ELSEVIER

Ecological Modelling 148 (2002) 27–46

ECOLOGICAL MODELLING

www.elsevier.com/locate/ecolmodel

Atmospheric urban pollution: applications of an artificial neural network (ANN) to the city of Perugia

P. Viotti [a,*], G. Liuti [b], P. Di Genova [b]

[a] *Department of Hydraulic, Transportation and Roads, University of Rome 'La Sapienza', Via Eudossiana 18, 00184 Rome, Italy*

[b] *Department of Civil and Environmental Engineering, University of Perugia, Via G. Duranti 93, S. Lucia Canetola, 06125 Perugia, Italy*

Received 13 November 2000; received in revised form 26 July 2001; accepted 1 August 2001

Abstract

Urban air pollution is a growing problem. Large cities in particular show, at some point of time, high concentrations of substances dangerous for human health. The difficulty in forecasting pollutant's concentration trends with a reasonable error is still an open problem. In this paper, a new approach is proposed. An artificial neural network (ANN) is used to forecast short and middle long-term concentration levels for some of the well-known pollutants. The results seem to be in good accord with the monitored data and allow its use as the forecasting model on a 24–48 h basis requiring only the meteorological conditions and the traffic level. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Atmospheric pollution; Artificial neural network; Backpropagation; Forecasting model; Perugia urban area

1. Introduction

In recent times, urban air pollution has been represented as a growing problem for communities. The danger represented by air pollutants has been largely demonstrated from toxicological studies both for short and long exposition times.

Some compounds like sulphur dioxide, atmospheric particulate, nitrogen compounds, carbon monoxide and ozone are considered as typical indicators of the air quality. This situation appears to be relevant in urban areas where particular atmospheric and orographic conditions can cause an accumulation of the pollutants.

Recently some community actions together with the renewal of the vehicles have brought about lower pollutant levels in urban areas, but new pollutants are becoming important (IPA, benzene, etc.); therefore, at the moment one of the best ways of controlling the pollutants' concentration seems to be the correct management of both urban development and traffic. The best management actually seems to be the possibility to forecast the pollutants' concentration trends 1 or 2 days earlier according to the meteorological conditions and the traffic expectation in defined ar-

* Corresponding author. Tel.: + 39-6-4458-5512; fax: + 39-6-4458-5037.

E-mail address: paolo.viotti@uniromal.it (P. Viotti).

PII: S0304-3800(01)00434-3

eas, so that the authorities have an opportunity to act. In fact the concentration of the pollutants in atmosphere in the urban areas are mainly due to the traffic emissions, connected to the traffic jam, the vehicular status, the geographic and local site characteristics, and meteorological conditions. The use of deterministic models or lagrangian ones to forecast the pollution levels in urban areas is made rather difficult for the complex fluidodynamic field which arise from the complex orography and for the heat phenomena which characterise the urban contest. One of the advantages of the artificial neural network (ANN) lies in the fact that the deterministic model needs a lot of information, whereas the neural network acts like a black-box model. The drawback of the neural approach is that no deep understanding on the physical phenomena is gained by using a neural network, since it resembles the behaviour of a black-box method. Moreover, the ANN, once trained, is fast at predicting the desired values. The last and the most convincing advantage is that the accuracy of the neural prediction is generally higher than the other kind of models.

In the present work, an ANN is used to forecast the pollutant concentration levels in the urban area of Perugia.

2. Model description

ANNs are a branch of artificial intelligence developed in the 1950s aiming at imitating the biological brain architecture. They are parallel-distributed systems made of many interconnected non-linear processing elements, called *neurons* (Hecht-Nielsen, 1990). A renewal of interest has grown exponentially in the last decade, mainly for the availability of suitable hardware (e.g. parallel computers, analogue/digital *neural* card for personal computers) that has made them convenient for fast data analysis and information processing.

Neural networks have found many applications on time series prediction (Lapedes and Farber, 1987; Werbos, 1988). Although their behaviour has been related to non-linear statistical regression (Bishop, 1995), the big difference is that neural networks seem naturally suited for problems that show a large dimensionality of data, such as the task of identification for systems with a big number of state variables. Many ANN-based models were developed for very different environmental purposes. Heymans and Baird (2000) have used network analysis to evaluate the carbon flow model built for the northern Benguela upwelling ecosystem in Namibia. Antonic et al. (2001) have estimated the forest survival after building the hydro-electric power plant on the Drava River, Croatia by means of a GIS constructed database and a neural network. A three-layer Levenberg–Marquardt feedforward neural network was used by Karul et al. (2000) to model the eutrophication process in the three water bodies in Turkey. Other examples in environmental field were provided by Giustolisi and Mastrorilli (1994) and Carosone et al. (1996); both adopted the perhaps most simple and widely used neural network, named *perceptron* with the *error backpropagation algorithm*.

An input layer, a hidden layer where the elaborations are performed, and an output layer constitute the most widely used perceptron, named single hidden layer perceptron. The single unit of the network, i.e. the model neuron, is shown in Fig. 1.

The neuron has N input lines and a single output. Each input signal is weighted, i.e. it is multiplied with the weight value of the corresponding input line (by analogy to the synaptic strength of the connections of the biologic neurons). The artificial neuron will combine these weighted inputs by forming their sum, and with reference to a threshold value and an activation function it will determine its output. All the inputs form a vector of values presented to the unit. If the neuron belongs to the input layer, the input vector presented to the network in a precise instant, is defined *pattern*. For supervised networks, as the perceptron is used (which belongs to the feedforward nets typologies), a pattern is formed by inputs related to the past together with the pollutant concentration to be forecasted, named real or desired output. A vector of values, related to an input parameter, presented in different instants to the unit is called the *input variable*. If the parameter is the desired output, it is called the *target variable*. The ANN works on a matrix containing more patterns. Particularly, the patterns represent the rows while the variables are the columns. This data set is a sample, i.e. a subset of the population representing the phenomenon studied. To be more precise, giving the ANN three types of subset of the available sample can create the forecasting model: the training set, the test set, and the validation set. Definitions on these sets are crucial and often confused in the neural networks field (Bishop, 1995, p. 372; Ripley, 1996, p. 354). In this paper, these definitions will be referred as follows:

- *training set*, the group of data by which we train the network, i.e. by which the network adjusts—taking randomly the patterns—its parameters (thresholds and weights), according to the gradient descent for the error function algorithm, in order to reach the best fitting of the non-linear function representing the phenomenon;
- *test set*, the group of data, given to the network still in the learning phase, by which the error evaluation is verified in order to effectively update the best thresholds and the weights;
- *validation set*, a set (or more sets) of new data (given in the generalisation phase, i.e. with the fixed net parameters) used to evaluate ANN generalisation, i.e. to evaluate whether the model has effectively approximated the general function representative of the phenomenon, instead of learning the patterns uniquely.

In mathematical terms, the process can be described by means of the following equations:

$$u = \sum_{i=1}^{N} w_i x_i \tag{1}$$

$$y = f(u - \theta) \tag{2}$$

where $x_1, x_2,\ldots,x_N$ are the input signals, $w_1, w_2,\ldots,w_N$ are the synaptic weights, u is the activation potential of the neuron, θ is the threshold, y is the output signal of the neuron and $f(\,)$ is the activation function. For convenience, Eqs. (1) and (2) can be reformulated by setting w_0 equal to zero and x_0 equal to -1. We then obtain:

$$\sum_{i=1}^{N} w_i x_i - \theta = \sum_{i=0}^{N} w_i x_i \tag{3}$$

$$y = f\left(\sum_{i=0}^{N} w_i x_i\right) \tag{4}$$

where it is evidenced that weights and thresholds constitute the characteristic network parameters. The most common form of activation function used in the construction of ANNs is the standard logistic or sigmoid function (Eq. (5)), because of the simplicity of its implementation:

$$f(x) = 1/[1 + \exp(-x)] \tag{5}$$

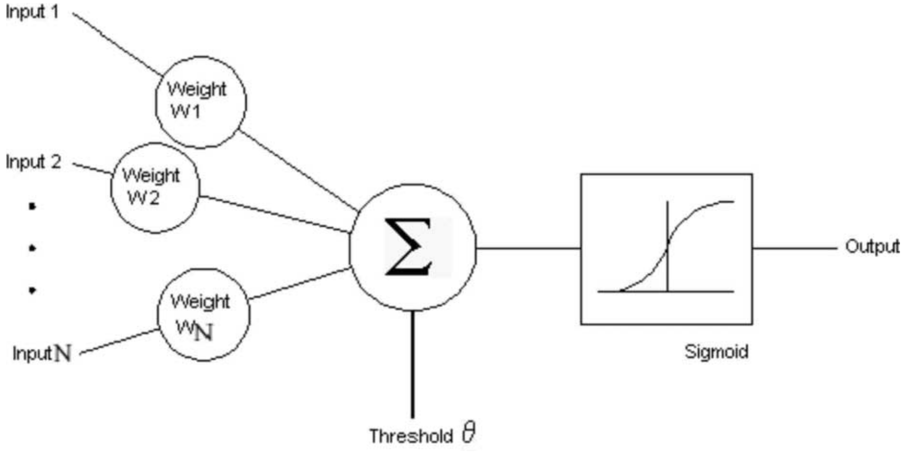

Fig. 1. The artificial neuron.

