A NEW FEEDFORWARD NEURAL NETWORK HIDDEN LAYER NEURON PRUNING ALGORITHM[1]

F. Fnaiech* *Member IEEE* - **N. Fnaiech*** - **M. Najim**** : *Fellow IEEE*
*E.S.S.T.T. Laboratoire CEREP, 5 Av. Taha Hussein, 1008 Tunis, TUNISIA..
Email: Farhat.Fnaiech@esstt.rnu.tn
**Equipe Signal & Image, ENSERB, BP 99, 33402 Talence Cedex, France.
Email: najim@tsi.u-bordeaux.f

ABSTRACT

This paper deals with a new approach to detect the structure (i.e. determination of the number of hidden units) of a feedforward neural network (FNN). This approach is based on the principle that any FNN could be represented by a Volterra series such as a nonlinear input-output model. The new proposed algorithm is based on the following three steps: first, we develop the nonlinear activation function of the hidden layer's neurons in a Taylor expansion, secondly we express the neural network output as a NARX (nonlinear auto regressive with exogenous input) model and finally, by appropriately using the nonlinear order selection algorithm proposed by Kortmann-Unbehauen, we select the most relevant signals on the NARX model obtained. Starting from the output layer, this pruning procedure is performed on each node in each layer. Using this new algorithm with the standard backpropagation (SBP) and over various initial conditions, we perform Monte Carlo experiments leading to a drastic reduction in the nonsignificant network hidden layer neurons.

1. INTRODUCTION

Multilayer perceptrons form a class of feedforward neural networks [3]. In spite of their popular use, there is still a lack of research work concerned with their architecture design. One major problem is to determine the optimal number of hidden units needed to mimic the system, by only using the input-output patterns. In [2], the authors used an iterative pruning algorithm to find the most appropriate network size. Their method is based on solving a linear system by least squares identification algorithm. Indeed, for large network size, the output matrix may have a deficient rank and infinite solutions may exist. The problem of model selection is handled differently in [9], where the authors consider it as a statistical problem and hence use the generalization of Akaike's information criterion (AIC) [1]. In fact, the problem is to find the optimal model in a family of networks and find the optimal parameter to approximate the system's conditional distribution which is computationally consuming. In [4], a statistical stepwise method for weight elimination has been used to solve the architecture pruning problem. In their statistical approach the authors fitted linear models to neural networks where the SBP becomes mathematically not justified. In this paper, we propose a new algorithm to solve the size pruning problem of NN hidden layer units. Among its different steps, this new approach uses in part the NARX model order determination algorithm developed first by Kortman-Unbehauen [5] [6]. Within this framework, the NN may be seen as a Volterra series [8], specifically as a NARX model. The paper is organized as follows: in the second section, we show the equivalence between a feedforward neural network and a NARX model. Then in section 3, we propose the new NN hidden layer's neurons pruning algorithm. Finally in section 4, some simulation results highlight the effectiveness of the proposed algorithm. In the appendix, we review the algorithm proposed by Kortmann-Unbehauen to the order selection of NARX models.

2. FEEDFORWARD NEURAL NETWORKS EQUIVALENT TO NARX MODEL

The general deterministic NARX input-output equation is given by:

$$y(k) = \bar{y} + \sum_{i=0}^{n} b_i u(k-i-d) +$$

$$\sum_{i=0}^{n} \sum_{j=i}^{n} b_{ij} u(k-i-d) u(k-j-d) \quad .... \quad +$$

$$\sum_{i=0}^{n} \,........\, \sum_{v=p}^{n} \sum_{q=v}^{n} b_{i......q} u(k-i-d) ...... u(k-q-d) \quad ...+$$

$$\sum_{i=1}^{m} a_i y(k-i) \; + \sum_{i=1}^{m} \sum_{j=i}^{m} a_{ij} y(k-i) y(k-j) \; +$$

[1] This work is supported by the Tunisian/ French project CMCU#00F1125.

