Feed-forward artificial neural networks: applications to spectroscopy

Dragan A. Cirovic
Bristol, UK

Applications of multi-layer feed-forward artificial neural networks (ANN) to spectroscopy are reviewed. Network architecture and training algorithms are discussed. Backpropagation, the most commonly used training algorithm, is analyzed in greater detail. The following types of applications are considered: data reduction by means of neural networks, pattern recognition, multivariate regression, robust regression, and handling of instrumental drifts.

1. Introduction

Artificial neural networks (ANN) are mathematical models comprised of highly inter-connected processing units, neurones. The theoretical bases of these systems were originally developed within neuroscience. They were subsequently embraced by other branches of science in an attempt to transfer rules governing natural intelligent systems into the sphere of artificial intelligence, in the first instance, and later into many others. The forms of exploitation of natural cognition principles by diverse branches of science often bear very little resemblance to their native counterparts. One of the most important concepts borrowed from the natural cognition research is that of parallel distributed processing (PDP). The PDP is a method of storing and retrieving information from the system which consists of a large number of very simple processing units, with an exceptionally high level of interconnection between the units. Such a system possesses the ability to adjust the strength of connections between its units during the learning stage. The connection strengths (network weights) are means of capturing information. The stored data can be retrieved subsequently by presenting the trained system with the input stimuli. The form of knowledge gathered by neural nets is fundamentally different from that used conventionally by computers, where information is stored in a sequential manner. The PDP systems generalize data, in the sense that they do not allow one-to-one correspondence between physical memory and information units. This, in turn, makes their information-retrieval times much shorter, and their physical storage requirements much smaller, than those offered by sequential systems.

The initial experiments with ANNs, which have more than a half-century-long tradition, did not offer much encouragement. This was owing partly to the lack of theoretical foundations, and partly to the inability of the technology of the time to cope with the high computational requirements of ANNs. The more recent advances in information technology coincide with an increase of interest in neural networks, and the last decade can be labelled the most fruitful period in the field. One of the first waves of theoretical advances in the area was reviewed in the book edited by Rumelhart and McClelland [1]. Since then, ANNs have proved to be useful in a number of fields, including pattern recognition and classification, function approximation, non-linear regression analysis, clustering, combinatorial optimization, robot control, and image compression. The width of scope of applications of ANN comes from their ability to approximate complex functions, which makes them well-suited for modelling non-linear relationships.

The developments in recent years have shown increasing growing interest in ANNs among chemists. This process has been catalyzed by a number of excellent tutorials and reviews [2-9]. The most often used network type was the multi-layer feed-forward (MLF) ANN with backpropagation training. It is thought that 90% or more of all applications of ANNs have utilized the feed-forward architecture. The backpropagation training procedure is so common for feed-forward networks that the term 'backpropagation' is regularly used to designate both the network architecture and the training procedure. The range of chemical applications of MLF ANN is very large. It includes helds as diverse as:
modelling of secondary molecular structure of proteins and DNA, molecular dynamics, quantitative structure-activity relationships (QSAR), quantitative structure-property relationships (QSPR), interpretation of spectra, calibration, and process control. In this article, precedence is given to the spectroscopic applications.

2. Theoretical foundations

MLF networks belong to a class of supervised learning nets. They require a training data set composed of input/output data pairs. For example, in multivariate calibration one may want to establish a relationship between IR spectra and the corresponding concentration profiles of chemical mixtures. Each sample is characterized by a spectrum (network input) and a concentration profile (the desired output). The inputs are presented to the net, one at a time, usually in random order, and the network outputs are compared to the target outputs in order to adjust network weights. One epoch involves the presentation of all samples to the network. Network training consists of performing a series of epochs until a certain convergence criterion is satisfied.

2.1. Network architecture

A MLF ANN consists of an input layer, an output layer, and a variable number of hidden layers. The input layer is not normally counted, because it is only formally present, in the sense that it does not do any processing; for example, a two-layer network consists of the input, hidden, and output layers (Fig. 1). All connections between the layers are allowed. Connections between the nodes of the same layer, as well as the auto-connections (loops), are prohibited. The connections between non-consecutive layers are called 'directed connections'. They are symbolically presented with bold arrows, as in Fig. 2. Although the recursive connections are feasible, they require adjustments to be made to the training algorithm. The geometrical disposition of neurons within a layer is an application-specific issue. In spectroscopic applications, layers usually have a linear topology.

