ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 453-458

Orthogonal Transformations for Optimal Time Series Prediction

Moisés Salmerón,* Alberto Prieto, Julio Ortega,†
Carlos G. Puntonet,‡ and Manuel Rodríguez Álvarez

Deparment of Computer Architecture and Computer Technology,
University of Granada. E.T.S. Ingeniería Informática,
Campus Aynadamar s/n, E-18071 GRANADA (Spain)

Abstract. Our objective is to present an hybrid algorithm for prediction applications. It relies both on classical matrix techniques and on emerging neural network techniques. We focus our study on RBF (Radial Basis Function) networks and on the SVD and QR-cp factorization procedures to automatically determine an optimal network configuration that incorporates desirable properties for prediction applications. The significance of this work is evident in the context of small to medium-sized companies that depend on decision-making policies that are not only reliable, but also computationally fast.

1 Introduction and Scope

The work in this paper is derived from the consideration of the problem of *time series prediction*. This kind of problem appears in numerous practical applications of scientific and economic interest, in fields such as finance, inventory control and production planning (e.g. [Kan95]). In general, what is required is an estimate (or *prediction*) of the future value $x(t+t_h)$ of a variable or process x, where $t_h > 0$ refers to the 'time horizon' (that is, the time period covered by the prediction). For relatively large values of t_h we talk of *long-range* or *long-horizon* prediction. In any case, information for *endogenous* prediction is computed from the actual and past values $x(t), x(t-2), \ldots$ up to a 'window size' W (that is, up to $x(t-W+1)$) available at time instant t, giving an estimate $\hat{x}(t+t_h)$ of the future value the process or series is assumed to take.

The structure of the paper is as follows: first, we define the general time series prediction problem and highlight its relevance for a wide range of practical, real-life situations. The mathematical definitions of the Radial Basis Function

*Corresponding author. E-mail: moises@ugr.es.
†Partially supported by Spanish MCyT Project TIC2000-1348.
‡Partially supported by Spanish MCyT Project TIC2001-2845.

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 453-458

(RBF) network paradigm are then reviewed, after which the main matricial decomposition schemes used in the paper are described in an operational sense. Based on the matrix decompositions approach, we propose a new means of creating optimal RBF neural architectures for time series prediction applications, one that solves most of the problems observed in previously described algorithms.

2 Time series prediction problems and RBFNNs

In our algorithm (thoroughly described also in [SOPP01]) matrices of size $2W \times W$ (where W is the prediction window size) are set up, in each time instant t, from past values using consecutive past windows as follows:

$$\mathbf{A}_W\big(x(\cdot),t\big) = \begin{pmatrix} x(t-3W+2) & x(t-3W+3) & \cdots & x(t-2W+1) \\ x(t-3W+3) & x(t-3W+4) & \cdots & x(t-2W+2) \\ \vdots & \vdots & \vdots & \vdots \\ x(t-W+1) & x(t-W+2) & \cdots & x(t) \end{pmatrix},$$

that is, $3W-1$ past values are used (including the actual value in t) to predict the value in $t+\tau_h$. This matrix can be used, e.g., for determining the relevant "lags" (time indexes) for prediction, yielding a reduced input model for (in this case) a neural model, although this input selection technique could be applied to any reasonable model. Matrices set up in this way can be useful for determining the relevant input nodes to the neural network; that is, the relevant *lags* for time series prediction, since the inputs to the RBF network are precisely lagged values of the same series we want to predict. A similar scenario holds for the neural nodes (RBF) activation values, over input vectors presented to the network. We are assuming that $x(t) \in \mathbb{R}$ for all $t \in \mathbb{N}$. The multivariate case of $\mathbf{x}(t) \in \mathbb{R}^n$ can be approached with an intuitive generalized setting of our discussion.

