Neural Network Modelling of Nonlinear Channels with Memory[1]

Mohamed IBNKAHLA and Francis CASTANIE

Institut National Polytechnique de Toulouse
ENSEEIHT, 2, Rue Camichel
31071 Toulouse Cedex, France
E-mail : ibnkahla@len7.enseeiht.fr

RÉSUMÉ

Le papier propose une approche basée sur les réseaux de neurones pour modéliser les canaux non linéaires avec mémoire. Ce papier présente uniquement les algorithmes et leur comportement pendant l'apprentissage. Plusieurs applications peuvent être trouvées dans [5].

ABSTRACT

The paper proposes a neural network approach for modelling nonlinear channels with memory. The paper only presents the algorithms and their learning bahavior. Several applications can be found in [5].

1 Introduction

Several nonlinear adaptive techniques have been proposed for modelling nonlinear channels with memory. These techniques include Volterra series, wavelet networks, neural networks, etc. See [11, 6] for a review. For example, adaptive Volterra series have been applied in [1] for modelling digital satellite channels. These Volterra models do not characterize each element of the channel. They provide a model only for the overall channel input-output transfer function. However, Volterra models provide an estimation of the global system memory and the complexity of the nonlinear transfer function.

Non adaptive parametric techniques also have been proposed for modelling nonlinear systems with memory (e.g. [3, 4, 7, 8, 9]). Nikias and Petropulu [7] review higher order statistic-based methods used for detection and characterization of nonlinearities. In particular, they present examples for the characterization of Volterra series coefficients. Block-oriented methods have been largely used for nonlinear system identification. These methods are based on the idea that the system to be identified is composed of several simple subsystems. For example, several authors studied Hammerstein systems which consist of the cascade of systems composed of a nonlinear memoryless element followed by a linear dynamic one. See for instance [4, 8, 9].

The key issue in adaptive system identification is to find the best model structure within which an optimal model has to be found by using an appropriate adaptive algorithm. This paper proposes adaptive neural network approaches for modelling nonlinear channels with memory. It is shown that a good choice of the neural network structure should follow directly from the application and the prior knowledge on the physical system to be modelled.

This paper presents only the algorithms and their learning behavior. In [5] two typical problems are addressed:

i) identification and characterization of digital channels which are composed of physically separable parts (e.g. linear filters with memory and nonlinear memoryless devices). An example is a digital satellite channel, composed of a linear filter followed by a memoryless nonlinear travelling wave tube amplifier (TWT) and a second linear filter. The neural network approach models the global nonlinear channel input-output transfer function and characterizes each component of the channel separately. The learning process uses only the channel input-output signals.

ii) modelling nonlinear channels which cannot be simply represented by separable parts. An example is the solid state power amplifier (SSPA) (nonlinear amplifiers with memory) used typically in satellite communications.

The analytic analysis of neural network algorithms applied for modelling nonlinear channels can be found in [2] and [5].

The paper is organized as follows. Section 2 gives an example of a nonlinear channel. Sections 3 and 4 present the neural network algorithms. Finally, section 5 is devoted to the algorithm learning behavior.

2 Example of a nonlinear channel

The satellite channel is a typical nonlinear channel. It consists of two earth stations connected by a satellite repeater through two radio links (uplink and downlink). As an example, figure 1 represents a complex base-band model. The transmission filter F0, the IMUX (input multiplexing) filter F1, and the OMUX (output multi-plexing) filter F2 are linear. The TWT is a memoryless nonlinearity with a complex transfer function which depends only on the input complex envelope. The TWT exhibits two kinds of nonlinearities, amplitude distortion (AM/AM conversion) and phase distortion (AM/PM conversion) [1, 10]. The TWT AM/AM and AM/PM conversions are represented by Saleh model

[1]This work has been supported in part by the French Space Agency (CNES), Toulouse, France, under contract 962/94/CNES/1232/00.

[10]: $A(r)=\frac{\alpha_a r}{1+\beta_a r^2}$, $\phi(r)=\frac{\alpha_p r^2}{1+\beta_p r^2}$, where r is the TWT input amplitude, $\alpha_a=2,\beta_a=1,\alpha_p=4.0033,$ and $\beta_p=9.104$. The TWT gain is defined as: $G(r)=\frac{A(r)}{r}=\frac{\alpha_a}{1+\beta_a r^2}$.

