

Regression-type Neural Networks for System Identification

Cesare Alippi, Vincenzo Piuri

Dipartimento di Elettronica ed Informazione - Politecnico di Milano
P.za L. Da Vinci 32, 20133 Milano - Italy
Tel: +39223993534 Fax: +39223993411

Abstract: In this paper we prove the effectiveness of using neural networks of regression type to identify time series and non linear dynamic systems. It is experimentally shown that, whenever the process generating the data is ruled by a linear model (such as an ARMA for time series), performances provided by the neural network are comparable with the optimal predictor given by the Kolmogorov-Wiener theory. On the other hand, performances outcome classical linear identification approaches when the system to be modelled is intrinsically non-linear. The work extends the one suggested by Narendra et Al. in [7] by considering a reduced set of training data and a black-box model for the system to be identified.

1. Introduction

The identification problem is to infer relationships between past input data and future ones of an unknown time serie or a dynamical system. The ultimate goal is to provide a model approximating the behaviour of the system.

In classical approaches the search for the optimal approximating model is carried out by considering a parameterised identification family (most of times linear such as AR, ARMA for time series or with an exogenous variable i.e. ARX, ARMAX for dynamical process) and selecting the best model within the family according to a defined figure of merit [5]. Nevertheless, a linear model does not always suffice to adequately approximate a non-linear model; moreover the final approximator is biased (i.e. the loss function is not zero and in the presence of strong non-linearities it is far from zero). Solutions proposed in the literature generally take into account linearised models of the process; parameters are determined by using the linear system identification theory in a set of working points of the process. More complex solutions are sometimes envisaged to cope with insufficient approximation accuracy which generally encompass expansions in functionals (e.g. Volterra's expansion): the bottleneck is the computational load.

To this end, neural networks, because of their modularity and computational simplicity, are natural candidates to approximate a given model. In particular, neural networks of regression type (input-hidden-output

feedforward topologies with sigmoidal activation functions in hidden units and a single output neuron characterised by a linear activation mapper) have been proved to be universal approximator [3], [4], in the sense they are able to approximate any real function (and therefore the unknown behaviour of the system with an arbitrary precision) under mild assumptions.

The non-linear parametric family extends the linear one by providing non-linear models; among them the NAR, NARMA, NARX, NARMAX subfamilies where N stands for non-linear.

The structure of the paper is as follow: in section 2 we provide tools to optimally determine a regression type neural predictor. It is shown that, whenever the training set is sufficiently exciting (i.e. it contains the relevant information to infer the law ruling the time series) and in correspondence with ARMA models, the neural network provides performances comparable with the ones obtainable by using the Kolmogorov-Wiener predictor.

In fact, we experimentally proved that the generalisation error is practically coincident with the optimal one provided by the theory, thus showing the validity of identification by learning. The analysis is then extended in section 3 to cover the case of systems ruled by a non-linear dynamic. Finally, the proposed methodologies will be used in section 4 to identify the input/output relationship of a brushless motor used to control the joint of an industrial robot. The input is the current of the motor, the output is its angular velocity (and consequently the joint position). We will show that identification by neural networks provided better performances and a simpler parameter identification procedure than the ones obtained with the optimal linear ARX model.

2. Prediction of ARMA models

Let we consider a time serie ruled by the linear ARMA model (suggested in [6])

$$\begin{aligned} y(t) = & 0.95 \cdot y(t-1) - 0.99 \cdot y(t-2) + \\ & + u(t) + 0.3 \cdot u(t-1) + \\ & + 0.4 \cdot u(t-2) + 0.5 \cdot u(t-3) \end{aligned} \quad (1)$$

being $u(t)$ a white gaussian noise with zero mean and λ^2 variance. The optimal one-step predictor, according to the Kolmogorov-Wiener theory, becomes

$$\begin{aligned}\hat{y}(t+1/t) = & -0.3 \cdot \hat{y}(t/t-1) - 0.4 \cdot \hat{y}(t-1/t-2) + \\ & -0.5 \cdot \hat{y}(t-2/t-3) + 1.25 \cdot y(t) + \\ & -0.59 \cdot y(t-1) + 0.5 \cdot y(t-2)\end{aligned}\quad (2)$$

while the associated prediction error coincides with the variance of the white noise and therefore $\text{Var}[y(t) - \hat{y}(t/t-1)] = \lambda^2$.

