Local Maximum Ozone Concentration Prediction Using LSTM Recurrent Neural Networks

Sabrine Ribeiro René Alquézar
Dept. Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Campus Nord, Mòdul C6, Jordi Girona 1-3,
08034, Barcelona, Spain.
{eslopes,alquezar}@lsi.upc.es

Abstract

This work describes the application of a powerful Recurrent Neural Network (RNN) approach, the Long Short Term Memory (LSTM) architecture proposed by Hochreiter and Schmidhuber, to a signal forecasting task in an environmental domain. More specifically, we have applied LSTM to the prediction of maximum ozone(O_3) concentrations in the East Austrian region. In this paper the results of LSTM on this task are compared with those obtained previously using other types of neural networks (Multi-layer Perceptrons (MLPs), Elman Networks (ENs) and Modified Elman Networks (MENs)) and the Fuzzy Inductive Reasoning (FIR) methodology. The different models used were ozone, temperature, cloud cover and wind data. The performance of the best LSTM networks inferred are equivalent to the best FIR models and slightly better than the other Neural Networks (NNs) studied (MENs, ENs and MLPs, in decreasing order of performance).

Keywords: Recurrent neural networks, Long Short Term Memory, Forecasting, Environmental modeling, Ozone concentration.

1 Introduction

This work deals with the prediction of ozone O_3 gas, which is considered one of the most common and damaging air contaminants. Air pollution is an important environmental problem that has worsened in most large cities, a situation driven by population growth, industrialization, and increased vehicle use.

Pure air is a gas mixture composed of 78% of nitrogen, 21% of oxygen and 1% of different components such as carbon dioxide, argon and ozone. The atmospheric contamination is defined as any change in the equilibrium of these components that produces a modification of the physical and chemical properties of the air. O_3 is a secondary pollutant which is formed in the troposphere when sunlight causes complex photochemical reactions involving two or more primary contaminants such as volatile organic compounds and nitrogen dioxide. The health consequences of exposure to high concentrations of O_3 are dangerous, since it can impair lung function and cause irritation of the respiratory system and eyes.

In order to provide adequate early warnings, it is valuable to have accurate and reliable forecasts of future high ozone levels. Therefore, the construction of ozone models that capture as precisely as possible the behavior of this gas in the atmosphere is of great interest for environmental scientists and government agencies.

In recent years, some studies using different approaches have been carried out to obtain models for forecasting ozone levels in local regions [1, 5, 13, 15, 16]. In particular, different types of neural networks, both feed-forward and partially recurrent nets, were investigated in [15] for predicting the maximum ozone concentration in the East Austrian region. More recently, new results have been reported on the same data using the Fuzzy Inductive Reasoning (FIR) methodology [5].

The main goal of this work has been to study the prediction performance on the same task of a powerful type of recurrent neural network called Long Short Term Memory (LSTM) and to compare its results with those reported previously on the same data. The LSTM architecture was proposed by Hochreiter and Schmidhuber [7] to overcome error back-flow problems that are typical in most recurrent neural networks (RNNs).

RNNs can be partially or fully recurrent. The former [6] are feed-forward nets that include feedback connections from a set of units called context units and

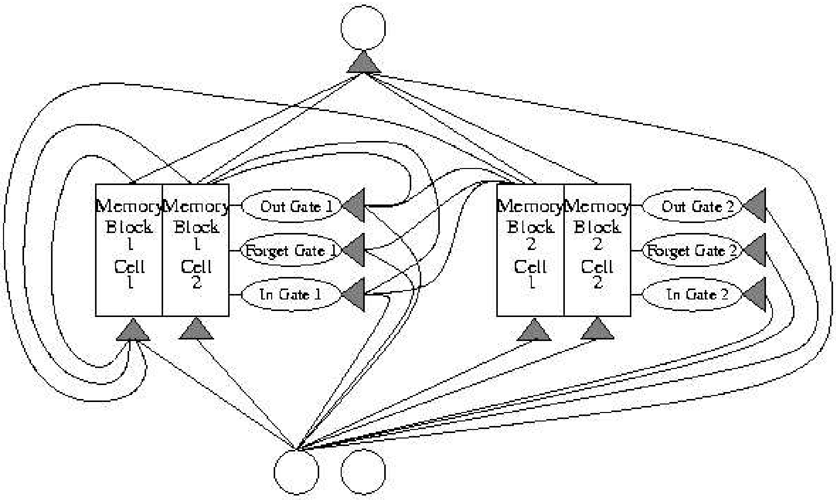

Figure 1: Example of LSTM net with 2 memory cell blocks of size 2 (from [3]).

