

Calibration of an array of voltammetric microelectrodes

R. Wehrens*, W.E. van der Linden

Department of Chemical Analysis, University of Twente, PO Box 217, 7500 AE Enschede, Netherlands

Received 8 January 1996; accepted 10 June 1996

Abstract

The calibration of a voltammetric sensor consisting of an array of individually modified electrodes is described. Linear calibration methods do not yield good results because of the inherent non-linear nature of the data. Neural networks can in principle model such dependencies, but their success is crucially dependent on the representation of the data. In this paper, neural networks and Principal Component Regression using several different data representations are compared. It is concluded that neural networks using unscaled first-derivative voltammograms yield the best results.

Keywords: Modified electrodes; Voltammetry; Sensors; Neural networks; Principal component regression; Calibration

1. Introduction

A voltammetric sensor consisting of an array of individually modified electrodes has significant advantages over sensors consisting of only one type of electrode. In more complicated matrices, such as samples consisting of more than one chemical species, the selectivity of the sensor may be insufficient to allow for a valid calibration: the jumps in the voltammogram may be too close to be discriminated individually. An array of modified electrodes may supply the complementary information necessary to be able to predict concentrations of multiple compounds in a sample. The modification may consist of a layer of another metal applied on top of the principal layer. Since electrochemical reaction

mechanisms are different for different metal surfaces, different voltammograms are expected revealing complementary information. Using such a sensor, each sample yields a matrix of data, the rows of the matrix being the applied voltages, and the columns the voltammograms for the individual electrodes.

In this paper, results are reported for the calibration of a sensor consisting of four different types of electrodes, coated with Au, Rh, Pt and Ir, respectively. The sensor is depicted in Fig. 1. Previously, results have been reported for a similar sensor using macroelectrodes [1]; here, however, microelectrodes are used. Apart from the fact that the small size makes it easy to use these sensors in practical applications, they also have a number of other advantages, most notably, a better signal-to noise ratio and a rapid attainment of steady-state currents. The last feature makes it possible to use them in flow-through cells [1]. On the other hand, their small size makes microelectrodes more sensitive to noise.

* Corresponding author. Present address: Department of Analytical Chemistry, Catholic University of Nijmegen, Toernooiveld 1, 6525 ED Nijmegen, Netherlands.

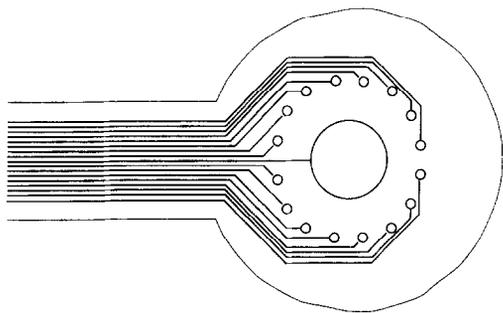


Fig. 1. Schematic drawing of the sensor.

Several problems may be encountered in calibrating such a sensor. First of all, the heights of the jumps in the voltammograms may not be proportional to the concentrations of the active compounds, as they would be in an ideal situation. Reaction products or reactants may adsorb to the electrode surface, thereby influencing other redox reactions. They may also react directly with each other or with other components in the sample. Furthermore, since only a counter electrode, not a real stable reference electrode, is incorporated in the sensor, the voltammograms may be shifted along the voltage axis.

2. Theory

The sensor data are calibrated using two methods: Principal Component Regression (PCR) and Artificial Neural Networks (ANN). A large amount of literature exists on both methods [2–4]; therefore, only a short summary will be given here. Matrices are denoted by uppercase letters, a prime indicates the transpose matrix.

2.1. Principal components regression

In ordinary least squares regression, the relation between predictor variables X and observed variables Y is modelled by

$$Y = XB + E$$

where B is the matrix of regression coefficients and E is the matrix of residuals, caused by noise, modelling errors etcetera. In simple cases, B can be found by

multiplying with the pseudo-inverse of X :

$$\hat{B} = (X'X)^{-1}X'Y$$

However, in many cases, this inverse does not exist because of collinearity of the data. One solution is to reduce the number of predictor variables, but this almost inevitably results in loss of information. Furthermore, it is often difficult to determine which variables to select.

