

Near infrared spectroscopy for classification of Iberian pig carcasses using an artificial neural network

C. Hervás,^a A. Garrido,^{b*} B. Lucena,^b N. García^a and E. De Pedro^b

^a*Departamento Matemática Aplicada, EU Politécnica, University of Córdoba, Spain.*

^b*Departamento Producción Animal, ETSIAM, University of Córdoba. Spain.*

Artificial neural networks (ANNs) have demonstrated their usefulness in near infrared (NIR) reflection and transmittance spectroscopy for quantitative prediction. The new approach presented here considers the use of ANNs for qualitative classification. Four forms of neural networks (a competitive network using the learning vector quantisation, LVQ learning rule; a backpropagation network using the extended delta-bar-delta, EDBD rule; a network with direct random search, DRS; and a simple competitive linear network, CL) have been tested for classification of 118 fat samples from Iberian pig carcasses into three different price groups. An ANN using the LVQ learning rule has been found to be the best in terms of classification error size. The classification ability of the LVQ network has been evaluated against discriminant analysis, one of the most used methods for NIR spectroscopic qualitative analysis.

Keywords: Artificial neural network, discriminant analysis, near infrared spectroscopy, qualitative analysis, Iberian pig.

Introduction

Over the last few years the number of papers devoted to artificial neural networks has increased significantly in the field of near infrared (NIR) reflectance and transmittance spectroscopy. ANNs have been used to find non-linear relationships between NIR reflectance and transmittance spectral data and analyte concentration.^{1–7} Most of these papers have been focused on determining whether ANNs could improve upon the traditional calibration methods (stepwise regression, principal component regression and partial least squares) and the results are quite diverse. It is true that most of the published examples show ANNs to be superior. However, it remains to be seen how much the added complexity of an ANN is worth, in general. From a practical point of view, one drawback with ANN

methods is that they are time-consuming.¹ Westergaard and Reeves⁶ compared three forms of neural networks, back-propagation networks (BPN), delta-bar-delta networks (DBD) and extended delta-bar-delta (EDBD) and they found that DBD and EDBD methods produced very similar results; the EDBD methods, however, required fewer training iterations saving time on the training phase, which could take days for BPN and DBD methods.

While ANNs have been used for quantitative NIR analysis a limited amount of work has been carried out on their use for NIR qualitative measurement. Previous work at the Animal Production Department (University of Córdoba)^{8,9} has shown that discriminant analysis using PCA scores from NIR transmission spectra of Iberian pig fat is more appropriate to routine use for the classification of Iberian hams in industrial quality assurance than

other traditional methods such as gas chromatography.

The objective of the present work is to evaluate the potential of ANNs as a method for classifying ham from Iberian pigs into different quality categories. The classification ability is evaluated against discriminant analysis.

Material and methods

Samples

One hundred and eighteen fat samples were collected from Iberian pig carcasses belonging to three different experimental groups: GI (pigs fed with acorns plus low amounts of a feed compound), GII (pigs fed with acorns plus medium amounts of a feed compound) and GIII (pigs fed only with feed compounds).

Sample preparation and NIR analyses

Frozen fat samples were defrosted in a microwave oven and then stored in an oven at 35°C in order to remain liquid.

A scanning monochromator (NIRSystems 6250) was used to measure NIR transmission spectra consisting of 700 datapoints at 2 nm intervals from 1100 to 2498 nm. The analysis was carried out by using a hand-made bronze sample holder. One ml of fat was placed between two quartz glasses having a path length of 1 mm. To reduce the effect of instrument noise, each spectrum was recorded after the average reading of 32 scans per sample. One single spectrum was recorded for each sample. NIR spectral data were manipulated using the ISI software package.¹⁰