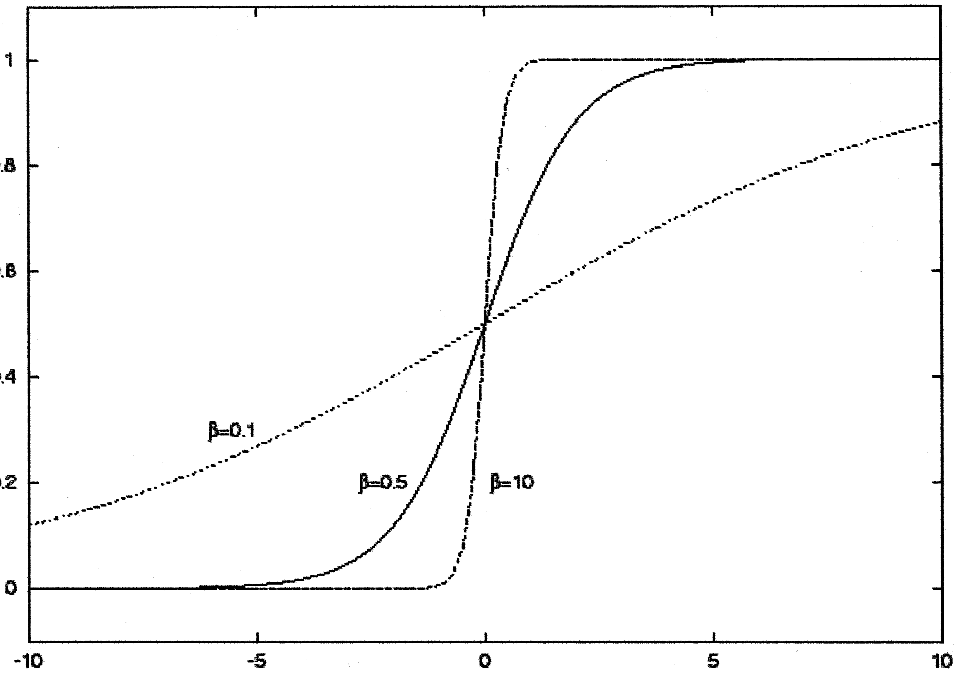

Fig. 2. Logistic functions varying with β.

where x is the problem variable. The non-linear activation functions are very important characteristics of an ANN, as they make the network a non-linear model, which is able to approximate complex non-linear functions, more interesting in usual applications, as the forecasting problem.

In the present work, a general form of the logistic function (Eq. (5′)), was used:

$$f(x) = \alpha/[1 + \exp[-\beta(x - \lambda)] \qquad (5')$$

where α is the amplitude, β is the slope parameter and λ is a threshold. The generalised logistic function instead of the standard one allows varying the sensibility of the function to the variations of the activation potentials in every layer. It is very difficult to choose the best values of these adjustable parameters for the logistic function and necessitates further studies. One of the criteria proposed by Pelliccioni and Pessa (1997) is based on an adjustment procedure, which develops as follows:

1. The high-sensitivity range of the activation potential received by output units layer must overlap significantly with the typical range of effective activation potential produced by hidden units.
2. The slope β must optimise the sensitivity to activation potential variations, i.e. enhancing it to capture the information contained in the data, but avoiding a very high sensitivity which could also capture noise (see Fig. 2).
3. The amplitude α is chosen to control the variability range of the output layer.

Another important property of the logistic function is that it is differentiable, an essential characteristic for the realisation of the backpropagation algorithm. This algorithm derives its name from the fact that error signals are propagated backward through the network on a layer-by-layer basis. The backpropagation algorithm is based on the selection of a suitable *error function* or *cost function*, whose values are determined by the difference between actual (network value) and desired (real or experimental value) outputs and which is also dependent on the network parameters such as weights and thresholds. The basic idea is that the cost function has a particular surface over the weight space and therefore, an iterative process such as the gradient descent method can be used for its minimisation.

The method of gradient descent is based on the fact that, since the gradient of a function always points in the direction of maximum increase of the function, then by moving towards the direction of the negative gradient induces a maximal 'downhill' movement that will eventually reach the minimum of the function surface over its parameter space. This is a rigorous and well-established technique for the minimisation of functions and has probably been the main factor behind the success of backpropagation. However, the method does not guarantee that it will always converge to the minimum of the error surface as the network can be trapped in various types of *minima*, such as temporary minima and local minima. Temporary minima are mainly due to the bad initialisation of weights and thresholds and can slow the learning process heavily: this situation corresponds often on a plateau or plane zone on the error function surface. A good initialisation, e.g. performed randomly on a little range around the distribution mean value for the input variables, can solve this problem easily. On the other hand, it is more difficult to escape from local minima and they often represent the learning limit (local optima) for the specific ANN created. They are as valleys on the error surface where the gradient descent can be trapped. The learning process can in some cases be far from the global optimum, and the problem can be solved only with a very good database, a best choice of the input configuration for the training, or with the usage of most powerful learning algorithms, e.g. the Levenberg–Marquardt, Gauss–Newton, Rprop, conjugate gradients, genetic algorithms, and others present in the ANN literature (Bertsekas, 1995; Masters, 1995; Bertsekas and Tsitsiklis, 1996).

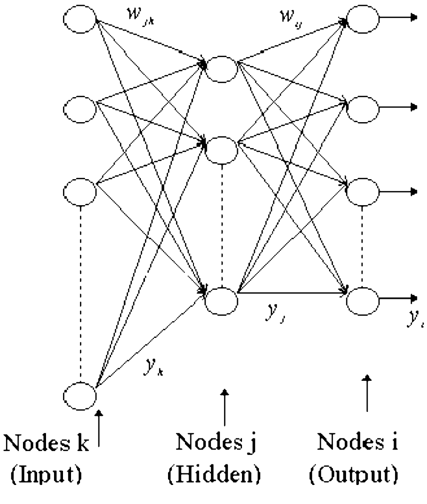

Fig. 3. Feedforward network general structure.

The backpropagation training consists of two steps of computation: a *forward pass* and a *backward pass*. In the forward pass an input pattern vector is applied to the sensory nodes of the network, i.e. to the units in the input layer. The signals from the input layer propagate to the units in the first layer and each unit produces an output, according to Eq. (4). The outputs of these units are propagated to the units in the subsequent layers and this process continues until, finally, the signals reach the output layer where the actual response of the network to the input vector is obtained (see Fig. 3).

During the forward pass, the synaptic weights of the network are fixed. During the backward pass, on the other hand, the synaptic weights are all adjusted in accordance with an error signal, which is propagated backward through the network against the direction of synaptic connections.

The mathematical analysis of the algorithm is as follows. In the forward pass, given an input pattern vector $y^{(\mathrm{p})}$, each hidden node j receives a net input:

$$x_j^{(\mathrm{p})} = \sum_k w_{jk} y_k^{(\mathrm{p})} \tag{6}$$

where w_{jk} represents the weight between the hidden node j and the input node k.

Thus the hidden neuron j produces an output:

$$y_j^{(\mathrm{p})} = f(x_j^{(\mathrm{p})}) = f\left(\sum_k w_{jk} y_k^{(\mathrm{p})}\right) \tag{7}$$

where $f(\,)$, is the activation function of the hidden layer. Each output node receives the input from the preceding hidden layer; the ANN used in the present work is constituted by a single output unit, i.e. by the value of the forecasted pollutant concentration, so that the entry to the output neuron can be written as:

$$x^{(p)} = \sum_j w_j y_j^{(p)} = \sum_j w_j \; f\left(\sum_k w_{jk} y_k^{(p)}\right) \tag{8}$$

where w_j represents the weight between the output node and the hidden node j. It therefore produces the final output:

$$\begin{aligned} y^{(p)} &= f\left(x^{(p)}\right) = f\left(\sum_j w_j y_j^{(p)}\right) \\ &= f\left[\sum_j w_j \; f\left(\sum_k w_{jk} y_k^{(p)}\right)\right] \end{aligned} \tag{9}$$

The presentation of all the patterns is usually called *epoch*. Many epochs are generally needed before the error becomes acceptably small. In the *batch mode* the error signal is calculated for each input pattern (as in the *incremental mode*) but the weights are modified only when *all* the input patterns have been presented (instead of the weights updating after every single pattern presentation as in the incremental mode). The error function E—reported in Eq. (10)—is calculated referring to the Mean Square Error (MSE) and the weights are modified accordingly:

$$E = \frac{1}{2}(d-y)^2 = \frac{1}{2}\left\{d - f\left[\sum_j w_j \; f\left(\sum_k w_{jk} y_k\right)\right]\right\}^2 \tag{10}$$

where d is the desired or real output (monitored variable value) and y is the ANN output or the forecasted value. In the batch mode E is equal to the sum of all the MSEs on all the patterns of the training set. E is obviously a differentiable function of all weights (and thresholds) and therefore we can apply the gradient descent method. For the hidden-to-output connections, e.g. in the batch mode, the gradient descent rule gives:

$$\Delta w_j = -\eta \frac{\partial E}{\partial w_j} \tag{11}$$

where η is a number called *learning rate*. Using the chain rule it can be written as:

$$\frac{\partial E}{\partial w_j} = \frac{\partial E}{\partial y^{(p)}} \frac{\partial y^{(p)}}{\partial x^{(p)}} \frac{\partial x^{(p)}}{\partial w_j} = -\sum_p \left(d^{(p)} - y^{(p)}\right) f'\left(x^{(p)}\right) y_j^{(p)} \tag{12}$$

Thus, the hidden-to-output connections are updated according to the following equation:

$$\Delta w_j = \eta \sum_p \left(d^{(p)} - y^{(p)}\right) f'\left(x^{(p)}\right) y_j^{(p)} = \eta \sum_p \delta^{(p)} y_j^{(p)} \tag{13}$$

where

$$\delta^{(p)} = \left(d^{(p)} - y^{(p)}\right) f'\left(x^{(p)}\right) \tag{14}$$

For the input-to-hidden connections the gradient descent rule is:

$$\Delta w_{jk} = -\eta \frac{\partial E}{\partial w_{jk}} \tag{15}$$

then using the chain rule, we obtain:

$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial y_j^{(p)}} \frac{\partial y_j^{(p)}}{\partial x_j^{(p)}} \frac{\partial x_j^{(p)}}{\partial w_{jk}} = \frac{\partial E}{\partial y_j^{(p)}} f'\left(x_j^{(p)}\right) y_k^{(p)} \tag{16}$$

