ELSEVIER

Catalysis Today 81 (2003) 485–494

CATALYSIS TODAY

www.elsevier.com/locate/cattod

Feedforward neural networks in catalysis
A tool for the approximation of the dependency of yield on catalyst composition, and for knowledge extraction

Martin Holeňa*, Manfred Baerns

Institute for Applied Chemistry, Berlin-Adlershof, Richard-Willstätter-Straße 12, Berlin D-12489, Germany

Received 17 July 2002; received in revised form 29 January 2003; accepted 3 February 2003

Abstract

Artificial neural networks are distributed computing systems implementing the functionality characterizing biological neural networks. This way of computing has become quite successful in practical applications as a tool for solving several traditional mathematical and data-analysis tasks, such as classification, clustering, approximation and prediction. In this paper, main principles of employing multilayer perceptrons for the approximation of unknown functions are outlined, and another possible use of multilayer perceptrons in combinatorial catalysis is indicated—their use for the extraction of knowledge from experimental catalytic input and output data. To counterbalance the abstractness of the subject, the method is illustrated by applying multilayer perceptrons to data on catalyst composition and catalytic results in the oxidative dehydrogenation of propane to propene.

Keywords: Artificial neural networks; Multilayer perceptron; Dependency; Approximation; Network training; Overtraining; Knowledge extraction; Logical rules; Oxidative dehydrogenation of propane

1. Introduction and objectives

In high-throughput experimentation for rapid discovering of catalytic materials for a defined catalytic reaction, huge amounts of data on composition, properties and performance of these materials are accumulated. In optimisation of the compositions of these materials and in finding fundamental relationships for catalysis, the assessment of all the data is required. This is only possible by their appropriate analysis, through which relationships between properties of solid materials and catalytic performance can be sought. A promising approach for this purpose is the application of a non-linear regression based on artificial neural networks (ANNs). Probably the first researchers who reported an ANN application in catalysis, for assessing data on oxidative dehydrogenation of ethylbenzene and on oxidation of butane, were Hattori, Kito and co-workers [1,2]; they had already an earlier experience in using ANNs for the estimation of acid strengths of mixed oxides [3,4]. Mentioning has to be made of several other publications: Sasaki et al. applied multilayer perceptrons (MLP) to data on decomposition of NO into N_2 and O_2 [5], Hou et al. to data on acrylonitrile synthesis via propane [6], Sharma et al. to Fisher–Tropsch synthesis data [7], and Huang et al. to data on oxidative coupling of methane [8].

* Corresponding author. Tel.: +49-30-6392-4451.
E-mail address: holena@aca-berlin.de (M. Holeňa).

0920-5861/03/$ – see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0920-5861(03)00147-0

The objectives of the present contribution are to outline the main principles of this approach, and to draw the reader's attention to the fact that the usefulness of artificial neural networks consists not only in their capability to establish an approximate dependence of the performance of a catalytic material on its properties, but also in the possibility to use them for knowledge extraction. For illustration, the approaches are applied to experimental data obtained in the search for an optimum composition of a catalytic material for the oxidative dehydrogenation of propane (ODP).

2. Methodology of investigating catalysts by means of feedforward neural networks

Artificial neural networks are distributed computing systems attempting to implement a greater or smaller part of the functionality characterizing biological neural networks. They have been developed mainly due to the discrepancy between the sequential and algorithmic way of information processing in traditional computers, and the parallel and associative way in which information is processed in biological systems, as well as between the digital representation of information used in the former, and the analog representation typical for the latter.

Nowadays, numerous comprehensive monographs exist, explaining various kinds of artificial neural networks, their underlying concepts and theoretical principles, e.g. [9–12]. In the context of catalysis, the relevance of employing neural networks for approximation and knowledge extraction was recently discussed in the overview paper [13], whereas for other applications in chemistry, the reader is referred to the overview books [14,15], the overview papers [16–18] and relevant parts of the collections [19–21].

The most basic concepts pertaining to ANNs are those of a neuron, whose biologically inspired meaning is some elementary signal processing unit, and of a connection between neurons enabling the transmission of signals between them. In addition to signal transmission between different neurons, signal transmission between neurons and the environment can also take place. The whole system of neurons, connections between them, and connections between the environment and neurons together forms the architecture of the artificial neural network. The neurons that receive signals from the environment but not from other neurons are called input neurons, whereas those sending signals to the environment but not to other neurons are called output neurons. Finally, neurons that receive and send signals only from and to other neurons are called hidden neurons. Those concepts are illustrated in Fig. 1.