Training and test set selection

Two measures are traditionally used to determine the quality of a given classification model. The first is to evaluate the percentage of correctly (or incorrectly) classified samples (objects) from a training set used to build the classification model. The second is evaluating the percentage of correctly (or incorrectly) classified samples in a test set which was not used for training. The selection of the training set is one of the more critical steps, not only for

neural networks, but for any approach of extracting knowledge from data (Zupan and Gasteiger¹¹). If no clear criterion exists, the training set can be chosen at random. ISI software provides an excellent opportunity to select the most representative set of samples from a file of available spectra, by using only the spectral data. The algorithm SELECT (Shenk and Westerhaus),^{12,13} included in ISI software, was designed to eliminate samples with similar spectra from a file of spectra. Singular value decomposition is used to calculate the eigenvalues and principal components for the spectral data. SELECT computes a suggested number of PCAs by counting the number of eigenvalues greater than the average eigenvalue divided by the square root of the number of samples. Each spectrum is first transformed by a (1,10,5) derivative, where the first number in parenthesis is the order of the derivative, the second number is the segment length in data points over which the derivative was taken, and the third is the number of data points in a running average smooth. The program calculates a standardised H (Mahalanobis) distance between all pairs of samples based on eigenvectors. A minimum distance is used such that if the distance between two spectra is less than the minimum distance ($H = 0.6$), one of the spectra is eliminated. The algorithm starts by identifying the spectrum that had the most neighbouring spectra closer than the minimum distance. The spectrum is retained and its neighbours discarded. The process then evaluates all remaining samples to identify the sample that has the most neighbours. Once again that sample is retained and its neighbours discarded. This process continues until no samples remain with neighbours closer than the minimum distance.

The application of the SELECT program to the original set of 118 spectra computed a suggested number of 12 principal components to represent all the information in the spectrum. A total of 64 samples were selected by SELECT as a training set and the remaining 54 samples were used as a test set.

Statistical analyses

Linear discriminant analysis was performed using the PROC DISCRIM procedure of the SAS software package.¹⁴ Neural networks development

was accomplished using the simulation package Neural Networks Professional II.¹⁵ Twelve PCA scores from the NIR data, as chosen by the SELECT program, were used as quantitative variables for linear discriminant analysis and as input to the networks. Four different types of artificial neural networks were explored for the purpose of classification: a competitive network using the learning vector quantisation LVQ learning rule,^{16,17} a back-propagation network using the extended-bar-delta EDBD learning rule,¹⁸ a learning algorithm without derivatives DRS¹⁹ and a simple competitive linear network CL.²⁰

In this paper, the authors use the percentage of samples incorrectly classified as a criterion to test the performance of the discriminant analysis and artificial neural network models.

A detailed description of the algorithms used for each network is outside of the scope of this paper, so the reader is referred to the references given for each network. However, since we found the LVQ network to be the best in terms of classification error, a short description of its structure and learning algorithm is given below for readers who are unfamiliar with that design.

LVQ network

The LVQ network was originally suggested by Kohonen.¹⁶ An LVQ network consists of an input layer, a Kohonen layer, which learns and performs the classification, and an output layer that produces the output following the classification made by the Kohonen layer (KL). The basic idea of the training is as follows.

The distance of the input vector to each unit in the KL is computed, and the nearest unit is declared to be the winner. If the winner unit is in the class of the input vector, it is moved towards it; on the other hand, if the winning unit belongs to another class, it is moved away from the input vector. This is referred to as *repulsion*.

In the basic LVQ learning, Euclidean distance is used. So the distance between the input pattern, x , and the weight vector of the i -th unit, w_i , is computed as follows:

$$d_i = \|w_i - x\| = \left\{ \sum_{j=1}^N (w_{ij} - x_j)^2 \right\}^{1/2}$$

and the winning unit adjusts its weight vector according to:

$$w' = \begin{cases} w + \lambda_1(x - w), & \text{if the winning unit is in the correct class} \\ w - \lambda_2(x - w), & \text{if winning unit is NOT in the correct class} \end{cases}$$

