Pruning neural network for architecture optimization applied to near-infrared reflectance spectroscopic measurements. Determination of the nitrogen content in wheat leaves

The Analyst

Cesar Mello,[a], Ronei J. Poppi,*[a] João Carlos de Andrade[a] and Heitor Cantarella[b]

[a] *Instituto de Química, Universidade Estadual de Campinas, C.P. 6154, CEP 13083-970 Campinas, SP, Brazil*
[b] *Instituto Agronômico, Centro de Solos e Recursos Agroambientais, C.P. 28, CEP 13001-970 Campinas, SP, Brazil*

Received 11th June 1999, Accepted 20th September 1999

The pruning neural network, based on the algorithm called optimum brain surgeon, was used for network architecture optimization. This network pruning procedure was applied for estimating the nitrogen contents in wheat leaves, using near-infrared diffuse reflectance spectroscopy. The results obtained with pruning were compared with those obtained by using ordinary procedures with neural networks, partial least squares, polynomial partial least squares and neural networks/partial least squares methodologies. Comparison of the results with those obtained by the conventional Kjeldahl method showed that the results with pruning neural networks were as good as those with ordinary neural networks and with PLS/neural networks, but better than those with the other methodologies. Although the comparison was performed for one data set, the pruning procedure has the advantage of introducing an automatic architecture optimization, which is cumbersome when performed by the other neural network procedures used in this work, generating a simplified model with better generalization abilities.

Introduction

Near-infrared diffuse reflectance (NIRR) spectroscopic measurements[1] allow the determination of a large number of substances in different matrices[2–9] without requiring any chemical reaction. Consequently, it provides a decrease in both reagent consumption and the time needed for performing the analysis, in addition to permitting *in situ* analysis.[10] These advantages confer particular characteristics on the technique, making it useful for routine analysis and on-line control in industrial processes.[11,12].

Analytical applications[13,14] of near-infrared spectroscopy have been explored since the early 1950s. However, owing to instrumental difficulties, its use as a quantitative approach only started 30 years later, together with the development and the widespread use of computers. As the coupling between equipment and microcomputers became easier and faster, obtaining and storing large amounts of data became possible. Thus, once the difficulties in obtaining the diffuse reflectance spectrum in the near-infrared region had been overcome, the next step was to extract quantitative information from them. At this point, it was observed that diffuse reflectance measurements in the near-infrared region could be related to the concentration of substances in the sample, although not with a linear relationship. This fact made the use of classical procedures of univariate calibration difficult or even impossible.

In recent years, with the development of chemometric methods, the application of multivariate calibration methods has become widely employed, enabling us to consider not only the reflectance spectra at a few characteristic wavelengths in calculations,[15] but also the full spectra. Neural networks[16] are among the possible non-linear multivariate calibration methods[17] which could be applied in these cases, and more recently neural networks with pruning[18] have been also considered.

The ability to model equally well data where the relation between dependent and independent variables is linear or non-linear is easily carried out with a neural network, one of the most commonly used multivariate calibration methods. In recent years, the number of neural network applications in chemistry has increased and diversified.[19,20] It is possible to find applications in modeling non-linear data in analytical chemistry,[21] in on-line processes,[22] and in design of new drugs[23] by QSAR.

The basic concepts of artificial neural networks

Neural networks (NN) are composed of basic units of information processing called neurons, which are arranged in lines or layers. An artificial neural network always has an input and an output layer and, between them, there is a variable number of hidden layers. This layer disposition and the number of neurons in each layer are called the neural network architecture. The input can be any multivariate signal, such as current readings at different potentials of a voltammogram, or absorbances at different wavelengths in spectroscopic measurements. The output of neural network responses is the independent variables (*e.g.*, concentrations), with which the neural network will be trained, in a typical calibration procedure.

The response or output, $\hat{Y}$, to an input vector $\boldsymbol{x}$, of a neural network with n_i input neurons, one hidden layer with n_h neurons and one output neuron with linear transfer function, can be written as

$$\hat{Y} = \sum_{j=1}^{n_h} W_j f\left(\sum_{i=1}^{n_i} w_{ji} x_i + b_j\right) + B \qquad (1)$$

where f is a linear, sigmoid or tangent hyperbolic transfer function, b_j and B are the biases of the model and $\boldsymbol{w}_{ji}$ *and* $\boldsymbol{W}_j$ *the weights of the hidden and output layers, respectively. In this study, the input vector* $\boldsymbol{x}$ is the NIRR spectra scores and the output is the nitrogen content in wheat leaves.