Although the partition of the set of neurons into input, hidden and output neurons still allows a large variety of architectures, the architecture encountered in practical applications is most often basically the same; it is a layered architecture, in which all signals propagate from layer to layer in the same direction, denoted as the forward direction. Such ANNs are called feed forward neural networks. Their most prominent example is the multilayer perceptron, in which a neuron-specific linear transformation of signals from the preceding layer is connected with each hidden or output neuron, whereas a network-specific non-linear function, called activation function, is subsequently applied to the results of the linear transformations connected with hidden neurons.

A particular MLP with one layer of hidden neurons used for approximating dependencies between

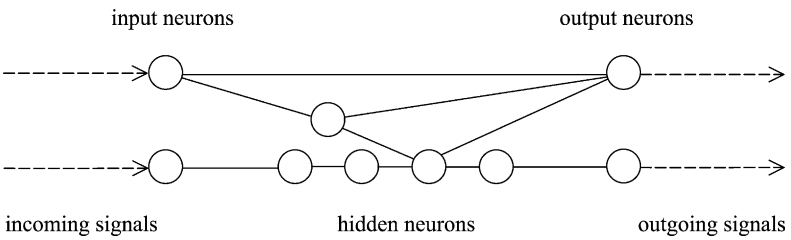

Fig. 1. A simple generic artificial neural network architecture.

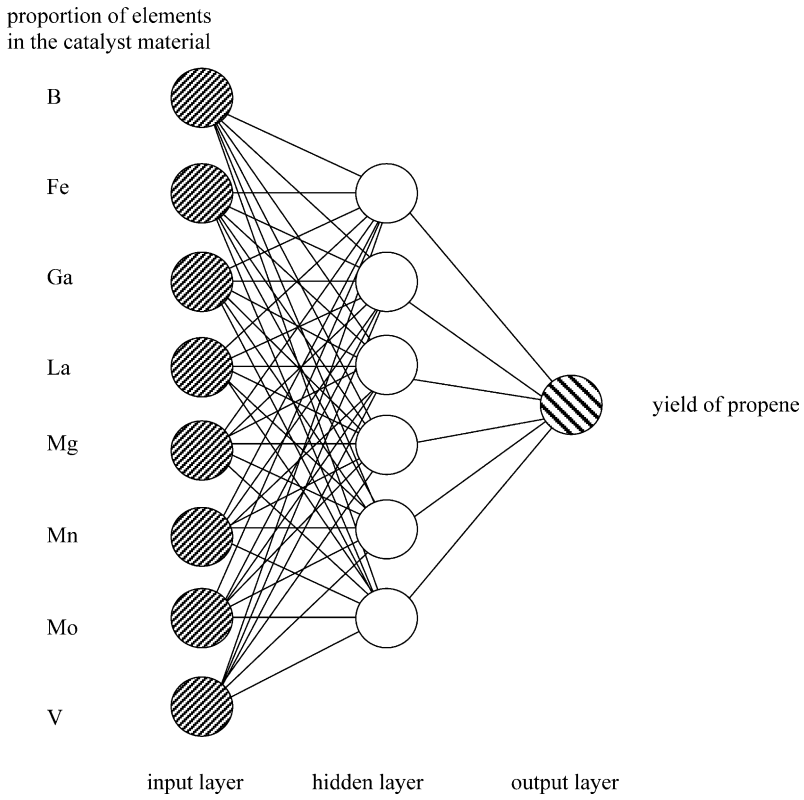

Fig. 2. Illustration of a multilayer perceptron with one layer of hidden neurons, used for correlating experimental data on the yield of propene in ODP (output) with catalyst composition (input). This architecture only indicates which elements compose the catalyst, but does not take into account any quantitative constraint on the composition.

composition and performance of catalytic data obtained from our experiments is shown in Fig. 2. An MLP with $\boldsymbol{n}_{\mathrm{I}}$ input neurons and $\boldsymbol{n}_{\mathrm{O}}$ output neurons computes a function from the space of $\boldsymbol{n}_{\mathrm{I}}$-dimensional vectors to the space of $\boldsymbol{n}_{\mathrm{O}}$-dimensional vectors. The precise form of that function depends on the specific linear transformations connected with individual hidden and output neurons. It is the set of computable functions that accounts for the most useful feature of multilayer perceptrons, i.e. their universal approximation property. This property means that even to a very general unknown dependency, and to any prescribed arbitrarily small distance in an appropriate function space, an MLP can always be found, with as few as one layer of hidden neurons, such that the function computed by this MLP lies within the prescribed distance from the unknown dependency.