where λ_1 and λ_2 are coefficients set by the user. This simple form of the LVQ algorithm suffers from the defect that some units tend to win too often while others never win. To avoid this defect a *conscience* mechanism was suggested by DeSieno.¹⁷ When a unit wins too often it develops a guilty conscience and penalises itself for having won too often. This variant of the basic algorithm is called LVQ1. The distance is computed as the sum of two terms:

$$d'_i = d_i + b_i$$

the bias term, b_i , is a function of the guilty conscience of each unit, and is calculated according to:

$$b_i = \beta d_{i_{\max}} (1 - Np_i)$$

where

- $d_{i_{\max}}$ is the maximum distance without bias,
- β is a constant which decreases as the training progresses,
- p_i estimates the unit win frequency, it is the guilty conscience,
- N is the number of units per class,

with the original Euclidean distance and the biased distance, LVQ1 determines both a global winner with the former and an in-class winner with the latter. The in-class winner is moved toward the input pattern according to the following:

$$w' = \begin{cases} w + \eta_1(x - w), & \text{if the in-class winner is also the global winner} \\ w + \eta_2(x - w), & \text{if the in-class winner is NOT the global winner} \end{cases}$$

while the global winner is moved away from the input vector according to:

$$w' = w - \eta_3(x - w),$$

if the global winner is NOT in the correct class.

Finally, the conscience, p_i , of each unit is computed:

$$p_i = \begin{cases} (1 - \alpha)p_i, & \text{if } i \text{ is NOT the in-class winner} \\ (1 - \alpha)p_i + \alpha, & \text{if } i \text{ is the in-class winner} \end{cases}$$

There is another variant of the LVQ algorithm, the LVQ2 algorithm. It is used for refining a model obtained with the LVQ or LVQ1 learning processes. LVQ2 is limited to cases in which the winning unit is in the wrong class and the second best unit is in the right one. A further limitation is that the training vector must be near the midpoint of the line joining these two units. Then the winning wrong unit is moved away from the input vector and the second best unit is moved toward it. In this way the procedure refines the boundary between the regions where the misclassifications are occurring.

The winning unit, w_1 , and the second best unit, w_2 , update their weight vectors as follows:

$$w_i' = w_1 - \eta_1(x - w_1)$$

$$w_i' = w_2 + \eta_1(x - w_2)$$

if x is near to $(w_1 + w_2)/2$. Nearness requires that x lies between the two planes perpendicular to the line joining w_1 and w_2 and passing through the points:

$$\frac{(w_1 + w_2)}{2} \pm \gamma(w_2 - w_1),$$

where γ is the *nearness* coefficient, and is fixed by the user.

In our training process we have used the LVQ1 learning rule for the first 3000 iterations, and then we have proceeded with the LVQ2 learning rule.

However, no further improvement has been achieved with the LVQ2 rule. The parameters of the learning process were modified following the schedule of Table 1.

Results and discussion

Classification problems can be attempted with different neural networks design and learning strategies. The first step taken in the present study was to compare the performance of four different networks: a competitive network using the learning vector quantisation LVQ learning rule, a back-propagation network using the extended-bar-delta EDBD learning rule, a learning algorithm without derivatives DRS, and a simple competitive linear network CL. The training set was used to set-up the parameters and structure for each network (Table 2). We have carried out an optimisation of the main parameters of the networks, such as number of hidden units, learning coefficient and data scaling in an heuristic way.

The results shown in Table 3 and Figure 1 make it clear that the best choice is the competitive network trained with the LVQ algorithm, even though the structure of this network is slightly more complicated than the structure of the EDBD one, the second best net. The LVQ network has 45 units (12 input units, 30 hidden units and three output units) while the EDBD has only 25 units (12 input units, 10 hidden units and three output units).

Table 1. Training parameters schedule.

Parameter	Iterations				
	0–999	1000–1999	2000–2999	3000–3999	4000–9999
η_1	0.06	0.02	0.02	0.03	0.01
η_2	0.05	0.02	0.02	—	—
η_3	0.04	0.02	0.02	—	—
α	0.001	0.001	0.001	—	—
β	1.0	0.3	0.1	—	—
γ	—	—	—	0.1	0.3

Table 2. Parameters of the networks.