The optimal network architecture depends on the complexity of the modelled relationship. Generally, error surfaces encountered in quantitative analysis are less complex than those used in qualitative analysis. The methods for determination of the optimal number of hidden layers can be divided into two groups. One of them starts from a very simple network and gradually increases the network complexity until the optimal architecture is found. The methods from the other group attempt to reach the optimum by moving in the opposite direction, i.e. they start from a large network and monitor the effects of reducing the number of hidden layers and connections between the nodes. In either case, a vast amount of redundant computation has to be performed in order to establish the optimum. However, for many practical applications such an extensive search for the optimal network architecture is not justifiable, and an empirical rule which says that the number of adjustable parameters should be approximately a half of the number of samples, can be used instead.

2.2. Network layers

In this article, network layers are denoted by the superscripts $p = 0, ..., N$ (see Fig. 3 and Table 1). The desired output is denoted by the superscript $d$. The $p$th layer's input, $\mathbf{x}^p$, and output, $\mathbf{y}^p$, are vectors with $m^p$ elements each, i.e.
The two successive layers, \((p-1)th\) and \(pth\) respectively, are related by means of a weight matrix

\[
\mathbf{W}^p = \begin{pmatrix} \mathbf{w}^p_1 & \cdots & \mathbf{w}^p_{m^p-1} \\ \vdots & \ddots & \vdots \\ \mathbf{w}^p_{m^p-1} & \cdots & \mathbf{w}^p_m \end{pmatrix}
\]

(2)

The input to the \(pth\) layer is the sum of weighted outputs from the previous layer

\[
\mathbf{y}^p = \mathbf{W}^p \mathbf{y}^{p-1}
\]

(3)

The output of the \(pth\) layer lies in a domain of the corresponding transformation function

\[
\mathbf{y}^p = f(\mathbf{y}^p)
\]

(4)

The simplest form of transformation function is the so-called threshold logic function

\[
f(\mathbf{y}^p) = \max\{0, \min(\mathbf{y}^p, 1)\}
\]

(5)

A limitation of this function is that it is not differentiable in its whole domain, which makes it inaccessible to gradient-based training methods. The continuous functions with a similar shape, such as sigmoidal (or logistic) and tanh are used instead (see Fig. 4). These functions are particularly convenient because their derivatives are very simple. The choice between the two functions is basically only dependent on the desired output range.

The adaptive ability of the network is greatly enhanced by the use of a constant term referred to as a threshold. This term causes a horizontal shift of the transformation curve, i.e. the threshold of the graphs in Fig. 4 is zero. A common way of implementing thresholds is to add to each network layer an extra node with a constant value of one (see Fig. 1. and 2).

2.3. Training algorithm

The most commonly used training algorithm, the backpropagation, is the first-order gradient method. It is known as a steepest decent optimization. Its object or error function, \(E\), is for the sample-wise minimization defined as a sum of squared residuals between the desired and the actual network output

\[
E(\mathbf{w}) = \sum_{i=1}^{m_N} (\mathbf{y}^d_i - \mathbf{y}^N_i)^2
\]

(6)

For the case of the batch (epoch-wise) minimization Eq. (6) is easily transformed into an average error for all samples. Eq. (6) employs a notational simplification which identifies a set of weight matrices \(\mathbf{W}^p (p = 1, \ldots, N)\) with a single weight vector, \(\mathbf{w}\), consisting of all weight elements, i.e.

\[
E(\mathbf{W}^1, \ldots, \mathbf{W}^N) = E(\mathbf{w})
\]

The backpropagation training is a process of step-wise modification of weights in the direction of the maximal gradient of object function, which is defined as follows:

\[
\Delta \mathbf{w}^p_i = -\eta \frac{\partial E(\mathbf{w})}{\partial \mathbf{w}^p_i}
\]

(7)

The learning rate, \(\eta\), is a positive real number. A high value for \(\eta\) accelerates training, but can easily lead to oscillations around the minimum. However, setting \(\eta\) to some low value causes an increase in the number of training epochs. The optimal value