3 Identification structure

This part proposes to identify the on-board devices (F1-TWT-F2). The neural network identification scheme is presented in figure 2. Figure 3 shows the neural network structure which duplicates the satellite channel structure (i.e. a memoryless nonlinear system between two linear systems). The neural net is composed of a linear filter (W1), a nonlinear network (NLN), and a second linear filter (W2). The nonlinear network structure also duplicates the TWT amplifier structure. It is composed of two sub-networks which correspond to the gain and phase conversions, respectively. Note that these gain and phase conversions only depend on the input signal amplitude (as in the TWT).

This structure characterizes separately each channel component i.e. the filters F1 and F2 and the TWT amplitude and phase conversions. Note that a classical multi-layer neural network cannot separately characterize each channel component but only can model the overall channel input-output behavior.

The neural net operates as follows. W1 filters the complex-valued input $x(n)$ (real FIR filtering). The output of W1 is $y(n)=\sum_{k=0}^{N_1-1} w_{1k}x(n-k)$ (II.1)

where N_1 is the length of W1.

The nonlinear network (NLN) has two nonlinear parts: 1) NLNG is the gain conversion; 2) NLNP is the phase conversion.

The squared amplitude $\rho(n)$ of the first filter output $y(n)$ is presented to both NLNG and NLNP. Their respective outputs $G(n)$ and $\phi(n)$ are expressed as:

$$G(\rho(n))=\sum_{k=1}^{N_G} w_{G2k}\, f\left(w_{G1k}\rho(n)+b_{G1k}\right)+b_{G2}, \quad \text{(II.2)}$$

$$\phi(\rho(n))=\sum_{k=1}^{N_P} w_{P2k}\, f\left(w_{P1k}\rho(n)+b_{P1k}\right)-f\left(b_{P1k}\right), \quad \text{(II.3)}$$

where $\rho(n)=r^2(n)=\|y(n)\|^2$, f is a nonlinear activation function, and N_G and N_P are the number of neurons in NLNG and NLNP, respectively.

Note that $\phi(0)=0$, i.e. the phase origin is 0 by construction.

The output of the NLN is given by:

$$z(n)=G(\rho(n))e^{j\phi(\rho(n))}y(n) \quad \text{(II.4)}$$

The NLN transfer function is similar to that of a TWT amplifier: nonlinear gain and phase conversions which depend only upon the input signal amplitude.

Finally, the second filter W2 output is given by:

$$s(n)=\sum_{k=0}^{N_2-1} w_{2k}z(n-k), \quad \text{(II.5)}$$

where N_2 is the length of W2.

4 Algorithm

The network learning is achieved at each iteration using the channel input-output complex signal pair.

The network weights are adjusted with a gradient descent algorithm which minimizes the actual squared error between the channel output $d(n)$ and the neural network output. The squared error at time n is:

$$J(n)=\|e(n)\|^2=e^2{}_R(n)+e^2{}_I(n)=\left\|d(n)-\sum_{k=0}^{N_2} w_{2k}(n)z(n-k)\right\|^2, \quad \text{(II.6)}$$

where R and I denote the real and imaginary parts, respectively.

The stochastic gradient recursion for the second filter is given by:

$$w_{2i}(n+1)=w_{2i}(n)+2\mu\left(e_R(n)z_R(n-i)+e_I(n)z_I(n-i)\right) \quad \text{(II.7)}$$

The stochastic gradient recursions for the first filter and the nonlinear neural network does not follow immediately from (II.6). This is because the error at time n depends upon $\{w_{2k}(n), k=0,1,\dots,N_2\}$ and $\{z(n-k), k=0,1,\dots,N_2-1\}$. Equations (II.1-4) show that $z(n-k)$ depends upon $\{w_{1j}(n-k), j=0,1,\dots,N_1-1\}$ and the nonlinear network weights at time n-k. Therefore the gradient calculation at time n requires the partial derivatives of $J(n)$ according to the N_2-1 past values of the weights.

For example, the gradient of $J(n)$ according to $w_{1i}(n)$ can be written as:

$$\frac{dJ(n)}{dw_{1i}(n)}=\frac{d\,J\left(w_{1i}(n),w_{1i}(n-1),\dots,w_{1i}(n-\lambda),\dots,w_{1i}(n-N_2+1)\right)}{dw_{1i}(n)}.$$

If μ is sufficiently small, the weights can be assumed to be slowly varying. Then, the gradient can be approximated by

$$\frac{d\,J(n)}{dw_{1i}(n)}\approx\sum_{k=0}^{N_2-1}\frac{\partial\,J\left(w_{1i}(n),w_{1i}(n-1),\dots,w_{1i}(n-\lambda),\dots,w_{1i}(n-N_2+1)\right)}{\partial w_{1i}(n-k)}$$

(II.8)

Therefore, the gradient in Eq. (II.8) depends upon the actual partial derivative of $J(n)$ according to $w_{1i}(n), w_{1i}(n-1),\dots$ and $w_{1i}(n-N_2+1)$. In order to simplify the algorithm, Eq. (II.8) is troncated and only the $J(n)$ most recent terms are used:

$$\frac{d\,J(n)}{dw_{1i}(n)}\approx\sum_{k=0}^{\lambda}\frac{\partial\,J\left(w_{1i}(n),w_{1i}(n-1),\dots,w_{1i}(n-\lambda)\right)}{\partial w_{1i}(n-k)}. \quad \text{(II.9)}$$

Thus, if $\lambda=0$, only the actual term is used. If $\lambda=N_2-1$, then all the terms in equation (II.8) are used.