A better solution is to compress the information in X into a few factors, linear combinations of the original predictor variables. In PCR, the factors are the eigenvectors of the square matrix $X'X$. Generally, only the first few of them are retained in the loading matrix \hat{P} . The scores of the samples on the new variables (\hat{T}) are easily calculated:

$$\hat{T} = X\hat{P}$$

Finally, the Y data are regressed on the scores:

$$Y = \hat{T}\hat{Q}'$$

where

$$\hat{Q} = (\hat{T}'\hat{T})^{-1}\hat{T}'Y$$

The regression coefficients for the original X variables can be calculated by

$$\hat{B} = \hat{P}\hat{Q}'$$

The number of principal components to be taken into account is quite crucial; one must take care not to choose too small a number, for then the dependencies between X and Y cannot be modelled, nor too big, since then noise is incorporated in the model. Usually, the correct number is determined using leave-one-out cross-validation, in which each sample in turn is excluded from the data set with which the model is built. The error is evaluated on the excluded samples. This is repeated for different numbers of principal components, and the number yielding the smallest error is taken to be optimal.

2.2. Artificial neural networks

Artificial neural networks are a non-parametric non-linear calibration method. Especially the so-

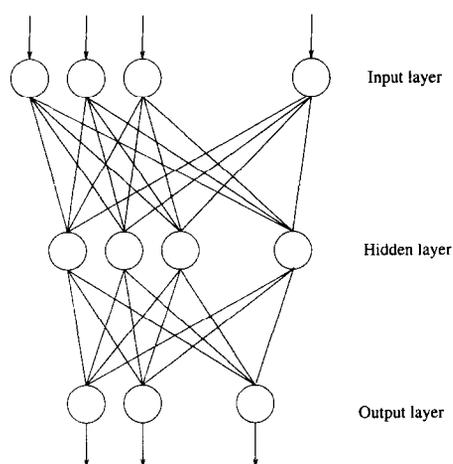


Fig. 2. Schematic representation of a neural network.

called feed-forward networks are often used. These consist of three or more layers of neurons: an input layer, each unit corresponding to a predictor variable in regression; one or more hidden layers; and an output layer corresponding to the dependent variables in regression (see Fig. 2). All units have at least one input and one output. Their output may consist of the sum of their inputs but usually a transfer function is applied to this sum. The non-linear modelling capabilities arise because of these transfer functions. In the hidden layer, sigmoid functions are often used; in the input and output layers, linear functions are used in quantification problems.

The output of a unit is sent with an attenuation factor (weight) to a unit in the next layer. These weights are randomly initialized before training. The model is built by repeatedly showing training instances (samples) to the network and adapting the weights so that the difference between the output units and the target values is minimized. Usually, the complete training set should be offered many times before a reasonable model is obtained. One pass of the randomly ordered instances in the training set is called an epoch. A vast number of different training algorithms exist [5]. The most well-known is called the back-propagation learning rule. In this work, another learning function, called resilient propagation (RPROP) [6,7] is used. This rule not only takes into account the value of the first derivative (i.e. the direction in which the weights should be adapted), but also the sign of the second derivative. This latter

parameter is used to determine the amount with which the weights are adapted. In preliminary experiments, this learning function gave slightly better results in far less epochs, as compared to back-propagation. In most cases, convergence was reached in less than 3000 epochs.

