Pergamon

Atmospheric Environment Vol. 32, No. 14/15, pp. 2627–2636, 1998

Printed in Great Britain

PII: S1352–2310(97)00447–0

1352–2310/98 $19.00 + 0.00

ARTIFICIAL NEURAL NETWORKS (THE MULTILAYER PERCEPTRON)—A REVIEW OF APPLICATIONS IN THE ATMOSPHERIC SCIENCES

M. W. GARDNER* and S. R. DORLING

School of Environmental Sciences, University of East Anglia, Norwich, Norfolk NR4 7TJ, UK

(*First received* 20 *February* 1997 *and in final form* 4 *September* 1997. *Published June* 1998)

Abstract—Artificial neural networks are appearing as useful alternatives to traditional statistical modelling techniques in many scientific disciplines. This paper presents a general introduction and discussion of recent applications of the multilayer perceptron, one type of artificial neural network, in the atmospheric sciences.

Key word index: Statistical modelling, neural network, backpropagation, artificial intelligence.

1. INTRODUCTION

Neural networks, or more precisely artificial neural networks, are a branch of artificial intelligence. Multilayer perceptrons form one type of neural network as illustrated in the taxonomy in Fig. 1. This article only considers the multilayer perceptron since a growing number of articles are appearing in the atmospheric literature that cite its use. Many of these papers describe the benefits that neural networks offer when compared to more traditional statistical modelling techniques. Most of the papers briefly describe the workings of neural networks and provide references, to books and papers, from which the reader may obtain further information. This review is aimed at readers with little or no understanding of neural networks and is designed to act as a guide through the literature so that they may better appreciate this tool.

This review is divided into several sections, beginning with a brief introduction to the multilayer perceptron followed by a description of the most basic algorithm for training a multilayer perceptron, known as backpropagation. A review of some of the recent applications of the multilayer perceptron to atmospheric problems will be presented followed by a discussion of some of the common practical problems and limitations associated with a neural network approach.

2. THE MULTI-LAYER PERCEPTRON: A BRIEF INTRODUCTION

Environmental modelling involves using a variety of approaches, possibly in combination. Choosing the most suitable approach depends on the complexity of the problem being addressed and the degree to which the problem is understood. Assuming adequate data and computing resources and if a strong theoretical understanding of the problem is available then a full numerical model is perhaps the most desirable solution. However, in general, as the complexity of a problem increases the theoretical understanding decreases (due to ill-defined interactions between systems) and statistical approaches are required. Recently, the use of neural networks, and in particular the multilayer perceptron, have been shown to be effective alternatives to more traditional statistical techniques (Schalkoff, 1992). Primarily it has been shown (Hornik *et al*., 1989) that the multilayer perceptron can be trained to approximate virtually any smooth, measurable function. Unlike other statistical techniques the multilayer perceptron makes no prior assumptions concerning the data distribution. It can model highly non-linear functions and can be trained to accurately generalise when presented with new, unseen data. These features of the multilayer perceptron make it an attractive alternative to developing numerical models, and also when choosing between statistical approaches. As will be seen the multilayer perceptron has many applications in the atmospheric sciences.

The multilayer perceptron consists of a system of simple interconnected neurons, or nodes, as illustrated in Fig. 2, which is a model representing a nonlinear mapping between an input vector and an output vector. The nodes are connected by weights and output signals which are a function of the sum of the inputs to the node modified by a simple nonlinear transfer, or activation, function. It is the superposition of many simple nonlinear transfer functions that

*Author to whom correspondence should be addressed.

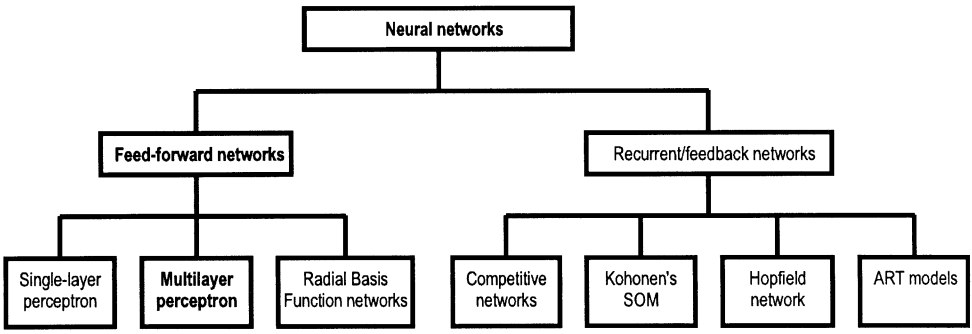

Fig. 1. A taxonomy of neural network architectures (after Jain *et al.*, 1996).

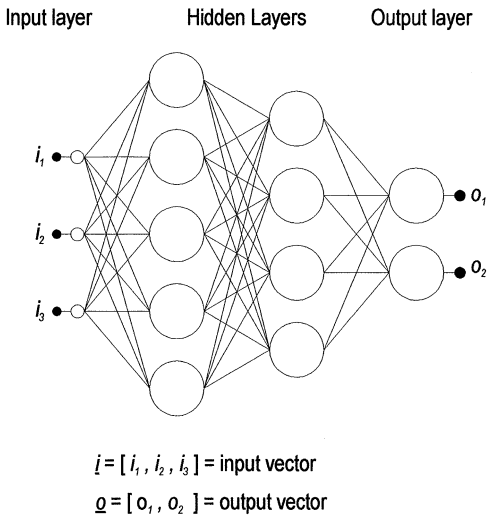

Fig. 2. A multilayer perceptron with two hidden layers.

