Neural networks in high-performance liquid chromatography optimization

Metting, H.J; Coenegracht, P.M J

Published in:
Journal of Chromatography

Publication date:
1996

ELSEVIER

Journal of Chromatography A, 728 (1996) 47–53

JOURNAL OF CHROMATOGRAPHY A

Neural networks in high-performance liquid chromatography optimization: response surface modeling

Harm J. Metting, Pierre M.J. Coenegracht*

Chemometrics Research Group, University Centre for Pharmacy, A. Deusinglaan 2, NL-9713 AW Groningen, Netherlands

Abstract

The usefulness of artificial neural networks for response surface modeling in HPLC optimization is compared with (non-)linear regression methods. The number of hidden nodes is optimized by a lateral inhibition method. Overfitting is controlled by cross-validation using the leave one out method (LOOM). Data sets of linear and non-linear response surfaces (capacity factors) were taken from literature. The results show that neural networks offer promising possibilities in HPLC method development. The predictive results were better or comparable to those obtained with linear and non-linear regression models.

Keywords: Optimization; Neural networks; Response surface modelling

1. Introduction

1.1. Optimization of the mobile phase composition

Optimization of the mobile phase composition is an important aspect of method development in HPLC. Retention mapping methods are important optimization methods because the global optimum can be found [1,2]. The aim of retention mapping methods is to describe the chromatographic behaviour of a solute in the design space by its response surface, which shows the relationship between the response, i.e., the capacity factor of a solute, and several input variables, i.e., the components of the mobile phase.

Retention mapping methods are also called simultaneous methods or regression methods because the model that describes the retention as a function of the mobile phase composition is often estimated by multiple (non-)linear regression. To construct the model a minimum number of experiments has to be performed i.e., the capacity factor has to be measured at a number of mobile phase compositions or design points. These measurements are performed 'simultaneously' according to an experimental design and used for modeling the response surface of every solute in the sample. The capacity factor of every solute in the sample can then be predicted at every mobile phase composition in the design space. This means that for every mobile phase composition a chromatogram can be predicted. The best chromatogram is selected by an optimization procedure, which will not be discussed here, because the focus of this article is on response surface modeling.

1.2. Response surface modeling

The factor space of a ternary eluent consisting of water or a buffer, methanol (MeOH) and acetonitrile (MeCN) can be represented by a triangle at the vertices of which the pure solvents are located. If

*Corresponding author.

0021-9673/96/$15.00
SSDI 0021-9673(95)00022-4

water (component x_1) is placed at the top of the triangle and the two organic modifiers (x_2, x_3) at the vertices of the ground side, then the solvent strength of the mobile phase systems decreases into the direction of the top of the triangle as the fraction of water of the system increases.

The design space, in which the design points are to be located, is found by constraining the factor space. For the non-isoeluotropic solvent system described by the mixture triangle, two constraints on the solvent strength are needed; an 'upper' constraint at low solvent strength in order to obtain a capacity factor of the last peak smaller than a given acceptable value, 20 for example, and a 'lower' constraint at high solvent strength that confines the capacity factor of the first peak to values greater than 1 (Fig. 1A).

The number of the design points that is required to construct the response surface by multiple linear regression, depends on the dimensions of the factor space and the order of the model. This discussion will be limited to the use of quadratic models. The quadratic mixture model for the capacity factor of one solute in a ternary mobile phase system is:

$$\ln(k') = a_1x_1 + a_2x_2 + a_3x_3 + a_{12}x_1x_2 + a_{13}x_1x_3 + a_{23}x_2x_3 + E \quad (1)$$

where x_1, x_2, x_3, are fractions of the components: water, MeOH, MeCN; a_1 to a_{23} are the coefficients to be estimated and E is the residual error. For the estimation of the coefficients and E we need in this case at least 7 experiments. The number of required experiments is determined by the necessity to estimate the coefficients a and the error term E of Eq. 1. For a quadratic mixture model in a quaternary system 10 model coefficients have to be estimated. The eleven experiments should be performed simultaneously and in a random order.