Whereas the number of units in the input and output layers is determined by the problem, the number of units in the hidden layer is variable, and may be optimized in much the same way as the number of principal components in PCR. Since training of a neural network starts from random weight values, there is no guarantee that the optimal network will be reached. Therefore, many training runs are many done. This cumbersome optimization of the network structure is one of the largest disadvantages of neural networks. Usually, the data set is split in two parts like the second validation method in the PCR calibration; one part for building the model and one independent part to validate the model. Care must be taken to prevent the network from merely remembering the training set examples. Therefore, the independent test set is continually evaluated during training. When the error in the test set starts to increase, this is evidence of overfitting, and further training is useless. If a lot of data are available, it is possible to split the data set in three parts, in which the second part is used to decide when to stop the training. The third part then is used to estimate the final prediction error with the optimal model. However, in many cases, the number of data is too small for this method. In this case, the number of epochs was fixed at a number at which preliminary experiments showed no overfitting, and the prediction error was estimated from the test set. The same training and test sets were used as in the PCR calibration.

3. Experimental

3.1. Chemicals

The samples consisted of mixtures of *ortho*-, *meta*- and *para*-dinitrobenzene, and monosubstituted nitrobenzene (Merck, pro Analyti). The solute was water containing $0.1000 \text{ mol l}^{-1}$ of both KNO_3 and ethanol (both Merck, pro Analyti). A full factorial design was

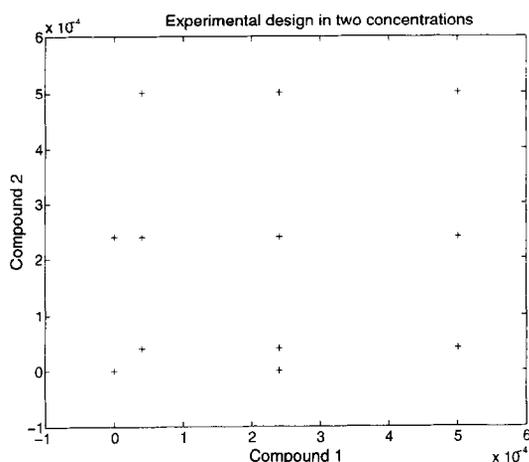


Fig. 3. Experimental design employed for any combination of two nitrobenzenes.

used, with concentrations of 4.02×10^{-5} , 2.41×10^{-4} and $5.01 \times 10^{-4} \text{ mol l}^{-1}$, yielding 81 samples. Additionally, 15 samples were measured in which one or more compounds were absent, and the others were at one of the above concentrations. A blank was also included. Plotting of concentrations of any two of the four compounds yields the picture of Fig. 3.

3.2. Equipment

The sensor consisted of 16 Ir microelectrodes; six of these were modified with an extra layer of another metal: 2 Au, 2 Rh, and 2 Pt. This sensor and its construction will be described in greater detail elsewhere. One electrode was selected for every type of top layer and thus four voltammograms, each consisting of 496 data points, were recorded in a single sweep.

3.3. Measurement procedure

The data set of 96 samples was measured by putting the sensor in a small beaker containing the sample, stirring, and measuring the voltammogram. Each sample was measured in duplicate. The scan rate was 20 mV s^{-1} , and the applied voltage varied between -1.0 and -2.2 V . The first measurement was discarded; the second was used to build or validate the calibration model.

3.4. Calibration

The data was divided into a training set (70 samples) and a test set (26 samples), which were used for both the neural network and the PCR calculations. The test set was balanced; each non-zero concentration level appeared eight times for each compound, as well as two samples in which the compound of interest was absent.

All neural networks used here had hidden units with sigmoid transfer functions; all other units had linear transfer functions. To each unit, a bias term was attached to stabilize calculations. All models were built using only the training set of 70 samples; the test set was used to assess performance and generalizing properties of the nets. For each combination of a representation and a neural-net-architecture, five batch training runs were done; the model with the smallest mean error was selected, except in cases where obvious outliers influenced the results. With the selected model, 10 training runs were performed to assess the predictive abilities of the network.

Errors of PCR and neural networks were evaluated as root mean square errors:

$$RMS = \sqrt{\frac{\sum_i d_i^2}{N}}$$

where N is the number of samples and d_i is the residual of the prediction. Data preprocessing and PCR were done in Matlab 4.0; neural network calculations were done using the Stuttgart Neural Network Simulator (SNNS) 4.0. Calculations were performed on a 486/66 MHz personal computer and on a Silicon Graphics R4400 SC. The learning and transfer functions used in the neural network calculations were the built-in functions of SNNS.