2.1. Approximating an unknown dependency of the performance of a catalyst on its descriptors

In catalysis, most often an interest exists in dependencies of products yields, catalyst activity, reactant conversions, and product selectivities on catalyst descriptors. Most typically, the descriptors are proportions of individual components in catalyst material, but they may include also various physico-chemical catalyst properties and reaction conditions. To obtain a neural network that computes a function approximating such dependence, two crucial steps are needed.

(1) Choosing an appropriate architecture, which—in the case of an MLP—means to choose an appropriate number of hidden neurons, or an appropri-

ate number of hidden layers plus an appropriate number of neurons in each of them.

(2) Connecting, in a process called network training, to each hidden and output neuron the linear transformation that fits best the available data. The quality of the overall fit of all those linear transformations may be assessed by various measures [13]. The one most commonly used is the mean squared error (M.S.E.), i.e. the mean squared distance between the output values that the network computes for a given sequence of inputs, and the output values that for those inputs have been experimentally measured (Section 3). Once a measure of the overall data fit is fixed, neural network training reduces to the task of finding the linear transformations optimal with respect to that measure. In the case of M.S.E., this means finding those linear transformations that lead to the minimal sum of squared distances between the computed and measured outputs. To this end principally any optimization method can be employed. In the early decades of artificial neural networks, a variant of gradient descent was very popular in that context, called back-propagation, due to how the elements of the M.S.E. gradient are calculated [9,22]. Nowadays, back-propagation is increasingly often competed by the Levenberg–Marquardt method, a hybrid method combining gradient descent and the Gauss–Newton quadratic optimization with Jacobian-based estimation of second derivatives [9,23,24]. Compared to back-propagation, the Levenberg–Marquardt method has much higher storage requirements and is substantially more difficult to implement, but its convergence speed increases from linear to quadratic near the minimum, whereas back-propagation converges always only linearly. It is this method that we have employed in the case study reported in Section 3. We have used an implementation of the Levenberg–Marquardt method available in the Matlab Neural Network Toolbox, with the default values of method control parameters.

In reality, the choice of the number of hidden neurons cannot be separated from network training; for assessing the appropriateness of a particular architecture for the available data, first a number of different networks with that architecture have to be trained. A better fit to the data used for training does not necessarily entail a better approximation of the ultimate unknown dependency. To assess the quality of the desired approximation, an independent test set of data is needed, obeying the unknown dependency but unseen by the network during training. The phenomenon of overtraining, i.e. a good fit of the training data accompanied with a bad fit to the test data, is the main problem faced during neural network training (Fig. 3). To reduce that phenomenon, various methods have been developed, such as early stopping or Bayesian

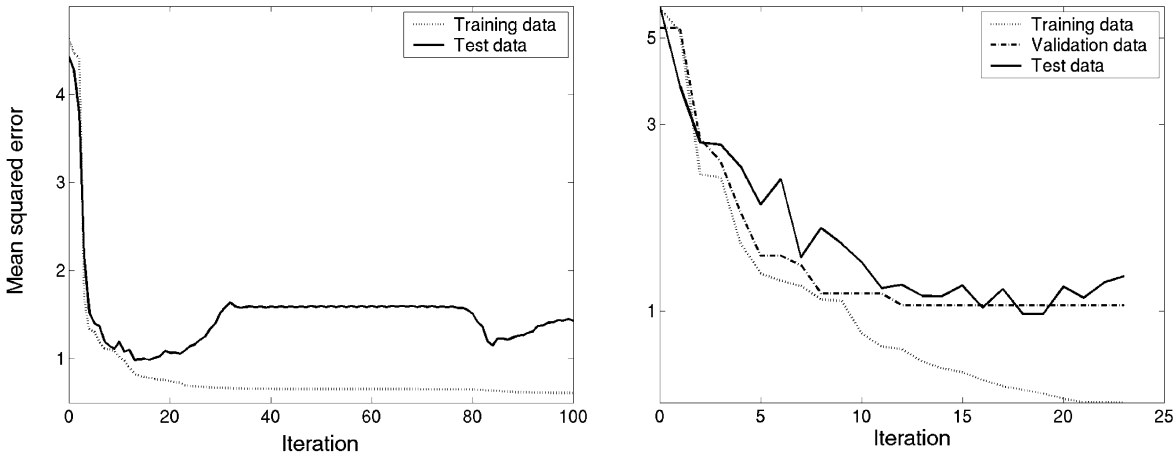

Fig. 3. An example of overtraining encountered when training an MLP with the ODP data using the Levenberg–Marquardt training method (left), and an example for reduction of overtraining using the early-stopping method (right) [13].

regularization. The former stops network learning in an iteration from which the quality of an overall fit on another independent set of data not used for training, called validation data, decreases. The latter views the linear transformations connected to hidden and output neurons as random variables with particular distribution, and overtraining as realizations that are not likely for that distribution; hence it can be reduced if a term penalizing improbable realizations is incorporated into the quality measure of the overall fit of the linear transformations (see [13] for details). Both these overfitting reduction methods have been combined with the Levenberg–Marquardt method in the case study in Section 3. Nevertheless, to fully exclude an influence of overtraining when choosing the number of neurons, the choice must be based on the fit to the test data (Fig. 4).