enables the multilayer perceptron to approximate extremely non-linear functions. If the transfer function was linear then the multilayer perceptron would only be able to model linear functions. Due to its easily computed derivative a commonly used transfer function is the logistic function, as shown in Fig. 3. The output of a node is scaled by the connecting weight and fed forward to be an input to the nodes in the next layer of the network. This implies a direction of information processing, hence the multilayer perceptron is known as a feed-forward neural network. The architecture of a multilayer perceptron is variable but in general will consist of several layers of neurons. The input layer plays no computational role but merely serves to pass the input vector to the network. The terms input and output vectors refer to the inputs and outputs of the multilayer perceptron and can be represented as single vectors, as shown in Fig. 2. A multilayer perceptron may have one or more hidden layers and finally an output layer. Multilayer perceptrons

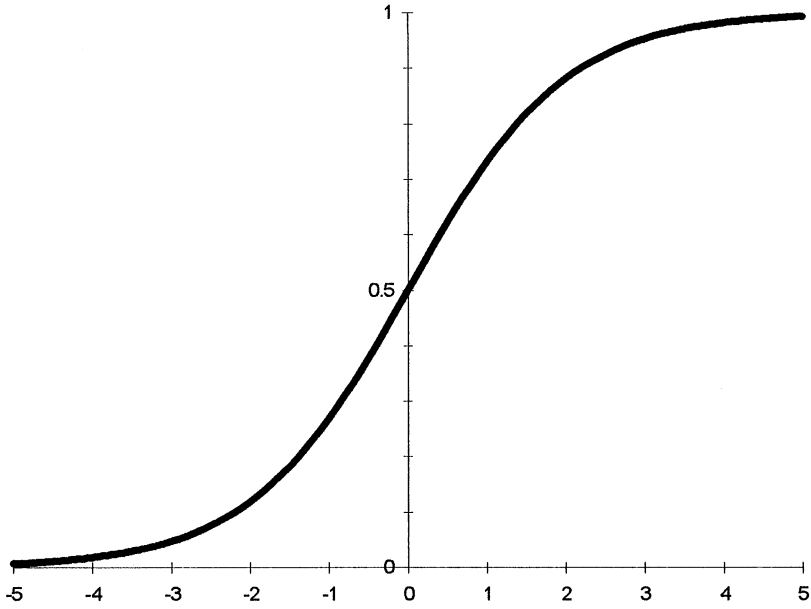

Fig. 3. The logistic function $y = 1/(1 - \exp(-x))$.

are described as being fully connected, with each node connected to every node in the next and previous layer.

By selecting a suitable set of connecting weights and transfer functions, it has been shown that a multilayer perceptron can approximate any smooth, measurable function between the input and output vectors (Hornik *et al*., 1989). Multilayer perceptrons have the ability to learn through training. Training requires a set of training data, which consists of a series of input and associated output vectors. During training the multilayer perceptron is repeatedly presented with the training data and the weights in the network are adjusted until the desired input–output mapping occurs. Multilayer perceptrons learn in a supervised manner. During training the output from the multilayer perceptron, for a given input vector, may not equal the desired output. An error signal is defined as the difference between the desired and actual output. Training uses the magnitude of this error signal to determine to what degree the weights in the network should be adjusted so that the overall error of the multilayer perceptron is reduced. There are many algorithms that can be used to train a multilayer perceptron. Once trained with suitably representative training data (see Section 6) the multilayer perceptron can generalise to new, unseen input data.

3. TRAINING A MULTILAYER PERCEPTRON—THE BACKPROPAGATION ALGORITHM

Training a multilayer perceptron is the procedure by which the values for the individual weights are determined such that the relationship the network is modelling is accurately resolved. At this point we will consider a simple multilayer perceptron that contains only two weights. For any combination of weights the network error for a given pattern can be defined. By varying the weights through all possible values, and by plotting the errors in three-dimensional space, we end up with a plot like the one shown in Fig. 4. Such a surface is known as an error surface. The objective of training is to find the combination of weights which result in the smallest error. In practice, it is not possible to plot such a surface due to the multitude of weights. What is required is a method to find the minimum point of the error surface.

One possible technique is to use a procedure known as gradient descent. The backpropagation training algorithm uses this procedure to attempt to locate the absolute (or global) minimum of the error surface. The backpropagation algorithm (Rumelhart *et al*., 1986) is the most computationally straightforward algorithm for training the multilayer perceptron. Backpropagation has been shown to perform adequately in many applications; the majority of the applications discussed in this paper used backpropagation to train the multilayer perceptrons. A full mathematical derivation of this algorithm can be found in almost all

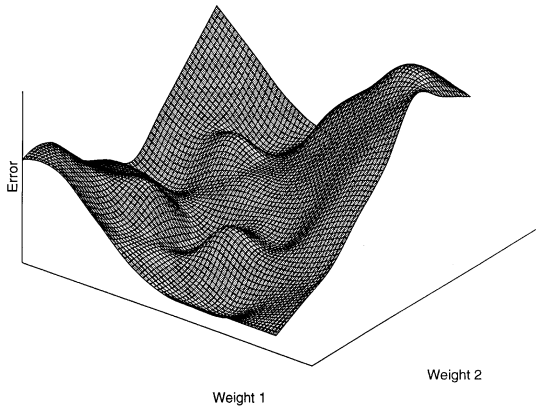

Fig. 4. An error surface for a simple multilayer perceptron containing only two weights.

neural network textbooks (e.g. Bishop 1995) so only the essential components of the algorithm will be discussed here. Backpropagation only refers to the training algorithm and is not another term for the multilayer perceptron or feed-forward neural networks, as is commonly reported.

The weights in the network are initially set to small random values. This is synonymous with selecting a random point on the error surface. The backpropagation algorithm then calculates the local gradient of the error surface and changes the weights in the direction of steepest local gradient. Given a reasonably smooth error surface, it is hoped that the weights will converge to the global minimum of the error surface.

The backpropagation algorithm is summarised below. Implementation details can be found in most neural network books (e.g. Bishop, 1995).