3.5. Data preprocessing

3.5.1. Principal component regression

Several preprocessing steps are needed before the calibration methods can be applied. First of all, a systematic noise, caused by the sampling procedure, was present in the voltammograms, that could easily be reduced to an acceptably low level by a moving average filter. Secondly, the format of the data should

be changed for PCR: each sample should yield a vector of data, not a matrix. This can be achieved by placing the voltammograms behind each other, a process called unfolding. In this case, for each of the different types of electrodes, one voltammogram was selected, yielding for each sample, a vector containing four voltammograms. To reduce computing time, only every fourth point of the voltammograms was retained. This yielded a data matrix of 96 samples times 246 data points.

Another possible representation is the first-derivative of the voltammograms (after smoothing), since the height of the jumps in the voltammogram is ideally proportional to the concentration of the active component. Because of the aforementioned complications, this relationship will not be exact, but the first-derivative representation does appeal to chemical common sense.

All PCR calculations were done for both scaled and unscaled input data. The scaled data were obtained by standardizing all measured currents at a specific voltage to zero mean and unit variance (autoscaling). Usually, this leads to a smaller number of significant factors [2], but the danger is that regions containing little information are artificially enhanced.

3.5.2. Neural networks

The main problem with representation for neural networks is data reduction. Since each input unit has a number of weights, associated with it, the training time becomes very large when a large number of input units are taken into account. Furthermore, because of the large number of degrees of freedom, the danger of overfitting increases. It is usually advisable to take the smallest possible network that achieves a certain predefined precision. This rule-of-thumb not only holds for the number of input units, but also for the number of hidden units in the network.

In this case, a small number of equidistant points (16 for each electrode) were taken from each voltammogram in order to reduce the number of connections in the network. Thus, 64 input units were used in this representation. The 15 differences between the points from one electrode were taken as the equivalent of the first-derivative representation in the PCR calculations, yielding 60 input units for

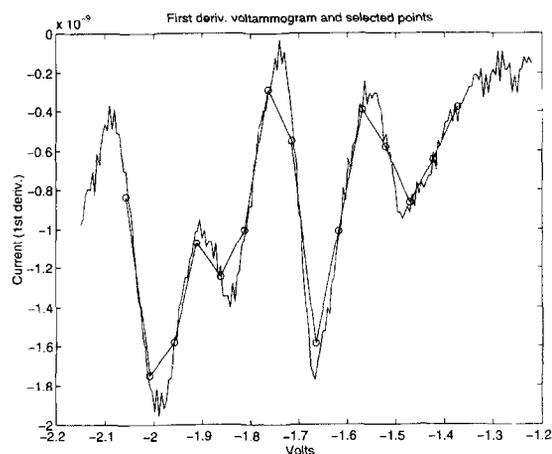


Fig. 4. First-derivative Au voltammogram, with all four components at $5.01 \times 10^{-4} \text{ mol l}^{-1}$. The 15 points describe the data well.

the neural networks. As can be seen in Fig. 4, the 15 points are able to describe the derivative voltammograms well.

For all neural network calibrations, the concentrations were scaled between 0.1 (zero concentration) and 0.9 (concentration 2500 units), since the outputs of the network could only vary between zero and one. Neural network input data were again scaled in two ways: autoscaling to zero mean and unit variance, as in the PCR calculations, and multiplication of the data with 10^7 so that the input values were mostly between -2 and 2 . This was done to avoid numerical instabilities in the weights, and this representation is essentially the same as the unscaled representation.

4. Results and discussion

4.1. Principal component regression

In Table 1, the results of a leave-one-out PCR cross validation using all 96 samples are given. The optimal numbers of PCs in the analysis are indicated in brackets; these numbers are also used in the analysis of the test and training sets separately, of which the results are given in Table 2. As a reference, the standard deviation around the mean concentration for all compounds was $1.886 \times 10^{-4} \text{ mol l}^{-1}$. In all cases, output values were scaled to unit variance and zero mean.