Particularly, with reference to $\partial E / \partial y_j^{(p)}$, it can be written as:

$$\frac{\partial E}{\partial y_j^{(p)}} = -\sum_p \left(d^{(p)} - y^{(p)}\right) \frac{\partial\left[f\left(x^{(p)}\right)\right]}{\partial y_j^{(p)}}$$

and, after simple passages, we obtain:

$$\frac{\partial E}{\partial y_j^{(p)}} = -\sum_p \left(d^{(p)} - y^{(p)}\right) f'\left(x^{(p)}\right) w_j \tag{17}$$

Therefore, Eq. (16) can be written as:

$$\frac{\partial E}{\partial w_{jk}} = -\sum_p \left(d^{(p)} - y^{(p)}\right) f'\left(x^{(p)}\right) w_j \; f'\left(x_j^{(p)}\right) y_k^{(p)} \tag{18}$$

from which the input-to-hidden connections updating is obtained as:

$$\Delta w_{jk} = \eta \sum_p \left(d^{(p)} - y^{(p)}\right) f'\left(x^{(p)}\right) w_j \; f'\left(x_j^{(p)}\right) y_k^{(p)}$$

and finally we obtain:

$$\Delta w_{jk} = \eta \sum_p \delta^{(p)} w_j \; f'\left(x^{(p)}\right) y_k^{(p)} = \eta \sum_p \delta_j^{(p)} y_k^{(p)} \tag{19}$$

with

$$\delta_j^{(p)} = \delta^{(p)} w_j \; f'\left(x^{(p)}\right) \tag{20}$$

Generally, with an arbitrary number of layers, the backpropagation update rule has the following form:

$$\{\text{Weight correction } \Delta w_{\mathrm{lm}}\} = \{\text{Learning rate } \eta\} \times \{\text{Local gradient } \delta_{\mathrm{l}}\} \times \{\text{Input signal of node } y_{\mathrm{m}}\}$$

This general rule for the adaptation of the weights is also known as the *generalised delta rule*. For further details about the specific characteristics and properties of the neural network used for the city of Perugia, one can refer to the thesis of Di Genova (Perugia, Department of Civil and Environmental Engineering, 2000).

3. Applications

The application of an ANN to the urban context in Perugia, particularly for the area of Fontivegge, near the F.S. Railway Station, is presented. The experimental data were obtained from the monitoring unit of the Presidio Multizonale di Prevenzione, located in Fontivegge and operative since 1997. The variables monitored were: sulphur dioxide, nitrogen oxides (NO, NO_2, NO_x), total suspended particulate and PM10, benzene, carbon monoxide, ozone, horizontal wind speed, moisture, pressure, temperature, total sun radiation (tsr), rain, and traffic (Mossone et al., 1999). The neural network was trained, tested and validated for many specific input configurations to forecast the concentrations of the single pollutant by varying the available information. Consequently, many models suitable for different circumstances, each one valid exclusively for a single air pollutant, have been implemented.

The elaborations performed in Perugia regarded the choice and the experimentation of:

- different ANN architectures;
- methodologies to scale and elaborate the available information, previously controlled and validated, for the ANN;
- testing and validation techniques for the realised models.

Among the many alternatives, by experience in Perugia, the following ANN characteristics were never modified.

- Number of layers: the used perceptron is always constituted by a single hidden layer.
- Number of output neurons: one, always corresponding to the foreseen pollutant concentration value.
- Learning rule: always the backpropagation standard.

On the other hand, the ANN modifiers have been mainly related to:

- the number of input and hidden neurons;
- data scaling;
- training and test sets choice;
- validation methodology (simple or crossed);
- learning rate (constant or variable);
- learning procedure (batch or incremental);
- activation function choice (logistic or hyperbolic).

The number of input neurons is directly connected to the quantity of information given to the ANN and is generally constituted from pollutant concentrations (including the one to be forecasted), meteorological and traffic data, even at different past hours (i.e. the number of inputs can be greater than the number of variables monitored). Too many inputs can slow the process and drive away the ANN from the inner function comprehension. On the other hand, few inputs make the convergence to optimal weight values really difficult or impossible. Statistical analysis on the available data and theoretical fundamentals on pollution processes can help find the right input configurations for the problem considered, as it will be showed in the following long-term forecasts on NO_x (e.g. model 4 in Fig. 16).

The hidden neurons choice characterises the number of coefficients for the non-linear inputs combination to give the final output. There are many empirical rules in the literature (Blum, 1992; Swingler, 1996; Berry and Linoff, 1997; Boger and Guterman, 1997) but they make no sense. The right choice could be suggested from a particular experience. It just has to be mentioned that too many hidden neurons in proportion to the input ones could cause the ‘overfitting’ or ‘overtraining’ phenomenon. In such a situation, the ANN learns too much from these good examples, though it does not learn the representative function of the problem—i.e. the ANN does not generalise—(e.g. model E, in Fig. 8) while few hidden neurons with respect to the input ones could miss the target of non-linearity approximation.

The available data and their correct utilisation problem perhaps represents the success key of the ANN model. It is clear that cities where the urban air quality monitoring system has been developed since many years with a large number of validated data required for the utilisation of an ANN to realise forecasting models. Like many statistically based problems, in fact, many continuous time series available are able to represent at the best the configurations of the complex physical phenomenon. The more complicate the physical and chemical processes related to a pollutant are, the more difficult it is to create a good forecasting ANN model. Primary pollutants (e.g. benzene, carbon monoxide and nitrogen monoxide) are directly dependent on the source emissions while secondary pollutants (e.g. ozone and nitrogen dioxide) are not dependent on sources but on chemical reactions involving their precursors and other atmospheric components, and are therefore difficult to be forecasted. ANNs are not magicians: if the information used for the training phase are poor in number and less relevant to the pollution process in examination, then the ANN cannot obviously extract the representative function for the problem, i.e. it cannot realise good forecasts. In that case, it could be useful as a highly advanced monitoring system, able to analyse many variables and operative since many years, or at least in a continuous way (with validated monitoring results) for many seasons. However, there are many difficulties linked to the availability of the hourly data. In this study, in Perugia (period considered: the years 1997–1999) the main causes for missing data were periodic instrumental calibration, unexpected breaks and various anomalies. This has brought about discontinuity in experimental data's trend. Further, the local administrations' choice of changing the variables to be monitored during the years in the observed site (the traffic was monitored in Fontivegge in 1997 and 1998 but not in 1999) is of no help in the perspective of a monitoring integrated system, whose target is to develop, together with an efficient environmental control, a powerful instrument able to give a clear information about the site characteristics and their evolution during the years, for every type of study required. The monitoring system and ANN models, in fact, can be seen as complementary: the former is necessary to create a good experimental database for the model; while the latter is useful to individuate the weak points of the monitoring management. In Perugia, it was possible to use only the 1997 and 1998 data for the middle and long-term forecasts, because the lack of traffic—which arises as the main pollution source—resulted decisive. It has to be mentioned that for the short-term forecasts, this choice does not constitute a problem for the ANN, which is able to extract the representation of the inner phenomenon thanks to the pollutants' inputs, introduced in the model in the short-term methodology. In the 1998, the instruments encountered many problems owing to which there were no continuous data available for many pollutants. For particulate and sulphur dioxide, the situation appears particularly serious, especially for the data related to the winter season (which is also the most problematic period because of the larger emissions of these pollutants), because of the lack of validated data in the year 1997 also. Nitrogen oxides and ozone presented lack of data in some long periods, while carbon monoxide results as the pollutant with more available validated data in the two useful years. Benzene has been monitored since the beginning of the 1999; therefore, it was only possible—without traffic inputs—to perform short-term forecasts. ANN reported results have to be seen in the eye of these considerations about the characteristics of the available data time series for the city of Perugia. In such a situation, the ANN forecasts

Table 1
Data matrix given to the ANN in the brief term forecast methodology

$A(t-n_A)$	$A(t-n_A-1)$	···	$A(t-1)$	$A(t)$	$B(t-n_B)$	···	$B(t)$	···	$X(t-n_X)$	···	$X(t)$	$X(t+h)$
$A(t-n_A-1)$	$A(t-n_A-2)$	···	$A(t)$	$A(t+1)$	$B(t-n_B-1)$	···	$B(t+1)$	···	$X(t-n_X-1)$	···	$X(t+1)$	$X(t+h+1)$
...	...	...	...	...	...	...	...	...	...	...	...	...

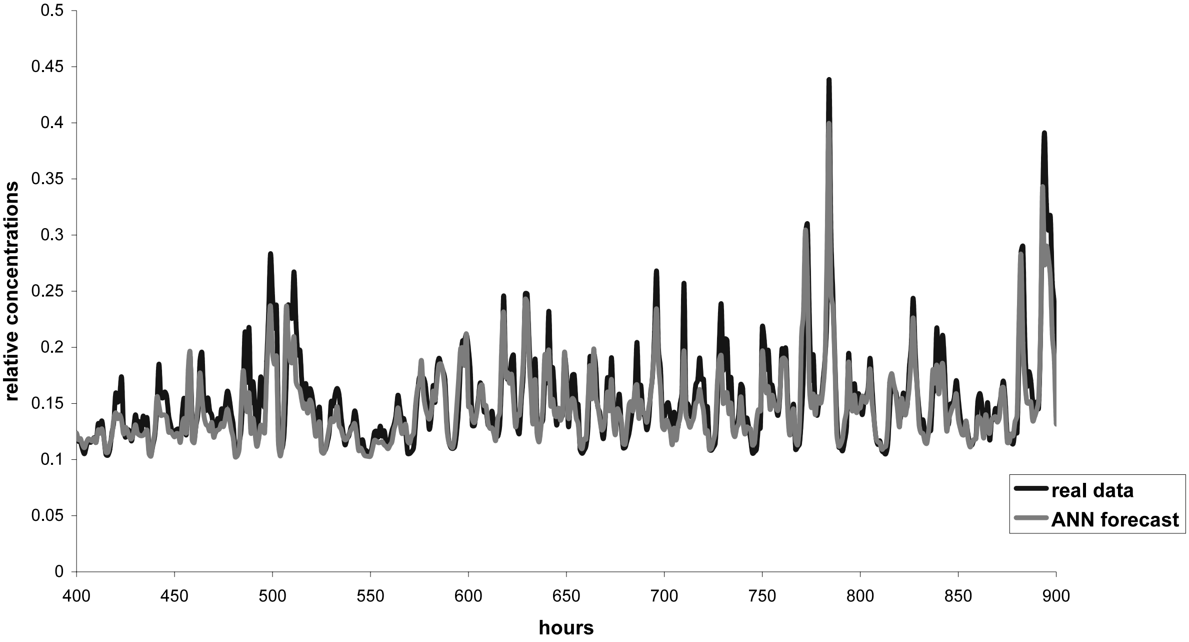

Fig. 4. Benzene: 1 h validation (model A).

can be considered satisfying, as the following results show.