As is well known, neural nets perform usually better with respect to classical statistical methodologies because they do not necessarily depend on large-sample theory assumptions as do classical statistical hypothesis testing, confidence intervals or (in the case of time series problems) stochastic processes theory approaches. The large-sample approaches are widely applied and powerful when their assumptions are met, which is not usually the case for some kinds of problems; for these, a methodology based on ANNs seems of more use to the decision-maker. Considerable work has yet to be done to make ANN behaviour more 'explainable' and to create deeper foundations, in the formal mathematical sense. Attempts are currently being made to formalize NN theory, but even while this is being done, we can usefully rely on ANNs as 'black-box' devices that behave as function approximators or prediction models, at least for practical purposes.

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 453-458

Radial Basis Funcion (RBF) networks

One of the simplest, but also most powerful, ANN models is the Radial Basis Function (RBF) network model. This consists of locally-receptive activation functions (or neurons) implemented by means of gaussian functions [MD89]. In mathematical terms, we have

$$o(\mathbf{x}) = \sum_{i=1}^{N} o_i(\mathbf{x}) = \sum_{i=1}^{N} h_i \exp\left\{-\frac{\|\mathbf{x}-\mathbf{c}_i\|^2}{\sigma_i^2}\right\} \quad (1)$$

where N denotes the number of nodes (RBFs) used; $o_i(\mathbf{x})$ gives the output computed by the i-th RBF for the input vector $\mathbf{x}$ as an exponential transformation over the norm that measures the distance between $\mathbf{x}$ and the RBF centre $\mathbf{c}_i$, whereas σ_i denotes the *radius* that controls the locality degree of the corresponding i-th gaussian response. The global output $o(\mathbf{x})$ of the neural network is, as can be seen, a linear aggregate or combination of the individual outputs, weighted by the real coefficients h_i.

In most RBF network applications, the usual methodology is to determine the coefficients h_i after setting up the location and radius for each of the N nodes. The locations can be set, as in [MD89], by a *clustering* algorithm, such as the *K-means algorithm*, and the radius is usually set after taking into account considerations on RBFs close to the one being configured. Adjustment of the linear expansion coefficients can be done using recursive methods for linear least squares problems, so that the mean square deviation over a training data set is minimized, and determined by the previously set parameter values. The recursive formulation allows for real-time implementation of the training algorithms for RBFs. These two steps configure the 'classical' procedure for training RBF networks, and it is clear that provision for some form of real-time (or 'online') adjustment also of the centre location and radius of each RBF can yield improved accuracy in the modelling of the datasets. On the other hand, a technique is sought that would allow also for input model reduction, that is another critical issue in neural network research.

The following sections describe the fundamentals of a procedure to do this, using matrix decompositions to achieve better network configurations for real-time operation. In the particular context of the RBF networks introduced above, the mapping of a time series prediction problem to the network is performed as follows: inputs are set up to a subset of past values (consecutive ones, in a first approximation) of the time series; the output is viewed as a prediction for the future value that we want to estimate. The computed error between the desired and network-estimated value is used to stochastically adjust the free parameters of the RBF network (such as the setting of node centres and/or widths, or the linear expansion coefficients that define the output).

3 Optimal Orthogonal RBF Prediction

We now consider our proposed method for optimal network determination, which is based on the matrix decompositions SVD and QR-cp. As has been

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 453-458

explained, our prediction algorithm will make use of matrices of time series or neural activation data. Now, it may happen that the transformation defined by a matrix $\mathbf{A}$ of size $m \times n$ is not 'optimal', non-optimality being understood in the sense that the vector subspace of $\mathbb{R}^m$ spanned by the n columns of $\mathbf{A}$ could in fact be generated by another set of fewer than n vectors. This corresponds to the case of *algebraic rank* less than n (rank $(\mathbf{A}) < n$), but also numerically we could have a full-rank matrix (rank$(\mathbf{A}) = n$) but $\mathbf{A}$ being very close (numerically) to a rank-deficient (rank strictly less than n) matrix. 'Closeness' is understood here in reference to a norm defined in the vector space $\mathfrak{M}_{a\times b}(\mathbb{R})$ that we have defined. The *numerical rank* of $\mathbf{A}$ is defined (relative to a given level of accuracy $\epsilon > 0$ and using the Euclidean, 2-norm) as [Bjö96]:

$$\text{Nrank}_\epsilon(\mathbf{A}) \stackrel{\text{def}}{\equiv} \min \{\text{rank } (\mathbf{B}) : \|\mathbf{A} - \mathbf{B}\|_2 \leq \epsilon\} \ , \tag{2}$$

where $\mathbf{B}$ is any matrix of the same dimensions as $\mathbf{A}$.

The problem of (numerically) solving a linear system of equations (that arises in the context of prediction by linear autoregressive methods, or in the neural network solution of time series problems [SOPP01]) is *discontinuous* in the sense that small changes to the matrices formed by the time series or neural data can lead to relatively large distorsions in the corresponding solution parameters. Roundings and other numerical errors can make a rank-deficient (rank less than n) matrix $\mathbf{A}$ appear as a full-rank (rank n) matrix $\mathbf{A}'$. The algorithms for solving the associated system (say, a linear matrix equation directly from an autoregressive model, or the linear matrix equation resulting from the output layer equation of an RBF neural network) could offer bad results if they failed to identify such a circumstance. In other words, algorithms must take into account the numerical rather than the algebraic rank of $\mathbf{A}$, so as to be able to detect the case of algebraic full-rank in the matrix argument $\mathbf{A}'$ but rank-deficiency of the same matrix in the numerical sense (Nrank$_\epsilon(\mathbf{A}') < n$ for a pre-established small ϵ), which would indicate that the 'real' abstract matrix $\mathbf{A}$ with which we are operating is, in fact, algebraically rank-deficient.

The most appropriate tool for our algorithm to be 'aware' of rank problems in the matrices that arise foir is an *orthogonal transformation*. Here we consider a special kind of orthogonal transformation called the *SVD* (*Singular Value Decomposition*) of $\mathbf{A}$ [Bjö96]. Formally, an orthogonal transformation consists of another matrix (denoted by $\mathbf{S}$ of size $m \times n$ whose rank is equal to the rank of $\mathbf{A}$:

$$\text{rank } (\mathbf{R}) = \text{rank } (\mathbf{A}) = k < n \ , \tag{3}$$

along with two orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$ of sizes $m \times m$ and $n \times n$ respectively, such that $\mathbf{A}$ can be expressed as

$$\mathbf{A} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^T \ , \tag{4}$$

where $\mathbf{S}$ is a diagonal matrix with elements in decreasing order (called the *singular values*) of $\mathbf{S}$); some of these elements may be zero, indicating that

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 453-458

A is rank-deficient; indeed, the number of nonzero singular values equals the (algebraic) range of matrix **A**.

Now, if we consider the above, taking into account the numerical issues (as in the computer solution of time series problems), a truly rank-deficient matrix will result in one or more negligible but non-zero singular values when the SVD of the noisy, real-life data (a distorted matrix $\mathbf{A}'$ of the assumed 'real' matrix) is computed. Algebraically, we could not claim **A** to be rank-deficient and could not proceed to any column selection (or its equivalent, to any data model reduction). A suitable strategy then could be to establish a *threshold* of numerical significance so as to ascertain whether the numerical matrix represents a real rank-deficient one. This is the essential idea behind the optimal prediction algorithm outlined in this paper.

Optimal network determination

Using the above ideas, we have derived an algorithm based on RBF networks that is very well suited for solving time series prediction problems. This algorithm, thoroughly described in the specialized reference [SOPP01], determines the 'optimum' number (in practical terms) of RBF neurons for computing the real-time recursive prediction of complicated time series such as the Mackey-Glass chaotic series [MG77]. It does so by decomposing the succesive matrices formed both from the past series input values, and from the neural activations of the RBF nodes. The basic operation is described in what follows, along with the experimental results.

