Optimal Minimal Neural Interpretation of Spectra

Claus Borggaard* and Hans Henrik Thodberg

The Danish Meat Research Institute, Maglegaardsvej 2, DK-4000 Roskilde, Denmark

A new method "optimal minimal neural-network interpretation of spectra" (OMNIS) based on principal component analysis and artificial neural networks is presented. OMNIS is useful whenever spectra are measured for the purpose of classification or quantitative determination. The spectra can be visible light, near-infrared (NIR) light, sound, or any other large amount of correlated data. OMNIS is unique in several respects: It employs principal component analysis as a preprocessor to a neural network. The neural network contains direct connections from input to ouput ensuring that OMNIS is a true generalization of PCR (principal component regression). The neural network size is optimized so that the resulting solution contains the minimum of connections necessary to interpret the data. Cross validation is used systematically to optimize the network. OMNIS is based on recent insights in neural network research showing that deliberate search for the minimum network compatible with the data is a unique way of obtaining the optimal generalization ability. As a result OMNIS gives the best cross validation. In comparison, with PCR and PLS (partial least squares) on two NIR calibration data sets, OMNIS is demonstrated to reduce the standard error of prediction by 50% to 75%.

I. INTRODUCTION

With modern sensor and computer technology, spectral data are collected with ever finer sampling, with ever larger precision, and in ever larger amounts. This leads to a need for efficient methods of interpreting the spectral data for the purpose of classifying samples or calibrating an instrument for quantitative determination.

The task of interpreting the spectra is conducted in the following standard framework. We have a training set consisting of a number of training examples. Each example is a spectrum (the input) and the associated interpretation (the output). We may also have a test set of additional examples. The goal of the modeling procedure is to construct an algorithm using only the information in the training set. The test set is used only to evaluate the final algorithm.

In the following we describe the components of OMNIS: PCA in section II and neural networks in section III. In section IV the components are integrated into OMNIS and the use of cross validation and the training procedure are explained. Section V covers experiments; section VI contains results and discussion. In section VII a method for finding the minimal network using pruning is applied. Finally, section VIII gives the outlook and conclusion.

II. CHEMOMETRICS RESOLVING THE CURSE OF DIMENSIONALITY

Multivariate Linear Regression. A spectrum is a sampling of a continuous function at a set of fixed wavelengths or energies. It is desirable to use a fine sampling in order not to lose information on the detailed structure of the spectrum. However, the fine sampling will give a large correlation between adjacent points in the spectrum. A multivariate linear regression (MLR) on the input of the training set often fails because the number of input variables is so large that there is not enough information in the outputs of the training set to determine each of the coefficients. This problem can be fixed by coarser sampling of the spectrum or selection of the most significant samplings of the spectrum, but both of these fixes are likely to result in a loss of useful information. This problem is called the curse of dimensionality, and it is resolved by chemometrics, a well-established standard technique embracing the methods of principle component regression (PCR) and partial least squares (PLS).

Principal Component Regression. PCR solves the problem by preprocessing the input data without using the output.[1] The preprocessing called principal component analysis (PCA) goes like this: The mean values of the inputs are subtracted, and one constructs the directions in the input space which maximizes the variance of the projected inputs. This is the first principal component, and the projection of a spectrum onto the first component is called the first score. This projection is next subtracted from the data and the search is iterated to find the second principal component, etc. The principal components describe the modes, or correlated variation, in the inputs. In PCR one performs a linear regression on the first few scores to predict the output.

Supervised and Unsupervised Learning. MLR is an example of supervised learning: The coefficients are found using the output as a teacher. PCA is an example of unsupervised learning. The principal components are deduced solely on the basis of the internal structure of the input. PCA does not deteriorate with increasing dimension—only supervised learning is haunted by the curse of dimensionality. PLS is similar to PCR. It also extracts some components, but these are found by a mixture of supervised and unsupervised learning according to an algorithm (see ref 1, pp 119–125). It is in fact possible to define a continuum of chemometric methods ranging from PCR over PLS to MLR, with increasing character of supervision in the preprocessing.[2] PLS is often preferred to PCR in applications. This is mainly because PLS is faster than PCA, which is an iterative process. PLS does not in general perform better than PCR.[3] We deliberately want to minimize the number of parameters trained by supervision, so we use the PCA components rather than the PLS components for preprocessing in OMNIS. In addition, the PLS components have a tendency to conserve information linearly correlated with the output and thus it can miss some useful nonlinear information.

III. NEURAL NETWORKS AND NONLINEAR INTERPRETATIONS

Nonlinearities. In many actual and potential applications of spectral measurements the desired interpretation is not a linear function of the spectrum. This means that the standard linear PLS and PCR algorithms are suboptimal or even useless methods. With the growing appreciation of nonlinear phenomena in general and the availability of powerful computers, this state of affairs has become unacceptable. Neural networks are the most promising candidate for tractable interpretation and modeling of a large class of nonlinear problems.