Once the values estimated by the neural network (the neural network output) have been obtained, the calibration error (E), defined as the sum of the squares resulting from the difference between the value estimated by the network (output) and the expected value given by conventional Kjeldahl method, can be calculated, as demonstrated in eqn. (2):

$$E(\boldsymbol{w}) = \sum_{i=1}^{m} (Y - \hat{Y})^2 \qquad (2)$$

where Y is the real value, $\hat{Y}$ is the neural network output value and m is the number of samples used to train the neural network. After obtaining the neural network output value, the next step is to correct the weights of all layers until the error (E) is minimized, which can be made through the error back-propagation procedure[17] or by the Marquardt-Levenberg method,[24] a variant of the Gauss–Newton method.[24] In this work the Marquardt–Levenberg approach was used, which is faster to converge and more robust.

The objective of the training procedure is to find a set of possible weights and biases which permit the network to predict values ($\hat{Y}$) as close as possible to the known outputs (Y), in order to minimize the error function $E(\boldsymbol{w})$. The correction of the weights ends when the error [$E(\boldsymbol{w})$] reaches previously established convergence criteria. At this point, the neural network is considered trained, and it is then possible to evaluate its generalization properties, using another sample group, with a different set of data from those used in the calibration step, known as the validation set. However, an artificial neural network, even after training, can present small errors in the calibration set and large errors in the validation set. In these cases, an overfitting due to an excessive quantity of neurons applied in the hidden layer has occurred. For instance, this is the case when the number of neurons in the hidden layer is close to the principal component number used in the principal component regression[17] (PCR) or to the polynomial order used in the polynomial regression. Whilst an over-complex architecture is likely to lead to overfitting, it can also occur as a result of a long training (overtraining). Therefore, the choice of the quantity of neurons in the hidden layer and the number of epochs (number of weights correction) used in the training are fundamental to avoid overfitting.

A long training can be avoid by selecting appropriate convergence criteria, based on the experimental error, since the number of epochs is basically governed by this term.

The correct selection of the quantity of neurons or even the architecture optimization of the hidden layer can be done by varying the quantities of these neurons and repeating the process of weight correction. One then, chooses the configuration which presents small errors for both calibration and validation.

Network pruning

A procedure for architecture optimization is the use of pruning neural networks (PNN). The basic idea of this method is to start a neural network with an excessive number of neurons inside the hidden layer and cut connections which have a slight influence on the error during the training step. Neurons to which all connections were cut will be deleted. Hence only those neurons useful for modeling will be maintained. The pruning technique[25] reduces the complexity of the neural networks and improves their ability of prediction, avoiding the use of over-parameterized models (excessive amount of neurons).

There are two methods of PNN the optimum brain damage[26] (OBD) and the optimum brain surgeon[27] (OBS). In both methods, the connections (or weights) are cut and the corresponding variation in error E, called saliency,[28] is evaluated. In OBD, the cut in the connections occurs during the training step and the neural network is not trained again. In contrast, in the OBS method, the network is trained after each cut, allowing new cuts to be made.

Optimum brain surgeon (OBS)

A term of regularization[29] is added to the error function $E(\boldsymbol{w})$, which is given by eqn. (2), related to the function cost [$C(\boldsymbol{w})$]:

$$C(\boldsymbol{w}) = E(\boldsymbol{w}) + \frac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{w} \qquad (3)$$