2.2. Extraction of knowledge about catalyst performance

The architecture of a trained neural network and the mapping computed by the network inherently represent the knowledge that has been contained in the data used to train the network, knowledge about relationships between the input and output variables, e.g. between catalyst composition and yield. However, except for low-dimensional relationships that can be visualised (Fig. 5), such a representation is not easily human-comprehensible, since it is very far from the symbolic and modular way of how people represent knowledge by themselves. That is why various knowledge extraction methods have been developed, attempting to transform the network architecture and the computed mapping into a human-comprehensible representation. In catalysis, interest may primarily exist in two kinds of knowledge:

1. Knowledge about optima of the approximated dependencies, e.g. knowledge about the catalyst composition corresponding to the maximal yield of a product in a complex reaction network. An illustration is presented in Section 3, Table 2.
2. Logical rules for the dependencies of the output variables on the input variables, i.e. implications of the general kind “If the input variables fulfil some particular input condition, then the output variables are likely to fulfil some particular output condition.”

An example of that kind of knowledge extracted in the application described in Section 3 is provided in Fig. 7 there. Methods for the extraction of knowledge from data by means of artificial neural networks are actually only a specific subclass of the data mining methods for the extraction of logical rules from data. The most important other representatives of those *methods*

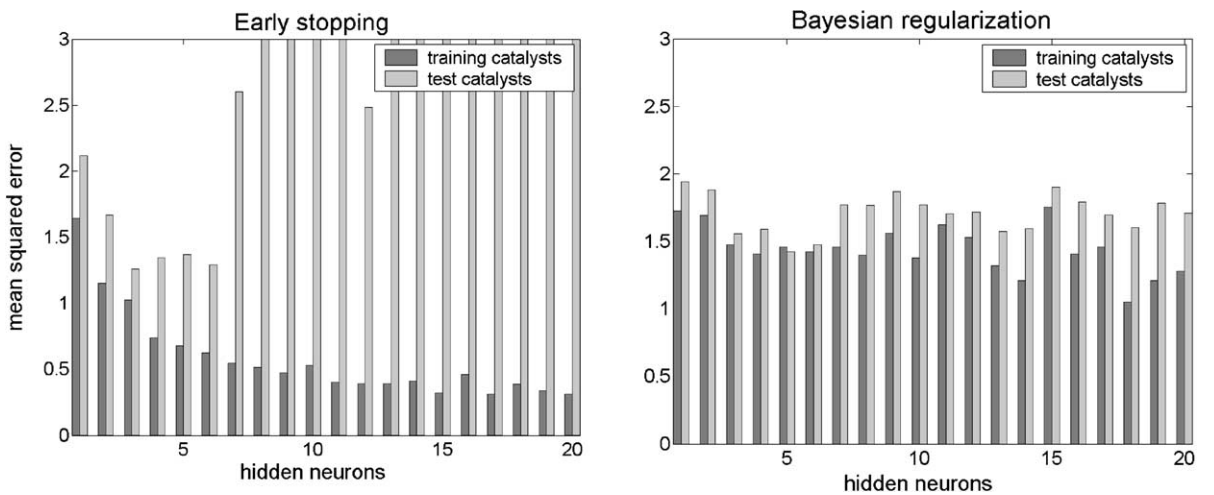

Fig. 4. Quality of fit for various numbers of hidden neurons in 1-hidden-layer MLPs trained with the ODP data using the Levenberg–Marquardt method and 2 different methods to reduce overtraining. The value for each number of hidden neurons has been averaged by means of a 22-fold cross-validation.