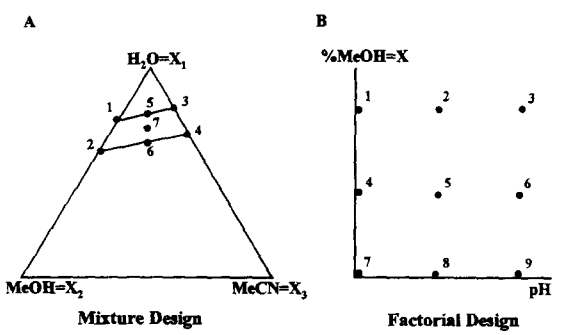

Fig. 1. (A) Mixture triangle for a three component phase system. Indicated is the design space with seven of the design points. (B) Factorial design for a two component system. Indicated is the design space with nine of the design points.

Simple polynomials, i.e. a quadratic model, can predict well the capacity factors of simple solutes in isoeluotropic ternary, quaternary, and non-isoeluotropic ternary and quaternary systems consisting of water and organic modifiers [2].

For the separation of ionogenic solutes, variation of the pH leads to dramatic changes of the selectivity. If the pH is introduced as an independent variable or factor, then a second factor, usually the fraction of an organic modifier, has to be varied simultaneously to define a suitable design space wherein the capacity factor has acceptable values [3]. The factor space can now be defined by two orthogonal variable axis: the pH and the fraction of the organic modifier (Fig. 1B).

Introduction of the pH as a variable affects not only retention, but also column efficiency and peak shape. This discussion is limited to the estimation of a model that describes the capacity factor as a function of pH and fraction modifier.

Polynomial models are inadequate to approximate the response surface and non-linear models have to be used, for example:

$$k' = \frac{a_1[H^+] + a_2}{[H^+] + a_3} \cdot \exp\{(a_4 + a_5[H^+] + a_6[H^+]^2)x + (a_7 + a_8[H^+])x^2\} + E \quad (2)$$

where x is the fraction of the modifier and a_1 to a_8 are the coefficients which have to be estimated by a non-linear regression method [3]. These methods usually need initial estimates of the coefficients and iteratively approximate the response surface.

1.3. Neural networks

In the last years artificial neural networks (ANNs) have generated a lot of interest. A variety of different

types of ANNs have been developed to simulate different activities of the human brain. These ANNs can be applied to various tasks of information processing: classification, modeling, association, and mapping [4,5]. They have been used for curve fitting [6]. In analytical chemistry neural networks have been applied to spectroscopy [7,8] and electrochemistry [9]. In HPLC optimization a neural network is used for peaktracking by means of spectral recognition [10]. In clinical chemistry an ANN was used to predict urinary calculus compositions [11].

In this paper the usefulness of ANNs for response surface modeling is investigated for the first time and compared to established methods like (non-) linear regression.

ANNs are mathematical systems that simulate biological neural networks. They consist of processing elements (neurons, nodes) organized in layers. A general description of a processing unit is given in Fig. 2. Backpropagation neural networks (BNNs) are most often used in analytical applications. BNNs model relations between sets of input and output data by minimizing the output errors during training. The model is stored in the connections (weights) between the processing layers. A problem in constructing a BNN is to find the optimal number of hidden neurons. Another problem of BNN training is overfitting, which occurs when the training data contain noise and the BNN is modeling the noise instead of the underlying features. Overfitting can be controlled by means of crossvalidation during training of the network.

2. Theory

The neural network used in this work is of the backpropagation neural network (BNN) type. A typical feedforward neural network with backpropagation has three layers: the input, the hidden, and the output layer (Fig. 3). Information in a BNN is stored in the weights, which are connections between neurons in successive layers and in the bias values of a neuron. W_{ji} is the weight-connection to neuron j in the actual layer from neuron i in the previous layer and $bias_j$ is the bias of neuron j. The activation of a neuron is defined as the sum of the weighted input signals to that neuron (Eq. 3)

$$Net_j = \sum_i w_{ji} x_i + bias_j \tag{3}$$

This activation is transformed to the neuron output by means of an activation function which is symmetrical sigmoid in this paper (Eq. 4).