The following presents the updating rule for a given λ. The algorithm (parametrized by λ) will be denoted by $A(\lambda)$.

NLNG update

Second layer weights:

$$w_{G2k}(n+1)=w_{G2k}(n)+2\mu\sum_{i=0}^{\lambda} w_{2i}(n)\delta_{G2}(n,i)x_{G1k}(n-i), \quad \text{(II.10)}$$

$$b_{G2}(n+1)=b_{G2}(n)+2\mu\sum_{i=0}^{\lambda} w_{2i}(n)\delta_{G2}(n,i), \quad \text{(II.11)}$$

where

$$x_{G1k}(n)=f\left(net_{G1k}(n)\right),\ net_{G1k}(n)=w_{G1k}(n)\rho(n)+b_{G1k}(n) \quad \text{(II.12)}$$

and

$$\delta_{G2}(n,i)=e_R(n)\left(\cos(\phi(n-i))y_R(n-i)-\sin(\phi(n-i))y_I(n-i)\right)$$
$$+e_I(n)\left(\sin(\phi(n-i))y_R(n-i)+\cos(\phi(n-i))y_I(n-i)\right)$$

(II.13)

First layer weights:

$$w_{G1k}(n+1)=w_{G1k}(n)+2\mu\sum_{i=0}^{\lambda} w_{2i}(n)\delta_{G2}(n,i)w_{G2k}(n-i)f'\left(net_{G1k}(n-i)\right)\rho(n-i)$$

(II.14)

$$b_{G1k}(n+1)=b_{G1k}(n)+2\mu\sum_{i=0}^{\lambda} w_{2i}(n)\delta_{G2}(n,i)w_{G2k}(n-i)f'\left(net_{G1k}(n-i)\right) \quad \text{(II.15)}$$

NLNP update

Second layer weights:

$$w_{P2k}(n+1) = w_{P2k}(n) + 2\mu \sum_{i=0}^{\lambda} w_{2i}(n) G(\rho(n-i)) \delta_{P2}(n,i) x_{P1k}(n-i)$$

(II.16)

where

$$x_{P1k}(n) = f(net_{P1k}(n)) - f(b_{P1k}(n)), \quad net_{P1k}(n) = w_{P1k}(n)\rho(n) + b_{P1k}(n)$$

(II.17)

and

$$\delta_{P2}(n,i) = e_R(n)\left(-\sin(\phi(n-l))y_R(n-l) - \cos(\phi(n-l))y_I(n-l)\right) + e_I(n)\left(\cos(\phi(n-l))y_R(n-l) - \sin(\phi(n-l))y_I(n-l)\right)$$

(II.18)

First layer weights:

$$w_{P1k}(n+1) = w_{P1k}(n) + 2\mu \sum_{i=0}^{\lambda} w_{2i}(n) G(\rho(n-i)) \delta_{P2}(n,i) w_{P2k}(n-i) f'(net_{P1k}(n-i)) \rho(n-i)$$

(II.19)

$$b_{P1k}(n+1) = b_{P1k}(n) + 2\mu \sum_{i=0}^{\lambda} w_{2i}(n) G(\rho(n-i)) \delta_{P2}(n,i) w_{P2k}(n-i) \left(f'(net_{P1k}(n-i)) - f'(b_{P1k}(n-i)) \right)$$

(II.20)