4. Results

4.1. Short-term forecasts

The methodology adopted for the short-term forecasts is the following: a matrix has been given to the ANN as configured in Table 1, where A, B, C… represent pollutants' concentrations, weather parameters, and traffic; X is the pollutant concentration to be forecasted; n_A, n_B,…,n_X represent the number of hours preceding the present hour t and h is the number of hours subsequent to the present hour t (forecast at h hour).

The correct choice of the number of hours preceding the present hour t for each input parameter was performed at each simulation by means of an accurate statistical analysis of the self-correlations and cross-correlations among the variables monitored, adopting the results obtained in the thesis of Celestini (Perugia, Department of Civil and Environmental Engineering, 1999). This way, the pollutant trend forecast using the exclusive input of the past hours, without giving to the network any information belonging to the forecasting time range, was tried. This methodology gives good results, as expected, for the 1 h forecasts, for which the correlations among the variables are still strong. Particularly for benzene the ANN has used at the best its potentiality extracting from only few hourly data (about 1100) and without traffic as the input in the function's inner representation. Some elaborations are reported in Figs. 4–8 and the results together with the ANN models' characteristics are compared in Table 2. For every elaboration, the training set (1100 examples) was used as a test set also (controlling that overtraining did not occur) and a validation set of 990 examples—for a more complete verification—was tried (in the figures mentioned only 500 data for a clearer reading are reported). The input variables were for all the models: one for NO_x, NO, NO_2, CO, O_3 and tsr; and three for benzene. The ANN forecasts resulted as excellent, giving a generalisation of the problem with a MSE less than 10% for all the models (considering that the training phase was operated with a MSE of about 12–16%). The batch training (model A) results better than the incremental one (model B) especially for the computational time: in model B realisation, it occurred for about an hour until it reached a training MSE equal to 0.164 and a validation one equal to 0.96 in comparison with

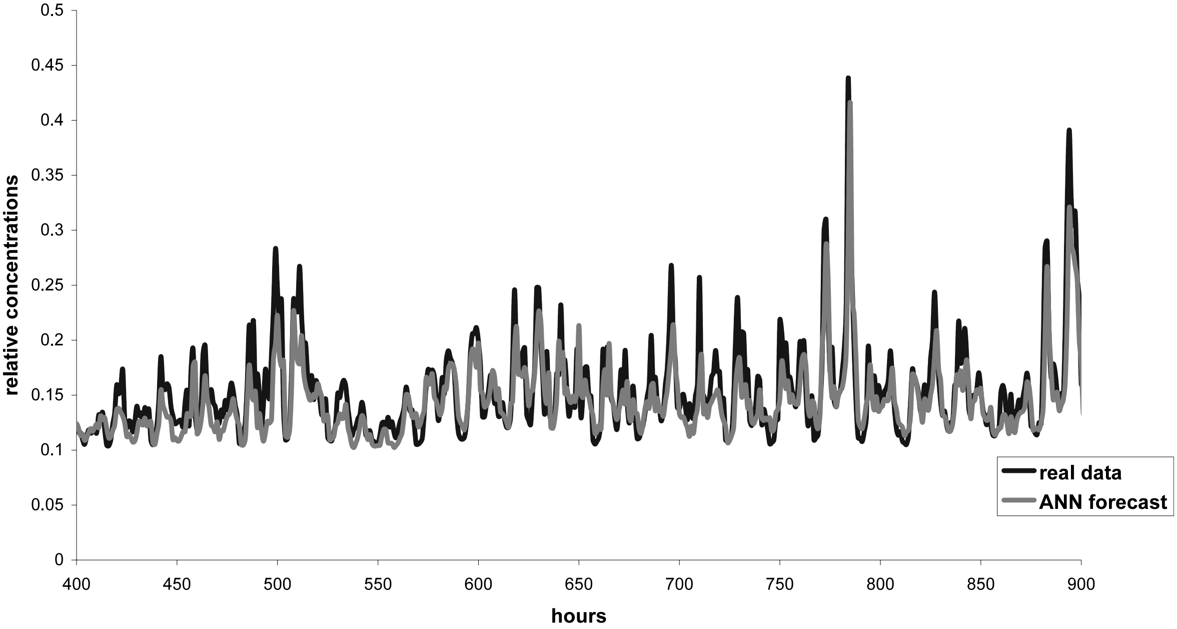

Fig. 5. Benzene: 1 h validation (model B).

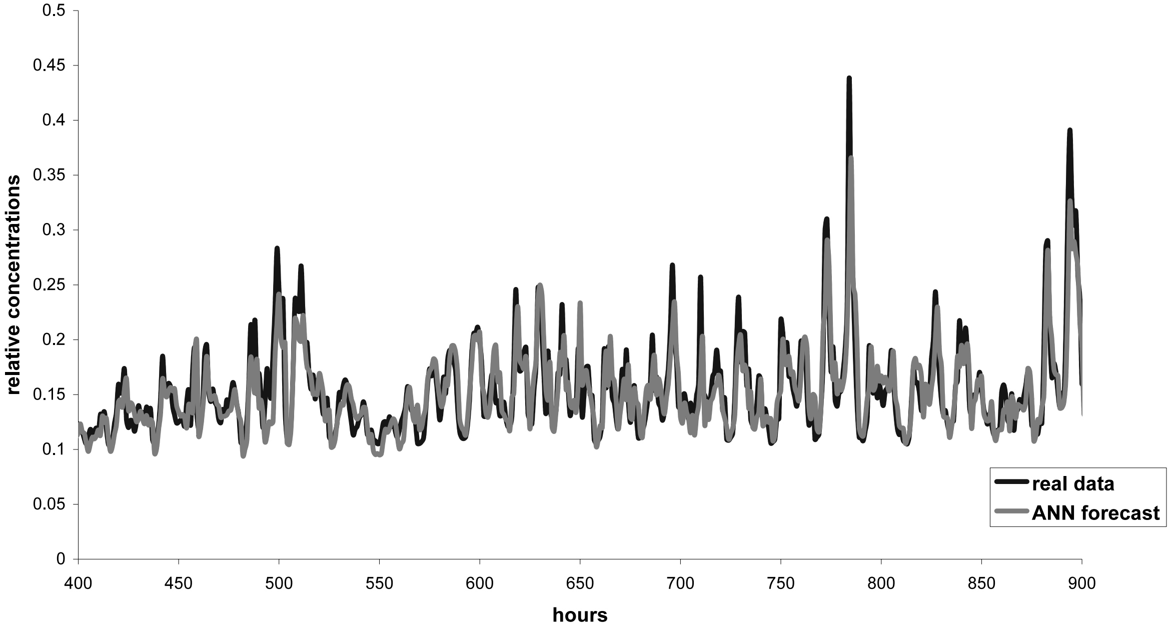

Fig. 6. Benzene: 1 h validation (model C).

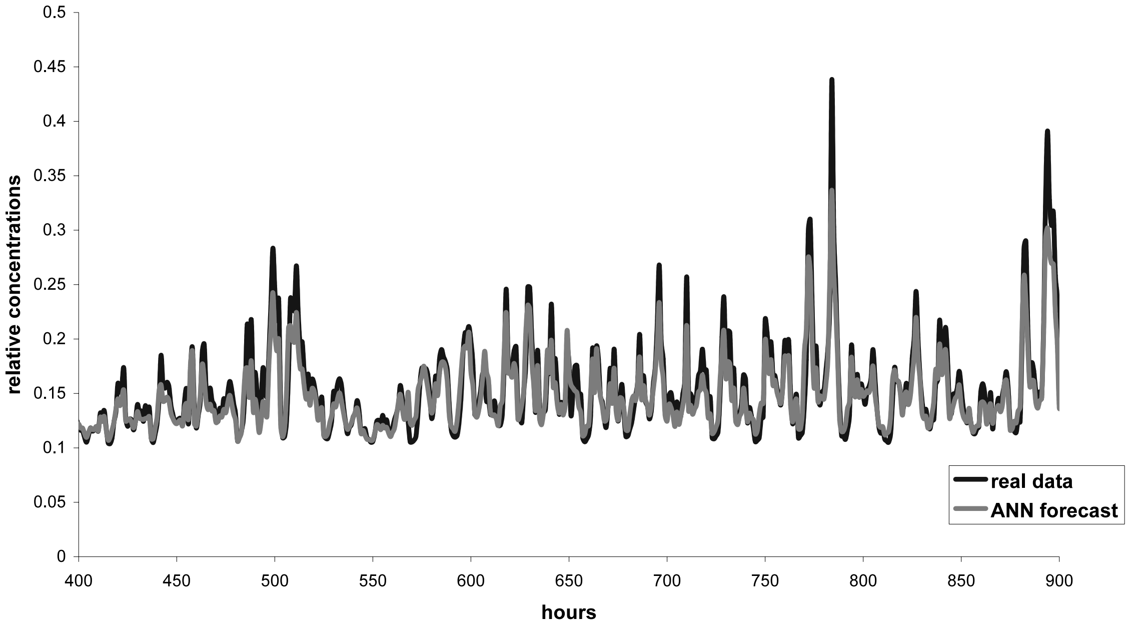

Fig. 7. Benzene: 1 h validation (model D).