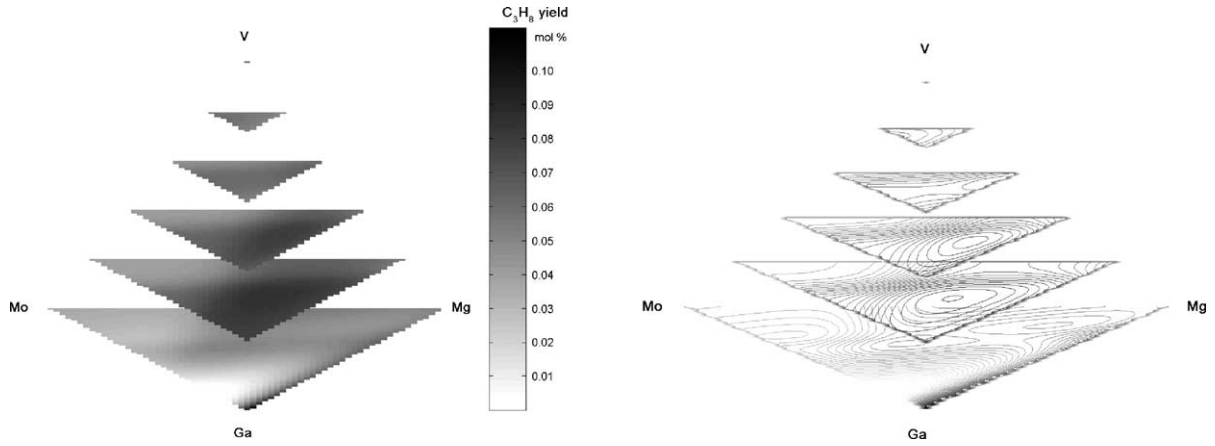

Fig. 5. Visualization of knowledge on the dependency of propene yield on the composition of catalysts limited to the elements Ga, Mg, Mo, V through colored slices and isoline-slices as computed by an MLP trained with ODP data.

are the extraction of association rules from empirical distributions, rule extraction based on statistical hypotheses testing, and the extraction of classification rules from various kinds of classification trees [25]. However, whereas the remaining rule extraction methods are direct methods, i.e. they extract rules directly from data, all the various existing ANN-based methods extract them not directly from data, but from an approximation computed by a neural network trained with the data [26–28]. The rule-extraction method that we have used has been sketched in [13]. Main principles on which the method relies are the density of piecewise-linear sigmoidals in the metric space of continuous sigmoidals, and the preservation of polyhedra by MLPs with piecewise-linear activation functions.

3. Application of an ANN to the oxidative dehydrogenation of propane

ODP can be carried out in the temperature range between approximately 620 and 800 K with redox-type metal oxides or with metal oxides or noble metals on which oxygen is adsorbed [29–31]; the lattice oxygen of the former materials and the adsorbed oxygen in the latter ones participate in oxygen abstraction. In the higher temperature range a significant proportion of homogeneous non-catalytic dehydrogenation occurs being initiated by catalytic surface reactions. In the present work to which the ANN was applied catalytic metal oxides of different qualitative and quantitative compositions were used. A total of 226 different compositions were prepared and tested for their catalytic performance, i.e. selectivity and yield towards propene at standard conditions (standard conditions: $T = 773\,\mathrm{K}$, $p_{C_3H_8} = 30.3\,\mathrm{kPa}$, $p_{O_2} = 10.1\,\mathrm{kPa}$, $p_{N_2} = 60.6\,\mathrm{KPa}$) while the contact time was adjusted in such a way that oxygen conversion was in the limits between 85 and 100% to make the output data comparable. The materials consisted of an α-alumina support which was coated with a shell of metal oxide mixtures selected from the oxides of B, Fe, Ga, La, Mg, Mn, Mo, V; the details of their selection and the optimisation of their composition by applying a genetic algorithm have been reported elsewhere [32,33]. The propene yield was used as objective function in the optimisation process (output data) and the composition of the catalytic material corresponded to the input data of the ANN.

Because the molar proportions of all elements whose oxides form the catalyst shell sum up to 100%, one of those proportions is always derivable from the remaining 7. Let us fix it to be the proportion of *V*. Due to that quantitative constraint, each multilayer perceptron that should approximate the dependence of propene yield on the proportions of shell metals must have seven input neurons, corresponding to the proportions of B, Fe, Ga, La, Mg, Mn, Mo (compared to Fig. 2, the input for *V* is now missing), and one output neuron, corresponding to the resulting yield. As far as

hidden neurons are concerned, we decided to restrict our attention to perceptrons with one hidden layer. This restriction was based on the one hand on theoretical properties of perceptrons with one and two hidden layers [34,35], on the other hand on our experience from training one- and two-hidden-layer perceptrons with other data from the same reaction, which was reported in [13]. In those experiments, architectures with only one hidden layer showed clearly better generalization ability. To find the most appropriate number of neurons in the single hidden layer, perceptrons with 1–20 neurons were trained and their generalization ability, measured with the M.S.E. of test data, was compared by means of a 22-fold cross-validation, using 216 catalysts as the training set and 10 catalysts as the test set. The Levenberg–Marquardt method has been used for network training, both early stopping and Bayesian regularization for overtraining reduction. For early stopping, 25% of the training set were split off as validation data.