Table 1
Results of PCR cross-validation

Representation	Input scaling	<i>Ortho</i> -DNB	<i>Meta</i> -DNB	<i>Para</i> -DNB	NB
Voltammogram	Y	0.713 (9)	0.956 (7)	0.946 (9)	1.689 (8)
	N	0.713 (7)	0.960 (5)	1.212 (7)	1.661 (6)
First-derivative	Y	0.627 (11)	0.868 (5)	0.621 (10)	1.607 (12)
	N	0.697 (9)	0.998 (5)	0.840 (8)	1.705 (2)

Values given are root mean square values (in $10^{-4} \text{ mol l}^{-1}$).

Values in brackets are the optimal number of principal components.

Table 2
Final PCR results on the independent test and training sets (26 and 70 samples respectively)

	Input scaling	<i>Ortho</i> -DNB		<i>Meta</i> -DNB		<i>Para</i> -DNB		NB	
		Train	Test	Train	Test	Train	Test	Train	Test
Voltammogram	Y	0.544	0.693	0.842	1.019	0.775	0.768	1.514	1.726
	N	0.650	0.952	0.884	0.970	1.244	1.410	1.524	1.769
First-derivative	Y	0.460	0.673	0.745	0.949	0.444	0.497	1.488	1.701
	N	0.524	0.775	0.879	1.602	0.639	0.668	1.649	1.744

Values given are RMS values (in $10^{-4} \text{ mol l}^{-1}$).

The number of PCs used is given in Table 1.

It is clear that in the case of nitrobenzene, not even a qualitative prediction can be obtained; the prediction error is only marginally smaller than the variance in the concentrations, i.e. using the mean value as a predictor would give almost the same error. *Ortho*-dinitrobenzene yields the best results; *meta*- and *para*-dinitrobenzene are more or less equal. It is interesting to see the effect of scaling. In all cases, autoscaling of the input variables leads to a larger number of significant components, in contrast with what was expected. Prediction errors are more or less independent of scaling in the voltammogram representation; however, in the first-derivative representation, scaling seems to have a beneficial effect.

Because the leave-one-out cross-validation error in some cases is a poor approximation of the real error that could be expected with new samples, the data set was split in a training and a test set. These were the same as the ones used in the neural network calibration. The training sets were used to build the models using the optimal number of principal components found in the LOOM cross-validation, and the test sets were used to evaluate the prediction errors. These final PCR results are given in Table 2.

In almost all cases, the errors of the test set are larger than the errors of the training set, as expected. Again, the autoscaled first-derivative representation

yielded the best results. These are plotted in Fig. 5. In the cases of *ortho*- and *meta*-DNB, outlying predictions influence the RMS values of the training set considerably. However, as can be seen in the figure, the differences are not very big and the majority of the test set predictions is in the range of the training set error, indicating that the test set is adequately described by the training data.

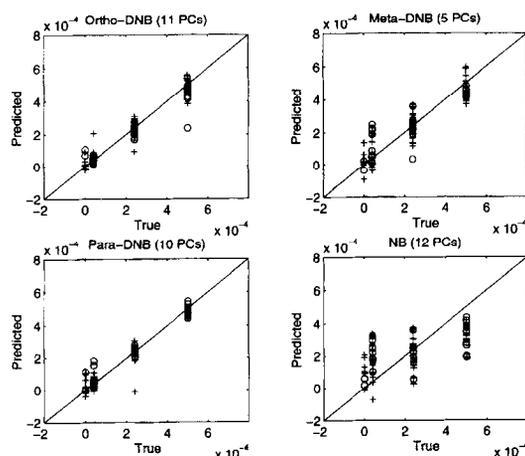


Fig. 5. True values versus predicted values for the four compounds. PCR, First-derivative representation (autoscaled inputs), optimal number of PCs. Training samples are indicated with plusses, test samples with circles.