Parameter	LVQ	EDBD	DRS ^a	CL
Learning coefficient	0.06	0.09	—	0.20
Momentum	0.00	0.10	—	0.20
Hidden units	30	10	20	20
Transfer functions	Linear	Sigmoid	Sigmoid	Linear
Input ranges	[−1, 1]	[−1, 1]	[−1, 1]	[−1, 1]
Output ranges	[0, 1]	[0, 1]	[0, 1]	[0, 1]

^aLearning coefficient and momentum are meaningless in the DRS network. This network has two parameters, the initial value of the self-adjusting variance that controls the size of the random steps taken in the weight space ($\sigma_0 = 0.125$), and the weight bound value ($w_{\max} = 15.0$).

However, the LVQ net can be considered simpler, since its learning algorithm needs less computational effort and its memory requirements are far less. It can be seen from Figure 1 that the LVQ network achieved the better results in the shorter time. Moreover, the effect of overfitting should be

noted. In the LVQ learning process, the classification error rises from its minimum when the training process passes 2000 iterations. The classification error attained during testing (Table 3) with the LVQ algorithm was three times less (3.64%) than that obtained with the EDBD algorithm (10.91%), and

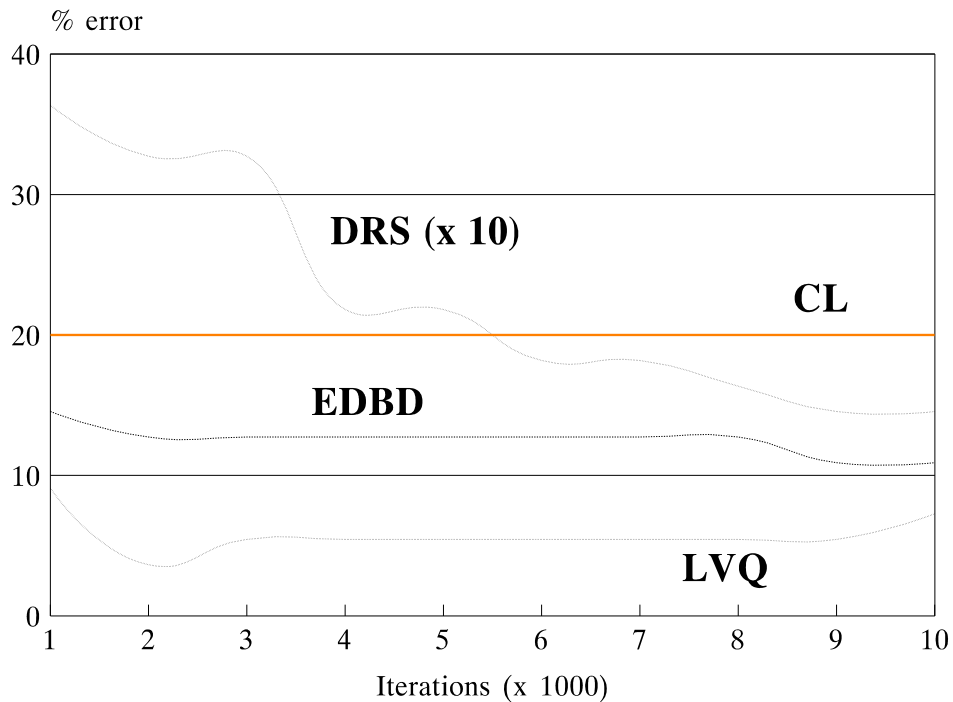


Figure 1. Number of iterations and classification errors during testing.

Table 3. Comparison of the four networks.

Network	Training error ^a	Testing error ^b	Iterations
LVQ	0.00%	3.64%	2000
EDBD	0.00%	10.91%	10000
DRS	0.00%	14.55%	85000
CL	13.85%	20.00%	1000

^aPercentage of samples incorrectly classified in the training set.

