Mapping Analytic Functions Using Neural Networks

Frank R. Burden

Chemistry Department, Monash University, Clayton, VIC 3168, Australia

Received January 4, 1994[⊗]

The practical limitations of the use of neural networks as model-free mapping devices are illustrated by application to Gaussian functions and benzene spectral data.

INTRODUCTION

Artificial neural networks (ANNs) have been used for the analysis of a variety of chemical data. They have been used for the three main purposes of classifying patterns, including mass-spectral data,[1] correlating chemical data with biological activity,[2] and for fitting continuous functions.[3] It is this last use that is addressed here with a particular view to the automatic fitting of continuous spectral functions making use of a minimum number of parameters. ANNs are commonly used to classify patterns in many other life situations such as recognizing human faces in a crowd or the letters of the alphabet in a document. In such cases the ANN is trained on a large number of given patterns which are presented to the network in the expectation that the network will correctly classify an indistinct pattern, much as the human eye can distinguish one alphabetic character from another even when the character is presented at an angle or is blurred in some manner. It is an attempt to mimic the ability of the human mind to quickly process visual information by emulating the processing ability of neurons. In the second case, the attempt to discover quantitative structure–activity relationships (QSAR) between chemical structures, or physiochemical measurements, and biological activity, there is still an element of pattern matching with an overlay of quantitative correlation. The recent use of continuous functions to represent the neuronic activation function has enabled the quantitative aspect to be introduced with some success especially where the results were not expected to be accurate to better than about 5% and the functional relationships were of low order, perhaps only weakly quadratic.

The further attempts to use ANNs to quantitatively represent more complex functions is founded on the Kolmorgorov–Sprecher theorem,[4,5] as interpreted by Hecht-Neilsen,[6] which indicates that any real continuous function can be exactly represented by an ANN with two hidden layers and a finite number of neurons. The theorem is of theoretical interest in that no method for constructing the activation function is given; however, Irie–Miyake[7] has proven that an infinite three-layered network can represent an arbitrary function provided that the hidden-layer activation functions and the mapping function are bounded and absolutely integrable.

There are some attempts in the chemical literature to fit spectral lines and simple functional forms with ANNs and with some claim to success.[8,9] On further investigation it is found that the root-mean-square error [RMS] is usually no better than 2–3% of the function maximum[10] which may be satisfactory in many situations but not in others, such as the potential functions of small molecules where 0.1% or better may be needed if spectral transitions are to be calculated from the function.

This limit of precision in representing simple analytic functions is shown here and which can, by inference, be seen as reducing the utility of the Kolmogorov–Sprecher theorem for a greater number of dimensions.

THEORY

Descriptions of artificial neural networks are given in an abundance of texts.[11] The ANNs used for mapping continuous functions usually have the form of a fully-connected feed-forward network with an input and an output layer together with one or two hidden layers and bias nodes in each layer, other than the output layer.

The training was carried out by presenting the network with input data which was randomly chosen from the input data set. Training was halted when the RMS was reducing at a rate less than 1×10^{-5} per cycle through the data or was oscillating. The potential problem of overtraining, which occurs when the number of weights is excessive, was anticipated by using a large training set and a separate testing set.

There are several parameters that must be chosen at the outset of the calculation.

(a) **The Architecture of the Network.** For function mapping this will depend on the complexity and dimension of the data set. Commonly the network will have one or two hidden layers, and the number of neurons in each layer determined experimentally by reference to the RMS. For the QSAR type of problem where there may be many inputs being mapped to one or two outputs representing biological activity, the architecture must be chosen with care to avoid overtraining, when the network effectively memorizes the training data. Rules for QSAR problems are discussed by Manallak and Livingstone[12] who use the parameter ϱ = no. of datapoints/no. of connections to select optimum network learning conditions. They suggest that ϱ should be greater than 2 and that a separate training and testing set be used. Lower values of ϱ are likely to cause the network to "memorize" the data rather than learn a general rule.

