A Connectionist System for Medium-Term Horizon Time Series Prediction

Samy Bengio, Françoise Fessant, and Daniel Collobert

France Télécom,
Centre National d'Études des Télécommunications,
LAB/RIO/TNT,
2, avenue Pierre Marzin,
22307 Lannion, FRANCE.
e-mail: {bengio,fessant,collober}@lannion.cnet.fr

Abstract

In this paper, we propose some improvements for the problem of time series prediction with neural networks where a medium-term prediction horizon is needed. In particular, the ionospheric prediction service of the french Centre National d'Études des Télécommunications needs a six-month ahead prediction of a sunspots related time series which has a strong influence on wave propagation in ionosphere. The proposed improvements consist in two different modular architectures and a way to increase the size of the training set. Experimental results are compared to those of a simple multi-layer perceptron.

1 Introduction

Suppose we have a given one-variable time series represented by the N values $\{x_1, x_2, \cdots, x_N\}$, where x_t is the series value sampled at time t. Prediction then consists to find the future values $\{x_{N+1}, x_{N+2}, \cdots\}$. It has been shown in [6] that if the series is deterministic, there exists an integer d (which is called the *embedding dimension*), an integer τ (which is an arbitrary delay) and a function $f(\cdot)$ such that for every $t > (d \cdot \tau)$:

$$x_t = f(x_{t-\tau}, x_{t-2\tau}, \cdots, x_{t-d\tau}) \tag{1}$$

Unfortunately, there exists no exact method to find neither d, τ or $f(\cdot)$ when the series is too small (less than 10^d samples for d and τ).

The ionospheric prediction service of the Centre National d'Études des Télécommunications (CNET) publishes monthly prediction reports about ionospheric propagation of radio-electric waves addressed to radio-diffusion services.

Ionospheric state depends directly on solar activity, so in order to make a prediction of its future state we have to predict the solar activity, which is here represented as the number of sunspots R [2]. The sunspots time series is known to be difficult to predict and has served as a benchmark in the statistics literature [9].

2 Problem Description

The CNET ionospheric prediction service has given us the $IR5$ time series which is a non-centered five-month mean of the monthly sunspots number mean MR:

$$IR5_t = \frac{1}{5}(MR_{t-3} + MR_{t-2} + MR_{t-1} + MR_t + MR_{t+1}) \tag{2}$$

where MR_t is the mean sunspots number of month t. The CNET needs a six-month ahead prediction of the $IR5$ index in order to publish and distribute a report to its users. The time series used in the present work starts in 1849 and ends in 1991, which gives us 1712 monthly data, over which the last 238 are kept for testing and comparison with their own heuristic. Figure 1 shows the $IR5$ time series.

Sunspots related time series such as the $IR5$ index are suspected to be non-stationary but there is not enough data to remove tendency and/or seasonality. This is very annoying because, when using learning techniques such as neural networks, to expect a good generalization (prediction) error while minimizing a training set error, one has to hypothesize the test set is drawn from the same probability distribution as the training set. As the series may have a tendency, this may not be the case. This is the reason why our test set is kept small in comparison to the training set size.

3 A Simple Connectionist Solution

The simplest way to use neural networks for prediction, which has already been used in many applications [4, 8], is to use a multi-layer perceptron

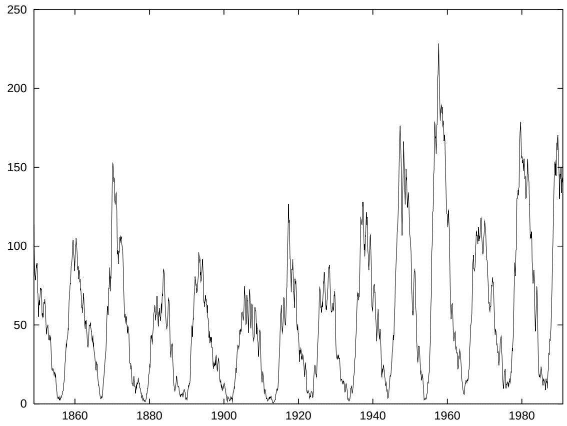

Figure 1: The *IR*5 Time Series

with one hidden layer, trained with backpropagation [5]. All input units are connected to all hidden units and all hidden units are connected to all output units. Given a series with N values, an embedding dimension d, a delay τ and an horizon h, training data consists of $N - (d \cdot \tau) - h$ input/output pairs such that:

$$\{x_{t-\tau}, x_{t-2\tau}, \cdots, x_{t-d\tau}\} \longrightarrow \{x_t, \cdots, x_{t+h}\} \tag{3}$$

where x_i is the i^{th} normalized value of the series[1]. The embedding dimension d, as well as the hidden layer size are usually set by cross-validation techniques. In most applications, h is set to 0 and τ is set to 1.

For the *IR*5 series, the best network we found had 40 input units, 23 hidden units, and 6 output units. τ was set arbitrary to 1. Hidden and output units used a nonlinear squashing function with output in $[-1, 1]$. Figure 2 summarizes this architecture. Results are reported later in the paper.

4 Improvements Over a Simple Use of Neural Networks for Prediction

In this section, we present three improvements over this simple use of a neural network for time series prediction. All three are based on the fact that generalization (prediction) error is related to training set error, network capacity (which itself is related to the number of free parameters), training

[1]The series is usually normalized to get a zero mean with values in $[-1, 1]$.

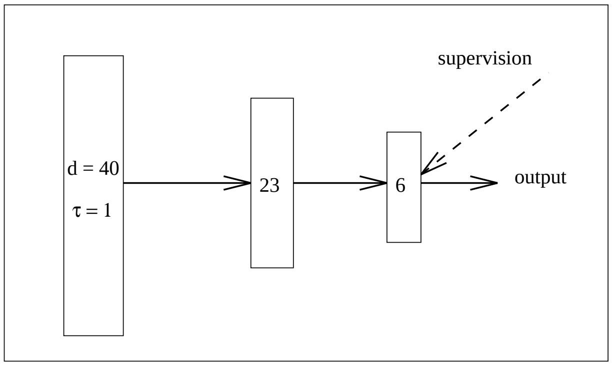

Figure 2: The Simple Multi-Layer Perceptron Solution

set size and the use of prior knowledge in system design [7]. The first two improvements are architectural: a better system design and less parameters. The last one shows how to use more training data.

4.1 Mix of Different Output Schemes

When one wants to predict x_{t+h} given $\{x_{t-\tau}, x_{t-2\tau}, \cdots, x_{t-d\tau}\}$ with $h > 0$, there are at least three simple ways to do it:

- a multi-layer perceptron (MLP) with one output unit x_{t+h},
- an MLP with $h+1$ output units $x_t, x_{t+1}, \cdots, x_{t+h}$,
- an MLP with one output unit x_t. In that case x_{t+h} is predicted using as input previously predicted values $\hat{x}_t, \cdots, \hat{x}_{t+h-1}$, which are estimates of $x_t, \cdots, x_{t+h-1}$.

Experimental results tend to show that the latter method is better when h is big ($h > 10$) whereas the first two are better when h is small.

We now propose a modular architecture which is composed of three small interconnected networks: the first two networks try to find independently a solution and the third one combines their results.

The first network tries to find x_{t+h} while the second tries to find also intermediate values $x_t, \cdots, x_{t+h-1}$, using the same input data. The outputs of the first and the second network are then used as input to a third network, which try to find x_{t+h}. The three networks are trained simultaneously and supervision is provided to all output units. This means that weights of the first two networks are influenced both by supervision provided by their respective network but also by the third network.

Since all three networks are small (the hidden layer sizes for experiments reported here were respectively 5, 5 and 11 for the three networks), the total number of free parameters in this model is still half the number of free parameters in the simple model. Once again , τ was arbitrary set to 1. Figure 3 shows this new architecture.