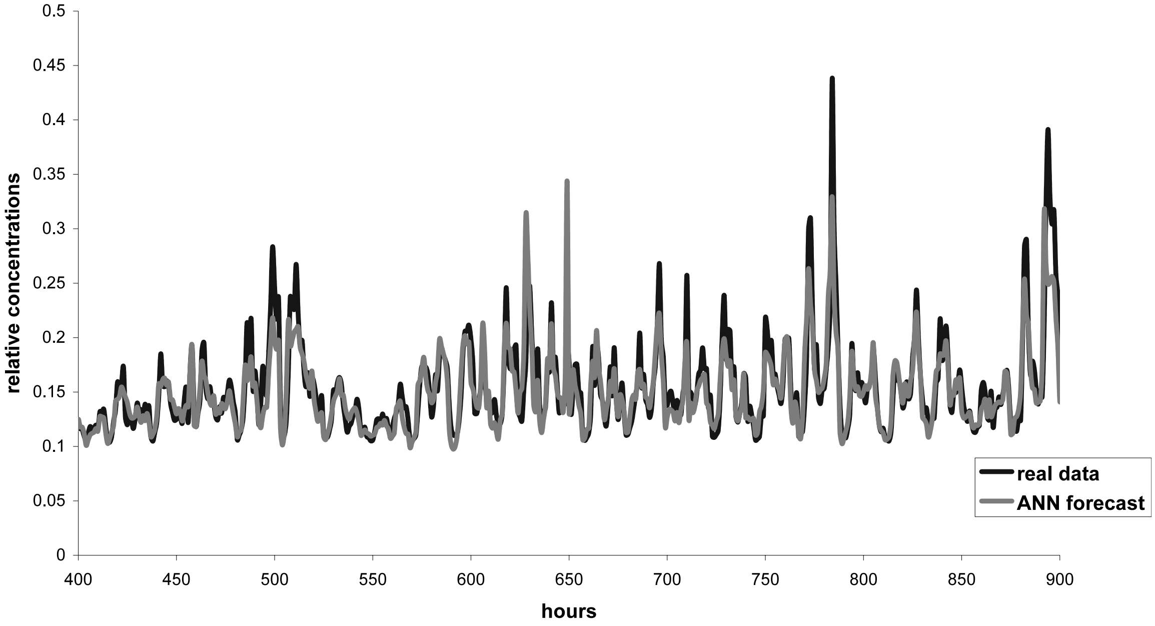

Fig. 8. Benzene: 1 h validation (model E).

about 20 min to gain a training MSE equal to 0.14 and a validation one equal to 0.92 for model A (see Table 2 and Figs. 4 and 5). The procedure with a variable learning rate (Darken and Moody, 1991) during the learning phase (model C) brings not only a little improvement in the MSE (the

ANN fits the curves better) but also a little worsening in the general trend of the forecast (the ANN does not fit the peaks very well). By modifying the scaling of the input variables (model D) the forecast seems lightly worse. It is important to note that with too many hidden neurons (model E) with respect to the input ones the results seem better but the ANN fell in a light overfitting, as showed in the trend comparison (Fig. 8). From these elaborations, it clearly appeared that the MSE is a good indication, but it is more necessary to compare the forecasting trend in order to evaluate the efficacy of a neural model.

Table 2
Benzene: 1 h forecasts results

	Model A	Model B	Model C	Model D	Model E
Training error	0.140	0.164	0.140	0.142	0.127
Input-hidden-output neurons	9-3-1	9-3-1	9-3-1	9-3-1	9-8-1
Data scaling [a]	(I)	(I)	(I)	(II)	(I)
Learning [b]	B	I	B	B	B
Learning rate [c]	$\eta = 0.3$	$\eta = 0.3$	$\eta_0 = 1.9$, $\eta_f = 0.17$	$\eta = 0.3$	$\eta = 0.3$
Epochs	50 000	200	50 000	50 000	50 000
Relative MSE	0.0921	0.0960	0.0947	0.0920	0.0917
Right classifications [d]	90.0%	88.8%	90.2%	89.0%	91.4%

[a] (I) represents the scaling among (0.1,0.9) of the data; (II) is the scaling procedure according to the mean value and the standard deviation of the variable distribution.

[b] B = batch mode; I = incremental mode.

[c] η = constant learning rate; decreasing learning procedure (Darken and Moody, 1991): η_0 = initial learning rate, η_f = final learning rate.

[d] According to a tolerance of 0.2 (i.e. they represent all the forecasted values which generate an MSE less by 20%).

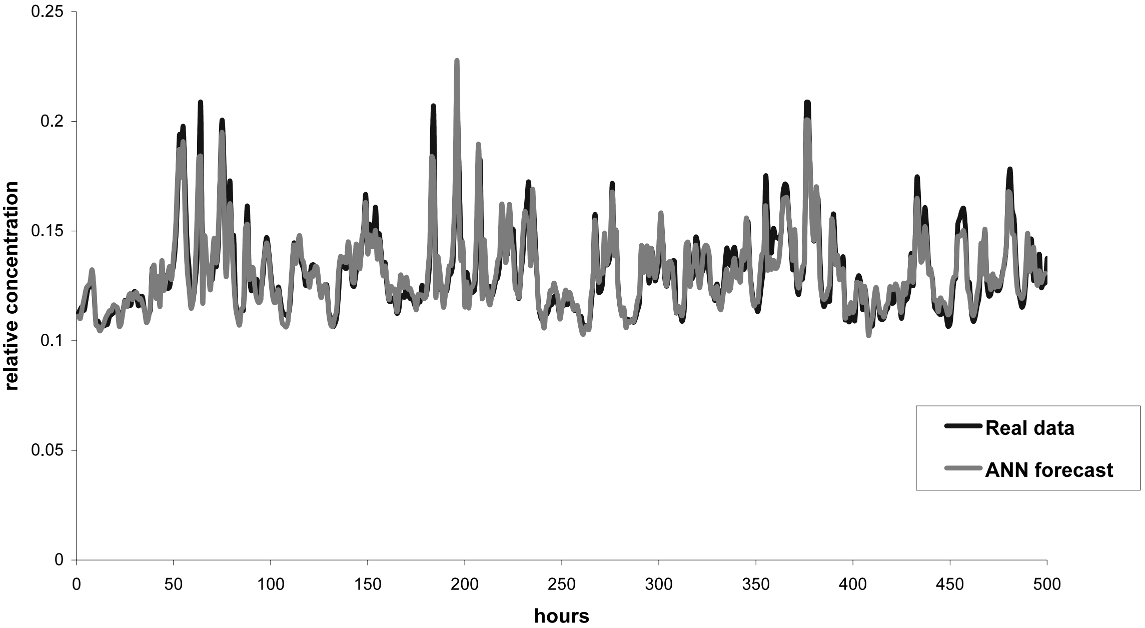

Fig. 9. Carbon monoxide: 1 h validation.

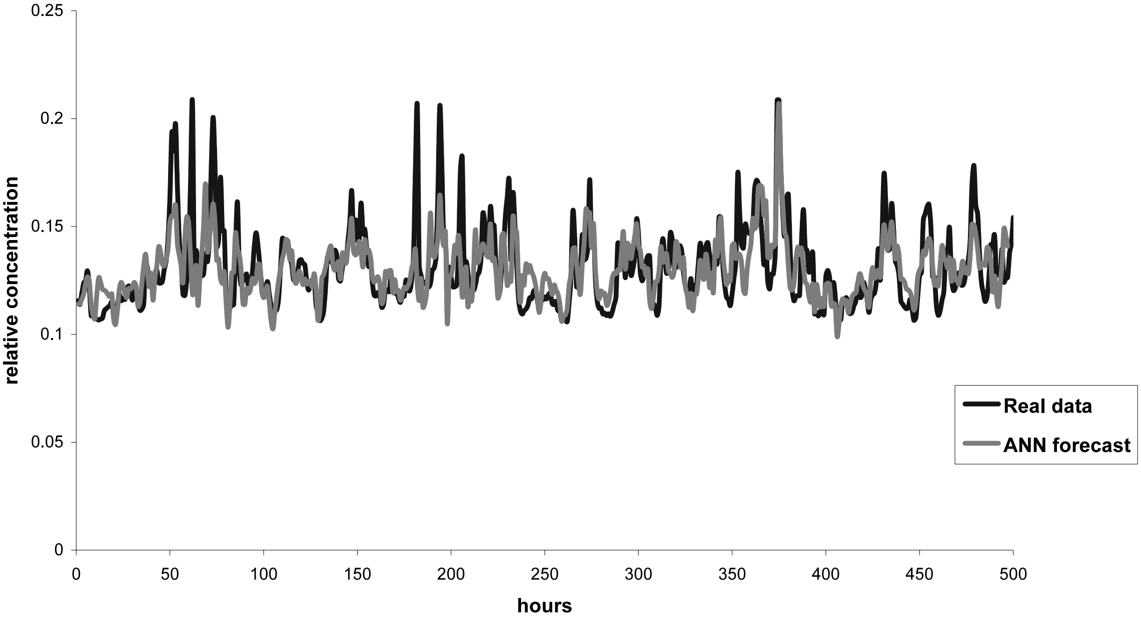

Fig. 10. Carbon monoxide: 3 h validation.

Table 3
Data matrix given to the ANN in the middle-long-term forecast methodology

$A(t-n_A)$	$A(t-n_A-1)$	···	$A(t-1)$	$A(t)$	$B(t-n_B)$	···	$B(t)$	···	$X(t)$
$A(t-n_A-1)$	$A(t-n_A-2)$	···	$A(t)$	$A(t+1)$	$B(t-n_B-1)$	···	$B(t+1)$	···	$X(t+1)$
...	...	...	...	...	...	...	...	...	...