The regularization term induces pruning processes and ensures numerical stability of the method, simply by punishing weights with low values through the regularization matrix $\boldsymbol{D}$. The next step is to enlarge the cost function in a Taylor series to the second order terms, around a possible minimum weight value $\boldsymbol{w_0}$, as follows:

$$C(\boldsymbol{w}) = C(\boldsymbol{w}_0) + \frac{1}{2}\delta\boldsymbol{w}^{\mathrm{T}}\boldsymbol{H}\delta\boldsymbol{w} \qquad (4)$$

where $\delta\boldsymbol{w} = \boldsymbol{w} - \boldsymbol{w}_0$ and $\boldsymbol{H} = (\boldsymbol{A} + \boldsymbol{D})$ is the Hessian[30] matrix with a regularization term. The term $\boldsymbol{A}$ is the second derivative matrix of the training error ($\partial^2 E/\partial \boldsymbol{w}^2$) containing all second order derivatives. After this phase, one weight must be eliminated and the cost function minimized. The elimination of the jth weight can be expressed as

$$\delta\boldsymbol{w}^{\mathrm{T}}\boldsymbol{e}_j = -\boldsymbol{w}_0^{\mathrm{T}}\boldsymbol{e}_j \qquad (5)$$

where $\boldsymbol{e}_j$ is the jth unit vector. The constrained extreme can be obtained by applying the Lagrange multipliers method:[30]

$$\tilde{C}_j(\boldsymbol{w}) = C(\boldsymbol{w}) + \lambda(\delta\boldsymbol{w} + \boldsymbol{w}_0)^{\mathrm{T}}\boldsymbol{e}_j \qquad (6)$$

where λ is the Lagrange multiplier.

Solving the above equation, one can obtain the vinculated extreme:

$$\delta\boldsymbol{w} = -\lambda\boldsymbol{H}^{-1}\boldsymbol{e}_j \qquad (7)$$

After this step, it is possible to return to eqn. (3), obtain $\delta E(\boldsymbol{w})$ and replace $\delta\boldsymbol{w}$ by the value obtained with eqn. (7), resulting in the following equation for the saliency, which is the change in the training error resulting from deletion of $\boldsymbol{w}_j$:

$$\delta E_j(\boldsymbol{w}) = \lambda\boldsymbol{w}_0^{\mathrm{T}}\boldsymbol{D}\boldsymbol{H}^{-1}\boldsymbol{e}_j + \frac{1}{2}\lambda^2\boldsymbol{e}_j^{\mathrm{T}}\boldsymbol{H}^{-1}(\nabla E + \boldsymbol{D})\boldsymbol{H}^{-1}\boldsymbol{e}_j \qquad (8)$$

This saliency equation will give mathematical criteria for cutting the connections.

Partial least squares

In addition to the evaluation of the neural networks and pruning neural networks approaches, other multivariate calibration methods such as the PLS,[31] PLS-polynomial[32] (POLYPLS) and PLS-neural network[33] (NNPLS) were also taken into account. The basic assumptions of these methods are first, to decompose

all independent variables and dependent variables, X and Y, respectively, into one product with two short matrices, plus one error matrix, as follows :

$$\boldsymbol{X} = \boldsymbol{TP} + \boldsymbol{E} \quad (9)$$

$$\boldsymbol{Y} = \boldsymbol{UQ} + \boldsymbol{F} \quad (10)$$

where $\boldsymbol{T}$ and $\boldsymbol{U}$ are the score matrices, $\boldsymbol{P}$ and $\boldsymbol{Q}$ are the loading matrices and, $\boldsymbol{E}$ and $\boldsymbol{F}$, are the error matrices of $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively. Such decomposition is extremely important in cases where the $\boldsymbol{X}$ matrix is improperly stored or when the number of samples is smaller than the number of independent variables. One can use the $\boldsymbol{T}$ matrix with a lower dimension than the $\boldsymbol{X}$ matrix without loss of useful information and also eliminating noise in data and colinearity. Once the decomposition has been done, the next procedure is to set, if possible, a linear relation between $\boldsymbol{U}$ and $\boldsymbol{T}$:

$$\boldsymbol{U} = \boldsymbol{bT} + \boldsymbol{e} \quad (11)$$

where $\boldsymbol{b}$ is an adjusted coefficient which is usually done using the NIPALS algorithm.[34] Finally, by replacing the $\boldsymbol{U}$ value in eqn. (10), one can obtain:

$$\boldsymbol{Y} = \boldsymbol{bTQ} + \boldsymbol{G} \quad (12)$$

For cases where the relationship between the scores of $\boldsymbol{X}$ and $\boldsymbol{Y}$ is not linear, the POLYPLS or the NNPLS can be used. Both have basically the same appliances, differing only in the relation between the scores, that are made by a variable degree polynomial or by a neural network approach.