### Table 1

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\sigma^p)</td>
<td>(pth) layer's input</td>
</tr>
<tr>
<td>(\mathbf{y}^p)</td>
<td>(pth) layer's output</td>
</tr>
<tr>
<td>(\mathbf{W}^p)</td>
<td>(pth) weight matrix</td>
</tr>
<tr>
<td>(m^p)</td>
<td>Number of elements in the (pth) layer</td>
</tr>
<tr>
<td>(E(\mathbf{w}))</td>
<td>Network error function</td>
</tr>
<tr>
<td>(\mathbf{w})</td>
<td>Vector of all network weights</td>
</tr>
<tr>
<td>(\Delta \mathbf{w}^p_i)</td>
<td>Gradient of the (ith) element for the (pth) layer</td>
</tr>
<tr>
<td>(\eta)</td>
<td>Learning rate</td>
</tr>
<tr>
<td>(\alpha)</td>
<td>Relative weighting of the momentum term</td>
</tr>
<tr>
<td>(\delta^p_i)</td>
<td>Delta factor for the (ith) node of the (pth) layer</td>
</tr>
<tr>
<td>(\mathbf{M}^p)</td>
<td>(pth) momentum matrix</td>
</tr>
</tbody>
</table>
of $\eta$ is a compromise between the two. Its actual value depends on the scale of the training set.

The analytical form of the gradient can be obtained by multiple application of the chain rule on error function. The elements of the gradient vector are partial derivatives of $E(w)$ in respect to the weight elements ($w_{ij}^p$). The computational requirements for the evaluation of a gradient element are proportional to the distance of the corresponding weight from the output layer. It can be shown that the gradient can be expressed in terms of a delta factor,

$$\delta_i^p = \frac{\partial y_i^p}{\partial w_{ij}^p} \sum_{j=1}^{m} \delta_j^{p+1} w_{ji}^{p+1},$$

which is independent of the actual position of the layer in the net. This rule is known as the generalized delta rule [1,2,4–6].

Backpropagation in its elementary form suffers from a very slow convergence rate, and it has high likelihood of oscillating around the minimum. In order to overcome these shortcomings, Rumelhart and McClelland [11] introduced a gradient momentum term into Eq. (7). The momentum is a weight change from the previous iteration, i.e. $M^p = \text{old} \Delta W^p$. One way of incorporating the momentum term into the backpropagation algorithm is as follows:

$new \ W^p = \text{old} \ W^p + \eta (1 - \alpha) \Delta W^p + \alpha M^p \quad (8)$

The relative weighting of the momentum term, $\alpha$, is a constant whose value is in the range 0–1. The effect of the momentum is to drive the response surface out of the local minimum extreme.

A further layer of sophistication of the backpropagation algorithm is to replace a single pair of learning/momentum coefficients by a set of weight-specific pairs of coefficients:

$$\Delta w_{ij}^p = -\eta_{ij}^p (1 - \alpha_{ij}^p) \frac{\partial E(w)}{\partial w_{ij}^p} + \alpha_{ij}^p M_{ij}^p \quad (9)$$

The self-adjustment is a preferred way of handling these coefficients. The general trend is to keep them proportional to their corresponding gradients. A number of different coefficient adjustment schemes has been proposed. The most widely used one, the so-called Delta-bar-delta rule [10,11], is based on a product $\Delta \bar{W}_{k-1} \Delta W$, where the term $\Delta \bar{W}_{k-1}$ is a weighted average gradient of the previous two iterations.

Although backpropagation is the most commonly used training method there are a number of alternatives. They range from the gradient-based methods such as the Kalman filters [10,11] to the gradient independent methods which utilize a genetic algorithm or simulated annealing. The main asset of gradient-based methods is their relatively short training time. However, gradient-independent methods have the advantage of being less likely to fall into a local minimum and the ability to model networks with discontinuous transfer functions.

The problem of the duration of network training is independent of the type of training algorithm. It is desired to achieve a balance between the under- and over-fitting of the network. This is usually done by monitoring the mean square error (MSE) of an...
independent validation data set during training. The point where this MSE curve reaches the minimum is assumed to correspond to an optimally trained network and any further reduction of the MSE accumulated in the training set would be regarded as over-fitting or memorizing of the training data set.