are trained by conventional back-propagation methods, that do not include any recurrence term in the learning rule. These models are able to learn simple tasks that only need very short term memory, since their training algorithm does not accomplish a real gradient descent calculation with respect to the weights, but rather makes only an approximation, which is based on not considering the terms originated by the influence of the weights in previous time steps (*truncated gradient*). The Elman Network (EN) [2] and the Modified Elman Network (MEN) [10] are among these partially recurrent architectures.

The fully recurrent nets are trained by gradient-descent based algorithms that correctly calculate the real gradient of the error [9]. In practice they have some drawbacks, since it has been demonstrated that the the error signal “flowing backwards in time” in gradient-based algorithms (Back-Propagation Through Time (BPTT), Real-Time Recurrent Learning (RTRL) and others [9]) tend to either, blow up or vanish [7]. The most usual case is the latter (if a sigmoid activation function is used, for example), in which the gradient magnitude decreases exponentially in time, preventing the net from learning long-term dependencies and from reaching the optimal task performance.

LSTM solves the vanishing gradient, because the combination of its architecture and its gradient descent training algorithm facilitates a constant error flowing in time. LSTM have been quite successfully applied to standard benchmarks related to classification problems [7, 4] and more recently to signal forecasting problems [3, 11].

In the following section we describe the LSTM architecture. In section 3 the experimental procedure is described and the obtained results are presented in section 4. Finally, some conclusions are given in section 5.

2 The LSTM Approach

LSTM [7, 4] belongs to a class of recurrent networks that has time-varying inputs and targets. That is, points in the time series or input sequence are presented to the network one at a time. The network can be asked to predict the next point in the future or classify the sequence or to perform some dynamic input/output association. Error signals will either be generated at each point of the sequence or at the end of the sequence.

A fully connected LSTM architecture is a three-layer neural network composed of an input layer, a hidden layer and an output layer. The hidden layer has a feedback loop to itself, i.e., at time step t of a sequence with n time steps, presented to the network, the hidden layer receives as input the activation values of the input layer and the activation values of the hidden layer at time step $t-1$. The basic unit in the hidden layer is known as a memory cell block. Figure 1 illustrates an LSTM net with a fully connected hidden layer consisting of two memory blocks, each one consisting of two cells. The LSTM showed has an input dimension of two and an output dimension of one. Only a limited subset of connections are shown. A memory cell block (Figure 2) consists of S memory cells and three multiplicative gates, called the input gate, output gate, and forget gate. Each memory cell has at its core a recurrently self-connected linear unit called “Constant Error Carousel” (CEC), whose activation is called the cell state. The CECs solve the vanishing error problem: in the absence of a new input or error signals to the

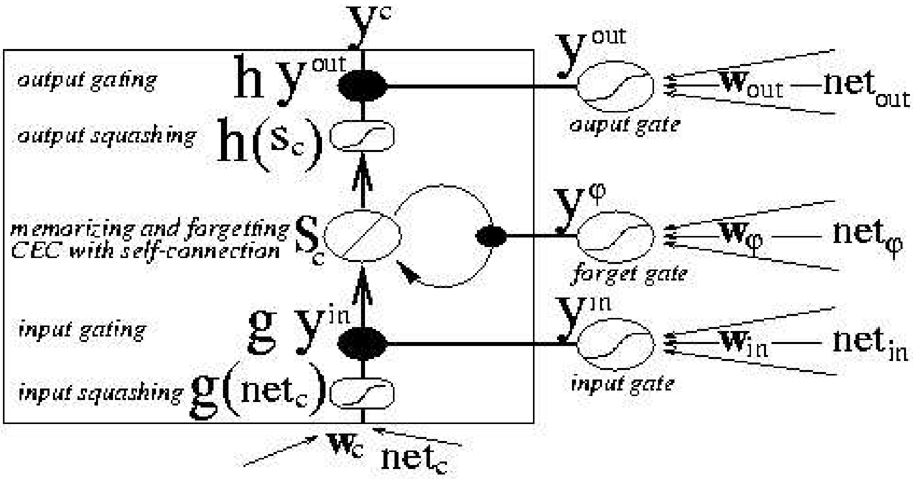

Figure 2: Memory block with its respective gates (from [3]).