$$y_j = \frac{2}{1 + e^{-Net}} - 1 \tag{4}$$

The error E of a network is defined as the squared difference between the target values t and the outputs y of the output neurons summed over p training patterns and j output nodes (Eq. 5).

$$E = \frac{1}{2} \sum_p \sum_j (y_{pj} - t_{pj})^2 \tag{5}$$

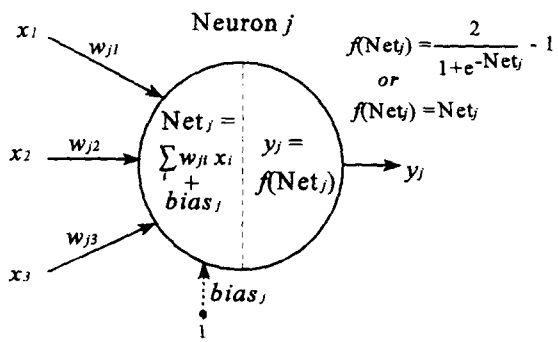

Fig. 2. General scheme of a processing unit. y_j is output of neuron j. w_{ji} is weight from neuron j to neuron i. $bias_j$ is bias of neuron j. Net_j is the sum of the weighted inputs of neuron j. $f(Net_j)$ is the transfer function of neuron j which may be sigmoid or linear.

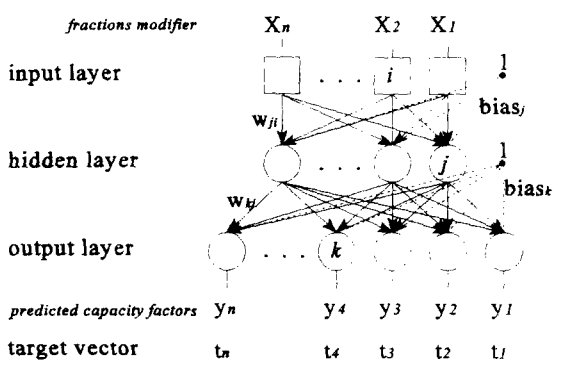

Fig. 3. Configuration of a feedforward backpropagation neural network. Squares indicate inputs, circles indicate nodes, arrows between the nodes represent weights. Dashed lines respresent the biases of the nodes.

During training of the network an input vector $\mathbf{x}_p$, for example the fractions of the modifiers in a design-point, is presented to the input layer. Each node in the input layer is connected to each node in the hidden layer and each node in the hidden layer is connected to each node in the output layer which produce the output vector $\mathbf{y}_p$.

The number of input nodes equals the number of elements of vector $\mathbf{x}_p$ (number of modifier fractions) and the number of output nodes equals the number of elements of vector $\mathbf{y}_p$ (*i.e.* the number of solutes in the chromatographic sample); the number of hidden units is an adjustable parameter and has to be optimized.

The goal of training the network is to change the weights between the layers in a direction that minimizes the error E according the steepest descent method (Eq. 6).

$$\Delta w_{ji}(n) = -\eta \frac{\partial E}{\partial w_{ji}} \tag{6}$$

where η is a positive constant known as the 'learning rate' and $\Delta w_{ji}(n)$ the current weight change for the weight w_{ji}. This gradient descent method can be enhanced by a 'momentum term' from previous weight changes as

$$\Delta w_{ji}(n) = -\eta \frac{\partial E}{\partial w_{ji}} + \alpha \Delta w_{ji}(n-1) \tag{7}$$

where α (momentum factor) is another constant which leads to

$$\Delta w_{ji}(n) = \eta \delta_j y_i + \alpha \Delta w_{ji}(n-1) \tag{8}$$

and

$$\Delta bias_j(n) = \eta \delta_j + \alpha \Delta bias_j(n-1) \tag{9}$$

(The bias term can be treated as a weight by assuming that it is connected to a node with an output of unit value)
where

$$\delta_j = \frac{1}{2}(1 - y_j^2)(t_j - y_j) \text{ for output units,} \tag{10}$$

and

$$\delta_j = \frac{1}{2}(1 - y_j^2)\sum_k \delta_k w_{kj} \text{ for hidden units.} \tag{11}$$

In the case of hidden units δ_k values are propagated backward from upper layer(s) (starting with the output layer) to lower layer(s) (hidden layer(s)). These forward and backward processes are repeated in cycles until the output vector $\boldsymbol{y}$ matches the target vector $\boldsymbol{t}$ within a minimum E.