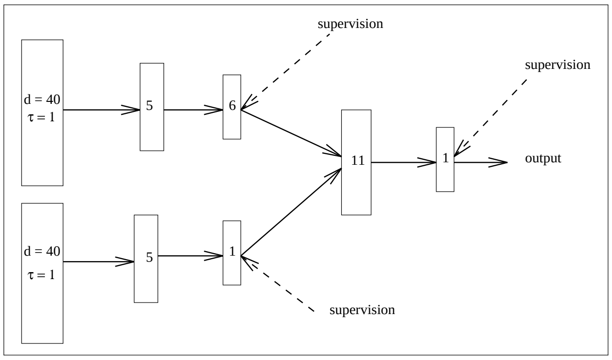

Figure 3: A First Architectural Improvement: Mix of Different Small Networks which have Different Outputs

4.2 Use of Different d/τ Combinations

The time series length being too small to determine adequately good values for d and τ, an alternate solution is to use a combination of many small networks each using different d and τ values. One way of doing this is to search for a new function which has the form:

$$\begin{aligned} x_{t+h} \quad = \quad g(\; & f_1(x_{t-\tau_1}, x_{t-2\tau_1}, \cdots, x_{t-d_1\tau_1}), \\ & f_2(x_{t-\tau_2}, x_{t-2\tau_2}, \cdots, x_{t-d_2\tau_2}), \\ & \vdots \\ & f_n(x_{t-\tau_n}, x_{t-2\tau_n}, \cdots, x_{t-d_n\tau_n}) \;) \end{aligned} \tag{4}$$

where $f_1(\cdot), \cdots, f_n(\cdot)$ are simple neural networks trained to predict x_{t+h} (and maybe intermediate values), each using its own d_i and τ_i, and $g(\cdot)$ is another network which is in top of the others and simultaneously trained. In the experiments we report here with the $IR5$ time series and a prediction horizon $h = 5$, we had three functions $f_i(\cdot)$:

- $f_1(\cdot)$ used $d = 40$, $\tau = 1$ and tried to predict x_{t+5}.

- $f_2(\cdot)$ used $d = 34$, $\tau = 2$ and tried to predict x_{t+1}, x_{t+3} and x_{t+5}.
- $f_3(\cdot)$ used $d = 30$, $\tau = 3$ and tried to predict x_{t+2} and x_{t+5}.

Figure 4 gives the actual architecture used in this paper. Again, the resulting network still has around half the number of free parameters compared to the simple network.

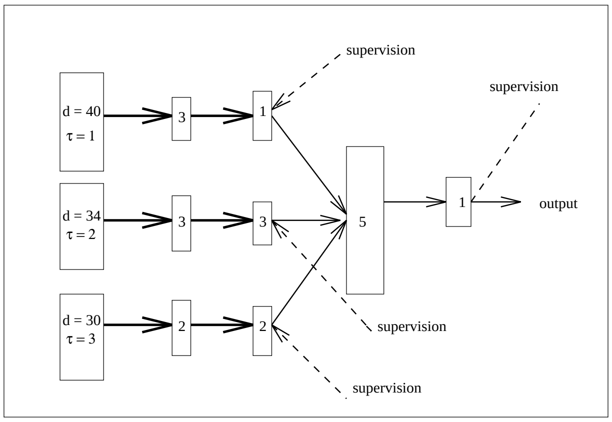

Figure 4: A Second Architectural Improvement: Use of Different d/τ Combinations

4.3 Increase of the Training Set Size

For the particular series of sunspots, data are in fact gathered every day, but only monthly means are usually used for prediction. We propose here a way to use more than just monthly means. The $IR5$ index, computed in equation (2) can be rewritten as follows:

$$NIR5_t = \frac{1}{152} \sum_{i=t-105}^{t+46} R_i \tag{5}$$

where $NIR5_t$ for a particular day t represents the $IR5$ index. The only difference with the $IR5$ series is that we now have a longer but daily series of the very same index. In fact, the $IR5$ series is a subset of the $NIR5$ one.