Experimental

Sample set

The sample set consisted of 58 wheat leaf samples, which were washed with a 1% v/v non-ionic detergent solution and rinsed 10 times with de-ionized water. The samples were oven-dried at 50 °C for 48 h, then ground using a Retsch mill with a 1.0 mm grid. Finally, the nitrogen content (in terms of mg N g^{-1} dry matter) was determined using the conventional Kjeldahl method.

Sample spectra

The diffuse reflectance spectra were recorded directly over a sample cell with an optical fiber probe, with low hydroxyl content to avoid near-infrared absorption, using a hand held spectrometer with a monochromator based on the acoustic optic tunable filter[35] (Brimrose) approach, set to operate in the 1500–2400 nm range, as shown in Fig. 1.

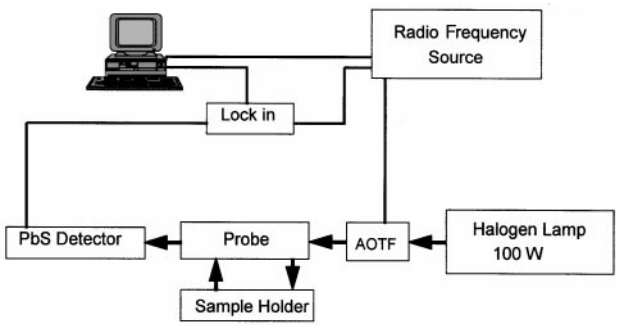

Fig. 1 Instrumental setup for NIR data collection.

Spectra preprocessing

Before the calibration phase, the spectra were preprocessed by using either a fast Fourier transform filter,[36] a principal component analysis (PCA)[37] or a multiplicative scattering correction (MSC).[38] For PLS calculations, the inputs were the whole preprocessed spectra using the fast Fourier transform filter and MSC, and for neural networks the inputs were the scores of the most significant principal components, obtained from the spectral data. The spectra set, after the application of the fast Fourier transform filter and MSC, are shown in Fig. 2.

Sample selection

The data set was split into two subsets, one to be used in the calibration procedure and other to validate the model. The calibration set had 34 samples and the validation set 24 samples. The selection of the samples into subsets was made according to the principal component analysis (PCA) approach. Plotting the scores of the first principal component (PC1) versus the second principal component (PC2), which represents 87% of data variance, it is possible to see that there are four groups of samples, as shown in Fig. 3. Samples representing each of the four groups were put into both the calibration and validation sets, maintaining in the calibration set samples with high and low nitrogen contents, to avoid extrapolation in the validation set. The sample selection inside each group was performed randomly.

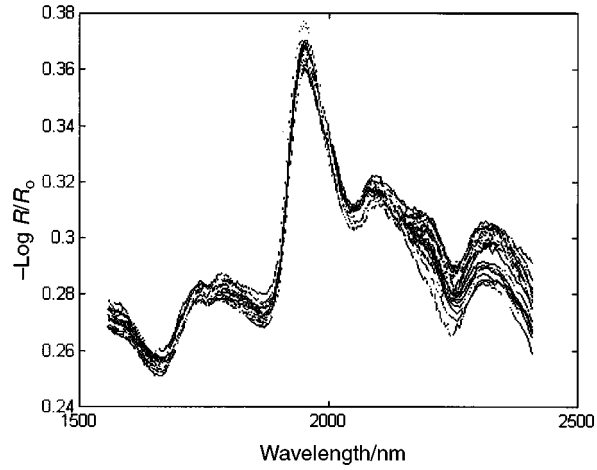

Fig. 2 NIR spectral data set after preprocessing.