cell, the CEC's local error back flow remains constant, neither growing nor decaying. Input and output gates regulate write and read access to a cell whose state is denoted S_c. The CEC is protected from both flowing activation and backward flowing error by the input and output gates respectively. When gates are closed (activation around zero), irrelevant inputs and noise do not enter the cell, and the cell state does not perturb the remainder of the network. The forget gate feed the self-recurrent connection with its output activation and is responsible for do not allow the internal state values of the cells grow without bound by resetting the internal states S_c as long as it needs. In addition to the self-recurrent connection, the memory cells receive input from input units, other cells and gates.

While the cells are responsible for maintaining information over long periods of time, the responsibility for deciding what information to store, and when to apply that information lies with the input and output gate units, respectively.

A single step involves the update of all units (forward pass) and the computation of error signals for all weights (backward pass). At time t, input gate activation $y^{in_j}(t)$ and output gate activation $y^{out_j}(t)$ are computed as follows:

$$\begin{aligned} net_{in_j}(t) &= \sum_m w_{in_j m} y^m(t-1), \\ y^{in_j}(t) &= f_{in_j}(net_{in_j}(t)) \end{aligned} \tag{1}$$

$$\begin{aligned} net_{out_j}(t) &= \sum_m w_{out_j m} y^m(t-1), \\ y^{out_j}(t) &= f_{out_j}(net_{out_j}(t)) \end{aligned} \tag{2}$$

The w_{jm} is the weight on the connection from unit m to unit j. The summation indices m may stand for input units, memory cells, or even conventional hidden units if there are any. All these different types of units may convey useful information about the current state of the net.

The forget gate activation $y^{\varphi_j}(t)$ is calculated like the other gates above:

$$\begin{aligned} net_{\varphi_j}(t) &= \sum_m w_{\varphi_j m} y^m(t-1), \\ y^{\varphi_j}(t) &= f_{\varphi_j}(net_{\varphi_j}(t)) \end{aligned} \tag{3}$$

where net_{φ_j} is the input from the network to the forget gate. For all gates, the squashed function f is the logistic sigmoid with range [0,1].

For $t > 0$, the internal state of the memory cell $S_c(t)$ is calculated by adding the squashed, gated to the input to the state at the previous time step $S_{c_j^v}(t-1)$, which is gated by the forget gate activation:

$$net_{c_j^v}(t) = \sum_m w_{c_j^v m} y^m(t-1),$$
$$S_{c_j^v}(t) = y^{\varphi j}(t) S_{c_j^v}(t-1) + y^{in_j}(t) g(net_{c_j^v}(t)) \tag{4}$$

where j indexes memory blocks, v indexes memory cells in block j, such that c_j^v denotes the v-th cell of the j-th memory block. The cell initial state is given by $S_{c_j^v}(0) = 0$.

The cell's input squashing function g used is a sigmoid function with range $[-1, 1]$. The cell output y^c is calculated by squashing the internal state S_c via the output squashing function h and then multiplying (gating) it by the output gate activation y^{out}.

$$y_{c_j^u}(t) = y^{out_j}(t)\ h(s_{c_j^u}(t)) \tag{5}$$

Here we used the identity function as output squashing function h.

Lastly, assuming a layered network topology with a standard input layer, a hidden layer consisting of memory blocks, and a standard output layer. The equations for the output units k are:

$$\begin{aligned} net_k(t) &= \sum_m w_{km} y^m(t-1), \\ y^k(t) &= f_k(net_k(t)) \end{aligned} \tag{6}$$

where m ranges over all units feeding the outputs units. As activation functions f_k use the linear identity function.

LSTM's backward pass [7] is basically a fusion of slightly modified truncated backpropagation through time (BPTT) [14], which is obtained by truncating the backward propagation of error information, and a customized version of RTRL [12] which properly takes into account the altered dynamics caused by input and output gates (see details in appendix of [7]).

LSTM's learning algorithm is local in space and time; its computational complexity per time step and weight is $O(1)$, that means $O(n^2)$ where n is the number of hidden units if we measure the complexity per time step. This is very efficient in comparison to the RTRL algorithm. The time step complexity is essentially that of BPTT, but unlike BPTT, LSTM only needs to store the derivatives of the CECs, this is a fixed-size storage requirement independent of the sequence length.