An example concerning the forecast of the carbon monoxide is showed in Fig. 9. The training set for the model was constituted from 5000 patterns and the test set from 1000 patterns, while the validation set were of 500 patterns. The total number of neurons was 76 (70 in the input layer, 5 in the hidden layer and 1 for the output); the learning rate was constant and equal to 0.3. About 10 000 epochs were performed using eight preceding hours taken from NO_x, NO, NO_2, CO, and traffic, six preceding data from SO_2, ozone and wind speed, two past hours from moisture, rain, total radiation and pressure and one hour from temperature and particulate. Though there was a huge quantity of inputs, the ANN performed very good results extracting the main information from the input architecture.

On the other hand, longer time forecasts demonstrated the lower degree of fitness of the ANN with the methodology adopted for short-term ones: e.g. the 3 h forecast of the carbon monoxide, with the same network characteristics of the 1 h forecast, is showed in Fig. 10. Therefore, different long-term forecasting problem planning was tried.

4.2. Middle and long-term forecasts

From the point of view of the applications, they constitute the more interesting forecasts. In order to foresee a pollutant trend for the next day, or the next week, under particular traffic trend or climatic conditions, weather and traffic variables were exclusively used as inputs to the neural net. Thus, it could be possible to evaluate the response of the air pollutants' concentrations under hypothetical or forecasted climatic circumstance (persistent or variable) or under particular traffic (which is the management or intervention variable) conditions. The matrix presented to the ANN according to this second method is represented in Table 3, where *A*, *B*,... are weather and traffic parameters (inputs); *X* is the pollutant concentration (target variable) and n_A, n_B,... are the number of hours preceding the hour *t* for the inputs.

By means of this methodology, letting the network learn the function linking the pollutant concentration, at a certain instant *t*, was tried with the weather or traffic conditions (at the same instant *t* or, at the most, at some previous hours). Although the results can be considered worse than the 1 h forecasts, in terms of absolute error, they present an evident and a more powerful value. Further, it is important to mention that the local administrations and the health and environmental protection institutions are usually more interested in catching the future pollutant trend (a well-performed task by the present perceptron model) rather than the precise concentration value every time. It was chosen to train the ANN with different data sets:

1. with hourly monitored data only;
2. with statistically elaborated data only ('mean day' data);
3. with monitored and statistically elaborated data together and with their different combination in the training and test sets.

The elaboration of the available data has brought to calculate the 'mean' or 'type day' for every month of the year, obtaining 288 data for every parameter (e.g. see Fig. 11).

Simulations carried out with monitored data only (case I) gave good results, even if it is important to mention that it is necessary to choose in this case very large data sets and with more accuracy in order to capture at the best the relationship between the target and the input variables. For further details one can refer to the thesis of Di Genova (Perugia, 2000). An example of NO_2 forecasting is reported in Fig. 12. The training set was of 4500 patterns, the test set of 4500 and the validation set for the model was of 500 patterns. The total number of neurons was of 13 (seven in the input layer, five in the hidden layer and one for the output). About 10 000 epochs were performed with a constant learning rate of 0.3 and one datum from every parameter (wind speed, temperature, moisture, rain, tsr, pressure and traffic) was taken.

As examples for the cases II and III, the elaborations with the 'mean day' data and the hourly

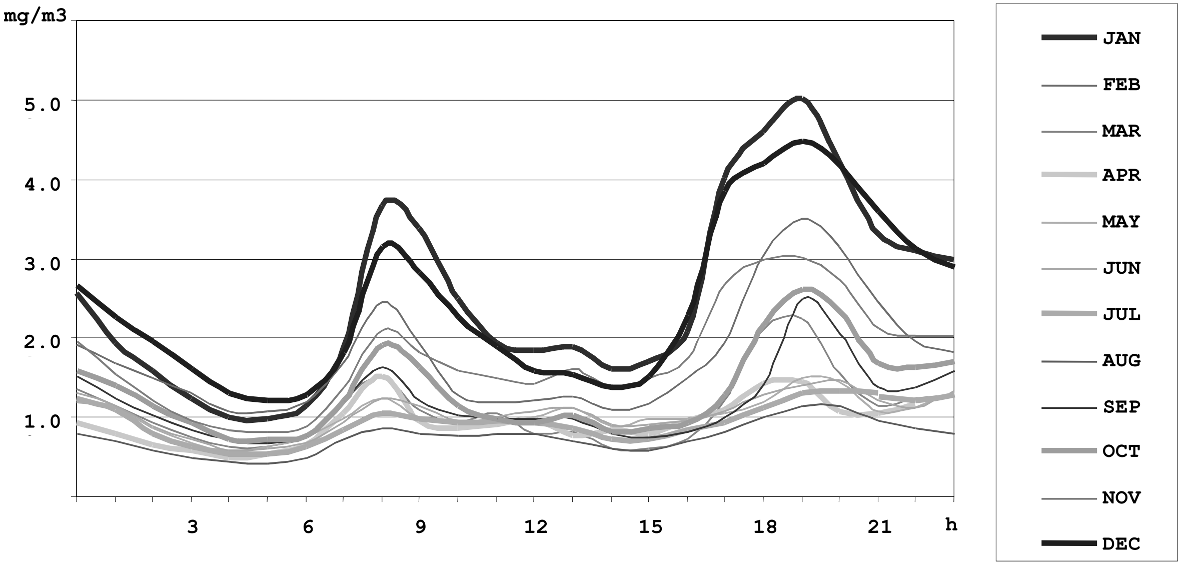

Fig. 11. Carbon monoxide: 'mean' or 'type day' (Perugia year 1997).

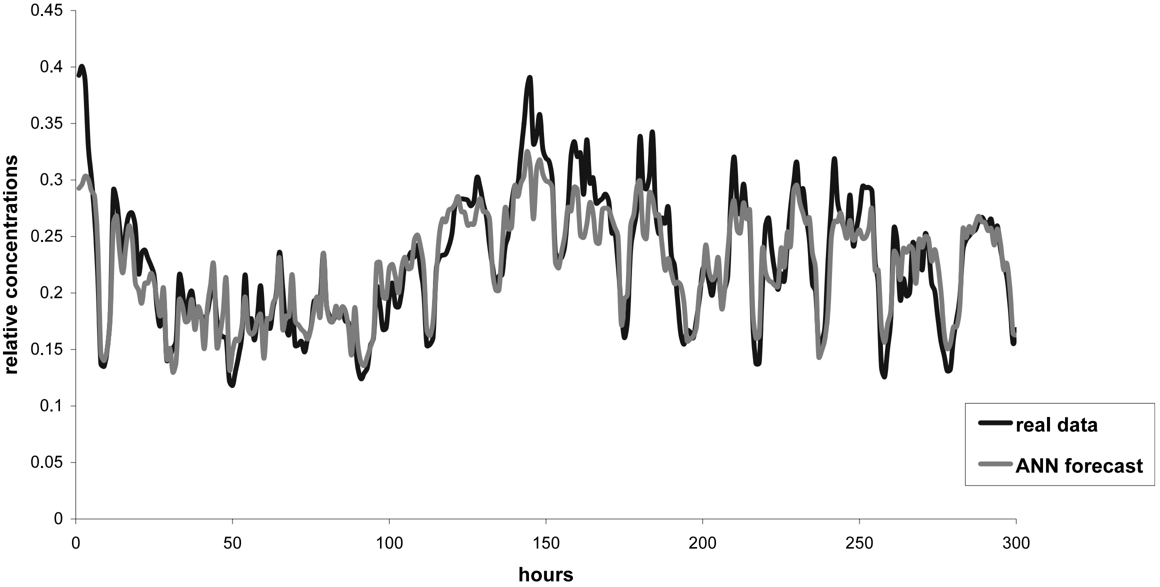

Fig. 12. NO_2: middle long-term validation (model trained with only hourly monitored data).

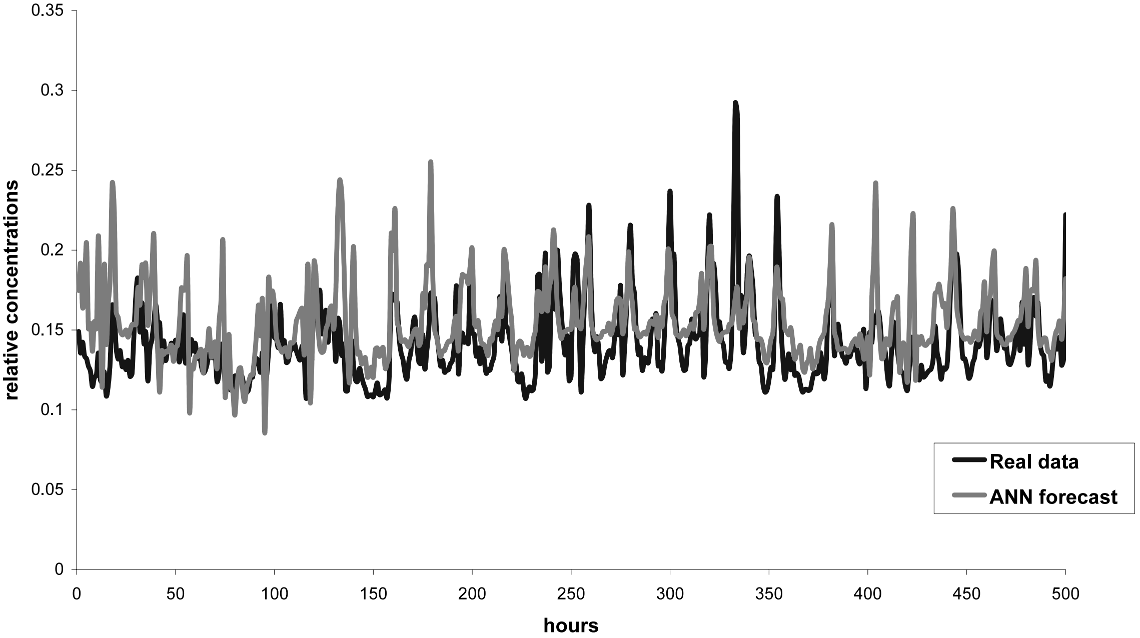

Fig. 13. NO_x: middle long-term validation (model 1).

monitored data to forecast NO_x concentration levels are reported (see Figs. 13–16). The results of the different simulations are compared in Table 4.