Neural Networks. We consider layered feedforward neural networks trained with the back-propagation learning algorithm.[4-6] These can be considered a generalization of MLR to arbitrary (smooth) nonlinear transformations. The network

0003-2700/92/0364-0545$03.00/0

consists of three layers of processing units (architectures are shown schematically in section IV):

a. a layer of input units which merely serves as distributors of the inputs to the network.

b. a layer of hidden units, which performs a weighted sum of the inputs followed by a nonlinear sigmoid transformation. The ith hidden unit has the output s_i:

$$s_i = \sigma(\sum_j (w_{ij}s_j - b_i))$$

The sum extends over the input units, and s_j is the output from input unit j. w_{ij} is the weight of the connection from input unit j to the hidden unit i, and b_i is the bias to the hidden unit. σ is the sigmoid function:

$$\sigma(y) = 1/(1 - e^{-y}) - 0.5$$

The connection weights and the biases are collectively referred to as the *weights*.

c. a layer of output units—often just one—which again performs a weighted sum of the inputs. The output units are receiving inputs from the hidden units *as well as directly from the input units*. For classification purposes the output units are performing a sigmoid transformation, while for quantitative determinations the output unit is linear.

The weights are the adjustable parameters. The back-propagation training algorithm compares the actual and desired output of the network and adjusts the weights of the network in order to reduce the error. Numerically, this takes place by performing a gradient descent in the cost function

$$\text{cost} = \tfrac{1}{2}\sum_i (s_i - t_i)^2$$

where the sum extends over the output units s_i, and t_i is the target value for unit i.

Direct Connections. With the direct connections from input to output and without the sigmoid transformation in the output layer, the network becomes a true generalization of MLR: If the connections from the hidden units are set to zero, the network reduces to MLR. In some cases only hidden units are needed, in others only direct connections, and in others both. When both types of connections are allowed in the same network, the task of deciding what is needed is left to the training algorithm. Hence the model is more adapative and flexible.

In ref 7 an alternative way of implementing direct connections is adopted: A linear hidden unit is used. In this model two subsequent linear transformations are necessary in order to give the same net effect as the direct connections. The linear hidden unit approach does not give more flexibility than the direct connections, but it adds two extra weights, namely the bias to the hidden linear unit and the weight of the connection from the hidden linear unit. Hence hidden linear units are less minimal than direct connections and should therefore be disfavored (see Ockham's razor in section IV). Another advantage of direct connections is that some workers have found that they give a speed-up in learing time.[8]

IV. OPTIMAL MINIMAL NEURAL INTERPRETATION OF SPECTRA (OMNIS)

An attempt to apply a neural network directly to the spectra falls victim to the curse of dimensionality. This section gives a better method.

Ockham's Razor. The curse of dimensionality is in fact part of a more general torment, the curse of complexity. This was realized as early as the 14th century by the famous William of Ockham. For this curse he frequently used the principle of economy in explanations, known as Ockham's razor. He used it under such forms as "*What can be done with less is done in vain with more*" and "*Plurality is not to be assumed without necessity*".

Minimal Neural Networks. Ockham's razor applies to supervised learning in chemometrics and neural networks. Recent research in neural networks can be summarized in the following rule:[9] *The neural networks with optimal generalization ability are characterized by having the fewest weights, while still processing the training data correctly.*

The generalization ability is the performance on a test set *not used in the development of the network*. The test set is assumed to be drawn from the same population as the training set. In accordance with this, OMNIS is designed to contain the optimal, minimal number of weights. This principle enters in two ways. First, OMNIS uses PCA to preprocess the spectra. This reduces the dimension of the input vector dramatically, leading to an equally dramatic reduction in the number of weights emerging from the first layer of the neural network. (Using PCA scores as input for a neural network is closely related to techniques that have been used for a number of years in image processing;[10] it has also been used for spectra under the name PC-ANN.[7] Second, OMNIS employs a systematic prescription for finding the optimal, minimal neural network, using cross-validation.

Transformation of Data. Before the spectra are passed through the principal component analysis, it is useful to transform them. For reflectance spectra one usually takes the logarithm in order to linearize the data. Sound spectra are often falling several orders of magnitude from one end of the spectrum to the other and should therefore be scaled to uniform variance or uniform noise level. These transformations influence the principal components, and the aim is to embrace the maximum amount of useful information in a small number of principal component scores. *One should not work too hard to linearize the data.* Neural networks can do this efficiently.

Before the scores are fed to the networks, they are scaled to unit variance or by maximum–minimum scaling.

Systematic Use of Cross-Validation. The training set is subdivided into two sets: C, the calibration set, and M, the monitoring set.

The division is done by sorting the examples by ascending output values and selecting every nth example for M. Useful values of n are 3–10. There are n different ways to do this division, which can be used cyclically in the exploration below.

OMNIS performs some explorations using C for training and M for cross-validation. One varies the initial transformation of the data, the number of input units (i.e. the number of scores), and the number of hidden units until a sufficiently optimal model is found, i.e. until the error on M is satisfactory.

Training. The network is initialized with random weights and trained. For every traversal of C the error on M is evaluated. This quantity decreases rapidly in the beginning. The training is stopped when the error on M starts to increase. Ten training sessions are performed—of course with different initial random weights. Finally, one selects the network with the smallest error on M.

The training is stopped before the network reaches the minimum error on C, because this is an efficient way to avoid overfitting. Overfitting can occur in spite of using the optimal, minimal number of input and hidden units, because the network is fully connected. Often a sparsely connected network can be found with equally good performance on C, and hence better generalization. In the beginning of the training, several weights are still untuned. Later on, they specialize to fit the details of the data.[11] By stopping the training before it reaches equilibrium, one prevents the network from exploiting the full potential of all its weights.

