 81 neurons of Figure 10, 27, or one third, belong to such problem cases.

Thus, it is this unique combination of a Kohonen network and a multilayer network trained by the back-propagation algorithm that has the highest predictive power as a classifier.

The multilayer neural network has an additional benefit: it provides a continuous scale of values between 0 and 1. These values can be taken as a quantitative measure of reactivity. Thus, although the network has been trained as a classifier it can, to a certain extent, be used to come up with a continuous measure of reactivity, 0 being non-reactivity, 0.5 medium reactivity, and 1.0 high reactivity.

An associative memory system (AMS) was also trained to store the relationship between structure and reactivity in an implicit manner.[6] There, too, no explicit mathematical function has to be given and the relationship may be nonlinear. In that respect it resembles a multilayer neural network. However, it was shown[6] that an AMS with a good predictive power can only be obtained by a tedious optimization procedure which forces one to find out which physicochemical variables are of little influence to eliminate them from the investigation, quite similar in analogy to the experimental design technique reported here. This elimination of variables nevertheless leads to a loss of information and is not necessary when a Kohonen network is used for selecting the training set.

Furthermore, an AMS offers little help for selecting a well-balanced training set. Here, the Kohonen network is of great value as the choice of the training set has a remarkable influence on the prediction obtained from the multilayer neural network.

One feature of the associative memory system is, however, quite interesting. It can provide an estimate of the reliability of its predictions.

13. Conclusions

The combined application of two fundamentally different types of neural networks has led to a procedure that allows one to automatically predict which bonds are preferentially prone to heterolysis in aliphatic molecules.

Several features are essential for this successful undertaking. First, the polar breaking of a bond has to be described by fundamental physicochemical variables. In our approach, we have chosen empirical variables in order to be able to treat large datasets with the computer resources available to us. Clearly, the general methodology outlined in this paper can also be applied when similar essential electronic and energy effects are calculated by quantum mechanical methods. The choice between empirical and quantum mechanical calculations of various degrees of sophistication will be dictated by the size of the molecules being studied, the extent of the dataset, and the access to computer resources.

The goal of this work was to train a multilayer neural network by the back-propagation algorithm to predict the polar breaking of single bonds in aliphatic molecules.

The next step in this endeavor is therefore to select a dataset for the training of such a neural network. We have shown that the self-organization of a Kohonen network is superior to an experimental design technique and even more so to a random selection for providing an appropriate training set. The selection of a training set on the basis of a Kohonen network can be recommended for any supervised learning method, including statistical and pattern recognition methods.

A Kohonen network is not only a good choice for the selection of a dataset but it can also preserve and indicate fundamental features that influence chemical reactivity.

In this paper we have shown how the classification of polar bond breakings into reactive and non-reactive can be used to develop a multilayer neural network able to perform that task automatically. Multilayer neural networks can also be used for modeling. Thus, if instead of a mere classification of bond breakings as reactive or non-reactive quantitative values on the probability of bond breaking are available, a neural network can be trained that predicts *quantitative* values on chemical reactivity.

Clearly, our approach relied on a manual classification of bond breakings as reactive or non-reactive. With increasing availability of databases on chemical reactions such information can be gathered automatically. This offers the opportunity of deriving predictions on the reactivity of bonds from the information in reaction databases by the combined application of two different types of neural networks, a Kohonen network and a multilayer neural network. This shows the benefits of the combination of an unsupervised and a supervised technique.

Acknowledgment. We gratefully acknowledge financial support by the Bundesminister für Forschung und Technologie (FRG). We extend our thanks to all co-workers that have contributed to the methods for calculating the physicochemical variables. J.Z. thanks the Ministry of Science and Technology of Slovenia for financial support of part of this work. We thank Dr. Peter Löw for initially compiling the dataset.