3. Applications

3.1. Data reduction

Modern analytical instruments are capable of quickly generating huge data sets. The various types of spectrometers are typical of such instruments. The acquired data are regularly characterized by a high level of redundancy. This results from the high correlation between the variables, e.g., the correlation between consecutive UV/Vis spectral channels in peak areas is close to one. In order to enhance the interpretability, to diminish the impact of noise, and reduce data storage requirements, the raw data are processed by dimension reduction techniques. The common ways of performing data reduction are principal component analysis (PCA) or Fourier transformation [12-15]. The omnipresence of these methods in chemical data processing does not qualify them as universal remedies. It is often the case that the direction of the maximal variance is not the most relevant one, or it may not be appropriate to project the data onto a set of mutually orthogonal functions. The linear data analysis overcomes these limitations by projecting data onto the directions of maximal covariance between the considered data sets partial least squares (PLS) [15-18]. This approach is easily extendable to a non-linear domain, e.g., polynomial or spline PLS [19,20].

In an attempt to merge the underlying principles of PLS and MLF ANNs with backpropagation training Wythoff [21] has proposed a new method for stepwise orthogonal decomposition. The method is called the Orthonet. It divides the network into two parts: the linear orthogonal feature (factor) extraction portion and the non-linear mapping portion. The first part consists of the input layer and the first hidden layer, while the second part comprises the remaining layers (see Fig. 5). The nodes of the first hidden layer have a linear transformation function. The weights between the input layer and the first hidden layer correspond to what is known as the 'loadings' in PLS. The network training procedure evaluates factors sequentially, one at a time. After the convergence, the factor is subtracted from the data set, which ensures their mutual orthogonality. During the training stage there is always only one feature node in the first hidden layer. The non-linear mapping portion of the network acts much like an ordinary feed-forward network. The trained network consists of all extracted factors, e.g., a three factor network has three nodes in the first hidden layer.

There are numerous alternatives to the Orthonet algorithm [21]. Some of them involve the simultaneous extraction of the orthogonal vectors. The main disadvantage of these implementations is that whenever one factor is modified the remaining ones have to be orthogonalized, which is easily achievable by means of the Gram-Schmidt or some other orthogonalization method. The drawback is that these orthogonalizations inevitably introduce undesired changes of the feature vector directions.

3.2. Interpretation of spectra

Interpretations of spectra belong to the pattern recognition class of ANN applications. The aim is to relate the spectra and corresponding chemical structures. For the case of IR spectra [22-24] the molecular structures are considered in terms of their functional groups. The presence or absence of functional groups is indicated by means of a certain coding scheme. The main problem in these applications is a disproportion between the abundances of functional groups, e.g., the abundance of CH and OH in a collection of organic compounds is likely to highly exceed that of the majority of other groups. A common solution is to treat different functional groups as Bayesian classes and to use their respective class frequencies as a weighting criterion [6]. An additional improvement in prediction rates can be achieved by utilizing networks based on orthogonal projections [21]. This results from the fact that many functional groups may account for only a relatively small percentage of total data variance.

The pattern recognition method proposed by Smits et al. [24] is aimed at dealing with huge databases of chemical components. This approach takes advantage of the observation that a system of small networks (modules), which are dedicated to specific classes of chemical components, is likely to be more efficient than a single network. In
3.3. Calibration

The relationship between backpropagation ANN and classical regression is neatly defined in the following quotations from Ref. [6]: "While classical regression begins with the assignment of an explicit model generated using analytical knowledge of the system under study, backpropagation develops an implicit model. The form of this model is constrained only by the bonds on the set of all functions that the chosen architecture can implement. Therefore, backpropagation learning can be considered to be a generalisation of classical regression". At the same time, it is important to realise that "Neural networks should only be used when the equations describing the variation in a system are unknown, or cannot be solved. Attempts to do otherwise will produce a solution which can at best match the predictive performance of the analytical solution on unknown inputs, and will probably be worse". Another important point to be made is that the extrapolation ability of ANN models is very low. For that reason, a trained network should not be used for making predictions outside the range spanned by the training set.

The early calibration work [25,26] was mainly concerned with the elucidation of the extent to which ANN regressors are affected by different levels of noise in predictors and responses and with defining their efficiency in modelling non-linearity in UV/Vis spectra. The analyzed sources of non-linearity were non-linear instrumental responses, concentration-dependent wavelength shifts, and absorption bandwidth changes. There have been a few attempts [27,28] to take advantage of network architectures with direct connections, or to minimize network size by eliminating less important connections [27]. The ANNs were shown to be efficient in modelling mass spectrometry data [29]. In this case, the significant gains in network training speed were achieved by discarding low intensity masses without sacrificing the prediction accuracy of the models.