It may be argued that using part of the training data as monitoring data is a waste of training data. However, the

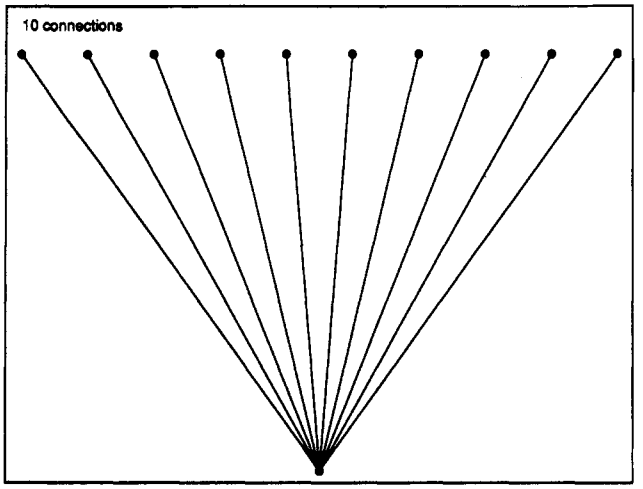

Figure 1. PCR (only direct connections). Input units are on the top, and the output below.

problem of overfitting is so severe in neural networks that the benefit of controlling it outweighs the loss of training data. Furthermore, M enters the model construction indirectly since it determines when to stop the training, and it selects among the 10 different models. Hence the error on M is in general smaller than the error on an independent test set. In section VII an alternative technique to remedy the overfitting problem (involving network pruning) is discussed and applied.

Evaluation of Performance. Once developed, the models can be tested. It is interesting to test for both generalization and robustness. The generalization is the performance on examples drawn from the same population as the training data, while the robustness is the performance on examples drawn from a somewhat different but related population. The latter is pertinent for real-life applications, since things often are not as they were in the calibration situation.

We use the following data sets to evaluate the network: T, a test set drawn from the same population as C and M, and E1 and E2, extrapolation sets drawn from two populations that are neighbors to C + M.

T tests interpolation. E1 tests extrapolations in the variable to be determined (e.g. fat), and E2 tests extrapolation in another variable hampering the determination (e.g. water).

It is important to use the error on T and not on M to evaluate the model. This strict principle should always be used when neural networks are evaluated.

In contrast to linear regression the neural network models are not unique. Starting the network with different random weights gives a different model and a different error on C, M, T, E1, and E2. Therefore, values quoted below for standard error of calibration (SEC) and standard error of prediction (SEP) contain uncertainties. These are estimated by performing several batches of 10 trained networks and studying the dispersions.

Architectures. The network architectures of Figures 1–4 were trained on C using M for monitoring. Then they were tested on T, E1, and E2.

A PLS model was constructed on the combined data sets C and M. This model was then asked to predict on the same three data sets, thus making it possible to compare PLS with PCR, OMNIS1, OMNIS2, and OMNIS3.

V. EXPERIMENTAL SECTION

Two sets of experimental data were used. One set is constructed on substances where we understand the nature of the nonlinearities. The other set is from a real-life calibration experiment on a meat/fat product.

Demonstration Experiment. The experiment was performed using a NIR spectrometer from LT Industries. The spectrometer

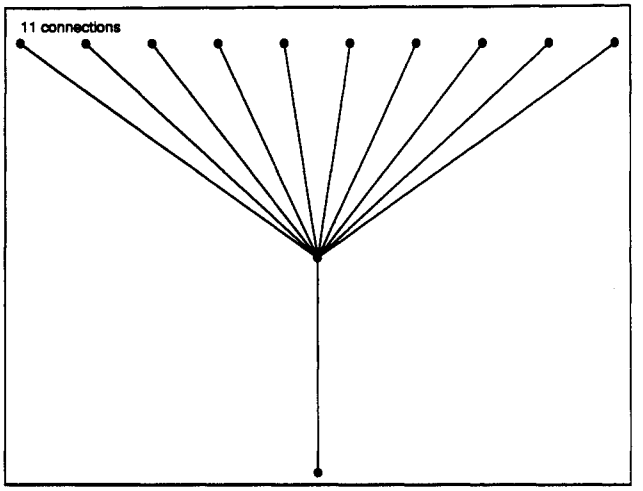

Figure 2. OMNIS1 (one hidden neuron, no direct connections).

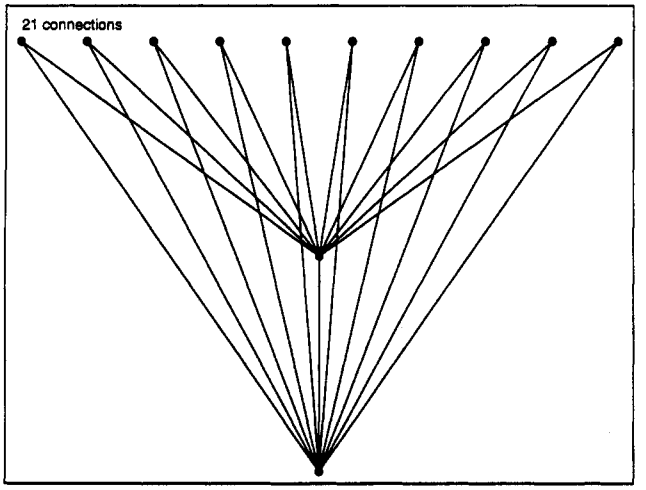

Figure 3. OMNIS2 (one hidden neuron + direct connections, PCR + OMNIS1).