Update of W1:

$$w_{1j}(n+1) = w_{1j}(n) + 2\mu . \sum_{i=0}^{\lambda} w_{2i}(n) \begin{pmatrix} \delta_{G2}(n-i) \sum_{k=1}^{N_G} w_{G2k}(n-i) f'(net_{G1k}(n-i)) w_{G1k}(n-i). \\ .2\left(y_R(n-i)x_R(n-i-j) + y_I(n-i)x_I(n-i-j)\right) \\ + G(\rho(n-i))\delta_{P2}(n-i) \begin{pmatrix} w_{P2k}(n-i) f'(net_{P1k}(n-i)) w_{P1k}(n-i) \\ .2\begin{pmatrix} y_R(n-i)x_R(n-i-j) \\ + y_I(n-i)x_I(n-i-j) \end{pmatrix} \end{pmatrix} \\ + G(\rho(n-i)) \begin{pmatrix} e_R(n) \begin{pmatrix} \cos(\phi(\rho(n-i)))x_R(n-i-j) \\ -\sin(\phi(\rho(n-i)))x_I(n-i-j) \end{pmatrix} \\ + e_I(n) \begin{pmatrix} \sin(\phi(\rho(n-i)))x_R(n-i-j) \\ + \cos(\phi(\rho(n-i)))x_I(n-i-j) \end{pmatrix} \end{pmatrix} \end{pmatrix}$$

(II.21)

Remarks:

1) The above algorithms have some common properties with the classical backpropagation algorithm such as error backpropagation, parallelism, etc.
2) The two nonlinear sub-networks NLNG and NLNP can be adjusted in parallel and simultaneously (i.e. the updating terms of each sub-network do not depend on the other).
3) The computational complexity (CC) increases as λ increases (e.g. the CC of $A(\lambda = 1)$ equals at least two times that of $A(\lambda = 0)$).
4) Note that the above algorithms can be slightly modified in order to model systems which are composed simply of two blocks rather than three, for example a linear filter followed by a TWT can be modelled by a W1-NLN structure. Note that simulation results have shown that if you use a W1-NLN-W2 structure to model a two block structure composed of F1-TWT (resp. TWT-F2), then W2 (resp. W1) converges to a constant.

5 Algorithm behavior

The number NLNG and NLNP neurons are $N_G = N_P = 9$ in the simulations below. The activation function is the hyperbolic tangent function. Various λ have been used so as to observe its effect on algorithm behavior and performance.

The input signal is a uniformly distributed white noise. Its power was chosen such that the TWT operates at saturation.

In order to accelerate the convergence, the algorithm starts with small N_1 and N_2 (e.g. $N_1 = N_2 = 5$). Then, 5 weights are added to W1 and W2 after each 25000 iterations (i.e. $N_1(n = 25000) = N_2(n = 25000) = 5 + 5 = 10$, etc.). The new weights are initialized at 0. When $N_1 = N_2 = 60$ (i.e. at the 275000^{th} iteration), the number of weights is fixed until the end of the learning process.

Figures 4-6 show the algorithm mean-square-error (MSE) learning for different learning rates μ. The MSE is calculated over blocks of 2500 iterations, i.e. $MSE(n, n \in [k, k+2500]) = \frac{1}{2500} \sum_{i=k+1}^{k+2500} \|e(i)\|^2$.

Comments on the convergence behavior:

1) At the beginning of the learning phase, the algorithms display a rapid convergence (the MSE has a stair-case shape). This is due to the increase of N_1 and N_2 at each of the 25000 iterations.
2) The MSE asymptotically approaches a small value. However, near convergence, the MSE oscillates. The two model filters begin alternative oscillations about their optimum values. These oscillations are more apparent when the learning rate μ is high.
3) Figures 4-6 show faster convergence as λ increases. Therefore, a tradeoff exists between convergence speed, MSE performance and computational complexity.
4) The algorithms were able to model each part of the channel (see e.g. [5]). However, although the memoryless portion of the satellite channel has been accurately modelled by the nonlinear network, the linear filters were mis-matched in delay, i.e. the impulse response of the first model filter W1 was delayed by Δ units of time relative to the impulse response of the first channel filter, whereas the second model filter W2 was advanced by Δ units of time relative to the impulse response of the second channel filter. Thus, the overall system delay was correct. Note that the delays can be removed if an instruction is added to the algorithm near convergence. This instruction consists of adding one unit advance (resp. one unit delay) to W1 (resp. to W2). If the learning MSE increases, then values of W1 and W2 (before adding the advance/delay) are kept. Otherwise, if the MSE decreases, this procedure is repeated (i.e. by adding a new unit advance/delay).

6 Conclusion

The paper presented neural network algorithms for modelling nonlinear channels with memory. A typical example of a nonlinear satellite channel was given. Simulation examples were given in order to illustrate the algorithms learning behavior.

References

[1] S. Benedetto, E. Biglieri, and V. Castellani, *Digital Transmission Theory*, Prentice Hall International, Englewood Cliffs, New Jersey, 1987.

[2] N. J. Bershad, M. Ibnkahla, and F. Castanié, "Statistical analysis of a two-layer backpropagation algorithm used for modelling nonlinear memoryless channels: the single neuron case", IEEE Trans. On Signal Processing, Vol. SP-45, No. 3, pp. 747-756, March 1997.