4.2. Neural network calibration

Since a neural network is a non-linear calibration method, it may be expected that the results will be better than PCR when non-linearities play a role. Given the nature of the measurements, these non-linearities are certainly expected here. A further advantage of neural networks is their apparent ease of use; no complicated mathematics are necessary to understand the basic principles behind them. However, finding an optimal network configuration and good training parameters is a difficult and tedious task. Furthermore, results may be very sensitive to the representation of both input and output data.

The optimal configuration as determined in 5–10 batch runs of training of the networks is given in Table 3. Each time, only the number of units in the hidden layer is varied. For the first-derivative representations, this number was between 3 and 14; for the voltammogram representations between 8 and 17. All other parameters have been kept constant. Learning parameters have been optimized beforehand, and are given in the caption of the table. Again, we can see that the first-derivative representation gives the best results with a reasonably compact model.

In all cases, autoscaling had a very bad effect on the results. The errors were much larger than in the unscaled data, although the optimal number of hidden units was smaller. This is probably due to variables that do not contain information, e.g. at the right end of the voltammogram: the variance of the measured current is quite small at these voltages. Therefore, autoscaling will inflate the differences between samples, where no difference should be observed.

Table 3
Optimal configurations of networks for the various representations

Representation	Input scaled	Optimal number of hidden units	RMS test set
Voltammograms	Y	10	1.453
	N	10	0.727
First-derivative	Y	4	0.990
	N	11	0.577

RMS values are mean values of 5–10 runs calculated with the same test set as used in the PCR calculations (in $10^{-4} \text{ mol l}^{-1}$).

Parameters of the RPROP learning rule in all experiments: $\Delta_0=0.2$, $\Delta_{\max}=10$, $\alpha=5$.

Table 4

Average prediction errors of the optimal neural network (test set only)

Ortho-DNB	Meta-DNB	Para-DNB	Nitrobenzene
0.573	0.466	0.456	0.814

The number of training epochs is 2500; the RMS values from 10 training runs are averaged.

RMS values are in $10^{-4} \text{ mol l}^{-1}$.

Variable selection would probably have a beneficial effect on the calibration.

With the unscaled data, the effect of changing the number of hidden units was quite small; however, with the scaled data no clear trend could be observed. The first-derivative representation using unscaled data was therefore selected as the optimal model. Each time, the training was stopped after 2500 epochs. After that, no significant decrease in prediction error could be observed. The results of the training of 10 runs are given in Table 4 and plotted in Fig. 6. Error bars of one standard deviation are plotted around the mean of the 10 runs for each sample. As can be seen in the figure, in some cases, the individual models give rather different predictions. Only the data in the test set are plotted because the error in the training set was an order of magnitude smaller and not relevant to the performance of the model.

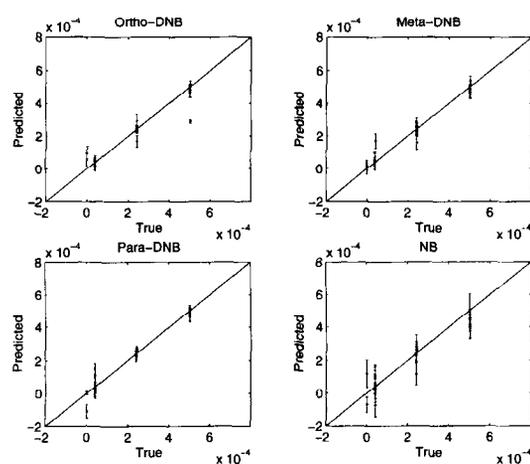


Fig. 6. True values versus predicted values for the four compounds (test set, 10 training runs). Error bars are drawn indicating plus or minus one standard deviation (in 10 predictions). ANN, First-derivative representation, 11 hidden units (unscaled inputs).

Interestingly, in some cases, similar errors can be observed as compared to the PCR results; most notably the prediction of the outlying *ortho*-DNB sample at $5 \times 01 \times 10^{-4} \text{ mol l}^{-1}$. The very small standard deviation indicates that all models exhibit the same behaviour for this sample. This may be caused by fouling of the electrode surfaces that induce shifts in the voltage scale. Especially, the predictions for nitrobenzene are better than with any other technique or representation and even comparable to the quality of the other predictions, although there is still a tendency to predict too low values for the highest concentrations. Using the optimal neural network model, the dinitrobenzenes are predicted best: the errors are comparable, taking into account the outlying observation in the *ortho*-DNB prediction.