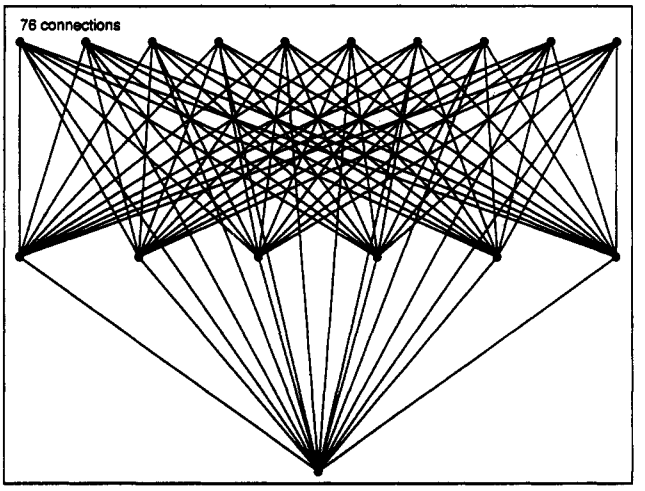

Figure 4. OMNIS3 (six hidden neurons + direct connections, PCR + neural net).

is equipped with an insertion probe that can be inserted into a lump of meat or immersed in a liquid. Light is emitted and detected through fiber optics. The spectrometer records a reflectance spectrum sampled at 1201 wavelengths from 900 to 1800 nm.

Ethanol and latex—plastic spheres of diameter 1060 nm—are mixed/suspended in water to give 99 different samples. The concentration domain covered by these 99 samples is shown in Figure 5. NIR reflection spectra were recorded from each of the 99 samples.

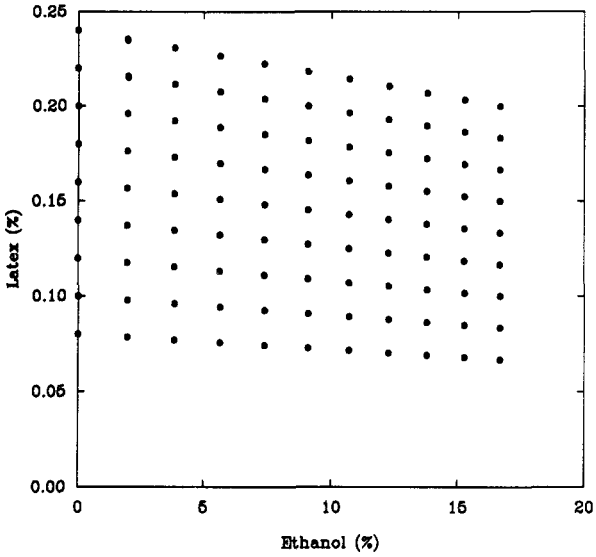

Figure 5. Latex versus ethanol concentration for the 99 samples studied in the demonstration experiment.

The interpretation task is to deduce the ethanol content from the NIR reflectance spectrum.

The absorption depends on the traveled distance of the reflected light, which in turn depends on the latex concentration. Hence the absorption of light from the ethanol depends both on the ethanol concentration and on the latex concentration.

$$\text{absorption} \approx \text{length} \times \text{concentration}$$

This is a nonlinear relation. A very understandable listing of the sources of nonlinear response are given by Gemperline et al.[7]

Real-Life Experiment. The experimental data consist of NIR spectra of 240 samples of ground pork meat with corresponding wet-chemistry results on fat, water, and protein contents. In this study we have restricted ourselves to making calibrations for determining the fat content.

The NIR spectra were recorded on a Tecator Infratec spectrometer that measures the absorbances at 100 wavelengths in the region 850–1050 nm. A Soxhlet method was used as the laboratory reference for the fat determination. Soxhlet values ranged from 2% to 59% fat. The data set was then split according to the scheme in section IV: The 8 objects with fat contents larger than 50% were put into the extrapolation test set E1. Spectra of the 17 samples with the highest water contents were put into the extrapolation test set E2. The remaining 215 samples were then split in a calibration set C of 129 objects, a monitoring set M of 43 objects, and a interpolation test set T of 43 objects.

The first 12 principal components (eigenvectors) were calculated from C, and all spectra in the 4 data sets were projected onto these principal components, thus reducing the size of each spectrum from 100 wavelengths to 12 scores. The scores of C are then used to create PCR calibrations and are used as input to the neural networks for determining the fat contents of the samples from the spectra.

Training Parameters for the Networks. The program is written in Pascal and runs on a IBM compatible PC. When networks are trained with this program, the learning rate for the back-propagation algorithm can be adjusted interactively by the user. In these studies learning rates ranged from 0.01 to 0.1. We scale the laboratory reference values in the same way as the inputs to the network are scaled (without this scaling the learning rate would have to be very different; see section III—transformation of data). The weights are updated after every 10 examples. No momentum was used. Before training commences, all weights are initially set at values from −0.5 to +0.5. The transfer function used at the hidden units is given in section III.

VI. RESULTS AND DISCUSSION

Demonstration Experiment. In order to illustrate the nonlinear character of the ethanol–latex data a PCA model was constructed on the whole data set composed of 99 spectra. In Figure 6 are plotted the scores for the first two principal components in this model. The data set was divided into C (50 samples) and M (49 samples). Nine calibrations were performed on the data from the demonstration experiment. PLS and PCR models using 2, 4, and 6 scores, based on C, were asked to predict the ethanol on M. The residuals of the PLS predictions (6-scores model) versus the predicted values on the test set are shown in Figure 7.