1. initialise network weights,
2. present first input vector, from training data, to the network,
3. propagate the input vector through the network to obtain an output,
4. calculate an error signal by comparing actual output to the desired (target) output,
5. propagate error signal back through the network,
6. adjust weights to minimise overall error,
7. repeat steps 2–7 with next input vector, until overall error is satisfactorily small.

The above implementation of the backpropagation algorithm is known as on-line training whereby the network weights are adapted after each pattern has been presented. The alternative is known as batch training, where the summed error for all patterns is used to update the weights. The benefits of each approach are discussed in Battiti (1992). In practice, many thousands of training iterations will be required before the network error reaches a satisfactory level—determined by the problem being addressed. As will be discussed later, training should be stopped

when the performance of the multilayer perceptron on independent test data reaches a maximum, which is not necessarily when the network error is minimised.

The error surface in Fig. 4 contains more than one minimum. It is desirable that the training algorithm does not become trapped in a local minimum. The backpropagation algorithm contains two adjustable parameters, a learning rate and a momentum term, which can assist the training process in avoiding this. The learning rate determines the step size taken during the iterative gradient descent learning process. If this is too large then the network error will change erratically due to large weight changes, with the possibility of jumping over the global minima. Conversely, if the learning rate is too small then training will take a long time. The momentum term is used to assist the gradient descent process if it becomes stuck in a local minimum. By adding a proportion of the previous weight change to the current weight change (which will be very small in a local minimum) it is possible that the weights can escape the local minimum.

4. MULTILAYER PERCEPTRON APPLICATIONS IN GENERAL

The multilayer perceptron has been applied to a wide variety of tasks, all of which can be categorised as either prediction, function approximation, or pattern classification. Prediction involves the forecasting of future trends in a time series of data given current and previous conditions. Function approximation is concerned with modelling the relationship between variables. Pattern classification involves classifying data into discrete classes. All of these applications are closely related and can be treated as modifications to the following generic model.

4.1. *The generic multilayer perceptron application*

The objective is to find an unknown function f which relates the input vectors in X to the output vectors in Y,

$$Y = f(X)$$

where $X = [n \times k]$, $Y = [n \times j]$, n is number of training patterns, k the number of input nodes/variables and j the number of output nodes/variables.

During training the function f is optimised, such that the network output for the input vectors in X is as close as possible to the target values in Y. The matrices X and Y represent the training data. The function f, for a given network architecture, is determined by the adjustable network weights.

4.2. *Function approximation and prediction*

Both function approximation and prediction are very similar. It is usually the case that only one variable is modelled from the input data, hence the multilayer perceptron will only have one output node, and the dimensions of matrices X and Y in the generic application are $n \times k$ and $n \times 1$, respectively. To use a multilayer perceptron for prediction involves training the network to output the future value of a variable, given an input vector containing earlier observations.

The multilayer perceptron approximates highly non-linear functions between X and Y and requires no prior knowledge of the nature of this relationship. This is one of the benefits multilayer perceptrons offer over conventional regression analysis. If the relationship between X and Y is non-linear then linear regression is clearly an inappropriate tool, although it may be possible to apply linear regression on a more local basis where the non-linearity can be dismissed. Non-linear regression is useful if the nature of the nonlinearity can be found and if the non-linearity is consistent over the entire range of measurements. Extremely non-linear relationships exist in the real world and it is inappropriate to attempt to understand these problems using traditional regression, attributing scatter to the presence of "noise". Under these circumstances the multilayer perceptron is a useful tool. A good theoretical example of this point is presented in Robinson (1991).

4.3. *Pattern classification*

Traditional pattern classifiers assign a class to every measurement vector in X. A classifier is said to partition the whole of measurement space, or the set of all possible input vectors, into j disjoint subsets each representing one of the j target classes. Classifiers aim to minimise the probability of misclassification via Bayes theorem, classifying a new example to the class that has the highest posterior probability. The posterior probability gives a measure of the likelihood of a particular measurement vector belonging to a particular class. An overview of Bayes theorem and pattern classification techniques can be found in Bishop (1995).

Multilayer perceptrons can be used for classification by assigning output nodes to represent each class. For example, if a classification of an air mass was either maritime, continental or polar, then the multilayer perceptron will require three output nodes. The target output vectors in the training data can be considered as binary vectors with a 1 indicating class membership and a 0 indicating no membership. In this way, the maritime class would be represented by an output vector of $[1, 0, 0]$, continental $[0, 1, 0]$, and polar $[0, 0, 1]$. Once trained the network can be presented with an unseen input vector. The output from the multilayer perceptron can be considered as *a posterior* probability; hence the final classification goes to the node with the highest value. If, for example, the output was $[0.6, 0.1, 0.2]$ then the most probable classification would be maritime.

Unlike function approximation and prediction, multilayer perceptrons used for classification will

have more than one output node. The dimensions of matrices X and Y in the generic application will be $n \times k$ and $n \times j$, respectively, where j is the number of possible classes, and n and k are as before.

Multilayer perceptrons have been shown to be superior to traditional classification approaches for several reasons (Benediktsson *et al.*, 1990). Firstly, and most importantly, this approach does not require any prior assumptions regarding the distribution of training data. Classifications that use Bayes theorem rely on a Gaussian (normal) distribution of the data which is often not the case in practical applications. Another benefit of the multilayer perceptron approach is that no decision regarding the relative importance of the various input measurements needs to be made; during training the weights are adjusted to select the most discriminating input measurements. Benediktsson *et al.* (1990) also make the point that with care the traditional classification algorithms can be made to classify more accurately than a multilayer perceptron, but that this requires "significantly more insight and effort on the part of the analyst". Multilayer perceptrons are no panacea.

5. MULTILAYER PERCEPTRON APPLICATIONS IN THE ATMOSPHERIC SCIENCES

There is no space to discuss in detail all atmospheric science applications of the multilayer perceptron. Instead a brief overview of applications from prediction, function approximation and pattern classification will be presented. It is hoped that these papers illustrate the main principles of applying the multilayer perceptron to real-world atmospheric problems. Other papers will be mentioned for reference purposes.