^bPercentage of samples incorrectly classified in the testing set.

was reached with five less learning steps. The number of iterations in Table 3 and Figure 1 represent the total number of patterns presented to the network until convergence was reached. It is considered that the convergence has been achieved when the classification error is gradually reduced to a minimum and then it stops or increases (Figure 1).

Finally, the performance of the LVQ ANN model was compared with that of a linear discriminant analysis (LDA) model. Table 4 summarises the performance of the LDA and ANN models on a test set and demonstrates the important aspects of this study. The performance of the LVQ neural network was better than that of the LDA model. The classification errors on the testing set were 3.6% and 12.96%, respectively, for LVQ and LDA models. This means that only two samples were wrongly classified by the LVQ model against seven samples wrongly classified by the LDA model.

One hundred percent of the samples from group III were correctly classified by using both LDA and LVQ models. Eighty-four percent and 80% of samples from groups I and II, respectively, were classified correctly by the LDA model. However, the use of the LVQ neural network allowed a better discrimination giving 96% (group I) and 93.3% (group II) of samples correctly classified (Table 4). An important point is that the LVQ neural network produces a model on which no samples from groups I or II were misclassified into group III. These results are very important for the Iberian pork industry because of the important differences in price between pigs from group III fed mainly with feed

compounds and the other two (I, II) fed mainly with acorns.

A number of parameters could have been of influence in the results obtained (e.g. number of inputs, the size and structure of the training set etc.). However, during the present work we deliberately wished to minimise such variation sources. The SELECT algorithm has been proven previously to be very useful in choosing representative samples from a large population which then can be used as a calibration set for linear regression NIR modelling.^{12,21} The preliminary results obtained in the present work demonstrate that SELECT could also be of valuable help in selecting samples for the training and design of neural networks.

It is important to highlight that, in addition to its better results, the network proposed is extremely simple, so it can work very quickly, even in real time. Slightly more complicated networks could reduce the overall error. Work is in progress to apply a pedagogical pattern selection strategy,²² where the patterns are presented to the network with different probabilities, depending on the problems found by the network in learning them, and in second order algorithms.^{23,24}

Conclusion

The results presented in this paper demonstrate that the artificial neural network investigated here could be used for qualitative analysis of Iberian pig hams giving a lower error of misclassification than a linear discriminant model.

Table 4. Comparison between discriminant analysis and LVQ network.

Linear discriminant analysis				LVQ network			
Belong to class	Classified as			Belong to class	Classified as		
	GI	GII	GIII		GI	GII	GIII
GI	21 (84.0%)	3 (12.0%)	1 (4.0%)	GI	24 (96.0%)	1 (4.0%)	0 (0.0%)
GII	3 (20.0%)	12 (80.0%)	0 (0.0%)	GII	1 (6.7%)	14 (93.3%)	0 (0.0%)
GIII	0 (0.0%)	0 (0.0%)	14 (100.0%)	GIII	0 (0.0%)	0 (0.0%)	14 (100.0%)

However, we can not conclude that neural networks are preferred to other methods in the general case. We have proved that in this particular case the results are clearly better than those obtained with LDA. The results encourage further effort in the application of neural networks to NIR spectroscopic qualitative measurement.

Acknowledgements

The authors would like to thank J.E. Guerrero and C. Borggaard for their helpful contribution to the preparation and discussion of results of this paper and to A. Sánchez-Palma for his work on data file conversion.