Setting τ to 30 implies we are searching for exactly the same kind of function as with the $IR5$ series with τ set to 1, except that the database size is around 30 times the $IR5$ one[2]. This is important, given that generalization

[2] And of course the fact that a month is not exactly equal to 30 days but this does not have any real impact on the goal.

error is related to training set size: the more data one have, the more free parameters one can handle without over-fitting, the more precise could be the resulting function.

In the experiments we report here, we used the $NIR5$ time series sampled every 7 days, which makes the database size around 4 times the original one.

5 Experimental Results

We compared the different connectionist models proposed in this paper for the $IR5$ time series to the actual heuristic used by the CNET prediction service.

For all networks, input dimension d, as well as hidden layers sizes, have been estimated using cross-validation. All experiments used the stochastic version of backpropagation algorithm, with a small weight decay to keep network capacity as small as possible. Results in Table 1 give the *Average Relative Cost* (ARV) obtained for the last 238 months of the series, which were not used to train the network. ARV is computed as follows:

$$ARV = \frac{1}{\hat{\sigma}^2} \frac{1}{N} \sum_{i \in P} (x_i - \hat{x}_i)^2 \tag{6}$$

where P is the test set, N is the test set size, $\hat{\sigma}^2$ is the estimated variance of the series, $\hat{x}_i$ is the i^{th} predicted value and x_i is the corresponding desired value. We also give the *Strong Error Percentage* (SEP) which represents the percentage of test set examples for which the distance between expected value and obtained value is more than 30 (for the $IR5$ time series, values range from 0 to 230). This statistic is important for the CNET because its users prefer to make many small errors rather than few important ones.

Table 1: Comparison of the Different Predictors

	ARV	SEP
actual CNET heuristic	0.113	5.04
Standard MLP	0.088	5.04
Modular Architecture (multiple output schemes)	0.075	1.68
Modular Architecture (multiple d/τ schemes)	0.073	1.68
Larger Training Set	0.071	1.68

6 Conclusion

In this paper we proposed three improvements over a simple use of multi-layer perceptron for time series prediction with a medium-term horizon. All gave better results for a particular and known difficult time series. We shall now try to mix some of these ideas, as well as use other models such as Elman recurrent networks [1] or hierarchical mixtures of experts [3].

References

[1] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, pp. 179–211, 1990.

[2] A. Izenman, "J. R. Wolf and the Zürich sunspot relative numbers," *The Mathematical Intelligencer*, vol. 7, no. 1, pp. 27–33, 1985.

[3] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," *Neural Computation*, vol. 6, no. 2, pp. 181–214, 1994.

[4] D. C. Park, M. A. El-Sharkawi, and R. J. Marks II, "Electric load forecasting using an artificial neural network," *IEEE Transaction on Power Systems*, vol. 6, no. 2, pp. 442–449, 1991.

[5] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing* (D. E. Rumelhart and J. L. McClelland, eds.), vol. 1, MIT Press, 1986.

[6] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence* (D. A. Rand and L.-S. Young, eds.), vol. 898 of *Lecture Notes in Mathematics*, (Warwick 1980), pp. 366–381, Springer-Verlag, Berlin, 1981.

[7] V. N. Vapnik, *Estimation of Dependencies Based on Empirical Data.* New-York, NY, USA: Springer-Verlag, 1982.

[8] A. Varfis and C. Versino, "Univariate economic time series forecasting by connectionist methods," in *Proceedings of the International Neural Network Conference (INNC)*, (Paris, France), pp. 342–345, 1990.

[9] A. S. Weigend, B. A. Huberman, and D. E. Rumelhart, "Predicting sunspots and exchange rates with connectionist networks," in *Nonlinear modeling and forecasting* (M. Casdagli and S. Eubank, eds.), pp. 395–431, Addison Wesley, 1992.