3.4. Instrumental drifts

Spectroscopic chemical data analysis is seriously affected by instrumental drifts. These are defined as differences between the spectra of the sample, recorded on the same instrument over a period of time, which cannot be eliminated by instrumental standardization. In pattern recognition, instrumental drifts make classes lying close to each other hard to distinguish [30], while in regression analysis analysis drifts manifest as bias. Recently, a neural network was employed in correction of pyrolysis mass spectrometer drifts. This method is purported to be robust on outliers, able to perform non-linear mapping of one spectrum onto another and to be applicable to other types of spectroscopy (i.e. IR, ESR and NMR) [31].

3.5. Robust methods

The least-squares criterion is at the core of the backpropagation training procedure (Eq. (6)). Consequently, MLF ANNs are affected by the presence of outliers in a data set as much as any of the classical least-squares-based regression methods. Robust regression techniques are chemometric methods dedicated to addressing these problems. They generally take one of two distinct routes: (1) outlier detection and elimination from the set or (2) robust regression. While the former is still mainly in the sphere of classical statistical analysis, the latter has been a focus of active ANN research [32–34]. In essence, the robust regression replaces the vulnerable least-squares method with some other regression method which is less susceptible to the presence of erroneous measurements. Many of these methods are founded on the concept of the median. The main difficulty associated with robust regression methods is that they are very computationally intensive [35].

3.6. Alternative approaches

An interesting alternative customization of feed-forward ANN was proposed by Frank [36]. Her method is termed the Neural Network based on PCR and PLS components non-linearized by Smoothers and Splines (NNPPSS). The NNPPSS is aimed at overcoming limitations of ordinary feed-forward ANN (both network architecture and training procedure) when it is used as a calibration tool on data sets having a low samples-to-
variable ratio. This method utilizes two-layer feed-forward network architecture. In contrast to the backpropagation training, it uses a fixed set of weights, and searches for the optimal number of hidden nodes and their optimal transformation function. The method uses either adaptive window smoothers or adaptive splines as its transformation functions. This choice of transformation function makes the method more robust to outliers and it is less affected by noise. In the first step, a set of network-weight vectors is calculated by means of some linear regression method [PLS or principal component regression (PCR)]. The algorithm then performs a non-linear fitting on a number of different subsets of weights vectors. The weight subset with the highest $R^2$ is accepted as the final solution.

4. Final remarks

MLF ANNs are well suited for the modelling of non-linear relationships and they are likely to be the methods of choice when a mathematical model is not available or cannot be obtained for the problem under consideration. However, there is no advantage in applying ANN-based techniques to problems which are well modelled by classical mathematical and statistical tools.

5. Glossary

*Artificial neural network (ANN).* The mathematical model consisting of simple inter-connected processing units which are highly inter-connected and are capable of storing data in a distributed form by means of network weights.

*Backpropagation.* The first-order gradient optimization method used for ANN training.

*Hidden layer.* The network layer between the input- and hidden layer.

*Input layer.* The temporary place-holder of the currently processed sample.

*Multi-layer feed-forward (MLF) ANN.* The type of network architecture which has neurones organized as a series of layers.

*Network epoch.* The sequence of sample presentations to the network which include all samples in the data set.

*Outlier.* The sample in a data set which has a much higher error level than the remaining samples

*Output layer.* The network layer which performs the final processing.

*Parallel distributed processing (PDP).* The approach to information management which stores and retrieves data by means of network connections.

Acknowledgements

The author would like to thank Health and Safety Executives (United Kingdom) for providing financial support. Professor Erno Pretsch (Zurich) is gratefully acknowledged for helpful suggestions.

References

A graphical criterion to examine the quality of multicomponent analysis
Implications for wavelength selection

Joan Ferré*, F. Xavier Rius
Tarragona, Spain

A graphical criterion approach has been developed to examine the quality of a set of sensors in multicomponent analysis. Criteria such as sensitivity and selectivity, used in wavelength selection problems, can be explained in terms of the confidence interval of the estimated concentrations. These confidence intervals describe (hyper)ellipsoids whose volume, shape and orientation are related to the optimization criteria. The effect of sensor selection on these criteria is discussed and guidelines for wavelength selection are given. The usefulness of the graphical criterion is shown in the simultaneous determination of 2-chlorophenol and 2,4-dichlorophenol in water.

1. Introduction

Spectroscopic multicomponent analysis consists of determining the concentrations of the K compo-