The effect of the 'momentum term' is damping of oscillations and a faster training in flat plateaus of the Error space.

3. Experimental

Data has been taken from literature [3,12]. For response surface modeling the following data sets have been used. (A) The capacity factors of 6 benzene derivatives measured at 17 different mobile phase compositions in a quaternary mobile phase system of water, MeOH, MeCN and THF (Table 2 in Ref. [2]). (B) The capacity factors of 15 mainly benzene derivatives measured at 9 different mobile phase compositions in a ternary phase system of water, MeOH and MeCN (Table 1 in Ref. [7]). (C) The capacity factors of 3 solutes measured at 36, 23, and 40 different mobile phase compositions respectively, in mobile phase systems of MeOH–buffers and MeCN–buffers varying in pH from 2.6 to 7 (Table 3 and Table 4 basic solutes and Table 5 in Ref. [3]).

3.1. Software

The artificial neural network program was written in C based on an algorithm which adjusts the learning rate factor and the momentum factor during training [14]. The optimum number of hidden nodes was determined by an algorithm of Syozo Yasui [13], which eliminates redundant hidden nodes by lateral inhibition. Multiple linear regression was performed with POEM version 3.0 (University Centre of Pharmacy, Groningen, Netherlands); non-linear regression was performed with Sigmaplot version 5.01 (Jandel Scientific, Erkrath, Germany).

4. Results

Before modeling the capacity factors are logarithmically transformed in order to get a homogeneous variance in y.

The response surfaces of the capacity factors of the benzene derivatives of datasets A and B were estimated by linear regression using Eq.1 for the ternary system of set B and a 10-term polynomial which is an extension of Eq.1 with the appropriate terms of x_4, for set A. The data is also modeled by a neural network. The number of input nodes equals the number of components in the eluent and the number of output nodes equals the number of solutes in the chromatographic sample. The number of hidden nodes was optimized by eliminating redundant hidden nodes by lateral inhibition [13].

Because of the small data sets no independent test sets can be derived from the data sets. Overfitting is controlled by crossvalidation using the leave one out method (LOOM), which means that the neural network is trained on $(p-1)$ patterns (design points) for t cycles and the pattern (design point) left out is used as a test pattern every Δt cycles intervals where the $\ln(k')$ value is predicted. This procedure is repeated p times leaving out the next pattern which gives p-test errors from the whole data set. From the test errors a mean test error is calculated as:

$$\text{Mean Test Err.} = \frac{1}{p}\sum_{n=1}^{p}\frac{1}{k}\sum_{j=1}^{k}(y_{nj}-t_{nj})^2 \qquad (12)$$

To follow the progress of the mean training error and the mean test error the network is trained for 5000 cycles using LOOM. Every 50 cycles the training error and the test error are recordered. From the mean test error curve the optimum number of training cycles can be obtained (Fig. 4). When the test error begins to increase overfitting may occur. The time necessary to find an optimal network using LOOM depends on the structure of the network, the number of design points and can be rather time consuming. For example the data set from Table 1 requires 13 min training time for 5000 cycles (LOOM) on a Pentium 100 MHz.

To compare the predictive power of the regression model with the neural network model MPRESS values (mean predicted error sum of squares) were