$$+\sum_{i=1}^{m}\ldots\ldots\sum_{v=p}^{m}\sum_{q=v}^{m} a_{i\ldots\ldots q}\, y(k-i)\ldots\ldots y(k-q)+ \\ \ldots\ldots\ldots\, + \\ \sum_{i=0}^{n}\sum_{j=1}^{m} c_{ij}\, u(k-i-d)\,y(k-j) \\ \sum_{i=0}^{n}\ldots\sum_{v=0}^{n}\sum_{q=1}^{m} c_{i\ldots vq}\, u(k-i-d)\ldots u(k-v-d)\,y(k-q)+\ldots \\ \sum_{i=0}^{n}\sum_{j=0}^{n}\ldots\sum_{q=1}^{m} c_{ij\ldots q}\, u(k-i-d)\,y(k-j)\ldots y(k-q) \tag{2.1}$$

where $y(k)$ is the output signal and $\overline{y}$ its mean value, $u(k)$ the input signal, n the order of the input regression, m is the order of the output regression, q the non linearity order and d the system delay.

In the sequel, we give some results on the equivalence between NN and NARX models. Equivalent results may be found in [8]. Eq. (2.1) is a general Volterra series expansion describing the input-output relation of a discrete time causal, nonlinear, time invariant system.

Let us consider a general feedforward neural network of L layers and assume a single neuron taken at the j^{th} hidden layer **Fig. 1**

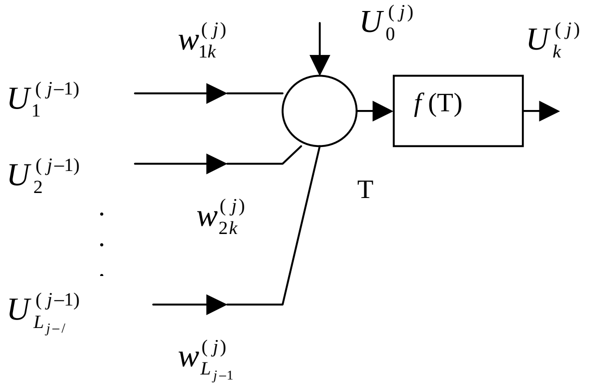

Fig. 1

If we consider a Taylor expansion of the sigmoid non linearity to the second order, the output of this unit becomes:

$$U_k^{(j)} = f(T) = \alpha_0 + \alpha_1 T + \alpha_2 T^2 \tag{2.2}$$

where $T = U_0^{(j)} + \sum_{s=1}^{L_{j-1}} w_{ks}^{(j)} U_s^{(j-1)}$. (2.3)

Here the signals $U_s^{(j-1)}$ are related to the layer (*j-1*). In order to develop these signals as NARX models, the following lemma is stated:

Lemma:

Let us define $A_s = w_{ks}^{(j)}, X_s = U_s^{(j-1)}$ and $L = L_{j-1}$, the following equality holds:

$$[\sum_{s=1}^{L} A_s X_s]^2 = \sum_{i=1}^{L}(A_i X_i)^2 + \sum_{i=1}^{L-1}\sum_{j=i+1}^{L} 2A_i A_j X_i X_j \tag{2.4}$$

Using this lemma and substituting Eq. (2.3) into Eq. (2.2), it is possible to derive the following output of the hidden unit of figure (2.1) as:

$$U_k^{(j)} = \alpha_0 + \alpha_1 U_{0k}^{(j)} + \alpha_2 (U_{0k}^{(j)})^2 + \\ \sum_{s=1}^{L_{j-1}} (\alpha_1 + 2\alpha_2 U_{0k}^{(j)}) w_{ks}^{(j)} U_k^{(j-1)} + \sum_{i=1}^{L_{j-1}} \alpha_2 (w_{ks}^{(j)})^2 (U_k^{(j-1)})^2 + \\ \sum_{r=1}^{L_{j-1}-1}\sum_{p=r+1}^{L_{j-1}} 2\alpha_2 w_{kr}^{(j)}) w_{kp}^{(j)} U_r^{(j-1)} U_p^{(j-1)} \tag{2.5}$$