5.1. *Prediction*

The multilayer perceptron has been applied within the field of air-quality prediction. The relationship between meteorology and pollution is complex, and potentially multi-scale in nature. Yi and Prybutok (1996) describe a multilayer perceptron that predicts surface ozone concentrations in an industrialised area of North America. The model takes nine input variables to predict the maximum daily surface ozone concentration. These variables include the morning ozone concentration, the maximum daily temperature, CO_2, NO, NO_2 and NO_x levels, and also wind speed and direction. Results from the multilayer perceptron were shown to be better than those obtained from regression analysis (using the same input data). The authors also suggest that the multilayer perceptron outperforms an ARIMA time-series modelling approach, however such comparisons between techniques must be made with care. For example, to fairly compare an ARIMA time-series model with a multilayer perceptron model, requires that both models are constructed using the same data. In Yi and Prybutoks' paper the multilayer perceptron was trained with all nine meteorological and chemical variables whilst the ARIMA model was constructed purely from the ozone time series.

Boznar *et al.* (1993) constructed a multilayer perceptron to predict atmospheric sulphur dioxide concentrations in a highly polluted industrialised area of Slovenia. The objective of the work was to develop a model that could make accurate short-term (hourly) predictions of sulphur dioxide concentrations in order to determine whether or not to reduce emissions from a coal-fired power station. Previous work had employed a numerical air dispersion model; due to complex topography in the region results were poor. Boznar *et al.* (1993) trained multilayer perceptrons to predict sulphur dioxide concentrations at various stations in the study area for which data was available. The input data to the model were taken from stations over a larger area and consisted of current and previous observations of sulphur dioxide concentrations and meteorological information. This illustrates how easily disparate sources of data can be brought together within a multilayer perceptron model. Results from the model were "encouraging" (no performance statistics were given) and an on-line implementation of the software is being made at one of the power stations where continuous air quality monitoring occurs.

Comrie (1997) compares ozone forecasts made by multilayer perceptron and regression models. The forecasts of summertime *daily* maximum (one hour) ozone concentration for various U.S. urban areas were made using average *daily* meteorological input data. Comrie concluded that the neural network approach found no dramatic improvements when compared to linear regression, with only small to moderate gains in model performance. In our view this is due to the nature of the daily data which does not represent the real non-linear ozone-meteorology relationship. It is likely that the neural network would significantly outperform regression at a sub-daily timescale when the non-linearity of the system is more apparent. By concentrating on the daily timescale the comparison is carried out on an essentially linear system, hence the close performance of the two approaches.

Predicting severe weather is one of the ongoing challenges facing meteorologists. The dynamics of severe weather phenomena are not easily included in current numerical weather prediction models due to their small scale and our limited understanding of them. Marzban and Stumpf (1996) trained a multilayer perceptron to predict the existence of tornadoes. The approach outperformed other techniques including discriminant analysis, logistic regression and a rule-based algorithm. McCann (1992) trained multilayer perceptrons to forecast the presence or absence of significant thunderstorms. Results indicated that the multilayer perceptrons were picking up patterns that skilled forecasters recognise as precursors to

thunderstorms. McCann makes the point that since the exact nature of the interaction between the components that produce a thunderstorm, basically the stability of the air and the presence of lifting mechanisms, are not well understood, it would be desirable to know what the multilayer perceptron has learnt. McCann concludes, after several attempts, that it is "practically impossible to understand the "black box". The conclusions of this study are not very satisfying to scientists. Acceptance comes from how well the networks have learned patterns". This point will be further addressed in Section 6.

Other applications of multilayer perceptrons for prediction include the forecasting of Indian monsoon rainfall (Navone and Ceccatto, 1994), Brazilian rainfall anomalies (Hastenrath and Greischar, 1993), daily solar radiation (Elizondo *et al*., 1994), crop damage by ozone (Benton *et al*., 1995), atmospheric dispersion of pollutants (Rege and Tock, 1996), solar activity (Macpherson *et al*., 1995), and carbon monoxide levels due to vehicle emissions at an urban road intersection (Moseholm *et al*., 1996).

5.2. *Function approximation*

Unlike with prediction, function approximation aims to use the multilayer perceptron to better and more fully model relationships between data. For example, Gardner and Dorling (1996) trained a multilayer perceptron to model the relationship between hourly surface ozone concentration and various local meteorological variables at a coastal location. The objective of this work was to assess the importance of meteorology in determining hourly surface ozone concentrations. The work illustrates that previous attempts to address this problem were inappropriate since the problem is non-linear. Linear regression was tending to underestimate the importance of meteorology.

The multilayer perceptron has been used to model non-linear transfer functions and has found much use in the retrieval of geophysical parameters from remotely sensed data (Thiria *et al*., 1993; Butler *et al*., 1996; Badran and Thiria, 1991; Clothiaux *et al*., 1994; Churnside *et al*., 1994). Krasnopolsky (1995) developed a multilayer perceptron to model the transfer function for special sensor microwave imager (SSM/I) surface wind speed retrieval. The multilayer perceptron approach was shown to be as good as previous empirical algorithms for retrieving wind speeds. However the performance of the multilayer perceptron under cloudy atmospheric conditions was shown to be better than the conventional algorithms. This finding was also supported by the work of Stogryn *et al*. (1994), who claim an improvement of more than a factor of two compared to alternative algorithms under non-clear conditions. This paper illustrates that a well-trained multilayer perceptron can adjust the relationship being modelled depending on specific conditions due to the many interactions between input variables—using a single transfer function for both cloudy and clear conditions was found to be inappropriate. One of the limitations of the multilayer perceptron approach was that wind speeds which were not well represented in the training data were subsequently poorly retrieved. The problem of selecting suitable training data will be discussed in Section 6.