Two types of results were derived. Firstly, the dependency of propene yield on catalyst composition was approximated by the ANN, and secondly logical rules with a prescribed consequent were extracted from the same trained neural network. Results on prediction of propene yield put aside as the test data are for the considered network shown in Table 1and Fig. 6. In general, there is a rather good agreement between prediction and available experimental data not used for training: the mean absolute difference between measured and predicted propene yield values is 0.3%, this represents a relative error of 5.4% with respect to the average measured propene yield.

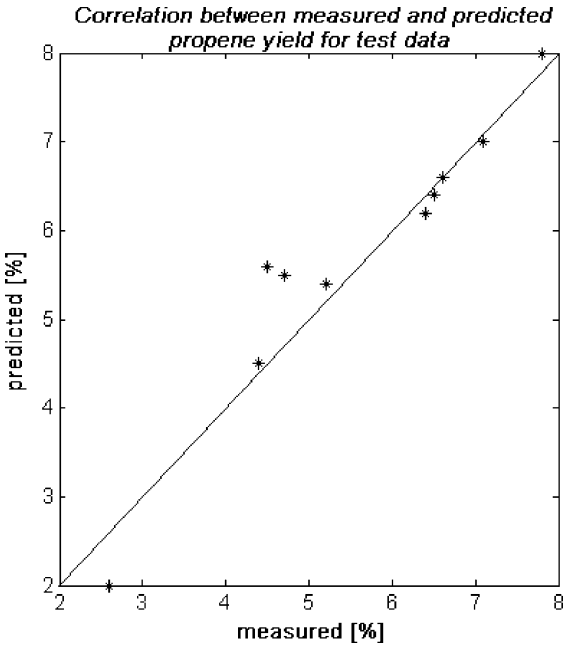

Fig. 6. Correlation between the measured and predicted propene yield values of 10 test catalysts for the neural network considered in Table 1.

Table 1
Example of validation of a network with test data

Composition of catalyst (mol.%)								Propene yield (%)	
B	Fe	Ga	La	Mg	Mn	Mo	V	Predicted	Measured
0	23	21	0	0	0	19	37	6.2	6.4
0	0	13	0	21	0	0	67	5.6	4.5
18	8.9	15	0	0	59	0	0	5.4	5.2
0	16	9.6	0	37	0	13	24	7.0	7.1
0	0	8.4	0	38	0	26	27	6.4	6.5
0	10	51	0	16	0	11	12	6.6	6.6
0	0	23	0	26	0	12	39	8.0	7.8
5.3	0	9.7	0	72	13	0	0	2.0	2.6
0	61	8.9	8.7	0	0	0	22	5.5	4.7
0	0	0	41	0	8.9	25	25	4.5	4.4

Standard reaction conditions: $C_3H_8/O_2/N_2$ feed gas ratio = 3/1/6, O_2 conversion >80%, T: 773 K, $p_{C_3H_8}$: 30.3 kPa, p_{O_2}: 10.1 kPa, p_{N_2}: 60.6 kPa. The network considered in the table has one hidden layer with five neurons, it has been trained with the Levenberg–Marquardt method, using early stopping for overfit protection. The training led to the mean squared error for training data 0.76 and for test data 0.23 (absolute values if propene yield is expressed in percent); a value for test data lower than the value for training data indicates that overfit is negligible.

Table 2
Example points of maximal function values computed with a feedforward network

Neural network		Proportion in catalyst (mol.%)								Propene yield (%)	
Overfit protection	Hidden neurons	B	Fe	Ga	La	Mg	Mn	Mo	V	Predicted	Measured
Early stopping	3	0	0	57	0	0	0	1.4	41	8.0	7.9
Early stopping	5	0	0	35	0	26	0	16	23	8.6	8.2
Early stopping	4	0	0	54	0	0	0	22	25	9.2	7.2
Early stopping	5	0	0	35	0	37	0	11	17	8.1	9.1
Bayesian regularization	5	0	0	57	0	0	0	16	27	8.8	7.9
Bayesian regularization	5	0	0	67	0	0	0	13	20	9.0	7.1

Standard reaction conditions as in Table 1.