For convenience the time index is omitted here. Equating Eq.(2.1) to Eq.(2.5), the coefficients of the NARX model can be expressed in terms of the NN weights and the polynomials coefficients of the activation function as:

$$\overline{y} = \alpha_0 + \alpha_1 U_{0k}^{(j)} + \alpha_2 (U_{0k}^{(j)})^2 \tag{2.6}$$

$$b_s = \left(\alpha_1 + 2\alpha_2 U_{0k}^{(j)}\right) w_{ks}^{(j)} \tag{2.7}$$

$$b_{rp} = \begin{cases} \alpha_2 \left(w_{kr}^{(j)}\right)^2 & si \quad r = p \\ 2\alpha_2 w_{kr}^{(j)}\, w_{kp}^{(j)} & si \quad r \neq p \end{cases} \tag{2.8}$$

$$u(k-s) = U_s^{(j-1)} \tag{2.9}$$

Note that all the terms related to the sequence $y(k)$ are null. Finally by inspecting Eq. (2.5) and Eq. (2.6)-(2.9) the output of a single neuron in a feedforward NN is a NARX model. Consequently, we can expect that at a given hidden layer j we have L_j NARX models, this yields of running the proposed pruning algorithm L_j times over j layers.

3. NN STRUCTURE DETECTION ALGORITHM

3.1 Introduction

Polynomial filters or Volterra models is an important class representing real world signals and systems [8]. Structure detection of these polynomial filters i.e. selection of the statistically significant terms is treated in the work of Kortman-Unbehauen [5] [6]. In the general nonlinear model of Eq.(2.1), the number of possible terms up to the degree q of the polynomial is given in [5] by:

$$r = \frac{(1+n+m+q)!}{q!(1+n+m)!} - 1 \tag{3.1}$$

Let us define:

$$\underline{p} = \begin{bmatrix} \overline{y} & \theta_1 & \theta_2 & \theta_3 & \ldots\ldots & \theta_r \end{bmatrix}^T \tag{3.2}$$

as the parameter vector, and

$$\underline{m}(k) = \begin{bmatrix} 1 & v_1 & v_2 & v_3 & \ldots\ldots\ldots & v_r \end{bmatrix}^T \tag{3.3}$$

as the signal vector. Here

$$v_1 = u(k), v_2 = u(k-1), \quad .....$$
$$v_{n+1} = u(k-n), \quad v_{n+2} = y(k-1),...$$
$$v_{n+m+1} = y(k-m), \quad v_{n+m+2} = u^2(k-1) \quad (3.4)$$
$$v_{n+m+3} = u(k)u(k-1),.. \quad v_r = y^q(k-m)$$

Eq. (2.1) can be expressed as:

$$y(k) = \underline{m}^T(k)\underline{p} \qquad (3.5)$$

Minimizing the sum of the squares of the equation error for N data points:

$$\min_p \sum_{k=1}^{N} e^2(k) = \min_p \sum_{k=1}^{N} \left[y(k) - \underline{m}^T(k)\underline{p}\right]^2 \qquad (3.6)$$

yields a least-square estimation of $\hat{\underline{p}}(k)$ [7].

3.2 New NN hidden layer neuron pruning algorithm:

In this section, we give the different steps of the new algorithm proposed.

Step 1: application definition, choice of N input/ output data patterns.

Step 2

1. arbitrary choice of a NN maximal structure namely NN($L_i, L_1,..., L_{l-1}, L_o$) L_i and L_o are fixed by the type of application L_{l-1} is number of units in layer number *l-1*
2. randomly initialize the initial synaptic coefficients
3. perform the training of the neural network.

Step 3:

1. start from the output layer *j=l-1;*
2. determine all the output signals $U_k^{L_o}$ for *k=1* to L_o function of all the hidden layer signals $U_p^{L_{l-1}}$ for *p=1* to L_{l-1}. Put the signals in the form of Eq.(2.5), thus define the vectors $\underline{m}_{L_{l-1}}^T$
3. use the Kortmann-Unbehauen algorithm (*step1* up to *step 8* in [5] summarized in Appendix A) to find significant terms entering in the computation of the output layer signals. Then, find the new neural network hidden layer namely *New* L_{l-1}.
4. using the same initial conditions as in step 2.2, reconstruct the new NN($L_i, L_1,..., NewL_{l-1}, L_o$)
5. perform the training of the new neural network
6. decrement $j = j-1$, and return to step 3.2
7. test *j*. If j=0 then go to step 4.

