Use of Modular Architectures for Time Series Prediction

Abstract

Recently, there has been a lot of papers published in the field of time series prediction using connectionist models. Nevertheless we think that one of the major problem which is rarely treated in the literature is related to the choice of input parameters (embedding dimension and delay). In this paper, we propose two modular approaches to this problem and apply them to a sunspot-related time series. Experimental results are then compared to a simple multi-layer perceptron in order to estimate performances of these models.

1 Introduction

Suppose we have a given univariate 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 consists to find the future values $\{x_{N+1}, x_{N+2}, \cdots\}$. If the series is obtained from a deterministic dynamical system, Takens [1] showed that 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 any $t > (d \cdot \tau)$:

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

Although one can approximate $f(\cdot)$ using a simple connectionist model such as a multi-layer perceptron trained by backpropagation [2], there exists no exact method to find neither d nor τ when N is too small (less than 10^d samples).

Since the pioneering work of Hu [3] and the first model using backpropagation for time series prediction by Lapedes and Farber [4], numerous papers have been published on the subject [5]. Most of them have overlooked the subject of input parameter selection (namely d and τ).

In this paper, we do not propose yet another heuristic to select d and τ (there are in effect already numerous heuristics, see a good review in [6]), which could lead to suboptimal solution in case of unsufficient data. We propose to mix many sets of parameters (each chosen using personal prefered heuristic for instance), using a modular connectionist approach.

2 Problem Description

Solar activity is usually measured as the number of sunspots R [7]. The sunspots time series is known to be difficult to predict and has served as a benchmark in the statistics and connectionist literature [8]. The France Télécom CNET ionospheric prediction service involves the $IR5$ time series which is a non-centered five-month mean of the sunspots number R:

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

where MR_t is the mean sunspots number of month t. Six-month ahead predictions of the $IR5$ index are requested in order to publish and distribute a report to CNET users. The solar time series starts in 1849 and ends in 1991, corresponding to 1712 monthly data, over which the last 238 are kept for testing and comparison with standard CNET heuristic. Figure 1 shows the $IR5$ time series.

3 Use of Different d/τ Combinations in a Modular Model

In this section, we propose a modular architecture which is composed of many small interconnected networks: each network tries to find independently a solution and the results are combined using another neural network. Modularity is heavily used in computer science: by decomposing a problem into modules, one expects each module to be simpler than the overall task; which can be called the *divide and conquer* strategy [9].

Since the time series length is too small to adequately determine correct values for d and τ, an alternate solution is to use on each small networks different d and τ values. Thus knowledge is incorporated into the structure. The resulting architecture is as follow:

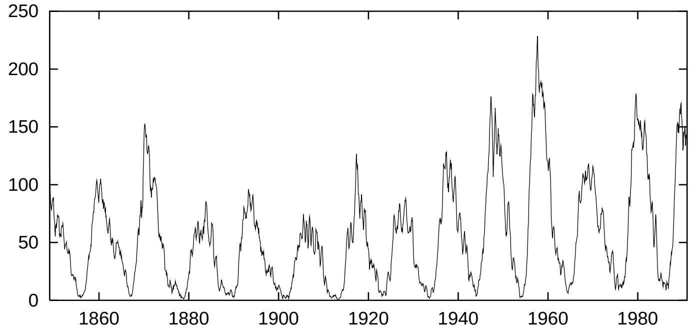

Figure 1: The *IR*5 sunspots series

$$\begin{aligned} x_{t+h} \;=\; g(\;& f_1(x_{t-\tau_1}, x_{t-2\tau_1}, \cdots, x_{t-d_1\tau_1}), \\ & f_2(x_{t-\tau_2}, x_{t-2\tau_2}, \cdots, x_{t-d_2\tau_2}), \\ & \vdots \\ & f_n(x_{t-\tau_n}, x_{t-2\tau_n}, \cdots, x_{t-d_n\tau_n}) \;) \end{aligned} \tag{3}$$

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. $g(\cdot)$ is another network which is on top of the others and simultaneously trained. In the experiments reported here with the *IR*5 time series and a prediction horizon $h = 5$, there are three functions $f_i(\cdot)$:

- $f_1(\cdot)$ uses $d = 40$, $\tau = 1$ and tries to predict x_{t+5}.
- $f_2(\cdot)$ uses $d = 33$, $\tau = 2$ and tries to predict x_{t+1}, x_{t+3} and x_{t+5}.
- $f_3(\cdot)$ uses $d = 24$, $\tau = 3$ and tries to predict x_{t+2} and x_{t+5}.

The exact architecture of each module is determined using early stopping via validation set error. Figure 2 gives the actual architecture used in this paper. The four networks are trained simultaneously and supervision is provided to all output units. Connections of the first three networks $f_1(\cdot)$, $f_2(\cdot)$, $f_3(\cdot)$ are influenced both by external supervision provided by their respective network (the cost function is minimized at each output layer) but also by the fourth network $g(\cdot)$, using the usual chain rule to find the exact error derivatives.