4 Operation and Experimental Results

The results in what regards to the Mackey-Glass time series prediction, are shown in the figures. Figure 1 indicates a substantial reduction to less than half of the neural resources originally required to 'learn' the prediction surface on an on-line basis, when our method based on SVD and QR-cp was tested.

The basic procedure involves forming a matrix **A** in which successive rows set up from neural activation $a_i(\cdot)$ values as real-time input patterns $\mathbf{x}$ are presented to the network. As the output layer is a linear expansion $\sum h_i \cdot a_i(\mathbf{x})$ of these activations, SVD enables the determination of the most relevant nodes (RBFs) in the network, just by examining the singular values of the related matrix. This is possible since the QR-cp phase transforms relevant singular values σ_i into a permutation matrix **P** that indicates the neurons (columns) responsible for most of the data 'energy' [SOPP01]. The activation matrix is set up every few iterations to allow for fast detection of dynamics changes in the input time series [SOPP01]. Therefore, a lot of matrix work is required for meaningful real-time operation of our algorithm, and the accuracy obtained must be balanced against this amount of computation. Parallelism can offer a way of doing so without too great a sacrifice of the prediction precision obtained. Thus, results such as those shown in Figure 1 can be attained efficiently in real-time if we resort to parallel architectures for the SVD and/or QR-cp subtasks. We must stress that the optimal lag configuration (0,6,12,18) determined by

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 453-458

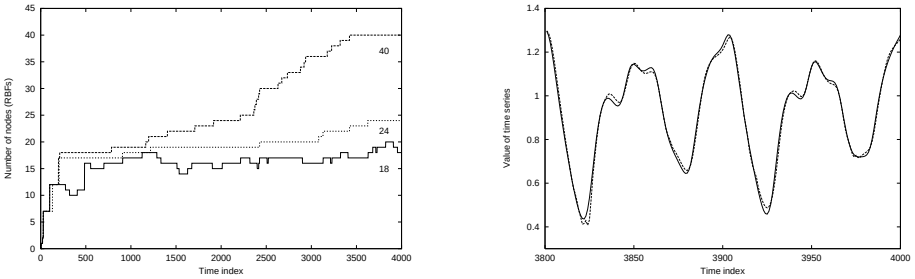

Figure 1: Left: comparison of the number of RBFs for on-line prediction of the Mackey-Glass series (upper plot = usual (RAN) algorithm; lower plots = two versions of the algorithm that use SVD and QR-cp for model reduction). Right: some real values of the series, along with the long-horizon values predicted using SVD and QR-cp methods.

our SVD and QR-cp method over input matrices (as suggested in the previous discussion) was the same proposed in the literature from ad-hoc, dynamical system theory considerations (e.g. [Kan95]). So, our algorithm determines adequate input models for prediction in an automated way.

5 Conclusions

It has been shown by experiments the fact (also claimed theoretically) that orthogonal decompositions can be a useful aid in any neural network construction algorithm. Thus, there is considerable potential in the use of these matrix formulations and associated theory, for the improvement of prediction systems based on neural models.

References

[Bjö96] Å. Björck. "Numerical Methods for Least Squares Problems". SIAM Publications. Philadelphia, U.S.A. (1996).

[Kan95] P.P. Kanjilal. "Adaptive Prediction and Predictive Control". Peter Peregrinus. Herts, U.K. (1995).

[MD89] J. Moody and C.J. Darken. Fast learning in networks of locally-tuned processing units. *Neural Computation* **1**, 281–294 (1989).

[MG77] M.C. Mackey and L. Glass. Oscillation and chaos in physiological control systems. *Science* **197**, 287–289 (1977).

[SOPP01] M. Salmerón, J. Ortega, C.G. Puntonet, and A. Prieto. Improved RAN sequential prediction using orthogonal techniques. *Neurocomputing* **41**(1–4), 153–172 (2001).