Let us now focus the attention on the neural predictor. The process generating the data, namely the ARMA model of (1), can be seen as a black box with a single measurable output y and no manipulable inputs. As a consequence, the neural predictor has to extract information uniquely from the available set of y 's.

A structure for the one-step neural predictor is given in figure 1; the neural predictor receives a set of past y 's and infers the next one. In multi-step prediction, say we want to predict the k -th step, the neural predictor receives the set of past y 's till time t and infers the $y(t+k)$ one and the procedure coincides with the one-step one.

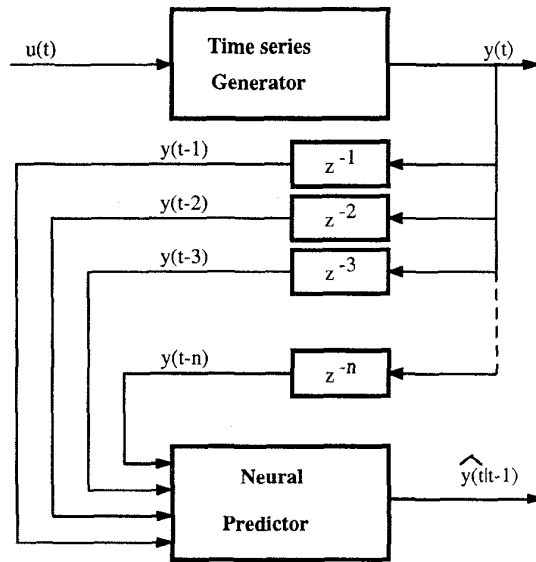


Figure 1: Structure of one-step predictor

It should be noted that the dimension N_I of the past y 's window (and therefore the input complexity of the neural predictor) is unknown, as well as the number N_H of hidden units; such dimensions need to be determined to obtain an optimal neural predictor in the generalisation ability sense.

Once chosen a supervised learning modality (based on a back-propagation like training procedure, for instance),

we have to properly construct a relevant training set to be extracted from the time series. To this purpose let

$$y_1, y_2, y_3, \dots, y_t \quad (3)$$

be the given time series. We will consider the case where the number of available data is large enough and therefore it is possible to split the data into two parts: the first lot of N_D data to accomplish the training procedure, the second one to validate the model by cross-validation. If the data set is extremely small, cross-validation techniques cannot be taken into account and FPE, AIC and RIS model selecting and validating criteria should be envisaged [2].

The construction of an appropriate training set requires now the definition of three entities

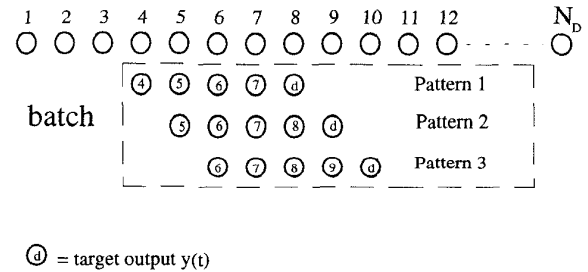
- The number of batches N_B to be extracted from the training data;
- The number of patterns N_P constituting a batch;
- The number of times N_L a generic batch is presented to the network during learning.

Operatively, during the training procedure, an index k is uniformly selected from the $[1, N_D - N_I - N_P + 2]$ interval and defines the starting point for the batch. Patterns are now extracted to create the N_P learning schemes (*input*, *output*), where *input* addresses the past y 's window and *output* the target output (the next value provided by the time series in a one-step prediction or the $y(t+k)$ in a k -step prediction).

Such learning schemes are now presented to the network one by one and weights are updated accordingly to the chosen learning algorithm. The training procedure operates now N_L times on the given batch.