(b) **The Activation Function for the Neurons.** Although the sigmoid function $S(x) = \{1/(1 + \exp(-x))\}$ is commonly used, there are other useful functions which can be tried such as the hyperbolic tangent, $\tanh(x)$, or

[⊗] Abstract published in *Advance ACS Abstracts,* September 15, 1994.

the simple limiter function, *Tri*(x) = 1 − *abs*(x); $-1 < x < 1$.

(c) **Learning Rate and Momentum.** The learning rate, η, is used to dampen the rate of change of the weights from iteration to iteration, whereas the momentum, α, adds in a small amount of the weight change from the previous presentation to keep the direction of change from varying too fast. These parameters are often varied automatically as the training progresses, starting with initial small learning rates.

Since neural networks are used in chemistry of pattern matching of infrared and NMR data,[13] for QSAR work,[14] and for function mapping,[9] it is not surprising that different architectures and activation functions might be needed for the different circumstances.

Even in the area of function mapping, there is a variety of common functions that can be mapped such as simple monatonic curves, curves with a few oscillations, and spectral traces which are approximately the sum of Gaussian or Lorentzian shaped absorption peaks such as those that occur in the UV or infrared regions of the spectrum. As will be shown, the spectral traces can be mapped more precisely when the neuron activation function has a similar shape to the spectral absorptions than when a sigmoid function is used.

There is some discussion in the literature on the use of Gaussian activation functions, $e^{-\alpha(x-x_0)^2}$. However these have generally been used with a different architecture to those discussed here[15] where the parameters of the Gaussian are predetermined according to the problem at hand, and there is no processing in the output layer. Figure 1 shows a plot of the sigmoid function $S(x) = 1/(1 + \exp(-x))$ together with its first and second derivatives, S' *and* S'' where

$$S'(x) = \frac{dS(x)}{dx} = S(x)(1 - S(x)) \qquad (1)$$

$$S''(x) = \frac{d^2S(x)}{dx^2} = S(x)(1 - S(x))(1 - 2S(x)) \qquad (2)$$

and $S'(x)$ has a similar shape to a Gaussian or Lorentzian function. $S'(x)$ can easily be incorporated into a back-propagation network by merely replacing $S(x)$ by $S'(x)$ and using $S''(x)$ in the back-propagation algorithm.

RESULTS

The results presented here show how the use of $S'(x)$ instead of $S(x)$ can improve the functional fit by an ANN by a factor of 2 to 3 as well as comparing the performance of ANNs with polynomial fits which have a similar number of coefficients to be evaluated. The data points are taken to be of similar density to that obtained from experimental spectral data which in the case of the benzene peaks is 2 cm^{-1} separation. This limits the size of the network to 1:5:5:1 if ϱ is to exceed 2 in the examples presented.

Fitting a Two-Dimensional Gaussian. The training and testing data sets were generated from the Gaussian function $0.1 + 0.8e^{-20(x-0.5)^2}$ by randomly selecting 100 values of x from the range 0−1. Two network architectures were used both for the sigmoid function, $S(x)$, and the sigmoid derivative function, $S'(x)$. Some results are shown in Table 1 from where it can be seen that the RMS error is lower for the derivative function in the 1:4:1 case but equivalent at 0.001, or 0.2% of the data mean, for the 1:5:5:1 case. The RMSs of some polynomial fits, $y = {}^{n}\sum_{i=0} a_i x^i$, were evaluated, and it was found that a polynomial of order 9, that is with 10 parameters, gives a similar fit to a 1:4:1 network with (2 × 4) + (5 × 1) = 13 parameters and a polynomial of order 11 has a lower RMS than a 1:5:5:1 network with (2 × 5) + (6 × 5) + (6 × 1) = 46 parameters which implies that in this case the ANN is, at least, inefficient in its use of parameters and may imply some interdependence between them.

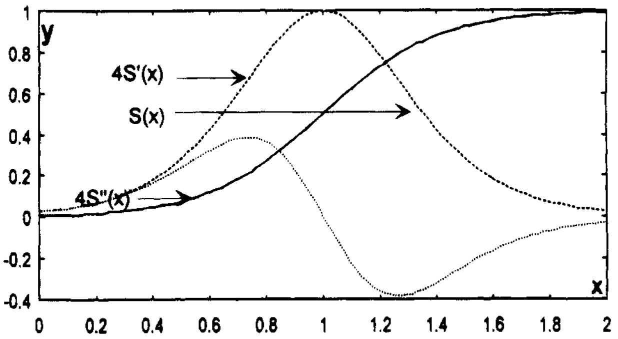

Figure 1. The sigmoid function and derivatives.