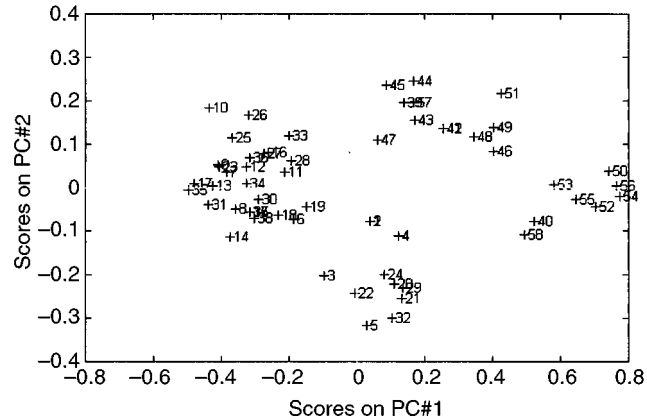

Fig. 3 Plot of scores on PC1 *vs.* scores on PC2.

Evaluation of the performance of the models

The relative performance of the different models was evaluated in terms of root-mean-square error of prediction (%RMSEP), defined as

$$\%\text{RMSEP} = \frac{1}{\bar{c}_p}\sqrt{\frac{\sum_{i=1}^{n_p}(y_i - \hat{y}_i)^2}{n_p}} \times 100 \qquad (13)$$

where $\bar{c}_p$ and n_p are the mean concentration and number of the samples used in the validation set, respectively, y_i is the real value and $\hat{y}_i$ is the predicted value obtained by different models.

Computer programs

The program used for network pruning was obtained from the neural network based system Identification Toolbox[39] for use with Matlab. The PLS and PCA calculations were from the PLS Toolbox[40] for use with Matlab. The program for noise reduction by fast Fourier transformation was implemented utilizing subroutines from Matlab 4.2.

Models

Neural network. The inputs employed for neural network training were the scores of the first five principal components scaled in the range from 0.2 to 0.8. The optimum network architecture and the training parameters are summarized in Table 1, where L indicates a linear transfer function and S a sigmoidal transfer function. For example, the notation SSSS means that there are four neurons with sigmoidal transfer functions. As already mentioned, the neural network was trained using the Marquardt–Levenberg method.

Pruning neural network. The inputs employed for neural network training are the same as mentioned above for the ordinary neural network procedure. The initial architecture network (before pruning) and training parameters are summarized in Table 1 and shown in Fig. 4. In this table, the notation SLSLSLSLSL means that there are 10 neurons in the hidden layer, the first with a sigmoidal transfer function and the next with a linear function, and so on. This procedure is very interesting because, in principle, it is possible to model linear and non-linear systems. Moreover, this procedure may be considered as diagnostic for non-linearity, because when at the end of the pruning procedure non-linear neurons remain, the system will be non-linear. Again, the initial neural network was trained using the Marquardt–Levenberg method. The optimization of the neural network architecture was carried out using the OBS algorithm.

The optimum architecture obtained after both training and pruning procedures is shown in Fig. 5.

PLS based models. The optimum number of factors (latent variables) utilized in all PLS models, obtained using leave-one-out cross-validation , was five. In the model elaborated with POLYPLS, the best relationship between ***u*** and ***t*** was established by a second degree polynomial.

In NNPLS, the best relationship between ***u*** and ***t*** was established by a neural network with an input layer with one neuron (one score at a time), a hidden layer with three neurons and an output layer with one neuron. The transfer function used between the hidden and output layers was sigmoidal and for the output layer the transfer function used was linear.

Table 1 Architecture of the neural network

Parameter	Value	
	Ordinary neural network	Pruning neural network
Input layer nodes	5	5
Input layer transfer function	LLLLL	LLLLL
Hidden layer nodes	4	10
Hidden layer transfer function	SSSS	LSLSLSLSLS
Output layer nodes	1	1
Output layer transfer function	L	L
Maximum number of iterations	500	100
Stop criterion	0.01	0.01
Regularization term (***D***)	—	0.01

Results and discussion

Five models were developed using NN, PNN, NNPLS, POLYPLS and PLS. The %SEC and %SEP values for each model are given in Table 2. In Fig. 6(A)–6(E) the correlation between the values of %N given by the Kjeldahl method and those predicted for each of five models is shown.