3 Experimental Procedure

The data available for this study is the same used by [15, 5] and stem from the East Austrian region. It was registered, mostly, in summer season, since O_3 occurs in highest concentrations during this season. The ozone values, measured in *ppb* (parts per billion), are calculated as the average of five measurement points providing three hours fixed average values. Additionally to the O_3 measurements, other three measures were provided, which are temperature (K), cloud cover (ranging from 0, *no clouds*, to 1 *completely cloudy*) and wind speed (m/s). These last values were originated from the weather prediction model of the European Center for Medium Range Weather forecast.

Ozone and weather data were available for two periods. The longest period, which covers the period 1-5-1996 to 30-9-1996 (149 values), was used as training set for inferring the models. The other one, which covers the period 7-7-1995 to 25-9-1995 (81 values), was used as test set to estimate the forecasting performance of the inferred models, that is to estimate their generalization ability.

Hence, in order to predict the maximum ozone concentration of a given day $Oz(t)$ (associated with discrete time step t), the models can make use of the ozone measurements from the previous days as well as the temperature, cloud cover and wind values both from the previous days an the current day. That is, given four finite sequences $Oz(1), \ldots, Oz(t-1)$ (ozone), $T(1), \ldots, T(t)$ (temperature), $C(1), \ldots, C(t)$ (cloud cover) and $W(1), \ldots, W(t)$ (wind), predict the value $Oz(t)$ of the maximum ozone concentration for the current day. For all the tested approaches (FIR, MLPs, ENs, MENs and LSTM), different models corresponding to different choices of the system input values were assessed. Specifically, Table 1 displays the set of input variables used in the predictions of the ozone at time t for each model, where the symbol $\bullet$ means that the corresponding value was used. However, it must be noted that, while FIR and MLPs are not capable to remember any previous information apart from the given input values, RNNs are theoretically capable of storing useful information from the observed past values (ENs and MENs to a limited extent and LSTM for much longer time lags) that may help to improve the predictions.

To prepare the data conveniently for LSTM, we have replaced the original target output $Oz(t)$ by the difference between $Oz(t)$ and $Oz(t-1)$ multiplied by a scaling factor fs, so that the target is calculated as $tg(t) = fs * (Oz(t) - Oz(t-1)) = \triangle Oz(t) * fs$ and fs scales $\triangle Oz(t)$ between -1 and 1. This scheme is illustrated in Figure 3.

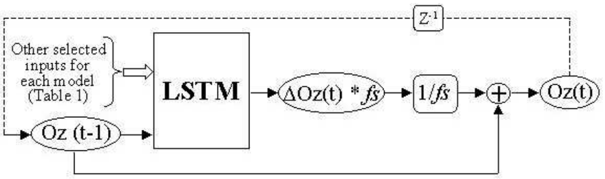

Figure 3: Setup for the output signals.

For each LSTM net that was built, there was one output unit and as many input units as required to introduce the values shown in Table 1 for each model. Nevertheless, for every model, just one hidden layer with j memory blocks of size 1 was used. This is a free parameter that must be set up aiming at finding a configuration that could result in the best overall performance for each model. Specifically, Table 2 shows the number of memory blocks selected for each model as well as the learning rate, momentum and number of epochs parameters that were given to the training algorithm.

For all the LSTM nets, the input units were fully connected to the hidden layer. The cell outputs were fully connected to the cell inputs, to all gates, and to the output unit. All gates, the cell itself and the output unit were biased.

The bias weights were fixed for input and output gates in successive blocks as: -0.5, -1.0, -1.5, and so