The elaborations showed that the use of averaged data only, especially in the present work for which it was available only one year (1997), was not correct to train the network (see models 1 and

2). The best solution seems to be represented by the accurate combination of statistically elaborated data together with the hourly monitored ones. It is also important to choose the more

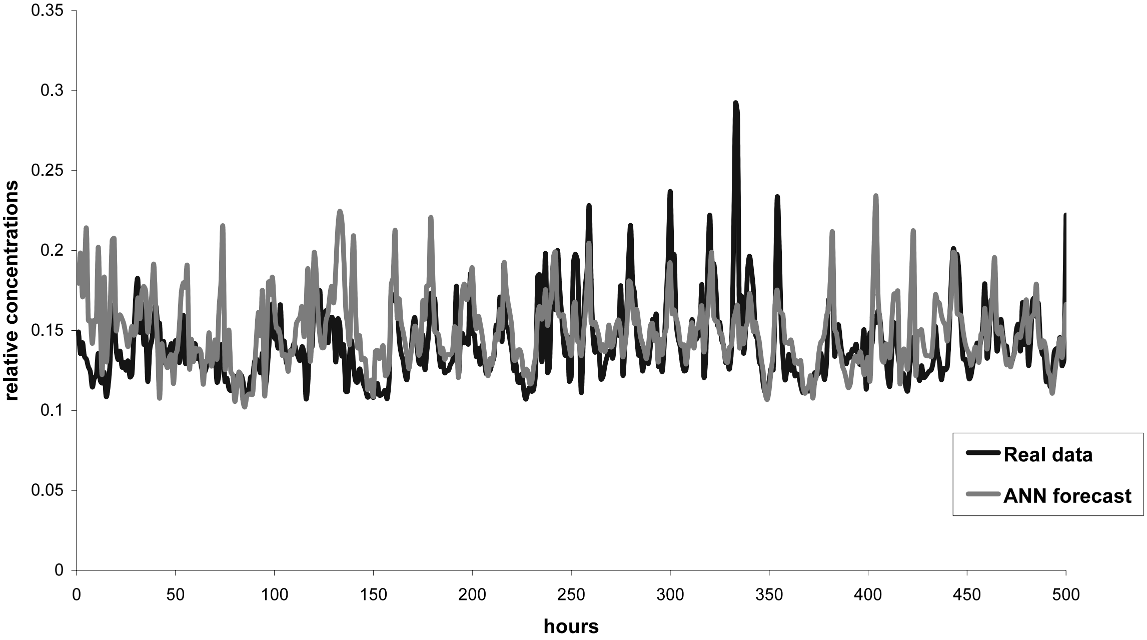

Fig. 14. NO_x: middle long-term validation (model 2).

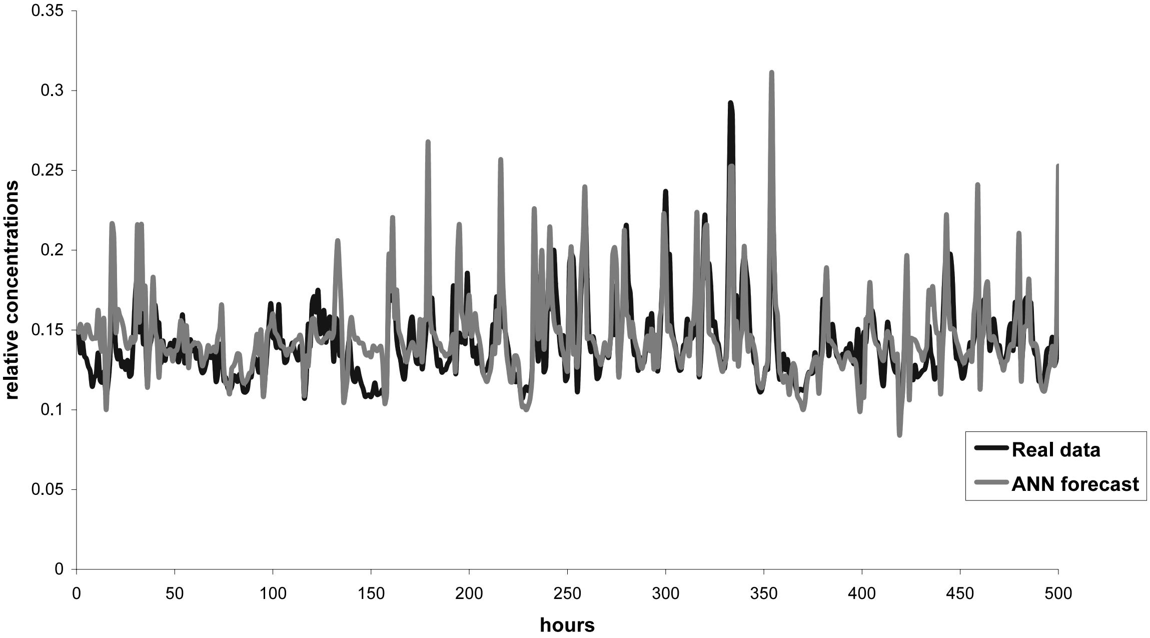

Fig. 15. NO_x: middle long-term validation (model 3).

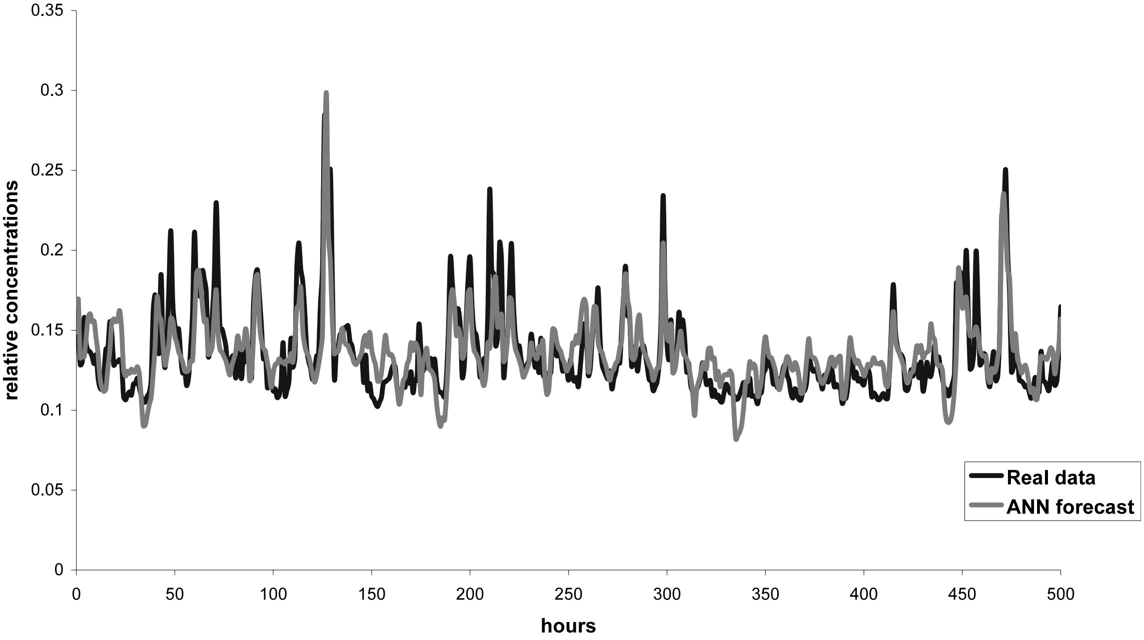

Fig. 16. NO_x: middle long-term validation (model 4).

Table 4
NO_x: long-term forecasts results

	Model 1	Model 2	Model 3	Model 4
Training set [a]	288 (a)	288 (a)	8000 (a,h)	4000 (a,h)
Test set	288 (a)	4000 (h)	8000 (a,h)	4000 (h)
Input variables [b]	1 (w,te,m,p,ts,r,tr)	1 (w,te,m,p,ts,r,tr)	1 (w,te,m,p,ts,r,tr)	4 (w,te,m,p,ts,tr)
Input-hidden-output units	7-5-1	7-5-1	7-5-1	24-5-1
Relative MSE	0.1422	0.1207	0.094	0.088
Right classifications	76.2%	82.8%	88.2%	92.4%

[a] (a) = averaged data only; (h) = hourly monitored data only; (a,h) = averaged and monitored data together.

[b] It is intended to set the number of inputs for every single parameter separately; between parenthesis are indicated the used parameters for the simulation: w = horizontal wind speed, te = temperature, m = moisture, p = pressure, ts = total sun radiation, r = rain, tr = traffic.

significant parameters among the meteorological ones for the present pollutant: in model 4 the rain data were not used because the cross-correlation coefficient between NO_x and rain was low with respect to NO_x and the other meteorological parameters, as reported by Celestini in his thesis (Perugia, 1999). It is noted that it is advisable to try to give more data related to the preceding hours for each parameter, i.e. more input neurons (in model 4 four inputs for every parameter were used) without exceeding in the specific case the number of inputs for the net architecture. For all the models 10 000 epochs were performed with a constant learning rate equal to 0.3.