More specifically, let i be an output unit of one of the small intermediate networks, $dest(i)$ the list of units which are connected to i, $y(i)$ the activation of unit i, E the total error to be minimized, E_i the error made at unit i and E_g the error made at the final output unit. Then one can compute:

$$\frac{\partial E}{\partial y(i)} = \beta \frac{\partial E_i}{\partial y(i)} + \alpha \sum_{j \in dest(i)} \frac{\partial E_g}{\partial y(j)} \tag{4}$$

α and β are positive numbers and sum to 1. They represent a weighting factor between the two terms. A good heuristic is to vary their value: at the beginning of the training, α should be small and β large, which will help to train faster, and the ratio should change smoothly in order at the end of the training to let the global error be the only important term.

4 Adaptative Mixture of Experts

The second modular architecture we propose is an adaptative mixture of experts, as introduced by Jacobs and Jordan [10]. Since its introduction, many papers has been published using this model or its extensions (such as Hierarchical Mixtures of Experts) for time series prediction. This

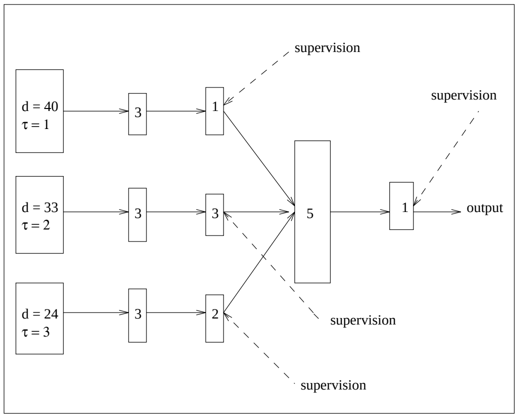

Figure 2: The modular neural network

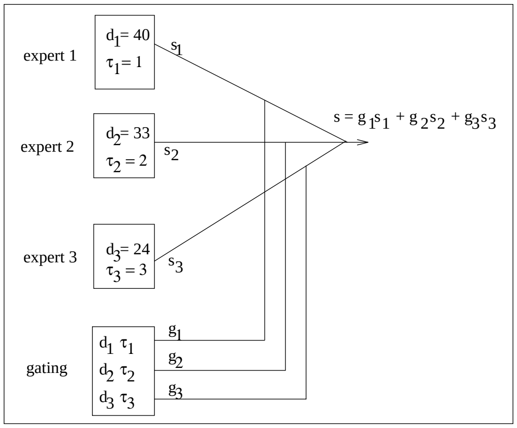

Figure 3: The adaptative mixture of experts

architecture combines associative and competitive learning. Different networks learn training patterns from different regions of the input space and model the distribution of training patterns. As illustrated in Figure 3, it consists of two types of modules: expert networks and a gating network. The expert networks compete to learn the training patterns and the gating network mediates the competition.

The expert networks have an arbitrary connectivity while the gating network have as many output units as there are expert networks. The activations of output units must be non negative and sum to one. During training, the weights of the expert and gating networks are simultaneously adjusted using the backpropagation algorithm. The error function is the log probability of generating the desired output vector under a mixture of gaussians.

The mixture of experts architecture is applied to the $IR5$ prediction problem. The tested architecture has three expert networks, each one being a multilayer perceptron with one hidden layer and viewing a different input window:

Expert 1: $d = 40$, $\tau = 1$, hidden units $= 8$.

Expert 2: $d = 33$, $\tau = 2$, hidden units $= 8$.

Expert 3: $d = 24$, $\tau = 3$, hidden units $= 8$.

Gating: uses all three input parameter sets, with 12 hidden units.

The main assumption for the use of this model is that, as it has been suggested in [11], there may be more than one underlying attractor generating the system; a dynamical noise could locally switch the system from one attractor to another. The mixture of experts could then select locally the best reconstruction space. This model has been used for instance in [12, 13] to select automatically the attractor generating each part of a signal made of random parts of many different signals. The main difference with our work is our proposition to let each expert have a different view of the input signal in order to help the specialization of each experts.

The model can also be compared to regime switching models [14, 15]: these models are based on the idea of piecewise linearization of non linear models over the state space by the introduction of tresholds. However these models are locally linear, whereas mixtures of experts could combined arbitrary complex and non linear models.

5 Experimental Results

The connectionist models proposed in this paper for the $IR5$ time series are compared to the actual heuristic used by the CNET prediction service and also to a simple model: a multi-layer perceptron with one hidden layer of units, trained by backpropagation (the best network found has 23 hidden units, $d = 40$ and $\tau = 1$).