	Ozone				Temperature				Cloud				Wind			
Model	$t-3$	$t-2$	$t-1$	t	$t-3$	$t-2$	$t-1$	t	$t-3$	$t-2$	$t-1$	t	$t-3$	$t-2$	$t-1$	t
OCW1	–	–	●	–	–	–	–	–	–	–	–	●	–	●	–	●
OTCW1	–	–	●	–	–	●	●	–	–	–	–	●	–	–	–	●
OT1	–	–	●	–	–	–	–	●	–	–	–	–	–	–	–	–
OTW1	–	–	●	–	–	–	–	●	–	–	–	–	–	●	–	–
OT2	–	●	●	–	–	–	●	●	–	–	–	–	–	–	–	–
OTC1	–	–	●	–	–	–	–	●	–	●	–	–	–	–	–	–
OTCW2	–	●	●	–	–	–	–	●	–	–	–	●	–	–	–	●
OT3	–	–	●	–	–	–	●	●	–	–	–	–	–	–	–	–
OT4	–	–	●	–	●	–	–	●	–	–	–	–	–	–	–	–
OT5	–	●	●	–	–	–	–	●	–	–	–	–	–	–	–	–
OTCW3	–	–	●	–	–	–	–	●	–	–	–	●	–	–	–	●
OT6	–	●	●	–	–	●	–	●	–	–	–	–	–	–	–	–
OTCW4	–	●	●	–	–	–	●	●	–	–	●	●	–	–	●	●
OT7	●	●	●	–	–	●	●	●	–	–	–	–	–	–	–	–
OTCW5	–	–	●	–	–	●	●	●	–	●	●	●	–	●	●	●
OTCW6	–	–	●	–	–	●	●	–	–	●	●	–	–	●	●	–

Table 1: Models setup.

Model	Epochs	MB	α	Momentum
OCW1	3,000	4	0.0035	0.98
OTCW1	1,000	2	0.01	0.70
OT1	1,500	1	0.001	0.90
OTW1	1,000	2	0.015	0.50
OT2	1,500	5	0.012	0.20
OTC1	1,000	4	0.001	0.99
OTCW2	500	1	0.001	0.98
OT3	1,500	6	0.00125	0.90
OT4	2,000	1	0.012	0.98
OT5	5,000	1	0.001	0.94
OTCW3	1,500	5	0.013	0.90
OT6	500	3	0.001	0.98
OTCW4	1,000	3	0.001	0.99
OT7	2,500	1	0.015	0.45
OTCW5	2,000	2	0.00125	0.98
OTCW6	3,500	3	0.00125	0.98

Table 2: LSTM setup for each model, showing the epochs ran, the number of memory blocks (MB), the learning rate (α) and the momentum values.

forth. The initialization of output gates pushes initial memory cells activations towards zero, whereas that of the input gates prevents memory cells from being modified by incoming inputs. As training progresses, the biases become progressively less negative, allowing the serial activation of cells as active participants in the network computation.

The forget gates were initialized with symmetric positive values: $+0.5$ for the first block, $+1.0$ for the second, $+1.5$ for the third, and so forth. The bias initialization must be positive in this case, since it prevents the cells from forgetting everything [3], i.e., when positive signal is used the gates are open what means that no gates are used.

The cell's input squashing function g used was a logistic sigmoid in $[-1, 1]$ and both the output squashing function h and the activation function of the output unit were fixed as the linear identity function. The random weight initialization was in the range $[-0.1, 0.1]$.

The error criterion used in order to evaluate the forecasting results was the root mean square error (RMSE):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Oz(t) - \widehat{Oz}(t))^2}{N}} \tag{7}$$

where $Oz(t)$ is the measured (target) value of the ozone concentration, $\widehat{Oz}(t)$ is the predicted value and N is the number of observations.

4 Results

Here the results obtained throughout the current work are presented and compared with those reported in previous studies on the same task [5, 15], where the former was carried out using FIR methodology and the later using several types of Neural Networks (NNs). FIR [8] is a qualitative modeling and simulation methodology that is based on observation of input/output behavior of a system to be modelled. The first goal of the FIR methodology is to identify qualitative causal relations between the system variables, and the second one is the prediction of future behavior on the basis of past observations. In addition to the error measure on the test set, FIR provides other measure called quality, which is based on an entropy reduction measure, and serves to point out which model has the best forecasting performance taking into account only the training set. The NNs used by [15] were a MLP [6], an EN [2] and a MEN [10]. MLPs are well-known feedforward nets; ENs are a class of partially recurrent neural nets, and MENs are an extension of ENs in which state neurons are connected with themselves, thus getting a certain inertia, which increase the capabilities of the network to dynamically memorize data.