Step 4 :

1. we shall obtain a new NN($L_i, NewL_1,..., NewL_{l-1}, L_o$)
2. using the same initial condition, train the new optimal net.

4. SIMULATION RESULTS

In the classic application of the circle in the square problem, the NN have to decide whether a point of coordinate (x, y) varying from –0.5 to 0.5 is in the circle of radius equal to 0.35. The starting NN structure chosen is first NN(2, 6, 1) and then NN(2, 10, 1). It is shown that when applying the new algorithm, we have a considerable reduction in the range of (~20% up to 40%) of the number of units in the hidden layer for many trials of initial condition (IC) realizations. We have also tested the new algorithm on other applications. We find out that on a Monte Carlo test of many initializing realizations, we have often a significant reduction in the number of hidden layers units.

IC realiz ation	#Start hidden units	BIC start NN	#Hid. units new NN	BIC new NN	results
1	6	-3.829	5	-4.096	good
2	6	0.4672	4	-2.953	good
3	6	-3.918	6	-3.898	bad
4	6	-4.101	4	-3.804	bad
5	6	-3.978	5	-4.007	good
6	6	-3.993	5	-4.006	good
7	6	-3.978	5	-4.013	good
8	6	-3.983	4	-4.034	good
9	6	-3.979	4	-1.380	bad
10	6	-4.060	5	-4.006	bad
11	10	-3.872	7	-3.956	good
12	10	-3.875	9	-3.900	good
13	10	-3.891	7	-3.964	good
14	10	-3.870	8	-3.931	good
15	10	-3.950	6	-3.995	good
16	10	-3.891	7	-3.964	good
17	10	-3.870	8	-3.931	good

Table. 1 results of the circle in the square problem

3.3 Comments:

The above table provides the results for the circle in the square problem. The results are considered good if the BIC* test of the new NN is lower than the BIC of the starting large NN. In some cases, in spite of the reduction of the number of hidden neurons, the BIC test criterion remains greater than for the starting net see rows 3, 4, 9 and 10 in the Table. 1.

5. CONCLUSION

In this paper, we have presented a new structure detection algorithm used to optimize the number of hidden units in feedforward NN. The proposed algorithm together with the back propagation training algorithm often leads to a fairly small network structure with satisfactory classification accuracy.

* BIC (Beysian Information Criterion) see Appendix A

APPENDIX

***Review of the Kortmann-Unbehauen NARX structure detection algorithm*:**

Before reviewing Kortmann-Unbehauen NARX structure detection algorithm [5] [6], let us define the information criteria used to get reliable decision on the order detection procedure:

a) the final prediction error technique (FPE) [1]

$$\lambda = FPE(v) = N.Ln\left[\frac{1}{N}\sum_{k=1}^{N}\varepsilon^2(k)\right] + N.Ln\left[\frac{N+v}{N-v}\right] \quad (3.7)$$

b) Akaike's information criterion (AIC) [1]

$$\lambda = AIC\ (v) = N.Ln\left[\frac{1}{N}\sum_{k=1}^{N}\varepsilon^2(k)\right] + 2.v \quad (3.8)$$

c) Khinchin's law of iterated logarithm criterion (LILC):

$$\lambda = LILC(v) = N.Ln\left[\frac{1}{N}\sum_{k=1}^{N}\varepsilon^2(k)\right] + 2.v.Ln[Ln(N)] \quad (3.9)$$

Beysian Information criterion (BIC):

$$\lambda = BIC\ (v) = N.Ln\left[\frac{1}{N}\sum_{k=1}^{N}\varepsilon^2(k)\right] + v.Ln(N) \quad (3.10)$$

Where v is the number of model parameters and N is the number of data points. $\varepsilon(k)$ represents the residuals i.e. $\varepsilon(k) = y(k) - \hat{y}(k)$. Therefore $\varepsilon(k)$ is an estimation of the equation error $e(k)$. The optimal model is chosen by decision criteria Eq. (3.7)-(3.10) when λ is minimal.