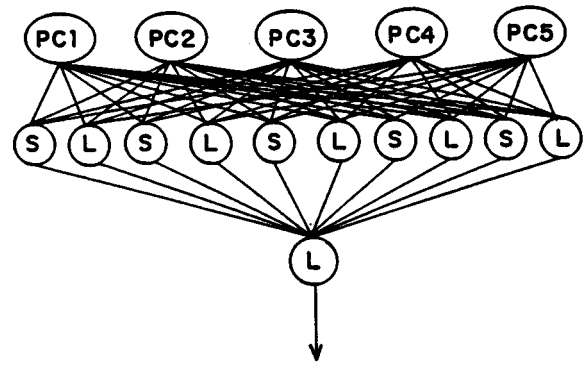

Fig. 4 Neural network initialization, where L indicates a linear transfer function and S a sigmoid transfer function.

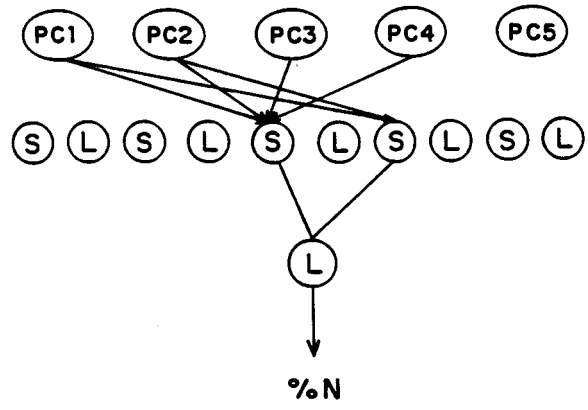

Fig. 5 Neural network after pruning, where L indicates a linear transfer function and S a sigmoid transfer function.

Table 2 Results obtained using different models

Model	%SEC	%SEP
PNN	4.8	6.2
Ordinary NN	5.0	6.5
NNPLS	6.1	7.6
POLYPLS	6.4	9.6
PLS	6.7	8.9

The analysis of the %SEP values, shown in Table 2, indicatest that the pruning neural network method was the most efficient calibration method, followed by ordinary neural networks, NNPLS, PLS and POLYPLS. The performance of neural networks with pruning was 5% better than that of ordinary neural networks, 18% better than NNPLS, 30% better than PLS and 35% better than POLYPLS. However, are these differences in %SEP values significantly different to justify the above conclusions?

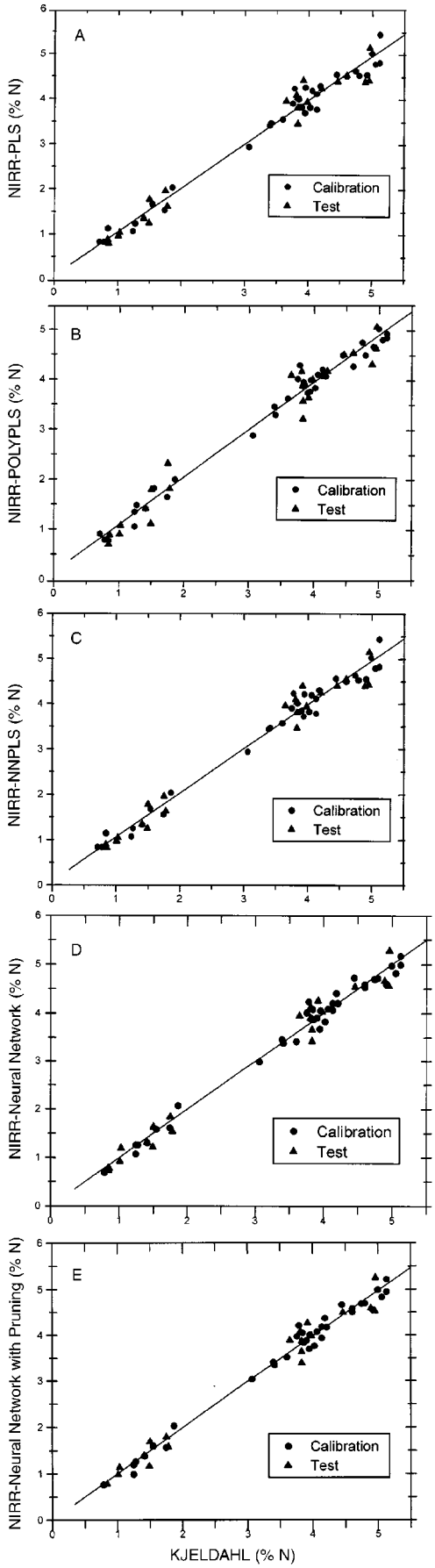

Fig. 6 Nitrogen content of wheat leaves predicted by five models *vs.* the Kjeldahl method: (A) PLS; (B) POLYPLS; (C) NNPLS; (D) ordinary NN; and (E) PNN.