	FIR		MLP		EN		MEN		LSTM	
Model	Quality	Test	Train	Test	Train	Test	Train	Test	Train	Test
OCW1	0.4306	13.6768	–	–	–	–	–	–	8.5868	12.1854
OTCW1	–	–	8.2406	10.8132	–	–	–	–	8.5095	11.5100
OT1	–	–	9.4942	11.2004	9.0740	10.3186	8.9964	10.5150	9.5170	11.4916
OTW1	0.3456	10.1110	–	–	–	–	–	–	9.3882	11.3920
OT2	0.2866	9.7577	–	10.9327	–	–	–	–	9.0629	10.8715
OTC1	–	–	–	–	–	–	–	–	8.7930	10.7130
OTCW2	–	–	–	11.3329	–	–	–	–	8.0063	10.6352
OT3	0.2830	9.8270	–	11.0501	–	–	–	–	9.5208	10.4955
OT4	0.3566	10.0220	–	–	–	–	–	–	8.6739	10.4449
OT5	0.3680	10.5508	–	11.1407	–	–	–	–	9.0524	10.4329
OTCW3	–	–	–	12.3596	–	–	7.3741	9.9579	8.2071	10.3020
OT6	–	–	–	11.3501	–	–	–	–	7.6893	9.9676
OTCW4	–	–	–	–	–	–	–	–	6.5071	9.9644
OT7	–	–	–	–	–	–	–	–	7.9025	9.9640
OTCW5	–	–	–	–	–	–	–	–	6.4669	9.9216
OTCW6	–	–	–	–	–	–	–	–	7.8340	9.7961

Table 3: RMSE errors for predictions.

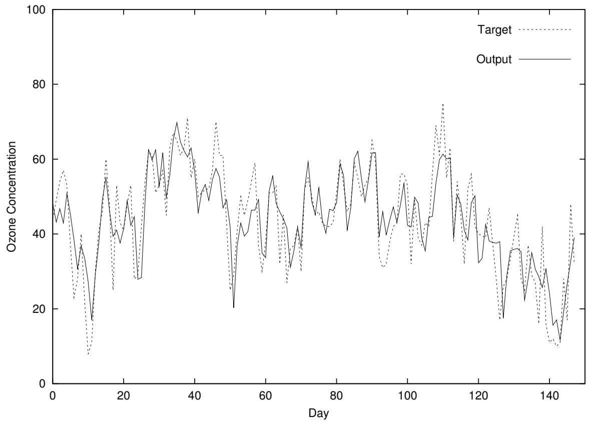

Figure 4: Prediction of training signals.

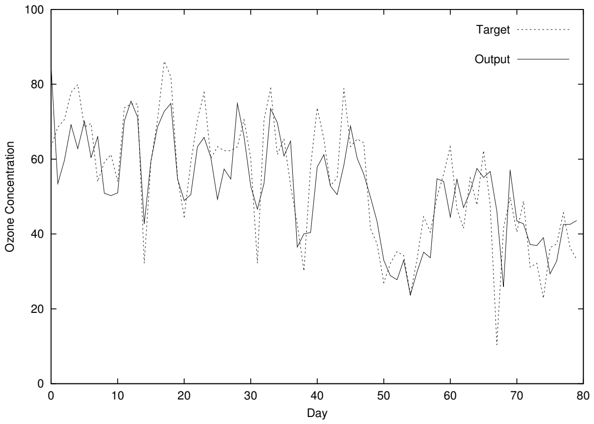

Figure 5: Prediction of test signals.

In Table 3, the RMSE errors for the training and test sets obtained by [5, 15] are compared to those achieved by LSTM. It must be noted, however, that in the case of FIR results, the RMSE of the training set (which is always zero because the training data is memorized) is replaced by the quality measure of the model estimated from the training set.

Figures 4 and 5 illustrate the prediction of output signal on training and test sets, respectively, carried out by the LSTM showing the best test RMSE. The target output signal is shown as dashed line and the predicted signal as solid line.

5 Conclusions

In this work we have tested Long Short Term Memory (LSTM), a powerful type of recurrent neural network, for forecasting the maximum ozone concentration in the East Austrain region from different sets of features. LSTM uses some basic structures called memory cell blocks to allow the net remember significant events distant in the past sequence of inputs. This information can be used to predict the future values of the sequence or time series being learned. In previous studies [7, 4], LSTM has demonstrated an impressive performance and has been shown to solve complex, artificial long time lag tasks that had never been solved by previous recurrent network algorithms. Here, we were interested in assessing the LSTM performance on a real forecasting problem, in which, however, it was thought that there is no need to store information over extended time intervals, but just to take into account the more recent history of previous input and output values.