References

1. C. Borggaard and A.J. Rasmussen, in *Near Infrared Spectroscopy. Bridging the Gap between Data Analysis and NIR Applications*, Ed by K.I. Hildrum, T. Isaksson, T. Næs and A. Tandberg. Ellis Horwood, pp. 73–78 (1992).
2. C. Borggaard and H.H. Thodberg, *Anal. Chem.* **64**, 545 (1992).
3. K. Kvaal, T. Næs, T. Isaksson and M.R. Ellekjær, in *Near Infrared Spectroscopy. Bridging the Gap between Data Analysis and NIR Applications*, Ed by K.I. Hildrum, T. Isaksson, T. Næs and A. Tandberg. Ellis Horwood, pp. 97–102 (1992).
4. R.L. Long, V.G. Gregoriou and P.J. Gemperline, *Anal. Chem.* **62**(17), 1791 (1990).
5. T. Næs, K. Kvaal, T. Isaksson and C. Miller, *J. Near Infrared Spectrosc.* **1**, 1 (1992).
6. M.O. Westerhaus and J.B. Reeves III, in *Near Infrared Spectroscopy. Bridging the Gap between Data Analysis and NIR Applications*, Ed by K.I. Hildrum, T. Isaksson, T. Næs and A. Tandberg. Ellis Horwood, pp. 79–84 (1992).
7. E. Wüst, H. Neemann and L. Rudzik, in *Near Infrared Spectroscopy. Bridging the Gap between Data Analysis and NIR Applications*, Ed by K.I. Hildrum, T. Isaksson, T. Næs and A. Tandberg. Ellis Horwood, pp. 67–72 (1992).
8. E. De Pedro, A. Garrido, I. Bares, M. Casillas and I. Murray, in *Near Infrared Spectroscopy. Bridging the Gap between Data Analysis and NIR Applications*, Ed by K.I. Hildrum, T. Isaksson, T. Næs and A. Tandberg. Ellis Horwood, pp. 345–348 (1992).
9. E. De Pedro, A. Garrido, A. Lobo, P. Dardenne and I. Murray, in *Leaping Ahead*, Ed by G.D. Batten, P.C. Flinn, L.A. Welsh and A.B. Blakeney. Royal Australian Chemical Institute, Victoria, Australia, pp. 291–295 (1995).
10. ISI. Infra Soft International, Port Matilda, PA, USA.

11. J. Zupan and J. Gasteiger, *Neural Networks for Chemists, An Introduction*. VCH, Weinheim (1993).
12. J.S. Shenk and M.O. Westerhaus, *Crop Sci.* **31**, 469 (1991).
13. J.S. Shenk and M.O. Westerhaus, *Crop Sci.* **31**, 1548 (1991).
14. SAS User's Guide Statistics, SAS Institute, Inc., Cary, NC (1991).
15. Neural Networks Professional II. Neural Ware, Inc., PA, USA.
16. T. Kohonen, *Self-Organization and Associative Memory, 2nd edition*. Springer-Verlag, New York, (1988).
17. D. DeSieno, in *Proc. Int. Conf. on Neural Networks*. Vol 1. IEEE Press, New York, pp. 117–124 (1988).
18. A.A. Minai and R.D. Williams, in *International Joint Conference on Neural Networks*. Vol. 1, IEEE, pp. 676–679 (1990).
19. N. Baba, *Neural Networks* **2**, 367 (1984).
20. J. Hertz, A. Krogh and R.G. Palmer, *Introduction to the Theory of Neural Computation*. Addison–Wesley, Redwood City, California (1991).
21. L. Moya, A. Garrido, J.E. Guerrero, L. Lizaso and A. Gómez, in *Leaping Ahead*, Ed by G.D. Batten, P.C. Flinn, L.A. Welsh and A.B. Blakeney. Royal Australian Chemical Institute, Victoria, Australia, pp. 111–116 (1995).
22. C. Cachin, *Neural Networks* **7**, 175 (1994).
23. S. Becker and Y. Le Cun, in *Proceedings of the 1988 Connectionist Models Summer School*, Ed by D. Touretzky, G. Hinton, and T. Sejnowsky. Morgan Kaufman, San Mateo, California, pp. 29–37 (1989).
24. N.B. Karayiannins and A.N. Venetsanopoulos, *Artificial Neural Networks*. Kluwer Academic, Boston (1993).

Received: 16 May 1995

Accepted: 19 January 1996