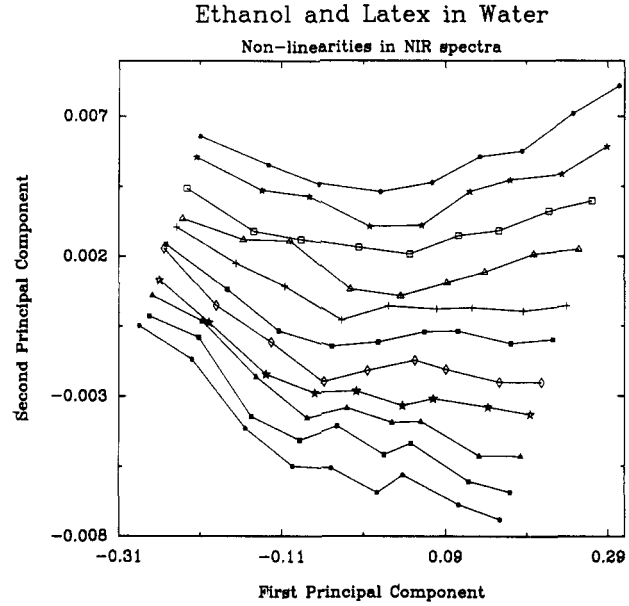

Figure 6. Scores plot for the first two principal components on the spectra from the demonstration experiment. The lines in this figure connect samples with equal ethanol concentration but with different latex concentrations. These lines clearly show that the data set is nonlinear

Table I. Standard Error of Prediction on Monitoring Set[a]

model	no. of scores used in model		
	2	4	6
PLS	1.18	0.78	0.71
PCR	1.61	0.80	0.71
OMNIS3	0.60	0.41	0.37

[a] Predicting ethanol in latex suspension.

Table II. Predicting Fat in Homogenized Meat Products

	PLS	PCR	OMNIS1	OMNIS2	OMNIS3
SEC (C)	2.83	2.76	1.68	0.82	0.47
SEP monitoring (M)	2.83	2.82	1.92	1.07	0.73
SEP interpolation (T)	2.86	2.92	1.85	0.95	0.65
SEP extrapolation (E1)	10.88	9.35	8.21	5.61	2.33
SEP extrapolation (E2)	2.93	3.21	2.40	1.00	0.59

Neural networks were trained on the same calibration set using the first 2, 4, and 6 principal components as input. In all cases 3 hidden neurons were used. In Table I the standard error of prediction for the 9 different calibrations are summarized.

When using PLS and PCR, cross-validation pointed at a 6-factor model as the optimal solution. We see that the neural network model performs better with only 2 factors than PLS and PCR do with 6 factors. This is clearly an advantage because models that use few principal components are likely to be more robust than models that use more, since the higher numbered components are more susceptible to noise.

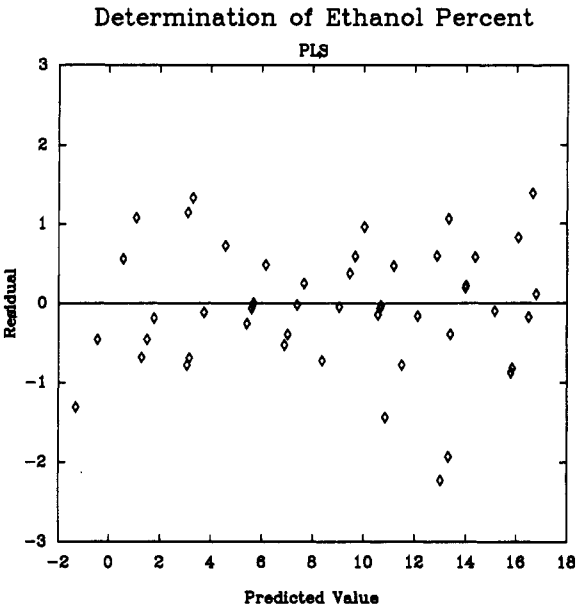

Figure 7. PLS predictions on the monitoring data set M in the demonstration experiment. The residuals (predicted minus laboratory result) are plotted versus the predicted results.

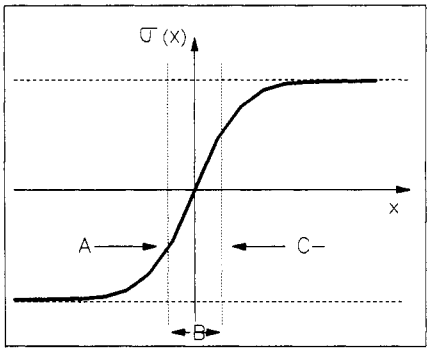

Figure 8. Data are passed through a sigmoid function at each hidden neuron.

Table III. Architecture of Models Used[a]

architecture (model)	direct connections	weights to be fitted
PCR is 10–0–1	yes	11
OMNIS1 is 10–1–1	no	12
OMNIS2 is 10–1–1	yes	22
OMNIS3 is 10–6–1	yes	83

[a] Weights are the number of connections plus the number of biases.

Real-Life Experiment. Results obtained with PLS, PCR, and OMNIS on the meat data are shown in Table II.

As mentioned in section IV, neural networks have a statistical nature. The OMNIS3 results in Table II are average values obtained from several training sessions. The standard deviation around these averages is approximately 0.05. The network complexity increases from PCR through the OMNIS models 1, 2, and 3. Table III summarizes the network architectures used. In this table the architecture 10–6–1 means that there are 10 input neurons, 6 neurons in the hidden layer, and 1 output neuron.