Let us resume the main steps of the Kortmann-Unbehauen algorithm for more details see.[5]:

Step 1: initialize maximal values of the backward shift of the input output : n, and m, and the maximal degree q of the polynomial to define the size of the nonlinear model Eq. (2.1). Calculate r by Eq.(3.1) Set the number of parameter *v=1*. Specify the threshold δ of the F-test. Choose the information criteria for decision.

Step 2*:* Correlate all r possible terms with the output signal $y(k)$ and determine the normalized correlation coefficients ρ_i [5] [6] for each term.

Step 3: Select the term with the highest (partial) correlation coefficient to determine the optimal variable which has to be added to the model, i.e. the one that contributes the greatest improvement to fit the model: $v_{opt_v} : \rho_{opt} = \max_i \{\rho_i\}$, increase the number of parameters by one to $\upsilon \rightarrow \upsilon + 1$.

Step 4: – Estimate the v parameter of the model with recursive least squares algorithm (RLS) and calculate $\varepsilon(k) = y(k) - \bar{y}(k) \quad k = 1,\ldots,N$

- Determine the static variable (overall F-test) [5].

- Calculate the multiple correlation coefficient R^2 [5], R^2 measures the proportion of the total variation about the mean value $\bar{y}$ of the output signal model.

- Compute the information criteria defined by Eq.(3.7)-(3.10): FPE_v AIC_v $LILC_v$ BIC_v.

Step 5: Check the value of the overall F- test for significance, and compare R^2 and the information criteria with their values obtained for *v-1* parameters of the previous model. The procedure quits if the F-test shows that the model equation is not significant or that the information criteria are greater than the previous model.

Step 6: Compute for each term in the model, excluding the mean value, the stochastic variable (partial F- test), $PF_i \ i = 1,\ldots,\upsilon - 1 \quad \upsilon > 2$

FPE_i AIC_i $LILC_i$ BIC_i, follow the rest of step 6 in [5]

Step 7 Examine the partial information criteria and the partial F- values for all current model variables. Check for the rejected terms and parameter estimation for the remaining model, see details in [5].

Step 8: Compute the normalized partial correlation coefficient of all possible remaining terms ρ_i - for more details see [5]-.

Keywords: Neural Network (NN), Standard Back-Propagation (SBP), pruning algorithm, structure detection, NARX.

6. REFERENCES

[1] H. Akaike "A new look at the statistical model – validation" *IEEE Trans. Auto. Control , Vol. AC-19, N°6, pp. 716-723, 1974.*

[2] G. Castellano, A. M. Fanelli and M. Pellilo, "An Iterative Pruning Algorithm for Feedforward Neural Networks" *IEEE ,Trans. Neural Networks ,Vol.8,N°3, pp.519-531, May, 1997.*

[3] A. Cichocki and R. Unbehauen, "Neural Network for Optimization and Signal Processing" John Wiley & Sons, England, 1993.

[4] M. Cottrell and B. Girard "Neural Modeling for Time Series: A Statistical Stepwise Method for Weight Elimination" *IEEE Trans. Neural Networks. Vol. 6, N°6, Novembre 1995.*

[5] M. Kortmann and H. Unbehauen, " Structure detection in the identification of nonlinear systems, " *APII, N°22, pp. 5-25, Special issue edited by M. Najim, Traitement de Signal (in French), 1988.*

[6] M. Kortmann, "Die Identifikation nichtlinearer Ein- und Mehrgröβensysteme auf der Basis nicht Modellansätze," Doctorate dissertation Nr 77, Ruhr-University, Bochum, 1988.

[7] L. Ljung, " System identification theory for the user," *Prentice Hall, 1987.*

[8] V.Z. Marmarelis and X. Zhao," Volterra Models and three layer perceptrons," *IEEE Trans. Neural Networks, Vol.8, N°6, pp. 1421-1433, November, 1997.*

[9] N. Murata, S. Yoshizawa and S- I. Amari " Network Information Criterion: determining the Number of Hidden Units for an Artificial Neural Network Model," *IEEE. Trans. Neural Networks, Vol.5, N°6, November 1994.*