To confirm that the above conclusions are correct, an *F*-test, at the 95% confidence level, according to eqn. (14), was used to compare RMSEP for the different calibration methods:

$$F(p_i p_j) = \left(\frac{\text{RMSEP}_{\text{method}}}{\text{RMSEP}_{\text{PNN}}} \right)^2 \qquad (14)$$

where p is the number of validation samples. The *F*-test critical value at the 95% confidence level is $F_{\text{critical}} = 1.98$.

When the *F*-test is applied to the %SEP values obtained from different modeling methods, it is observed that PNN are significantly better than PLS ($F = 2.1$) and POLYPLS ($F = 2.4$). However, for the PNN, NNPLS and ordinary NN there are no significant differences in the %SEP values at this confidence level.

The results obtained with the *F*-test, at the 95% confidence level, showed that the performance of five modeling algorithms can be classified basically in two groups. The first, with lower errors (PNN, NNPLS and ordinary NN), and the second one, with larger errors (PLS and POLYPLS). Models that belong to a determined group do not show differences in %SEP values at this confidence level.

Conclusion

The evaluation of modeling methods simply by comparing prediction errors can result in misleading conclusions. Therefore, the evaluation of the model performance requires the application of statistical significance tests, such as the *F*-test.

The modeling methods tested were classified into two groups. One, with lower prediction errors, formed by PNN, ordinary NN and NNPLS methods, and the other, with larger errors, formed by PLS and POLYPLS methods. In each of these groups there are no significant differences in the values of %SEC and %SEP at the 95% confidence level.

Hence, the models based on neural networks show superior performance for determining nitrogen content in plants using diffuse reflectance spectroscopy in the near-infrared region. The three models based on neural networks do not show significant differences in the %SEP values, and the choice of one of them must be made by considering factors such as computational time, facility of implementation and software availability. Therefore, it is possible to consider that pruning neural networks offers a significant application potential in multivariate calibration compared with the other methods studied, since its use does not need a network architecture optimization procedure, reducing the possibility of overfitting. In addition, only the pruning neural network offers the possibility of determining if the system is linear or non-linear, and of performing the modeling for both sets of data.

Acknowledgements

The authors thank the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) and the Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) for fellowships and financial support of their research projects. Thanks are also due to Professor C. H. Collins for critically reading the manuscript.