A further example is given in Table 2—the prediction of optimal catalyst compositions corresponding to the maximal propene yield for six different ANN-based approximations. The maxima were computed using the sequential quadratic programming method [36–38]. Experimental values of propane yield were for those compositions obtained through subsequent experimental verification. Hence, differently to the experimental values in Table 1, those experimental results were not used as a test set during network training. The mean absolute difference between measured and predicted propene yield values is now 1.1%, respecting a relative error of 14% with respect to the average measured propene yield. This is substantially higher than in Table 1, but not surprising—the points of local extrema (maxima and minima) of an approximation are likely to belong also to the points of its extremal deviation from the approximated function. It is interesting to note that the optimal catalyst composition always contains the

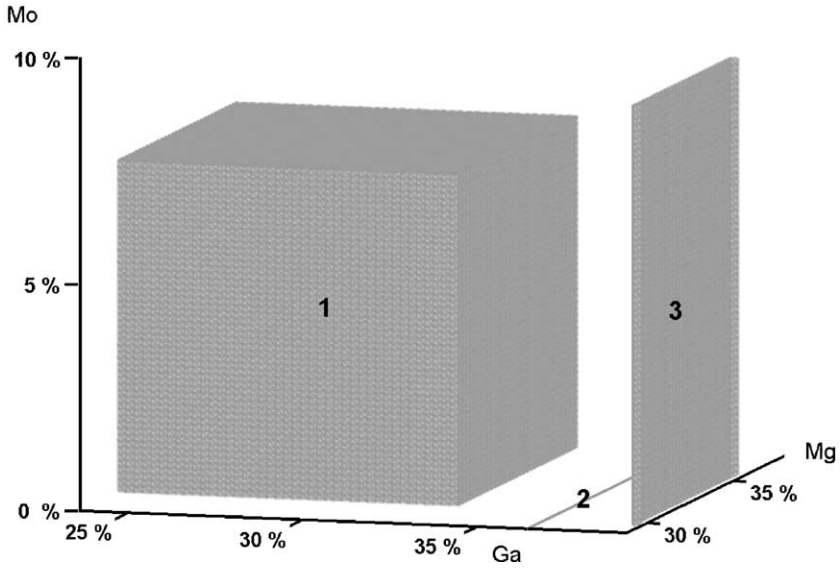

Fig. 7. A visualization of the three-dimensional projection to the dimensions corresponding to Ga, Mg and Mo of the antecedents (left-hand sides) of the following three rules, numbered 1–3, which have been extracted from the approximation computed by the neural network considered in Table 1. *Rule 1*: If the concentrations fulfil $24\% \leq \text{Ga} \leq 33\%$ and $31\% \leq \text{Mg} \leq 39\%$ and $0\% \leq \text{Mo} \leq 7\%$ and the remaining concentrations are 0%, then yield of propene $\geq 8\%$. *Rule 2*: If the concentrations fulfil $\text{Ga} \approx 36\%$ and $28\% \leq \text{Mg} \leq 38\%$ and the remaining concentrations are 0%, then yield of propene $\geq 8\%$. *Rule 3*: If the concentrations fulfil $0\% \leq \text{Fe} \leq 12\%$ and $\text{Ga} \approx 38\%$ and $29\% \leq \text{Mg} \leq 36\%$ and $0\% < \text{Mo} < 9\%$ and the remaining concentrations are 0%, then yield of propene $\geq 8\%$.

oxides of V, Mo, Ga and in two cases of Mg. This is surprising since we concluded from earlier work that MgO is an absolute requirement and Mo oxide is not unconditionally necessary. These highly interesting results need further experimental evidence.

Finally, logical rules have been extracted from the ANN, which are illustrated in Fig. 7.

4. Conclusions

The basic principles of artificial neural networks were explained. Such networks are suited for establishing relationships between catalyst properties and catalytic performance. In the presents paper, only the property "composition" was considered; in principle, however, also other properties may be included. Such properties may be optimized with respect to maximal performance. This has been illustrated in a case study for the catalytic oxidative dehydrogenation of propane to propene. It is anticipated that this approach will eventually be also used for deriving generic fundamental relationships in catalysis, thus proving that high-throughput experimentation is not only assisting in rapid catalyst development but may also lead to more fundamental insights in this area of science.

Acknowledgements

This work was funded by the German Ministry of Education and Research (BMBF) and the State of Berlin, Department of Science, Research and Culture (grant no. FKZ 03C3013). Discussions within the COMBICAT consortium supported by the EC of the European Union were highly appreciated.