We have shown that LSTM is capable of capturing the dynamic behavior of the system under study as accurately as other inductive approaches tested previously (other types of neural nets [15], FIR [5]). The performance of the best LSTM networks inferred are equivalent to the best FIR models and slightly better than the other neural networks applied (Modified Elman Nets, Elman Nets and Multi-Layer Perceptrons, in decreasing order of performance). However, the errors obtained using the best models are still quite high (an RMSE of 9.7961 for LSTM, 9.7577 for FIR and 9.9579 for MEN). Therefore, it seems that either the available data is not good enough (few observations, noisy measurements, use of model forecasts instead of real measured data for some variables) or some important variables to predict the ozone concentrations are not taken into account. On the other hand, the fact that a powerful RNN like LSTM is not able to improve significantly the results obtained by static approaches like FIR and MLPs may confirm the hypothesis that just a very few previous values of the involved variables are enough to predict the local maximum ozone concentration.

6 Acknowledgements

We are thankful to Jürgen Schmidhuber and Felix Gers, researchers from Instituto Dalle Molle di Studi sull'Intelligenza Artificiale (Lugano, Switzerland), for providing us with the last version of LSTM simulator and for being helpful with our uncountable questions related to it. Ozone data were contributed by the Austrian Environmental Protection Agency (Umweltbundesamt, UBA) and by the government of Lower Austria. This research was supported by Spanish CICYT under project TAP99-0747.

References

[1] G. Acuña, H. Jorquera and R. Perez. Neural Network Model for Maximum Ozone Concentration Prediction.*In Proc. of the International Conference on Articial Neural Networks*, pp.263-268, 1996.

[2] J.L. Elman. Finding Structure in Time. *Cognitive Science* **14**, pp.179-211, 1990.

[3] F.A. Gers. Long Short Term Memory in Recurrent Neural Networks. Thesis N 2366, Lausanne, 2001.

[4] F.A. Gers, J. Schmidhuber and F. Cummins. Learning to Forget: Continual Prediction with LSTM. *Neural Computation* **12** (10), pp.2451-2471, 2000.

[5] P. Gómes, A. Nebot, F. Mugica and F. Wotawa, Fuzzy Reasoning for the Prediction of Maximum Ozone Concetration. *13th European Simulation Symposion 2001*, pp. 535-542, France, 2001.

[6] J. Hertz, A. Krogh and R.G. Palmer. *Introduction to the Theory of Neural Computation.* Addison-Wesley, Redwood City CA, 1991.

[7] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation* **9**, pp.1681-1726, 1997.

[8] G. Klir, Architecture of Systems Problem Solving, Plelum Press: New Yourk, 1985.

[9] B.A. Pearlmutter. Gradient Calculations for Dynamic Recurrent Neural Networks: A Survey. *IEEE Trans. on Neural Networks* **6** (5), pp.1212-1228, 1995.

[10] D. Pham and X. Liu. Neural Networks for Identification, Prediction and Control. *Springer Verlag*, 1995.

[11] S. Ribeiro and R. Alquezar, A Comparative Study on a Signal Forecasting Task applying Long Short-Term Memory (LSTM) Recurrent Neural Networks, *VI Simpósio Ibero-Americano de Reconhecimento de Padrões*, Brasil, 2001.

[12] A. J. Robinson and F. Fallside. The Utility Driven Dynamic Error Propagation Network. Tech. Report, Cambridge University Engineering Department, 1987.

[13] A. Stohl, G. Wotawa and H. Kromp-Kolb. The IMPO Modeling System Description, Sensitivity Studies and Spplictions. Tech. Report, Universität für Bodenkultur, Institut für Meteorologie und Physik, Türkenschanzstraše 18, A-1180 Wien. 1996.

[14] R.J. Williams and J. Peng. An efficient gradient-based algorithm for on-line training of recurrent network trajectories. *Neural Computation* **2**, pp.491-501, 1990.

[15] D. Wieland and F. Wotawa, Local Maximum Ozone Concentration Prediction Using Neural Networks. *Procs. AAAI99, Workshop on Enviromental Decision Support Systems and Artificial Intelligence*, pp.47-54, 1999.

[16] F. Wotawa and G. Wotawa, From Neural Networks to Qualitative Knowledge in Ozone Forecasting. *AI Comunications* 14, pp.23-33, 1999.