References

1 P. William and K. Norris,, *Near-infrared Technology in Agricultural and Foods Industries*, American Association of Cereal Chemists, St. Paul, Minnesota, 1990.
2 D. L. Wetzel, *Anal. Chem*., 1983, **55**, 1165A.
3 M. Twoney, G. Downey and P. B. McNulty, *J. Sci. Food Agric*., 1984, **67**, 77.
4 D. H. McQueen, R. Wilson and A. Kinnunen, *Talanta*, 1995, **42**, 2007.
5 E. Dreassi, G. Ceramelli, P. Corti, P. L. Perruccio and S. Lonard, *Analyst*, 1996, **121**, 219.
6 D. Jouan-Rimbaud, M. S. Khots, D. L. Massart, I. R. Last and K. A. Prebble, *Anal. Chim. Acta*, 1995, **315**, 243.
7 R. Tkachuk, *J. Am. Oil Chem. Soc*., 1981, **58**, 819.
8 M. Hana, W. F. McClure and T. B. Whitaker, *J. Near Infrared Spectrosc*., 1995, **3**, 133.
9 D. Wienker, W. Broek, W. Melssen, L. Buydens, R. Feldhoff, T. Kantimm, T. H. Fehre, L. Quick and K. Cammann, *Anal. Chim. Acta*, 1995, **317**, 1.
10 M. Martin, A. Garrison, M. Roberts, P. Hall and C. Moore, *Process Control Qual*., 1993, **5**, 187.
11 D. Lambert, B. Descartes, J. R. Llinas, A. Espinosa, S. Osta, M. Sanchez and A. Martens, *Analusis*, 1995, **23**, M9.
12 D. Lambert, B. Descartes, A. Espinosa, M. Sanchez, S. Osta, J. Gil, A. Martens and M. Valleur, *Analusis*, 1995, **23**, M20.
13 W. Kaye, *Spectrochim. Acta*, 1954, **6**, 257.
14 W. Kaye, *Spectrochim. Acta*, 1955, **7**, 181.
15 H. Martens and T. Naes, *Multivariate Calibration*, John Wiley & Sons, Chichester, 1989.
16 J. Zupan and J. Gasteiger, *Neural Networks for Chemists: an Introduction*, VCH, Weinhein, 1992.
17 S. Sekulic, M. B. Seasholtz, Z. Wang, S. E. Lee, B. R. Holt and B. R. Kowalski, *Anal. Chem*., 1993, **65**, 835A.
18 J. Hertz, A. Krogh and R. G. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley, Menlo Park, CA, 1991.
19 J. Zupan and J. Gasteiger, *Anal. Chim. Acta*, 1991, **248**, 1.
20 G. Katerman and J. R. Smits, *Anal. Chim. Acta*, 1993, **277**, 179.
21 P. J. Gemperline, J. R. Long and V. G. Gregoriou, *Anal. Chem*., 1991, **63**, 2313.
22 P. Marteau, N. Zanier-Szydlowski, A. Aoufi, G. Hotier and F. Cansell, *Vib. Spectrosc*., 1995, **9**, 101.
23 X. H. Song, R. Q. Yu, *Chemom. Intell. Lab. Syst.*, 1993, **19**, 101.
24 J. R. S. Jang, C. T. Sun and E. Mizutani, *Neuro Fuzzy and Soft Computing*, Prentice Hall, Englewood Cliffs, NJ, 1997.
25 R. J. Poppi and D. L. Massart, *Anal. Chim. Acta*, 1998, **375**, 187.
26 Y. Le Cun, B. Boser, J. S. Denker and S. A. Solla, in *Advances in Neural Information Processing Systems*, ed. D. S. Touretzsky, Morgan Kaufman, San Mateo, CA, 1990, vol. II, p. 598.
27 S. E. Fahlman and C. Libiere, in *Advances in Neural Information Processing Systems*, ed. D. S. Touretzsky, Morgan Kaufmann, San Mateo, CA, 1990, vol. II, p. 524.
28 C. Goutte, in *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing IV*, Piscataway, NJ, 1994.
29 B. Hassibi and D. G. Stork, in *Advances in Neural Information Processing Systems*, ed. J. D. Cowan and C. L. Giles, Morgan Kaufmann, San Mateo, CA, 1993, vol. V.
30 J. Larsen and L. K. Hansen, in *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing IV*, Piscataway,NJ, 1994.
31 K. R. Beebe and B. R. Kowalski, *Anal. Chem*., 1987, **59**, 1007A.
32 S. Wold, N. Kettaneh-Wold and B. Skagerberg, *Chemom. Intell. Lab. Syst*., 1989, **14**, 333.
33 S. J. Quin and T. J. McAvoy, *Comput. Chem. Eng*., 1992, **16**, 379.
34 P. Geladi and B. R. Kowalski, *Anal. Chim. Acta*, 1986, **185**, 1.
35 D. C. Tran, *Anal. Chem*., 1992, **64**, 917A.
36 P. Bloomfield, *Fourier Analysis of Time Series*, Wiley, New York, 1976.
37 M. J. Adams, *Chemometrics in Analytical Spectroscopy*, Royal Society of Chemistry, Cambridge, 1995.
38 T. Isaksson and B. R. Kowalski, *Appl. Spectrosc*., 1993, **47**, 702.
39 N. Norgaard, *Neural Network Based Systems Identification Toolbox*, Technical Report 95-E-773, Institute of Automation, Technical University of Denmark, Lyngby, 1995.
40 B. M. Wise, N. B. Gallagher, *PLS_Toolbox for Use with Matlab, Version 1.5.1*, Eigenvector Technologies, Manson, 1995.

Paper 9/05570C