At all hidden neurons, data are passed through a sigmoid function as shown in Figure 8. Here we have subdivided the function into three separate domains: region A with positive second derivative; region B which is almost linear; region C with negative second derivative. OMNIS1 attempts to solve the problem often seen in chemometrics where the residual-versus-predicted plot shows a banana-like swarm of points. The hidden neuron in OMNIS1 gives a postprocessing of the response that can model four different shapes shown in Figure 9.

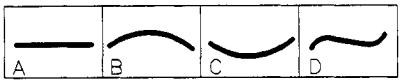

Figure 9. Shapes of a residual plot that can be described by only one hidden neuron with a sigmoid response (OMNIS1).

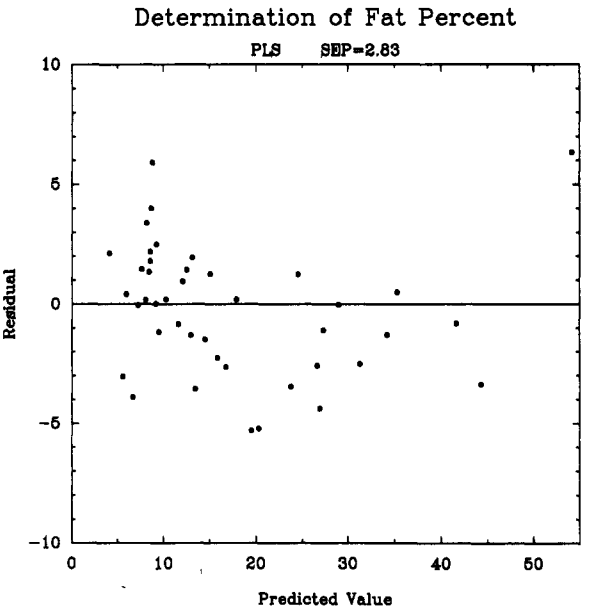

Figure 10. PLS prediction (residual) for predicting fat in ground meat products on the test set T. Notice the banana-like shape, which shows that the data set is nonlinear.

The OMNIS1 modelings of the four shapes in Figure 9 are performed by adapting one or more of the three regions of the sigmoid function in Figure 8. If the data are completely linear, as in the case of Figure 9A, the trained OMNIS1 network will only use the near-linear part of the sigmoid denoted B in Figure 8. Data that displays residuals as in Figure 9B,C will be modeled by OMNIS1 by using only regions A and C, respectively. When the shape of the residual plot is as in Figure 9D, OMNIS1 will use most of the dynamic range of the sigmoid function in order to correct for the nonlinearities. The validity of these assumptions has been checked by creating artificial data sets that give residual plots as shown in Figure 9 and histograming the inputs to the sigmoid function on the trained OMNIS1 network.

In Figure 10 the differences between PLS-predictions and the laboratory values on the test set T are shown versus the predicted values. It should be possible to cure the bending of this curve by using OMNIS1. In Figure 11 OMNIS1 has been applied to the test set. We see that the curve in Figure 10 is somewhat straightened as expected. The remaining residual after applying OMNIS1 does not fit into any of the four categories in Figure 9, thus demonstrating the limitations of OMNIS1.

OMNIS2 only uses one hidden neuron but includes the PCR solution. OMNIS1 and OMNIS2 are attractive because they have very few parameters and because they make a global optimization of all the weights. This means that every new training of these networks with new initial weights gives exactly the same performance.

The residuals from the predictions of OMNIS3 are shown in Figure 12. This neural network gives an impressive fit to the laboratory data.

OMNIS3 reduces SEP by 77% on the interpolation test set T relative to the PLS model. On the extrapolation sets E1

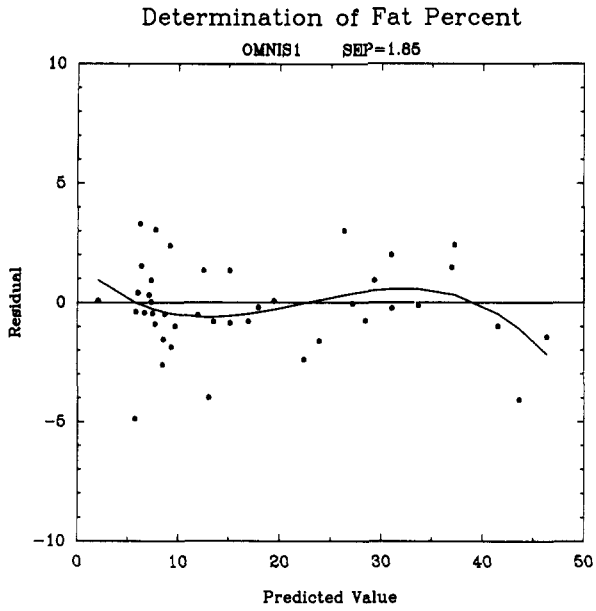

Figure 11. OMNIS1 applied to the same data set as in Figure 10. This simple network has improved performance by 35%. The cubic spline has a shape which cannot be modeled by OMNIS1 (see Figure 9).