5.3. *Pattern classification*

Global climate change may be indicated in the changing distribution of clouds and cloud types, especially in the polar regions. Multilayer perceptrons have been applied to the classification of satellite images to distinguish between clouds and ice or snow. Multilayer perceptrons have been shown to produce better classifications than discriminant analysis (Welch *et al*., 1992; Tovinkere *et al*., 1993; Bankert, 1994). Cloud classification has been taken a step further by Peak and Tag (1992), who have developed a system to produce synoptic style analyses directly from satellite imagery. The system was developed for the U.S. Navy to aid with the on-ship interpretation of satellite imagery, in particular for identifying the location of fronts, cyclones, severe weather and sea states. The system uses a combination of multilayer perceptrons for pattern recognition and an expert system to produce the final analysis. An introduction to expert systems can be found in Simons (1984).

Multilayer perceptrons have also been applied to the classification of atmospheric circulation patterns. Cawley and Dorling (1996) describe a system which attempts to reproduce a manual classification of atmospheric circulation patterns over the British Isles known as the Lamb Weather Types. Inputs to the multilayer perceptron consist of surface pressure observations over a grid of points centred over the British Isles. The multilayer perceptron marginally outperformed a rule-based classification scheme. Such an automated approach to circulation classifications is desirable since a manual scheme will contain discontinuities as authors change over time. The frequency of the Lamb Weather Types will be subject to variations due to climate change. The ability to classify the masses of output from climate simulation models in such an automated manner would be of great use. Verdecchia *et al*. (1996) report on a similar approach to determine the presence of a "blocking situation" (a region of stationary high pressure) over Europe.

The benefits afforded by the multilayer perceptron, when used for pattern classification, have been clearly observed in work producing land cover classifications (Foody *et al*., 1995; Benediktsson *et al*., 1990; Chen *et al*., 1995; Foody, 1995). The generalisation capability of the multilayer perceptron, even when trained on a limited amount of data, has been used to reduce the time and effort involved in producing land cover classifications from multi-spectral data (Hepner *et al*., 1990). More recently, output from the multilayer perceptron has been interpreted as a fuzzy or soft

classification, and attempts to model mixtures of land cover types have been made (Foody, 1996).

Another application of the multilayer perceptron to pattern recognition is in the classification of convergence lines from radar imagery (Hagelberg and Helland, 1995). Convergence lines represent preferred locations for thunderstorm development. The ability to automatically detect such features from radar images would be of great use in aviation weather forecasting. Finally, Miniere *et al.* (1996) report on the use of a multilayer perceptron to classify electron and proton whistlers (lightning-related phenomena), in real time, aboard magnetospheric satellites. This paper also presents a brief overview of multilayer perceptron applications in space physics.

6. LIMITS, PROBLEMS AND SOLUTIONS—BACKPROPAGATION AND THE MULTILAYER PERCEPTRON IN PRACTICE

The benefits of using multilayer perceptrons have been illustrated. One of the reasons often cited for not using multilayer perceptrons in practice, and artificial neural networks in general, is that they are difficult to implement and interpret. Although this is true to a certain degree, there is an abundance of useful information available that can assist in the process, enabling common pitfalls to be avoided. Commercially available software will often provide built in solutions to protect the user from many of the following issues, however it is still important to have a basic understanding of the problems.

Further discussion can be found in recent textbooks (e.g. Bishop, 1995) and in the neural networks newsgroup frequently asked questions (FAQ) available via the Internet (Sarle, 1997). Many of the problems are common to all statistical modelling techniques and will be mentioned to illustrate that neural networks are not the solution to all problems facing statistical modelling. It must be stressed that there have been many successful applications of the multilayer perceptron when trained using the most basic backpropagation algorithm. With care, some trial and error, and a theoretical understanding of backpropagation, good results can be obtained.

The initial problem faced when using a multilayer perceptron is deciding on the network architecture—the number of layers and nodes in those layers. There are no rules to help in this process. The number of input and output nodes is determined by the problem at hand. Technically only one hidden layer is required to approximate any smooth measurable function between inputs and outputs (Hornik *et al.*, 1989). The optimum number of nodes required in the hidden layer is problem dependent, being related to the complexity of the input and output mapping, the amount of noise in the data and the amount of training data available. If the number of nodes in the hidden layer is too small the backpropagation algorithm will fail to converge to a minimum during training. Conversely, too many nodes will result in the network overfitting the training data resulting in poor generalisation performance.

The principal of generalisation requires further attention. Given some training data and a network with *too many* nodes and hidden layers, it is highly probable that the network will eventually learn all the training patterns in the training data a case of overfitting. In a problem with noisy data the network would eventually learn all the noise in the training data. Since the noise can be assumed to be randomly distributed, similar training patterns with different degrees of noise would be seen as dissimilar by the network. When such an overtrained network is presented with a new pattern it is likely that the network will incorrectly classify it, since the pattern (and its associated random noise) will not have been observed in the training data. Reducing the number of hidden layers and nodes serves to act as a noise filter, forcing the network to ignore the small-scale noise and to learn the underlying patterns in the training data, the desired situation.

It is important to remember that the usual purpose of training the multilayer perceptron is to achieve good generalisation on unseen data, for example in prediction applications. Maximum generalisation performance will occur before the overall network training error reaches a minimum, as shown in Fig. 5. A network trained on a *noisy* set of data until the global minimum is reached is overtrained. One way to ensure this does not occur, and that the generalisation performance will be good, is to divide the training data into several sets—a training set, a validation set and a test set. The training set is used to actually train the network. The validation set can be used to assess the generalisation ability of the network whilst training is occurring. Training is stopped when the generalisation performance reaches a maximum. This technique is known as early stopping and is particularly useful when training multilayer perceptrons with real world, noisy data. Finally, the test set is used

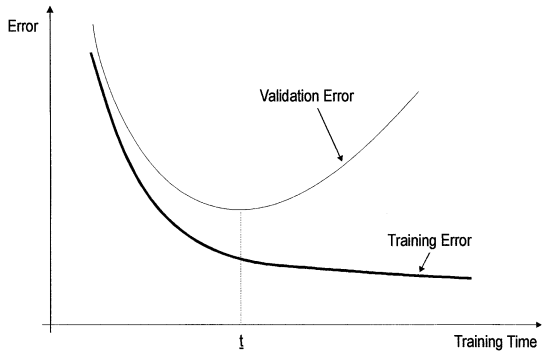

Fig. 5. Training and validation errors with respect to training time.

to assess the overall performance of the trained network.