An example of the batch extraction procedure is shown in figure 2. Data are given as in expression (3) by substituting the datum $y(i)$ with the generic i -th circle. In particular, it has been chosen $k=4$, $N_I=4$, $N_P=3$.

The determination of an optimal network requires to match the information content present in the application with the topological complexity of the neural structure and can be done by applying an optimisation procedure, a spectral decomposition technique [1] for instance.



ⓓ = target output $y(t)$

Figure 2: Extraction of a training batch

The method requires an overdimensioned network to be successfully trained (e.g. we considered a 6-3-1 topology) and locally optimises each layer by reducing the number of input neurons first, and the number of hidden units then. The methodology relies on the information content flowing through the layer to be optimised. In particular, the most relevant singular values associated with the output of the layer coincides with the minimal topology.

A set of $N_D=100$ data were considered for the training phase while the 700 consequent data created the validation set (the variance λ^2 of the noise was set to be one). Training was based on the classical back-propagation algorithm with the choice $N_B=50$, $N_P=5$, $N_L=10$.

The spectral decomposition approach was then applied and led to a network with two inputs (and this is reasonable if we note that the AR component of the ARMA model has time window with dimension two), while the number of hidden neurons was set to be two as well. Cross-validation proved that such topology was the best one (in the generalisation ability sense).

The most critical interval (within the validation set) was the 300-400 one. Comparison between the optimal predictor as given in (2) and the best neural one are given in figure 3 and 4. In particular figure 3 presents the neural network's performance while figure 4 the ones

provided by the Kolmogorov-Wiener predictor. It can be seen that the neural prediction almost coincides with the optimal one. To quantify the validity of the approach, we measured the mean squared errors $E(W)_{tr}$ and $E(W)_{val}$ over the training and the validation sets. Results are given in table 1.

	100 training data	remaining 700 data
Kolm.-Wiener	$E(W)_1=1.4$	$E(W)_2=0.88$
NN 6-3-1	$E(W)_{tr}=2.46$	$E(W)_{val}=2.52$
NN 2-2-1	$E(W)_{tr}=0.79$	$E(W)_{val}=1.02$

Table 1: Comparison between neural predictors and the optimal one

It can be seen that the neural network with structure 6-3-1 is not optimal since the validation error (i.e. the correct estimation of the prediction error variance) is far from the optimum (i.e. 1). On the other hand, the 2-2-1 topology provides performances comparable with the optimal one, thus proving the validity of the approach. Several experiments were carried out by both considering AR and ARMA models for the process

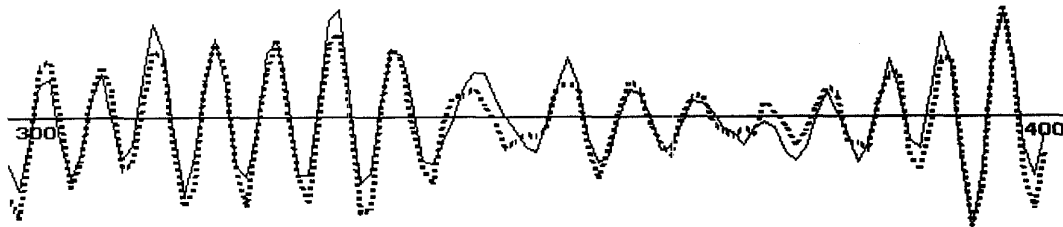


Figure 3: Performances provided by the neural network in the worst interval

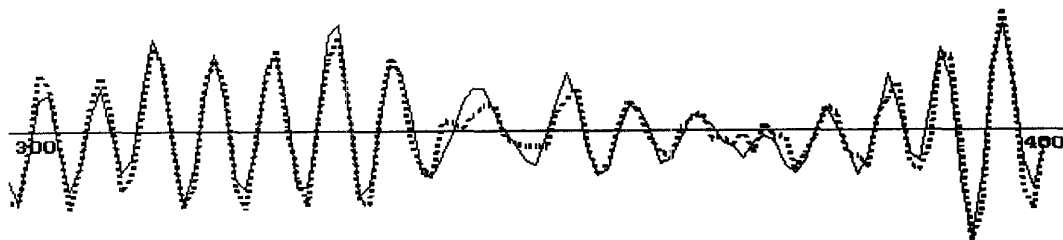


Figure 4: Performances provided by the one-step Kolmogorov-Wiener predictor.

generating the data. In all cases, whenever the network topology is optimised, the neural predictor provides performances comparable with the Kolmogorov-Wiener optimal predictor.