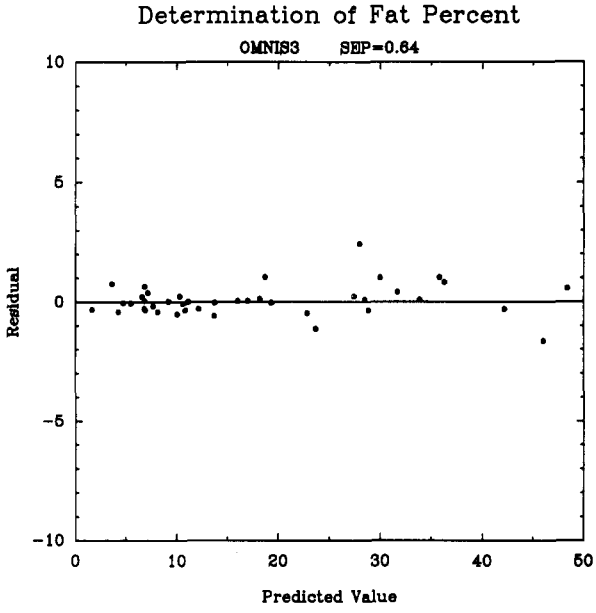

Figure 12. OMNIS3 predictions of the fat contents on the test set T (ground meat products). The standard error has been reduced by 77% compared to the PLS model in Figure 9.

and E2 the reduction is even larger, 79% (see Table II). This is to be expected, since the network learns the nonlinearities, which show up more dramatic in E1 and E2. Neural networks can extrapolate a nonlinear dependency.

If data sets E1 and E2 are added to the calibration and monitoring sets C + M, PLS gets worse on the test set T while OMNIS3 improves slightly. The results of this study are shown in Table IV as standard error of prediction on the test set T. Hence the neural network can transfer knowledge from E1 and E2 to the intermediate range.

VII. SEARCH FOR THE MINIMAL NEURAL NETWORKS

Ockham's razor is the name of a thorough pruning technique developed by one of us.[12] It removes connections tentatively one at a time and retrains the network. If the network error on the training set is no worse than before, apart from a small allowable error increase, the connection is considered

Table IV. Predictions on Test Set T[a]

	training set	
model	C + M	C + M + E1 + E2
PLS	2.83	3.07
OMNIS3	0.65	0.63

[a] The performance of the neural network model improves when the range of the training set is increased. PLS gets worse.

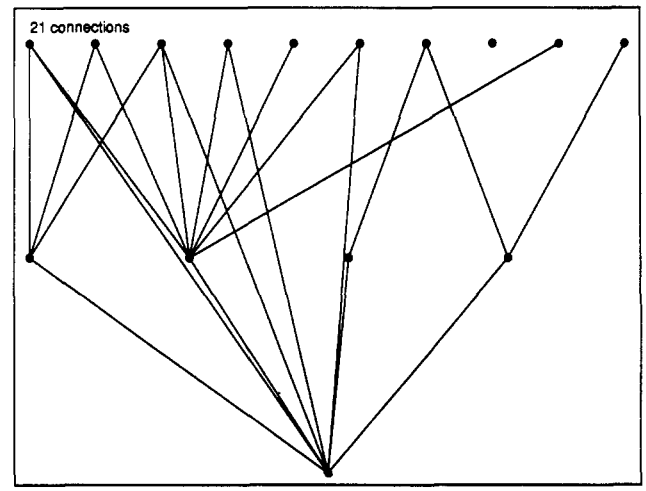

Figure 13. Result of pruning the network in Figure 4. This network puts more emphasis on the first scores.

pruned; otherwise the network with the connection is reinstalled. This process continues until the error on the training set starts exceeding the acceptable level. The advantage of this method is that it employs both C and M for training.

Finding a minimal network to describe the training data can be considered a search. To include the power of randomness in the search, several networks with different initial weights are trained and pruned. Finally the smallest networks are chosen as candidates for the best solution. This technique has been applied to the OMNIS3 network (10–6–1 architecture) described in section V. The pruning decreased the number of connections and biases to be estimated from 83 to only 26, see Figure 13.

The ability of this much simpler network to generalize is on the same level as the much larger OMNIS3. This suggests that OMNIS3 is already optimal.

The pruning technique solves the search for the optimal network using the following elements with a reminiscence of biological dynamics: randomness in the initial configuration of the networks, evolution through pruning, and selection of the smallest.

Applying Ockham's razor is a very time-consuming process. In the case of spectroscopic data it does not improve the results significantly compared to the methods described previously.

VIII. CONCLUSION AND OUTLOOK

It has been demonstrated that neural networks can be used for quantitative determinations in NIR spectroscopy. On data with a nonlinear relationship between spectral response and analyte concentration, a neural network trained on principal component scores produces calibrations that are substantially better than those obtained with traditional chemometric methods.

Neural networks are able to make calibrations on NIR data with fewer scores as input than PLS and PCR. This means that noisy spectra are unlikely to cause as much trouble when neural networks are used as when PLS and PCR are used, because spectral noise always shows up in the higher principal components. PLS uses the higher scores to compensate for

nonlinearities in the first few of these. The following questions/objections are often put forward when attempting to use neural networks for quantitative analysis:[13,14]

1. Neural networks are difficult to design and will not generalize satisfactorily.

2. Can neural networks extrapolate? Will neural networks predict better on data outside the calibration set range than PLS and PCR?

3. Neural networks are prohibitively slow to train.

The first of the above points has certainly proven wrong in these two data sets (and in other cases we have tried). Measured by standard error of prediction the neural network performs a factor 4 better than PLS and PCA in a real-world application on both interpolation and extrapolation data sets. We have demonstrated that neural networks are able to predict well on new objects that fall outside the calibration set range. In the "real-life experiment" OMNIS3 out-performed both PLS and PCR completely.