There are two other factors that determine the level of generalisation that will be obtained from a multilayer perceptron. The first concerns the relationship that the multilayer perceptron is attempting to approximate. In particular, this function must be smooth, by which a small change in an input only results in a small change in the output. Secondly, the training data must be adequately extensive and also representative. Multilayer perceptrons perform well when used for interpolation, but poorly if used for extrapolation. The training data must fully represent all cases about which the multilayer perceptron will be required to generalise from. If both these conditions are satisfied, and the multilayer perceptron is trained with a suitable training algorithm, then the performance of the network will be good. Some further issues relating to the selection of training data can be found in Lawrence (1991).

Given a set of data, the question arises of how best to divide the data into training, validation and test sets, such that the conclusions drawn from the multilayer perceptron are valid and performance is representative. If, for example, the data are divided randomly there is a chance that the data in one of the sets may be biased towards extreme or uncommon events. This will cause problems when assessing the performance of the network. A network can only generalise on the range of data inputs for which it was trained. This is a particular problem when working with small data sets, when good or bad performance could be related to the selection of the various data subsets. One way around this problem is known as V-fold cross-validation, and involves randomly dividing the data into "V" independent subsets (Stone, 1974). The multilayer perceptron is then trained on each subset and its performance on the remaining data is recorded. The stability of the training can then be used to gauge the confidence that can be placed on the network performance when trained with all the data. An excellent overview of the problems associated with data modelling, and determining measures of performance, can be found in Breiman *et al.* (1984).

Often, especially in the atmospheric sciences, successfully modelling the average behaviour of a system is not the desired goal. It is important that the model can resolve infrequent, extreme events which are often of great importance. Several approaches to overcome this problem have been suggested, including data pre-processing, and by making adjustments to the training algorithm so that well learnt patterns are temporarily ignored from the training process (Lister *et al.*, 1993).

Decisions need to be made in choosing suitable values for the learning rate and momentum parameters required by the backpropagation training algorithm. The best values to use are problem-specific. Le Cun *et al.* (1993) describe a process which can determine the maximum possible value of the learning rate. If a network fails to converge an increase in momentum and a decrease in the learning rate may help. If the network oscillates around a solution or becomes unstable then decreasing the learning rate may result in convergence. The selection of suitable learning parameters requires a degree of judgement and is perhaps the most unattractive aspect of the backpropagation algorithm. More sophisticated algorithms avoid the need to specify such sensitive training parameters.

There are many activation functions which can be used in place of the logistic function. Virtually, all non-linear functions could be used, however for the backpropagation algorithm the function must be differentiable. Bounded functions are preferred, since these will prevent weights from taking very large values, slowing convergence during training. One alternative to the logistic function is the hyperbolic tangent (tanh) function. Different layers in the network can have different activation functions. It is often useful, for example, to have an unbounded function (usually the identity function $y = x$) at the output layer. This enables the outputs to take a range of values without being bounded to the limits of the function, for example between 0 and 1 for the logistic function.

Rescaling input data between 0 and 1 is often reported in papers describing applications of the multilayer perceptron. In theory this is unnecessary since any rescaling can be compensated for within the network by adjusting the input to hidden layer weights. In practice, initial network weights are chosen randomly. Therefore, if one input has a large range and another has a small range, but both exhibit a similar amount of variance, then the network may ignore the small input due to the large contribution from the other input. It is therefore advisable to standardise the inputs to the multilayer perceptron, by dividing the input by the standard deviation of all values in the training data. This may also assist with the interpretation of network weights.

Backpropagation is slow and is particularly affected by the "curse of dimensionality" (after Bellman, 1961). As the dimensionality of the data increases, the amount of training data required by the multilayer perceptron rapidly increases. Similarly, as the data become more complex the number of weights, and hence the size of the network, rapidly increases. Both these problems serve to slow the speed at which the backpropagation algorithm will converge. One solution to this problem is to reduce the dimensionality of the input data, which can be achieved through the use of principal components, or by removing redundant variables from the input data. This process is known as feature selection (Setiono and Liu, 1996). An alternative solution is to use a different algorithm. The number of algorithms that exist, and all their derivatives, make it very difficult to know which algorithm to turn to. There is not space here to discuss all such

algorithms. A good description of many algorithms can be found in Gibb (1996) or Battiti (1992).

Once trained, multilayer perceptrons can represent relationships, often with surprising accuracy, that are not fully understood by the traditional theory. To complete this introduction attention must be paid to the "black-box" nature of the multilayer perceptron. No consensus opinion and related theory has been proposed which enables an analysis of the network weights to determine exactly what the multilayer perceptron has learnt. Techniques exist to determine the relative importance of the input variables (Sarle, 1997). These techniques range from a straightforward analysis of the network weights in the input-hidden layer, through to more complex algorithms which take into account the hidden-output layer weights (Garson, 1991). Other approaches involve calculating partial derivatives of the output with respect to the various inputs. All these approaches have their limitations and will often provide conflicting results. If the problem that the multilayer perceptron is applied to is one of prediction, or classification, or the exact nature of the input–output relationship is not important, then the "black box" limitation is of no consequence. If the multilayer perceptron is being applied to problems where the desire is to increase the knowledge of a physical process, and the interaction of driving mechanisms, then the "black-box" limitation will restrict the usefulness of the multilayer perceptron. One particularly promising avenue of research involves pruning the network after training, removing redundant weights. The final network, with much fewer weights, may be more easily analysed. Reed (1993) provides a valuable review of pruning algorithms.