3. Identification of dynamical systems

Identification of dynamical systems with neural networks has been suggested for the first time in [7]; under the stationarity hypothesis for the process generating the data, it was shown how neural networks of regression type are able to solve the problems. The proposed methodology presents anyway two fundamental limits which prevent the use of neural identification techniques for most of real applications:

- it is required a partial knowledge about the presence and placement of non linearities within the process (e.g. on inputs, on the AR component);
- a large number of data (in [7] at least 100.000 data) are requested.

Both requirements are unfeasible in many real applications since the model of the process is generally unknown and must be treated as a black-box; besides, a limited number of data is usually available since there is a cost to extract and measure process variables. Such limits can be solved by applying the training methodology proposed in this section.

In the first extension the process is treated as a black-box, and it is reasonable to assume that the function ruling the identifier is similar to a non linear ARX or NARX. At the end of the training procedure the discrepancy between the output of the real process and that of the neural must be small independently of the input signal.

A scheme for the neural identifier is given in figure 5 which extends identification carried out with ARX models. In fact, the output of the identifier depends on past inputs and past outputs (where past is related to a temporal window) and can be expressed as

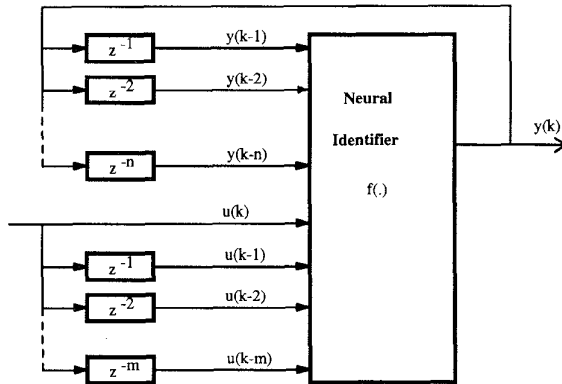


Figure 5: The working scheme of a neural identifier

$$y(k) = f(u(k), \dots, u(k-m), y(k-1), \dots, y(k-n))$$

Once more, as in the neural predictor, m and n are unknown and need to be finely tuned to optimise performances.

During learning, the links between the real process and the identifier are as depicted in figure 6 (in the hypothesis the training procedure is based on an identification error scheme such as in the mean squared error), during the recall phase the process and the identifier are completely disconnected and the identifier works in the figure 5 configuration.

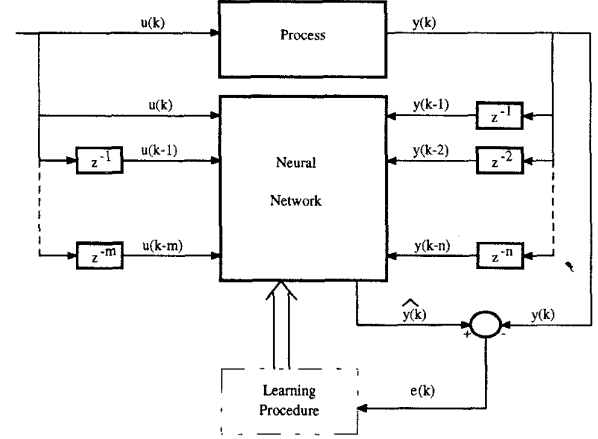


Figure 6: The neural network and the process during training

The problem concerning the large number of data required by [7] can be solved by using the same training procedure suggested in section 2 for the neural predictor. In particular, experiments show that the number of patterns per batch N_B and the number of training cycles per batch N_L should be 1. In this way, all training schemes have the same relevance. By applying this training procedure, a regression type neural network suffices to solve the benchmarks suggested in [7] (instead of a two layered structure); besides, the number of neurons is smaller.