References

[1] T. Hattori, S. Kito, Catal. Today 23 (1995) 347.
[2] S. Kito, T. Hattori, Y. Murakami, Appl. Catal. A 114 (1994) L173.
[3] T. Hattori, S. Kito, H. Niwa, Y. Westi, A. Satsuma, Y. Murakami, Studies in surface science and catalysis 90 (1994) 229.
[4] S. Kito, T. Hattori, Y. Murakami, Ind. Eng. Chem. Res. 31 (1992) 979.
[5] M. Sasaki, H. Hamada, Y. Kintachi, T. Ito, Appl. Catal. A 132 (1995) 261.
[6] Z.Y. Hou, Q. Dai, X.Q. Wu, G.T. Chen, Appl. Catal. A 161 (1997) 183.
[7] B.K. Sharma, M.P. Sharma, S.K. Roy, S. Kumar, S.B. Tendulkar, S.S. Tambe, B.D. Kulkarni, Fuel 77 (1998) 1763.
[8] K. Huang, C. Feng-Qiu, D.W. Lü, Appl. Catal. A 219 (2001) 61.
[9] M.T. Hagan, H. Demuth, M. Beale, Neural Network Design, PWS Publishing, Boston, 1996.
[10] S. Haykin, Neural Networks. A Comprehensive Foundation, IEEE, New York, 1999.
[11] K. Mehrota, C.K. Mohan, S. Ranka, Elements of Artificial Neural Networks, MIT Press, Cambridge, 1997.
[12] H. White, Artificial Neural Networks: Approximation and Learning Theory, Blackwell Scientific Publishers, Cambridge, 1992.
[13] M. Holeňa, M. Baerns, in: J.N. Cawse (Ed.), Experimental Design for Combinatorial and High Throughput Materials Development, Wiley, New York, 2003, p. 163.
[14] J. Zupan, J. Gasteiger, Neural Networks for Chemists, Wiley–VCH, Weinheim, 1993.
[15] J. Zupan, J. Gasteiger, Neural Networks in Chemistry and Drug Design: An Introduction. Wiley–VCH, Weinheim, 1999.
[16] M.A. Henson, Comput. Chem. Eng. 23 (1998) 187.
[17] W.J. Melssen, J.R.M. Smits, L.M.C. Buydens, G. Kateman, Chemom. Intell. Lab. Syst. 23 (1994) 267.
[18] J.R.M. Smits, W.J. Melssen, L.M.C. Buydens, G. Kateman, Chemom. Intell. Lab. Syst. 22 (1994) 165.
[19] J.W. Clark (Ed.), Scientific Applications of Neural Nets, Springer, Berlin, 1998.
[20] P.G.J. Lisboa (Ed.), Neural Networks: Current Applications, Chapman & Hall, London, 1992.
[21] A.F. Murray (Ed.), Applications of Neural Networks, Kluwer Academic Publishers, Dordrecht, 1994.
[22] D.E. Rumelhart, G.E. Hinton, R.J. Williams, in: D.E. Rumelhart, J.L. McClelland (Eds.), Parallel Data Processing, vol. 1, MIT Press, Cambridge, 1986, p. 318.
[23] M.T. Hagan, M. Menhaj, IEEE Transactions on Neural Networks 5 (1994) 989.
[24] D. Marquardt, SIAM J. Appl. Math. 11 (1963) 431.
[25] M. Holeňa, in: A.K. Hyder, E. Shahbazian, E. Waltz (Eds.), Multisensor Fusion, Kluwer Academic Publishers, Dordrecht, 2002, p. 511–532.
[26] R. Andrews, J. Diederich, A.B. Tickle, Knowledge based systems 8 (1995) 378.
[27] S. Mitra, Y. Hayashi, IEEE Trans. on Neural Networks 11 (2000) 748.
[28] A.B. Tickle, R. Andrews, M. Golea, J. Diederich, IEEE Trans. on Neural Networks 9 (1998) 1057.
[29] O.V. Buyevskaya, M. Baerns, Catalysis 16 (2002) 155.
[30] F. Cavani, F. Trifforo, Catalysis Today 24 (1995) 307.
[31] V.A. Mamedov, V. Cortes-Corberan, Appl. Catal. A 127 (1995) 10.
[32] O.V. Buyevskaya, A. Brückner, E.V. Kondratenko, D. Wolf, M. Baerns, Catalysis Today 67 (2001) 369.

[33] D. Wolf, O.V. Buyevskaya, M. Baerns, Appl. Catal. A 200 (2000) 63.
[34] K. Hornik, Neural Networks 4 (1991) 251.
[35] G. Brightwell, C. Kenyon, H. Paugam-Moisy, in: M.C. Mozer, M.T. Jordan, T. Petsche (Eds.), Advances in Neural Information Processing Systems, MIT Press, Cambridge, 1997.
[36] R. Fletcher, Practical Methods of Optimization, vol. 1; Unconstrained Optimization, vol. 2, Constrained Optimization, Wiley, New York, 1980.
[37] P.E. Gill, W. Murray, M.H. Wright, Practical Optimization, Academic Press, London, 1981.
[38] K. Schittkowski, Ann. Operat. Res. 5 (1985) 485.