7. CONCLUSION

The multilayer perceptron has been shown to be a useful tool for prediction, function approximation and classification. The practical benefits of a modelling system that can accurately reproduce any measurable relationship is huge. The benefits of the multilayer perceptron approach are particularly apparent in applications where a full theoretical model cannot be constructed, and especially when dealing with non-linear systems. The numerous difficulties in implementing, training and interpreting the multilayer perceptron must be balanced against the performance benefits when compared to more traditional, and often inappropriate, techniques.

There are many commercially and freely available software packages that enable users to implement neural networks relatively easily (Sarle, 1997). Such software is increasingly allowing neural networks to be implemented in a similar manner to regression or discriminant analysis, shielding the user from difficult parameter selection. Unfortunately, most packages provide only a limited number of training algorithms, which cannot be adjusted or adapted by the user. Writing your own code has the benefit of enabling quick implementation of the most recent adaptations to training algorithms. Since there are many alternatives to standard backpropagation this has obvious benefits. Coding the algorithm by hand requires a full understanding of the mechanisms by which the neural network learns and provides more insight into the technique. Whichever method is adopted, the full benefits that neural networks offer can only be realised through a fundamental understanding of the basic theory.

Acknowledgements—We are grateful to Gavin Cawley, School of Information Systems, University of East Anglia and the anonymous reviewers for their helpful comments, and also to the School of Environmental Sciences, University of East Anglia, for supporting this work.

REFERENCES

Badran, F. and Thiria, S. (1991) Wind ambiguity removal by the use of neural network techniques. *Journal of Geophysical Research* **96**(C11), 20521–20529.

Bankert, R. L. (1994) Cloud classification of AVHRR imagery in maritime regions using a probabilistic neural network. *Journal of Applied Meteorology*, **33**, 909–918.

Battiti, R. (1992) First- and second-order methods for learning: between steepest descent and Newton's method. *Neural Computation*, **4**, 141–166.

Bellman, R. E. (1961) *Adaptive Control Processes.* Princeton University Press, Princeton, NJ.

Benediktsson, J. A., Swain, P. H. and Ersoy, O. K., (1990) Neural network approaches versus statistical methods in classification of multisource remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing* **28**(4), 540–552.

Benton, J., Fuhrer, J., Gimeno, B. S., Skarby, L. and Sanders, G. (1995) Results from the UN/ECE ICP-crops indicate the extent of exceedance of the critical levels of ozone in Europe. *Water, Air and Soil Pollution* **85**, 1473–1478.

Bishop, C. M. (1995) *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford.

Boznar, M., Lesjak, M. and Mlakar, P. (1993) A neural network-based method for the short-term predictions of ambient SO_2 concentrations in highly polluted industrial areas of complex terrain. *Atmospheric Environment.* B **27**(2), 221–230.

Breiman, L., Freidman, J. H., Olshen, R. A. and Stone, C. J. (1984) *Classification and Regression Trees.* Wadsworth, California.

Butler, C. T., Meredith, R. V. Z. and Stogryn, A. P. (1996) Retrieving atmospheric temperature parameters from DMSP SSM/T-1 data with a neural network. *Journal of Geophysical Research* **101**(D3), 7075–7083.

Cawley, G. C. and Dorling, S. R. (1996) Reproducing a subjective classification scheme for atmospheric circulation patterns over the United Kingdom using a neural network. *Proceedings International Conference on Neural Networks 1996*, ICANN'96.

Chen, K. S., Tzeng, Y. L., Chen, C. F. and Kao, W. L. (1995) Land-cover classification of multispectral imagery using a dynamic learning neural network. *Photogrammetric Engineering and Remote Sensing* **61**(4), 403–408.

Churnside, J. H., Stermitz, T. A. and Schroeder, J. A. (1994) Temperature profiling with neural network inversion of microwave radiometer data. *Journal of Atmospheric and Oceanic Technology* **11**(1), 105–109.

Clothiaux, E. E., Penc, R. S., Thomson, D. W., Ackerman, T. P. and Williams, S. R. (1994) A first-guess feature-based algorithm for estimating wind speed in clear-air doppler radar spectra. *Journal of Atmospheric and Oceanic Technology*, **11**, 888–908.

Comrie, A. C. (1997) Comparing neural networks and regression models for ozone forecasting. *Journal of Air and Waste Management* **47**, 653–663.

Elizondo, D., Hoogenboom, G. and McClendon, R. W. (1994) Development of a neural network model to predict daily solar radiation. *Agricultural and Forest Meteorology* **7**, 115–132.

Foody, G. M. (1995) Land cover classification by an artificial neural network with ancillary information. *International Journal of Geographical Information Systems* **9**(5), 527–542.

Foody, G. M. (1996) Approaches for the production and evaluation of fuzzy land cover classifications from remotely sensed data. *International Journal of Remote Sensing* **17**(7), 1317–1340.

Foody, G. M., McCulloch, M. B. and Yates, W. B. (1995) Classification of remotely sensed data by an artificial neural network: issues related to training data characteristics. *Photogrammetric Engineering and Remote Sensing* **61**(4), 391–401.

Gardner, M. W. and Dorling, S. R. (1996) Neural network modelling of the influence of local meteorology on surface ozone concentrations. *Proceedings 1st International Conference on GeoComputation*, University of Leeds, pp. 359–370.

Garson, G. D. (1991) Interpreting neural network connection weights. *Artificial Intelligence Expert*, **April**, 47–51.

Gibb, J. C. (1996) Back propagation family album, Technical Report C/TR96-05, Department of Computing, Macquarie University, NSW 2109, Australia.