Experiments use the stochastic version of backpropagation algorithm. The networks are trained starting from initial random weights and the training is stop using an early stopping method to prevent overfitting. Results in Figure 4 give the *Average Relative Cost* (*ARV*) obtained for the last 238 months of the series, which were not used to train the networks. *ARV* is computed as follows:

$$ARV = \frac{1}{\hat{\sigma}^2}\frac{1}{N}\sum_{i\in P}(y_i - \hat{y}_i)^2 \qquad (5)$$

where P is the test set, N is the test set size, $\hat{\sigma}^2$ is the estimated variance of the series, $\hat{y}_i$ is the i^{th} predicted value and y_i is the corresponding desired value. We give also the variance of the ARV, σ^2_{ARV}.

	ARV	σ^2_{ARV}
CNET heuristic	0.113	0.024
Simple model	0.088	0.019
Modular Architecture	0.064	0.008
Expert Mixture	0.076	0.011

Figure 4: Comparison of the predictors

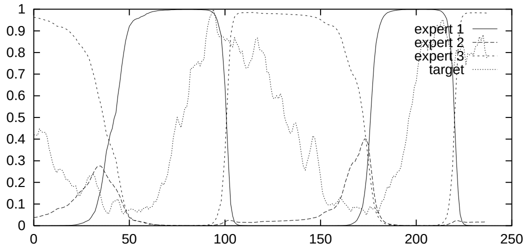

Figure 5: The expert repartition for the test set

Both modular models perform better than a simple multi-layer perceptron, in terms of both *ARV* and its variance σ^2_{ARV}. The mixture of experts gives worse results than the first model, probably because of the linear nature of the expert combination. However, as it can be seen in Figure 5, it reveals a strong evidence of input space separation related to the series gradient. This could help to analyze the series and to select future architectures.

6 Conclusion

In this paper we proposed two modular solutions to the problem of input parameter selection for time series prediction. Both gave us better results than a simple multi-layer perceptron for a specific and known to be difficult time series. Other modular architectures could be tested, such as the newly proposed *IOHMM* model [16]. Moreover, another direction which should be explored is related to the stationarity of a time series: there is currently no good method to preprocess a series such that it always results in a stationary process, which is important in order to expect good generalization performance.

References

[1] 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, 1981, Springer-Verlag, Berlin, pp. 366–381.

[2] 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.

[3] M. Hu, *Application of the adaline system to weather forecasting*, Electrical Engineering Degree Thesis Technical Report 6775-1, Stanford Electronic Laboratory, 1964.

[4] A. Lapedes and R. Farber, *Nonlinear signal processing using neural networks: prediction and system modelling*, Tech. Rep. LA-UR-87-2662, Los Alamos National Laboratory, Theoretical Division, 1987.

[5] A. S. Weigend and N. A. Gershenfeld, eds., *Time Series Prediction: Forecasting the Future and Understanding the Past*, Addison-Wesley, 1993.

[6] H. Abarbanel, R. Brown, J. Sidorowich, and L. Tsimring, *The analysis of observed chaotic data in physical systems*, Reviews of Modern Physics, 65 (1993), pp. 1331–1392.

[7] A. Izenman, *J. R. Wolf and the Zürich sunspot relative numbers*, The Mathematical Intelligencer, 7 (1985), pp. 27–33.

[8] 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., Addison Wesley, 1992, pp. 395–431.

[9] F. Fogelman Soulié, *Neural network architectures for pattern recognition*, in From Statistics to Neural Networks, Theory and Pattern Recognition Applications, V. Cherkassky, J. Friedman, and H. Wechsler, eds., Springer Verlag, 1994, pp. 243–262.

[10] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, *Adaptive mixtures of local experts*, Neural Computation, 3 (1991), pp. 79–87.

[11] T. Meyer and N. Packard, *Local forecasting of high-dimensional chaotic dynamics*, in Nonlinear Modeling and Forecasting, M. Casdagli and S. Eubank, eds., Addison Wesley Publishing Company, 1992, pp. 249–263.

[12] L. Xu, *Signal segmentation by finite mixture model and EM algorithm*, in International Symposium on Artificial Neural Networks, 1994, pp. 453–458.

[13] ———, *Channel equalization by finite mixtures and the EM algorithm*, in Neural Networks for Signal Processing V, F. Girosi, J. Makhoul, E. Manolakos, and E. Wilson, eds., IEEE Press, 1995, pp. 603–612.

[14] J. Hamilton, *Analysis of time series subject to changes in regime*, Journal of Econometrics, 45 (1990), pp. 39–79.

[15] H. Tong, *Non-linear Time Series, A Dynamical System Approach*, Clarendon Press, 1990.

[16] Y. Bengio and P. Frasconi, *An Input Output HMM architecture*, in Advances in Neural Information Processing Systems: Proceedings of the 1994 Conference, G. Tesauro, D. S. Touretzky, and T. K. Leen, eds., MIT Press, 1995.