As a second more complex system to be identified (see also [8]) we consider here

$$y(k) = \frac{y(k-1) + [y^2(k-2) \cdot u(k-1)]^2 + [y^2(k-3) \cdot u(k-2)]^2 + u(k)}{1 + y^2(k-3) \cdot y(k-2) + y(k-3) \cdot y^2(k-2)} \quad (4)$$

The optimal network topology is determined by using the spectral decomposition approach and lead to $m=3$, $n=2$ for a total of 6 inputs. This is in complete agreement with expression (4) where the temporal window dependency is the same. The final network structure is a 6-5-1 topology while training is characterised by $u(t)=WN(0,1)$, $N_D=100$ e $N_B=80000$.

The use of a white gaussian noise as an input stimulus is fundamental since it is able to influence the whole process dynamic.

The number of training data is greatly reduced with respect Narendra's methodology. In our approach just 100 data are required to successfully train the neural network instead of thousands. The number of training data was determined by measuring the validation error with respect to different training set.

Validation performances are presented in figure 7 and 8 where inputs are respectively $u(t)=WN(0,1)$ and

$$u(t) = 0.3 \cdot \sin\left(\frac{2 \cdot \pi \cdot k}{25}\right) + 0.4 \cdot \sin\left(\frac{2 \cdot \pi \cdot k}{32}\right) + 0.3 \cdot \sin\left(\frac{2 \cdot \pi \cdot k}{40}\right) \quad (5)$$

As we can see a very simple network is able to identify the process behaviour.

4. Identification of the input/output relationship of a brushless motor

The problem of identifying the model of a brushless motor represents an interesting application. The motor is used to control a joint of a six-joints robot arm (Comau-SMART 3S) able to handle an object at the end-effector. The control input for the motor is a current while the output is its angular velocity which allows positioning of the joint.

The non-linearities of the process are mainly concentrated on coulomb friction and hystereses on speed reducers.

For a fine control, non linearities need to be taken into account. This would require an adequate and accurate model for the whole process. However, it is, unfortunately, either unknown or too complex.

The goal is to identify the input-output relationship by considering the motor as a black-box. During data measurement, all the joints are kept fixed apart from the one to be identified. A set of 1000 input/output schemes were measured by considering different loads for the gripper. We obtained good identifications and we will focus the attention on the 6th joint for which also classical identification techniques have been used.

Such techniques led to an ARX(3, 3) model. Neural identification was carried out by firstly considering an overdimensioned network of regression type characterised by $N_D=300$ and $N_B=30000$ and then reducing the topological complexity by means of the spectral decomposition approach. The final optimal neural network structure was 18-10-1 (12 past outputs, 5 past inputs and the actual one).

Performances on the validation set are plotted for the ARX(3,3) and the neural network in figures 9 and 10, respectively. To quantify the improvement in performance, we computed the global mean squared error in both cases which was 1,42 for the neural network optimal structure and 2,12 for the optimal ARX model.

Results prove how the neural model was able to better identify the model. The residual error was then computed: the ARX model and the neural identifier are shown in figures 11 and 12, respectively. The ARX model is biased; therefore, being an optimal model, it is not able to catch non-linearities. The residual error associated with the neural identifier is closer to a white noise.



Figure 7: Validation performance in correspondence to $u(t)=WN(0,1)$

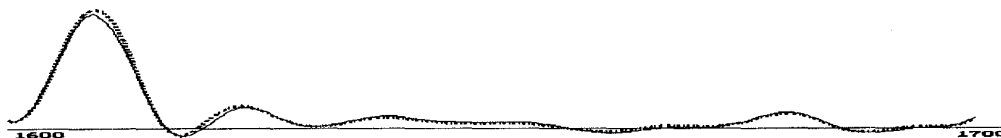


Figure 8: Validation performance in correspondence to the input given in (4)