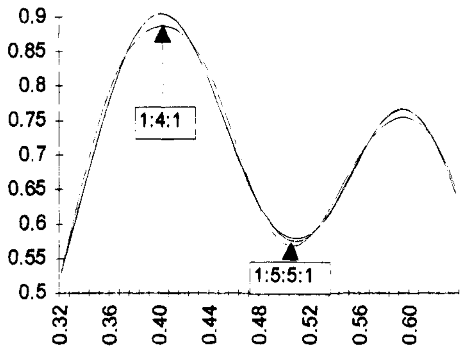

Figure 2. Fitting around the maxima of the Gaussian function $0.1 + 0.8e^{-100(x-0.4)^2} + 0.65e^{-125(x-0.6)^2}$.

Fitting Two Gaussian Functions. The function used was $0.1 + 0.8e^{-100(x-0.4)^2} + 0.65e^{-125(x-0.6)^2}$ and treated as above. There are now two centers about which the function must be fitted, and it is the purpose of the weights associated with the bias nodes to achieve the centering of the activation functions, whereas the other weights control the width. The results shown in Table 1 show that the 1:4:1 network with the sigmoid activation function has an RMS of 0.071, which is 12 times larger than for the single Gaussian. However there is an improvement by a factor of 10 to 0.008, or 1.6% of the data mean, in going from the 1:4:1 network to the 1:5:5:1 network and a further gain of 2.5 in going to the $S'(x)$ activation function. Figure 2 shows plots of the fitted data around the Gaussian maxima where there is a large curvature and the worst fit and is typical in that the network function is of lower curvature.

Fitting the Sigmoid Derivative, $S'(x)$. Where an attempt is made to fit the sigmoid derivative function of eq 2, it might be expected that a simple network using $S'(x)$ itself as the neuron activation function would produce a near perfect fit. Of course, at the limit of using a single neuron with only two weights (parameters) to be determined then a perfect fit will be obtained. However, if the output layer neuron also incorporates an activation function then similar results for fitting a Gaussian are achieved which shows that it is not

Table 1. Root-Mean-Square Error for the Fitting of Various Functions

architecture	1:4:1	1:4:1	1:5:5:1	1:5:5:1
no. of weights, including bias	13	13	46	46
activation function	$S(x)$	$S'(x)$	$S(x)$	$S'(x)$
1 Gaussian	0.006	0.003	0.001	0.001
2 Gaussians	0.071	0.01	0.008	0.003
2 benzene peaks	nonconvergence	nonconvergence	0.010	0.006

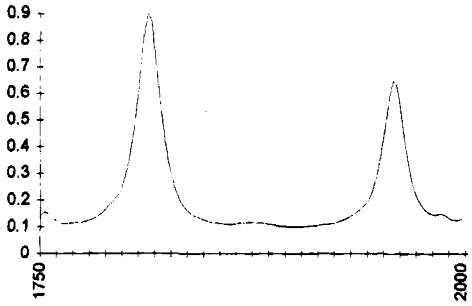

Figure 3. The benzene infrared spectrum in the range 1750–2000 cm^{-1}.

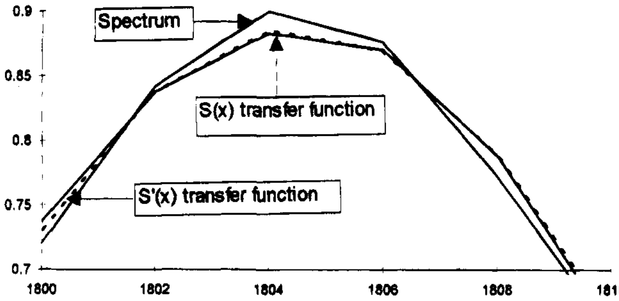

Figure 4. Fitting of the benzene peak at 1804 cm^{-1} around its maximum.

sufficient to have a match between the function to be fitted and the activation function in order to achieve a precise fit. If a major improvement in the fit cannot be obtained in this case, then it is probably fruitless to search for a better activation function in the general case.