Another feature that can be used to assess the applicability of data representations is the sensitivity to overfitting. Using the unscaled first-derivative data, no overfitting occurred; the error continued to decrease with the number of training epochs. However, in some of the other representations, overfitting occurred even after less than 1000 epochs. This is clearly an undesirable situation and an indication that the network is taking noise for information.

5. Conclusion

In this article, it is shown that it is possible to use a sensor consisting of individually modified microelectrodes to quantify multiple components in a sample. Although the precision of the predictions is not extremely good, it is probably sufficient for control and monitoring purposes. It should be kept in mind that the sensor design is not focused on a specific set of chemicals, but is generally applicable.

The data set presented here constitutes a difficult calibration problem: voltammograms of four different types of electrodes are used to quantify four chemically very similar compounds, with very similar reactions occurring at the electrodes. The non-linearities that arise because of adsorption processes on the electrode surface or reactions between products and/or reactants, clearly cannot be accounted for by a linear method like PCR. Neural

networks, on the other hand, are better suited to model these types of dependencies.

The representation of the data is of prime importance. In this work, a chemically appealing representation proved to give the best results. However, other representations lead to more compact models. Also, criteria other than the size of RMS values may be of importance, most notably, the robustness of the sensor. A representation that leads to a larger prediction error may still be preferable if it has much less need for recalibration. In the ideal situation, each sensor could be fitted with an identical calibration model, possibly executed in hardware. This, however, will probably not be achieved. An alternative is to make use of a basic model that only needs some “fine-tuning” now and again, e.g. some epochs of training on a reference data set. This line of research will be pursued further in our laboratory.

It is interesting to see if there are methods to make use of the so-called “second-order advantage” [8]: linear calibration of second order data (i.e. data in which one sample yields a response matrix instead of a response vector or scalar). These methods, in principle, can account for unknown interferents. However, they are only valid in linear cases. Recently, an approach applied to sensor technology has been published [9].

Acknowledgements

The authors would like to thank J. Simons for the construction of the sensor, J. Kalkman for measuring the data set and M. Bos for stimulating discussions.

References

- [1] J. Simons, M. Bos and W.E. van der Linden, Data processing for amperometric signals, *Analyst*, 120 (1995) 1009–1012.
- [2] H. Martens and T. Næs, *Multivariate Calibration*, Wiley, New York, 1989.
- [3] J.R.M. Smits, W.J. Melssen, L.M.C. Buydens and G. Kateman, Using Artificial Neural Networks for Solving Chemical Problems, I. Multilayer Feedforward Networks, *Chemometrics Intell. Lab. Sys.*, 22 (1994) 165–189.
- [4] P.J. Gemperline, J.R. Long and V.G. Gregoriou, Non-linear multivariate calibration using principal components regression and artificial neural networks, *Anal. Chem.*, 63 (1991) 2313–2323.

- [5] D.E. Rumelhart and J.L. McClelland, Parallel Distributed Processing, Explorations in the Microstructure of Cognition, Foundations, Vol. 1, MIT Press, London, 1986.
- [6] M. Riedmiller and H. Braun, A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. In: Proceedings of the IEEE International Conference on Neural Networks, 1993.
- [7] SNNS Manual version 4.0 (1994); SNNS Manual and Software Obtainable by Anonymous ftp from ftp.informatik.uni-stuttgart.de..
- [8] E. Sanches and B.R. Kowalski, Tensorial calibration: II. Second-order calibration, *J. Chemometrics*, 2 (1988) 247–263.
- [9] K.S. Booksh, Z. Lin, Z. Wang and B.R. Kowalski, Extension of trilinear decomposition method with an application to the flow probe sensor, *Anal. Chem.*, 66 (1994) 2561–2569.