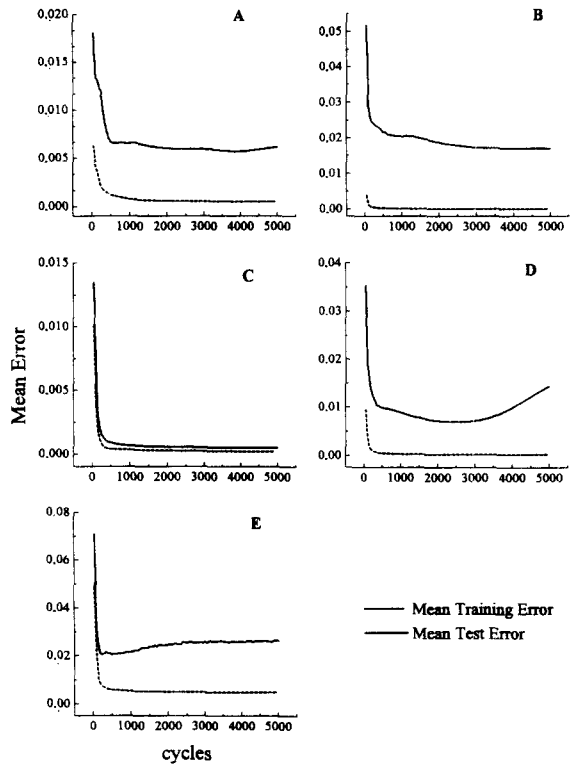

Fig. 4. Plot of the mean training error (dashed lines) and the mean test error (solid lines) against the number of cycles during training of the BNN to determine the optimal number of cycles using LOOM. (A) data-set from Table 1; (B) data-set from Table 2; (C) data-set containing solute 1 from Table 3; (D) data-set containing solute 2 from Table 3; (E) data-set containing solute 3 from Table 3.

calculated from the predicted $\ln(k')$ at the optimum number of cycles using LOOM.

$$\text{MPRESS} = \frac{1}{p}\sum_{n=1}^{p}(y_{nj}-t_{nj})^2 \qquad (13)$$

In Eq. 12 and Eq. 13, y_{nj} represents the predicted capacity factor of solute j of the nth observation with a model constructed by training the neural network without the nth observation (LOOM) and t_{nj} is the measured capacity factor for that observation and k represents the number of output neurons (number of solutes).

MPRESS values were calculated at a training time of 500 cycles for all data sets and at 4000 cycles for data sets from Table 1 and Table 2 and at 2500

Table 1
MPRESS values of polynomial and neural network models based on ln(k')

Model	Solute					
	1	2	3	4	5	6
MM10	0.0141	0.0271	0.0118	0.0141	0.0135	0.0153
NN4 500	0.0076	0.0106	0.0059	0.0041	0.0059	0.0041
NN4 4000	0.0059	0.0082	0.0057	0.0026	0.0076	0.0036

Models: MM10=10 term quadratic mixture model; NN4 500=neural network model with 4 hidden nodes at 500 cycles; NN4 4000=neural network mode with 4 hidden nodes at 4000 cycles (optimum). Solutes: six benzene derivatives (number objects=17) (Table 2 of Ref. [2]).

Table 2
MPRESS values of polynomial and network models based on ln(k')

Model	Solute							
	1	2	3	4	5	6	7	8
MM6	0.0267	0.0467	0.0167	0.0411	0.0411	0.0267	0.0189	0.0422
NN8 500	0.0111	0.0122	0.0089	0.0133	0.0233	0.0156	0.0122	0.0167
NN8 4000	0.0091	0.0158	0.0079	0.0164	0.0236	0.0066	0.0061	0.0201
Model	Solute							
	9	10	11	12	13	14	15	
MM6	0.0656	0.0533	0.0411	0.0489	0.1167	0.1678	0.1756	
NN8 500	0.0222	0.0178	0.0133	0.0233	0.0344	0.0511	0.0489	
NN8 4000	0.0194	0.0187	0.0156	0.0245	0.0560	0.0504	0.0244	

Models: MM6=6 term quadratic mixture model; NN8 500=neural network with 8 hidden nodes at 500 cycles; NN8 4000=neural network with 8 hidden nodes at 4000 cycles (optimum). Solutes: 15 mainly benzene derivatives (number objects=9)(Table 1 of Ref. [12]).

cycles for data sets from Table 3 (except data set containing solute 3 which has a minimum mean test error at 500 cycles). After 500 training cycles the test error does not decrease very much (Fig. 4) and the training time is much shorter (85 s for the data set from Table 1).