Fitting Absorption Peaks from the Infrared Spectrum of Benzene. Figure 3 shows two of the infrared absorbance peaks of benzene, at 1804 and 1951 cm^{-1}, taken by an FTIR spectrometer at 4 cm^{-1} resolution and which could be closely fitted with two Lorentzian functions centered at the midpoint of each peak. Figure 4 again shows the inadequate fit near a peak maximum where there is high curvature. Any attempt to fit more complex areas of the spectrum merely produces results of far less precision.

CONCLUSION

The expectation that artificial neural networks can be used to provide precise model-free mapping of complex functions is not realized in practice. The size of the network is limited by the number of data points available and by the need to avoid overtraining caused by an excess of weights. In the case where the function to be fitted is representable by known analytic functions then the ANN can be made more efficient by choosing an appropriate activation function for the neurons. However, in such cases, the ANN is unlikely to compete with a direct fitting procedure using either the relevant analytic function or a spline method, which can achieve a very close fit though at the expense of using a set of at least three polynomial coefficients for every data point. It would seem that ANNs are best suited to fitting functions of low order in a multidimensional space such as occurs in QSAR or for pattern recognition problems. As the simple examples illustrated show, ANNs are not suitable where the functions, such as potential functions for predicting spectral transition frequencies, need to be represented to 0.1% or better across the range.

REFERENCES AND NOTES

(1) Lohninger, H.; Stancl, F. Comparing the performance of neural networks to well-established methods of multivariate data analysis: the classification of mass spectral data. *Fresenius J. Anal. Chem.* **1992**, *344*, 186–189.

(2) Salt, D. W.; Yildiz, N.; Livingstone, D. J.; Tinsley, C. J. The use of artificial neural networks in QSAR. *Pesticide Sci.* **1992**, *36*, 161–170.

(3) Kosko, B. *Neural Networks in Signal Processing*; Prentice Hall: New Jersey, 1992.

(4) Kolmogorov, A. N. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. *Doklady Kademii Nauk SR* **1957**, *144*, 679–681.

(5) Sprecher, D. A. On the structure of continuous functions of several variables. *Transl. Am. Math. Soc.* **1965**, *115*, 340, 355.

(6) Hecht-Nielsen, R. Kolmogorov mapping neural network existence theorem. *IEEE ICNN* **1987**, *3*, 11–13.

(7) Irie, B.; Miyake, S. Capabilities of three-layered Perceptrons. *IEEE ICNN* **1988**, *1*, 641–648.

(8) Salt, D. W.; Yildiz, N.; Livingstone, D. J.; Tinsley, C. J. The Use of Artificial Neural Networks in QSAR. *Pesticide Sci.* **1992**, *36*, 161–170.

(9) Bishop, C. M.; Roach, C. M. Fast Curve Fitting using Neural Networks. *Rev. Sci. Instrum.* **1992**, *63*, 4450–4456.

(10) Maggiora, G. M.; Elrod, D. W.; Trenary, R. G. Computational Neural Nets as Model-Free Mapping Devices. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 732–741.

(11) Muller, B. *Neural Networks: an introduction.* Springer-Verlag: New York, 1995.

(12) Manallak, D. T.; Livingstone, D. J. Artificial Neural Networks: Applications and Chance Effects for QSAR data analysis. *Med. Chem. Res.* **1992**, 2, 181–190.

(13) Ball, J. W.; Jurs, P. C. Automated Selection of Regression Models Using Neural Networks for ^{13}C Spectral Predictions. *Anal. Chem.* **1993**, *65*, 505–512.

(14) Andrea, T. A.; Kalayeh, H. Applications of Neural Networks in Quantitative Structure–Activity Relationships of Dihydrofolate Reductase Inhibitors. *J. Med. Chem.* **1991**, *34*, 2824–2836.

(15) Wasserman, P. D. *Advanced Methods in Neural Computing*; Van Nostrand Reinhold: New York, 1992; Chapter 8.