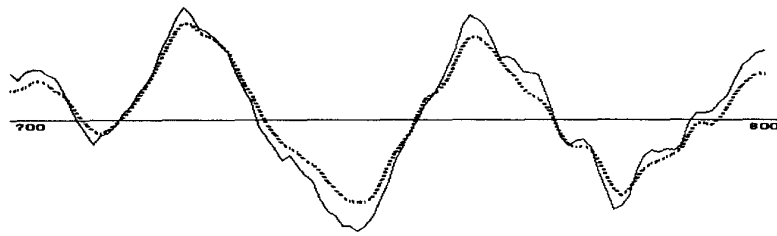


Figure 9: The ARX(3,3) model's performance on the validation set

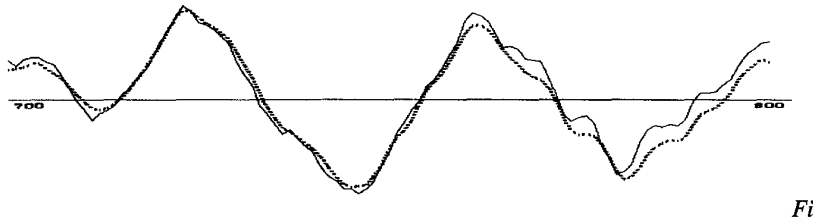


Figure 10: The neural network performance on the validation set

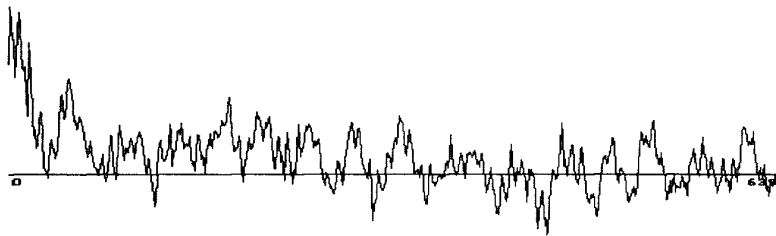


Figure 11: residual error for the ARX(3,3) model

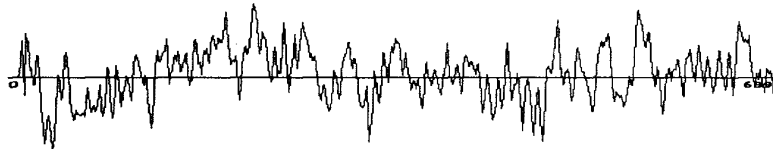


Figure 12: residual error for the neural identifier

5. Conclusions

In this paper we proved how neural networks of regression type can be successfully used to identify time series and dynamical systems. In particular, whenever the time series is ruled by a linear approach, the neural network provides comparable performances with the ones obtainable with the optimal predictor set by the Kolmogorov-Wiener theory. Identification by neural networks are an interesting and valuable tool to simply identify the process behaviour.

References

- [1] C. Alippi, V. Piuri: "Topological Minimisation of Multi-Layered Feed-Forward Neural Networks by Spectral Decomposition", Proc. of the IEEE International Joint Conference on Neural Networks IEEE-IJCNN92, Beijing, China, November 1992.
- [2] C. Alippi: "Extending the FPE and the Effective number of parameters to biased estimators: how to select and validate optimal neural topologies", Internal report of the Department of Brain and Cognitive sciences, MIT, Cambridge, March 1995.
- [3] G. Cybenko: "Approximation by superposition of sigmoidal function", Math Control System Signals, 1989.
- [4] K. Hornik, M. Stinchcombe, H. White: "Multilayer feedforward networks are universal approximators", Neural Networks, vol 2, 1989.
- [5] L. Ljung: System identification: theory for the user, Prentice Hall, New Jersey, 1987.
- [6] M. Marseguerra, S. Minoggio, A. Rossi, E. Zio: "Neural network predictions and fault diagnosis applied to stationary and non stationary ARMA modeled time series", Progress on Nuclear Energy, 1992
- [7] K.S. Narendra, K. Parthasarathy: "Identification and control of dynamical systems using neural networks", IEEE Transaction on Neural Networks, vol. 1, no. 1, 1990.
- [8] F. Previdi, M. Serraton: Identificazione e Predizione di sistemi dinamici usando reti neurali, MSc thesis, Dip. Elettronica, Politecnico di Milano, 1993.