As for the third objection, calibrations OMNIS1, OMNIS2, and OMNIS3 were produced on a 20-MHz PC using Turbo Pascal software. Training time varied from 10 to 30 min corresponding to 300–1000 traversals of the calibration set. This included the initial calculation of the PCA eigenvectors on the training set. The main reason for these very short training times is that we have used direct connections from input to output. Most nonlinear effects in NIR spectroscopy can be thought of as relatively small perturbations on a linear solution. Therefore it is only the direct connections that need adjusting to get the largest part of the calibration in place. Also the networks we have used are very small. This has been possible because we have used PCA to reduce the number of inputs.

Earlier attempts to use artificial neural networks for calibrations[13] have not been quite so optimistic about the future of this technique. However, the use of direct connections in neural networks and principal component scores as input ensures that the performance (generalization) is never worse than that obtained with PCR, because PCR is part of the model.

REFERENCES

(1) Martens, Harald; Naes, Tormod. *Multivariate Calibration*; John Wiley & Sones: New York, 1989; pp 97–111.
(2) Stone, M.; Brooks, R. *J. J. R. Statist. Soc. B* **1990**, *52* (No. 2), 237–269.
(3) Thomas, Edward V.; Haaland, David M. *Anal. Chem.* **1990**, *62*, 1091–1099.
(4) Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. In *Parallel Distributed Processing*; Rumelhart, D. E., McClelland, J. L., Eds.; MIT Press: Cambridge, MA, 1986; Vol. 1, pp 318–362.
(5) Lippmann, Richard P. *IEEE ASSP Mag.* **1987**, April, 4–22.
(6) Jansson, Peter A. *Anal. Chem.* **1991**, *63*, 357A–362A.
(7) Gemperline, Paul J.; Long, James R.; Gregoriou, Vasilis G. *Anal. Chem.* **1991**, *63*, 2313–2323.
(8) Lang, Kevin J.; Witbrock, Michael J. *Connectionist Models*, Proceedings of the 1988 Summer School; Morgan Kaufmann Publishers, Inc.: San Mateo, CA, 1989; pp 52–59.
(9) Baum, E. D.; Haussler, D. *Neural Computation 1*; MIT Press: Cambridge, MA, 1989; 151–160.
(10) Cottrell, Garrison W. *Connectionist Models*, Proceedings of the 1990 Summer School; Morgan Kaufmann Publishers, Inc.: San Mateo, CA, 1991; pp 328–337.
(11) Weigend, A.; Huberman, B.; Rumelhart, D. *Int. J. Neural Syst.* **1990**, *1* (No. 3), 193–209.
(12) Thodberg, Hans H. *Int. J. Neural Syst.* **1991**, *1* (No. 4), 317–326.
(13) Long, J. R.; Vasilis, G. G.; Gemperline, P. J. *Anal. Chem.* **1990**, *62*, 1791–1797.
(14) Zupan, J.; Gasteiger, J. *Anal. Chim. Acta* **1991**, *248*, 1–30.

RECEIVED for review August 23, 1991. Accepted November 27, 1991.

Water-in-Oil Microemulsions as Solvents for Laser-Excited Multiphoton Photoionization

Mitchell E. Johnson and Edward Voigtman*

Department of Chemistry, University of Massachusetts at Amherst, Amherst, Massachusetts 01003-0035

Laser-excited multiphoton photoionization spectroscopy in polar liquid solutions is impeded by relatively high noise levels. Detection limits are 2–3 orders of magnitude worse than in apolar solvents. Water-in-oil microemulsions of water/AOT [sodium bis(2-ethylhexyl) sulfosuccinate]/*n*-heptane were evaulated as alternative solvents for polar analytes. Dispersion of water/surfactant aggregates in heptane added excess low-frequency noise relative to pure heptane, but it was possible to filter out the excess noise. Root mean square noise improvements of as much as 30 times were possible. Signal amplitude was also reduced in microemulsions for the test compounds 1-naphthalenesulfonic acid, 4-amino-1-naphthalenesulfonic acid, and *p*-phenylenediamine, however, such that no improvement in the detection limit was realized. Photoionization efficiency is very sensitive to the microenvironment. Self-buffering of the water pool is proposed as a possible mechanism for signal suppression. However, precision and linearity of response were improved, particularly in determination of the acid salts. The potential exists for use in fundamental studies of photoionization and for analyses with buffered or high ionic strength solutions.

* Author to whom correspondence is to be addressed.

INTRODUCTION

Laser-excited multiphoton photoionization spectroscopy (MPI) has been shown to be a useful and sensitive method for the determination of molecular species in the condensed phase.[1-4] Briefly, photoionization is the process whereby a molecule absorbs two photons whose combined energy is enough to promote an electron into the ionization continuum. The geminate ion-electron pair thus produced lasts only a few tens of nanoseconds in liquid solvents; however, if the ion-electron pair is produced in an electric field gradient, a current proportional to the number of charge carriers will be induced in the electrodes and can be detected and amplified by conventional methods.

One barrier to the general usefulness of MPI stems from the detection of the ionization event in polar solvents. When the bias voltage is applied across polar analyte solutions, residual ionic impurities and auto-ionized solvents carry a certain amount of current, the leakage current.[2] The dc component of the leakage current is easily removed, but the noise associated with it is not. This excess noise is responsible for detection limits that are at least 2–3 orders of magnitude worse for analytes in polar solvents than in nonpolar solvents.[1,5-7] The amount of noise and its spectral character depend strongly on the polarity and grade of the solvent,[8] but

0003-2700/92/0364-0551$03.00/0