Ozone concentrations are more difficult to be foreseen because of the complex mechanisms, which regulate the dynamics of this pollutant in atmosphere, classified as secondary. Therefore, it is very problematic to forecast ozone levels without information about its precursors. The following simulation, among the many performed, can be in reality considered good (see Figs. 17 and 18). It used a training set of 3500 patterns (including 288 patterns of the 'mean day'), a test set of

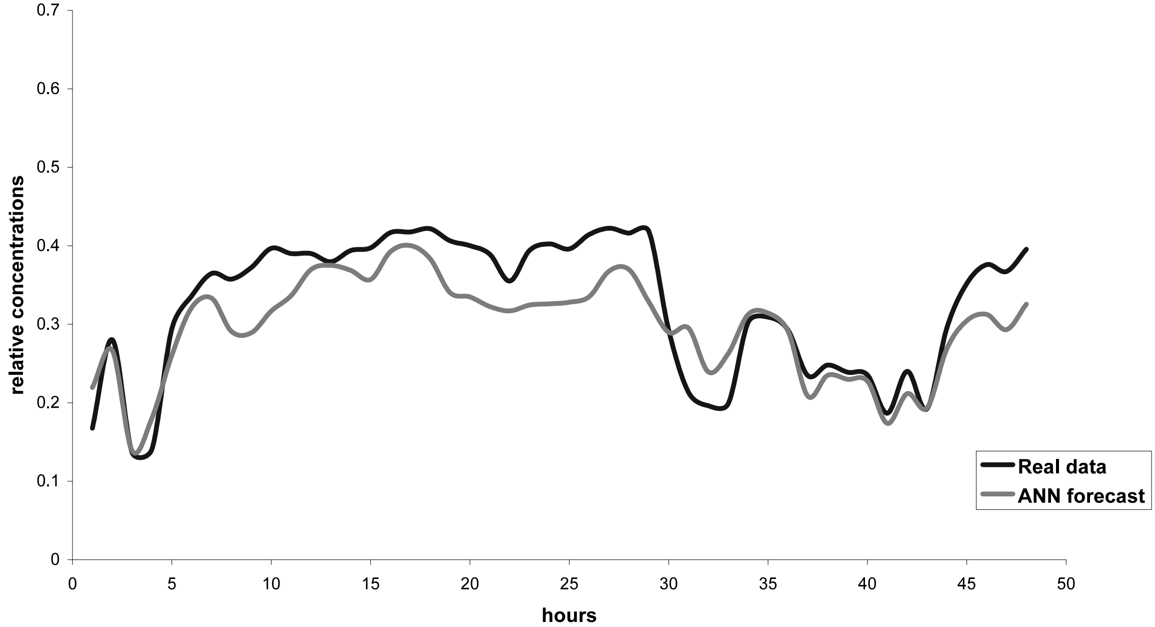

Fig. 17. Ozone: middle long-term validation (48 h).

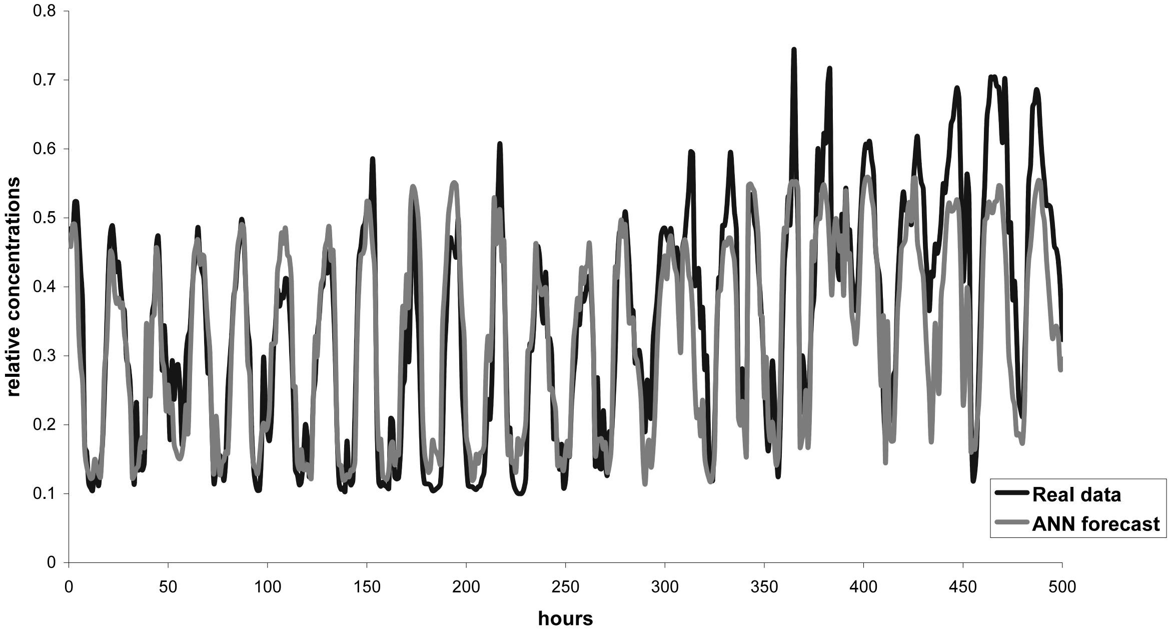

Fig. 18. Ozone: middle long-term validation (500 data).

3500 patterns and two validation sets (500 and 48 patterns). The number of neurons was 13 (seven as inputs, five in the hidden layer and one as output). About 10 000 epochs were performed at a constant learning rate of 0.3. The results for the two validations reported a relative MSE equal to 0.126 (48 h) and a relative MSE equal to 0.19 (500 h).

5. Discussion

Although the monitored period considered (2 years, i.e. only two time series) seems to be quite short and the lack of continuous hourly data occurred to many parameters, from the present study, it has appeared that the ANN with a single hidden layer based on the standard backpropagation algorithm described above, using eventually only the simple sigmoid (Eq. (5)) as activation function, resulted as a very efficient model to forecast both short and middle long-term air pollutant concentrations in urban area, especially for primary pollutants (i.e. substances directly connected to the emission sources, e.g. carbon monoxide, particulate, benzene). The data processing required a dimensionless input and target variables presented to the network: scaling (0.1,0.9) of the variables, as used largely for standard backpropagation based ANN—provided that it does not concern classification tasks as in our problem (Haykin, 1994)—represented, from the different experimented alternatives, a rapid solution in the learning phase and a quite good result in the fitting of the function to approximate.

6. Conclusions

The perceptron with backpropagation algorithm model have shown very good performances for the 1 h forecasts. It is necessary to mention that in order to use the model for forecasting aims (both short and middle long-term forecasts) single pollutant ANNs have to be built. For the middle (24 h) and long-term forecasts, ANNs can be used introducing hypothesis about the values of the meteorological and traffic parameters. In this case, although the ANN forecasts appear to be worse than the 1 h ones, their results, in term of MSEs, are better than the usual deterministic models ones and furthermore, they are more rapid in the forecasting phase. For local administrations and health and environmental protection institutions, which are usually more interested in catching the future pollutant trend rather than the precise concentration value, this methodology appears to be very useful. The ANN has given good results in the middle and long-term forecasting of almost all the pollutants. A shrewd preliminary analysis of the available data appears fundamental because it can give additional input for the learning phase (as the mean day data) and good indications about the time series (self- and cross-correlations).

References

Antonic, O., Hatic, D., Krian, J., Bukocev, D., 2001. Modelling groundwater regime acceptable for the forest survival after the building of the hydro-electric power plant. Ecol. Model. 138, 277–288.

Berry, M.J.A., Linoff, G., 1997. Data Mining Techniques. Wiley, New York.

Bertsekas, D.P., 1995. Non-linear Programming. Athena Scientific, Belmont, MA.

Bertsekas, D.P., Tsitsiklis, J.N., 1996. Neuro-Dynamic Programming. Athena Scientific, Belmont, MA.

Bishop, A., 1995. Neural Networks for Pattern Recognition. Oxford University Press, Oxford, UK.

Blum, A., 1992. Neural Network in C + + . Wiley, New York.

Boger, Z., Guterman, H., 1997. Knowledge extraction from artificial neural network models. IEEE Systems, Man and Cybernetics Conference, Orlando, FL.

Carosone, F., Cenedese, A., Viotti, P., 1996. Stima di portate fluviali da serie storiche di precipitazione mediante reti neurali. Gruppo Nazionale Difesa Catastrofi Idrogeologiche, Scritti in onore di Mario Ippolito, Napoli, 16–17 Maggio 1996.

Darken, C., Moody, J., 1991. Note on learning rate schedules for stochastic optimisation. In: Lippman, R.P., Moody, J.E., Touretzky, D.S. (Eds.), pp. 832–838.

Giustolisi, O., Mastrorilli, M., 1994. Realizzazione di un modello connessionista applicato a un problema di idrologia urbana. XXIV Conv. di Idraulica e Costruz. Idrauliche, Italy.

Haykin, P., 1994. Neural Networks, A Comprehensive Foundation. Macmillan, New York.

Hecht-Nielsen, R., 1990. Neurocomputing. Addison-Wesley, MA.

Heymans, J.J., Baird, D., 2000. A carbon flow model and network analysis of the northern Benguela upwelling system, Namibia. Ecol. Model. 126, 9–32.

Karul, C., Soyupak, S., Çilesiz, A.F., Akbay, N., Germen, E., 2000. Case studies on the use of neural networks in eutrophication modelling. Ecol. Model. 134, 145–152.

Lapedes, A., Farber, R., 1987. Non-linear signal processing using neural networks. Technical Report no. LA-UR-2662 Los Alamos National Laboratory, Los Alamos, NM.

Masters, T., 1995. Advanced Algorithms for Neural Networks: A C+ + Source Book. Wiley, New York.

Mossone, M., Segoni, M., Barbagianni, N., Peirone, E., Salciarini, S., Pompei, M., Areni, M., 1999. La rete di monitoraggio della qualità dell'aria del Comune di Perugia: prima relazione annuale 1998–1999. Laboratorio Chimico U.S.L. n.2 di Perugia, Perugia.

Pelliccioni, A., Pessa, E., 1997. Methodological considerations on use of perceptron model to forecast CO and NO_2 concentrations in urban areas. In: Power, H., Tirabassi, T., Brebbia, C.A. (Eds.), Air Pollution V: Modelling, Monitoring and Management. Computational Mechanics Publications, pp. 667–676.

Ripley, B.D., 1996. Pattern Recognition and Neural Networks. Cambridge University Press, Cambridge.

Swingler, K., 1996. Applying Neural Networks: A Practical Guide. Academic Press, London.

Werbos, P., 1988. Generalization of Backpropagation with application to a recurrent gas market model. Neural Networks 1, 339–356.