The MPRESS values for the polynomial models and the corresponding network models of data A and B are presented in Table 1 and Table 2. As the quality of fit of the polynomial models is already good [2], it is even more surprising to see even better predictive power of the network model. A possible explanation may be the fact that in the regression method each solute has its own model. The network, however, constructs one model for all solutes as the whole data matrix, consisting of the capacity factors of all solutes at all design points used for training. In this way the information in the data matrix is utilized more completely, because the peak sequence in the different chromatograms, which is implicitly present in the data matrix, can contribute to the model.

Table 3
MPRESS values of non-linear models based on ln(k')

Model	Solute		
	1	2	3
NR	0.0083	0.0157	0.0038
NN7 500	0.0008	0.0130	0.0038
NN7 2500	0.0006	0.0121	

Models: NR=non-linear regression; NN7 500=neural network with 7 hidden nodes(optimal for all solutes) at 500 cycles; NN7 2500=neural network with 7 hidden nodes(optimal for all solutes) at 2500 cycles. Solutes: 1=2,3,4,-trihydrobenzophenone (number objects=36), 2=basic solute org5222 (number objects=23), 3=*o*-nitrophenol (number objects=40), (Tables 3, 4 and5 of Ref. [3]).

Non-linear response surfaces of the capacity factors of the solutes of data set C were estimated by non-linear regression using Eq. 3 and by a neural network. The network configuration was optimized by eliminating redundant hidden nodes. From each of the tables of Ref. [5], which contain the capacity factors of 10, 7 and 10 solutes respectively, one solute was selected. The network was trained with all data from each table, so that one network model is

obtained for the capacity factors of all solutes of a given table. Each output node of the trained network predicts the capacity factor of one solute.

The MPRESS values for non-linear models estimated by regression and by the neural network are presented in Table 3. For the first solute the network model outperforms the regression model, but in both other cases the performance of the regression model and the network are of the same order.

5. Conclusions

Neural networks offer promising possibilities in HPLC method development. Response surface for linear and non-linear changing capacity factors can be estimated with results better than or comparable to those obtained with linear and non-linear regression models. Optimization of the number of hidden nodes by lateral inhibition provides an objective way for configuring the network and reduces the risk of overfitting.

References

[1] J.L. Glajch and J.J. Kirkland, J. Chromatogr., 485 (1989) 51.
[2] P.M.J. Coenegracht, A.K. Smilde, H.J. Metting and D.A. Doornbos, J. Chromatogr., 485 (1989) 195.
[3] P.J. Schoenmakers, S van Molle, C.M.G. Hayes and L.G.M. Uunk, Anal. Chim. Acta, 250 (1991) 1.
[4] J. Gasteiger and J. Zupan, Angew. Chem. Int. Ed. Engl., 32, (1993) 503.
[5] J. Zupan and J. Gasteiger, Anal. Chim. Acta, 248 (1991) 1.
[6] Li Zheng, Cheng Zhaonian, Xu Li and Li Tonghua, Anal. Chem., 65 (1993) 393.
[7] M. Bos and H.T. Weber, Anal. Chim. Acta, 247 (1991) 97.
[8] B.J. Wythoff, S.P. Levine and S.A. Tomellini, Anal. Chem., 62 (1990) 2702.
[9] M. Bos, A. Bos and W.E. van der Linden, Anal. Chim. Acta., 233 (1990) 31.
[10] P.M.J. Coenegracht, H.J. Metting, E.M. van Loo, G.J. Snoeijer and D.A. Doornbos, J. Chromatogr., 631 (1993) 145.
[11] M. Volmer, B.G. Wolthers, H.J. Metting, T.H. de Haan, P.M.J. Coenegracht and W. van der Slik, Clin. Chem., 40 (1994) 1692.
[12] P.M.J. Coenegracht, A.K. Smilde, H. Benak, C.H.P. Bruins, H.J. Metting, H. de Vries and D.A. Doornbos, J. Chromatogr., 550 (1991) 397.
[13] S. Yasui, Proceedings 2nd International Conference on Fuzzy Logic and Neural Networks, Izuka, July 17–22, 1992, p. 775.
[14] L.W. Chan and F. Fallside, Comput. Speech Lang., 2 (1987) 205.