Hagelberg, C. and Helland, J. (1995) Thin-line detection in meteorological radar images using wavelet transforms. *Journal of Atmospheric and Oceanic Technology* **12**(3), 633–642.

Hastenrath, S. and Greischar, L. (1993) Further work on the prediction of Northeast Brazilian rainfall anomalies. *Journal of Climate* **6**, 743–758.

Hepner, G. F., Logan, T., Ritter, N. and Bryant, N. (1990) Artificial neural network classification using a minimal training set: comparison to conventional supervised classification. *Photogrammetric Engineering and Remote Sensing* **56**(4), 469–473.

Hornik, K., Stinchcombe, M. and White, H. (1989) Multilayer feedforward networks are universal approximators. *Neural Networks* **2**, 359–366.

Jain, A. K., Mao, J. and Mohiuddin, K. M. (1996) Artificial neural networks – a tutorial. *Computer* **March**, 31–44.

Krasnopolsky, V. M., Breaker, L. C. and Gemmill, W. H. (1995) A neural network as a non-linear transfer function model for retrieving wind speeds from the special sensor microwave imager. *Journal of Geophysical Research* **100**(C6), 11033–11045.

Lawrence, J. (1991) Data preparation for a neural network. *Artificial Intelligence Expert* **November**, 34–41.

Le Cun, Y., Simard, P. Y. and Pearlmutter, B. (1993) Automatic learning rate maximisation by online estimation of the Hessian's eigenvectors, eds. Hanson, S. J., Cowna, J. D. and Giles, C. L. *Advances in Neural Information Processing Systems*, vol. 5, pp. 156–163, Morgan Kaufmann, California.

Lister, R., Bakker, P. and Wiles, J. (1993) Error signals, exceptions, and back propagation. *Proceedings of 1993 International Conference on Neural Networks* 573–576.

McCann, D. W. (1992) A neural network short-term forecast of significant thunderstorms. *Forecasting Techniques* **7**, 525–534.

Macpherson, K. P., Conway, A. J. and Brown, J. C. (1995) Prediction of solar and geomagnetic-activity data using neural networks. *Journal of Geophysical Research-Space Physics* **100**(A11), 21735–21744.

Marzban, C. and Stumpf, G. J. (1996) A neural network for tornado prediction based on doppler radar derived attributes. *Journal of Applied Meteorology* **35**, 617–626.

Miniere, X., Pincon, J. L. and Lefeuvre, F. (1996) A neural network approach to the classification of electron and proton whistlers. *Journal of Atmospheric and Terrestrial Physics*, **58**(7), 911–924.

Moseholm, L., Silva, J. and Larson, T. (1996) Forecasting carbon monoxide concentrations near a sheltered intersection using video traffic surveillance and neural networks. *Transportation Research* **1D**(1), 15–28.

Navone, H. D. and Ceccatto, H. A. (1994) Predicting Indian monsoon rainfall: a neural network approach. *Climate Dynamics* **10**, 305–312.

Peak, J. E. and Tag, P. M. (1992) Towards automated interpretation of satellite imagery for navy shipboard applications. *Bulletin of the American Meteorological Society* **73**(7), 955–1008.

Reed, R. (1993) Pruning algorithms—a survey. *IEEE Transactions on Neural Networks* **4**(5), 740–747.

Rege, A. R. and Tock, R. W. (1996) A simple neural network for estimating emission rates of hydrogen sulfide and ammonia from single point sources. *Journal of Air and Waste Management Association* **46**, 953–962.

Robinson, R. (1991) Neural networks offer an alternative to traditional regression. *Geobyte* **February**, 14–19.

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986) Learning internal representations by error propagation, in: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, eds. D. E. Rumelhart and J. L. McClelland, Vol. 1, MIT Press, Cambridge, MA.

Sarle, W. (1997) *comp.ai.neural-nets Frequently Asked Questions*, ftp://ftp.sas.com/pub/neural/FAQ.html.

Schalkoff, R. (1992) *Pattern Recognition: Statistical. Structural and Neural Approaches*. Wiley, New York.

Setiono, R. and Liu, H. (1996) Improving backpropagation learning with feature selection. *Applied Intelligence* **6**, 129–139.

Simons, G. L. (1984) *Introducing Artificial Intelligence*. NCC publications, Manchester.

Stogryn, A. P., Butler, C. T. and Bartolac, T. J. (1994) Ocean surface wind retrievals from special sensor microwave imager data with neural networks. *Journal of Geophysical Research* **99**(C1), 981–984.

Stone, M. (1974) Cross-validatory choice and assessment of statistical prediction. *Journal of the Royal Statistical Society* **B36**(1), 111–147.

Thiria, S., Mejia, C., Badran, F. and Crepon, M. (1993) A neural network approach for modelling non-linear transfer functions: application for wind retrieval from spaceborne scatterometer data. *Journal of Geophysical Research* **98**(C12), 22827–22841.

Tovinkere, V. R., Penaloza, M., Logar, A., Lee, J., Weger, R. C., Berendes, T. A. and Welch, R. M. (1993) An intercomparison of artificial intelligence approaches for polar scene identification. *Journal of Geophysical Research* **98**(D3), 5001–5016.

Verdecchia, M., Visconti, G., D'Andrea, F. and Tibaldi, S. (1996) A neural network approach for blocking recognition. *Geophysical Research Letters* **23**(16), 2081–2084.

Welch, R. M., Sengupta, S. K., Goroch, A. K., Rabindra, P., Rangaraj, N. and Navar, M. S. (1992) Polar cloud classification using AVHRR imagery - an intercomparison of methods. *Journal of Applied Meteorology* **31**, 405–420.

Yi, J. and Prybutok, R. (1996) A neural network model forecasting for prediction of daily maximum ozone concentration in an industrialised urban area. *Environmental Pollution* **92**(3), 349–357.