[3] Giunta, G. Jacovitti, and A. Neri, "Bandpass nonlinear identification by higher order cross correlation", IEEE Trans.

On Signal Processing, Vol. 39, No. 9, pp. 2092-2095, September 1991.

[4] W. Greblicki, "Nonlinearity estimation in Hammerstein systems based on ordered observations", IEEE Trans. on Signal Processing, Vol. 44, No. 5, May 1996.

[5] M. Ibnkahla, N. Bershad, J. Sombrin, and F. Castanié, "Neural network modelling and identification of nonlinear channels with memory: algorithms, applications and analytic models", IEEE Trans. On Signal Processing, to appear.

[6] A. Juditsky et al., "Nonlinear black-box modeling in system identification: Mathematical foundations", Automatica, Vol. 31, No. 12, pp. 1725-1750, 1995.

[7] C. L. Nikias and A. P. Petropulu, "Higher-order spectra analysis- A nonlinear signal processing framework", Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[8] S. Prakriya and D. Hatzinakos, "Blind identification of LTI-ZMNL-LTI nonlinear channel models", IEEE Trans. Signal Processing, Vol. 43, No. 12, pp. 3007-30013, December 1995.

[9] J. Ralston, A. Zoubir, and B. Bouashash, "Identification of a class of nonlinear systems under stationary non gaussian excitation", IEEE Trans. Signal Processing, Vol. 45, No. 3, March 1997.

[10] A. Saleh, "Frequency-independent and frequency-dependent nonlinear models of TWT amplifiers", IEEE Trans. Communications, Vol. Com-29, No. 11, November 1981.

[11] J. Sjoberg et al., "Nonlinear black-box modeling in system identification: a unified overview", Automatica, Vol. 31, No. 12, pp. 1691-1724, 1995.

Figures:

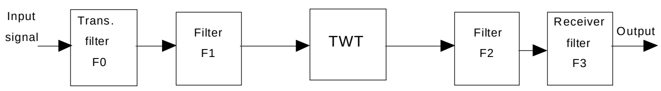

Figure 1: A simplified satellite channel.

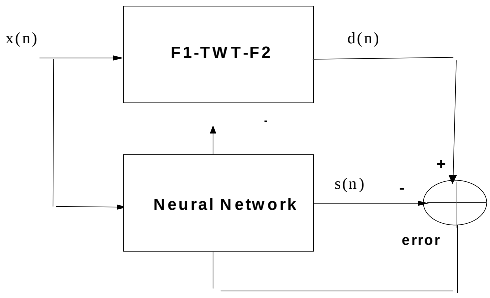

Figure 2: Identification structure.

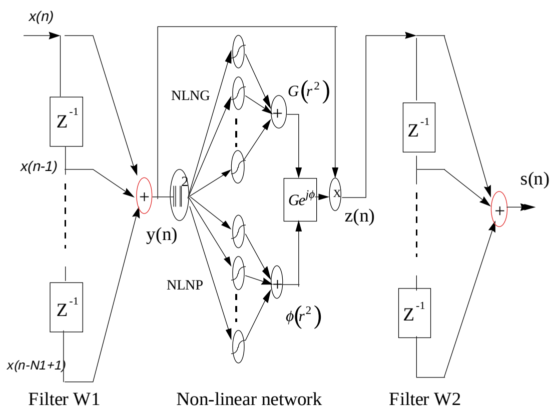

Figure 3: Neural network structure

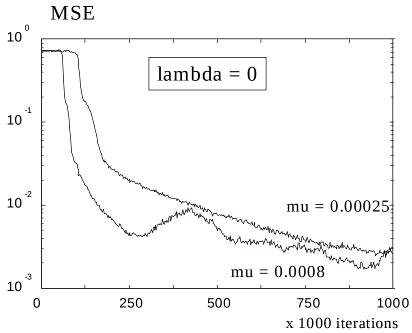

Figure 4: Learning curves, $\lambda = 0$.

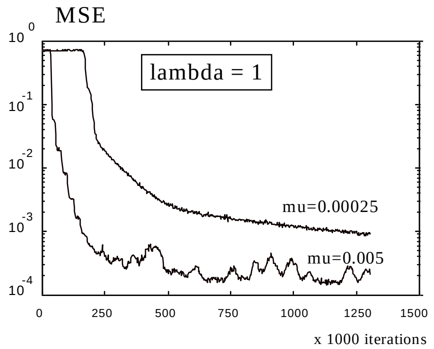

Figure 5: Learning curves, $\lambda = 1$.

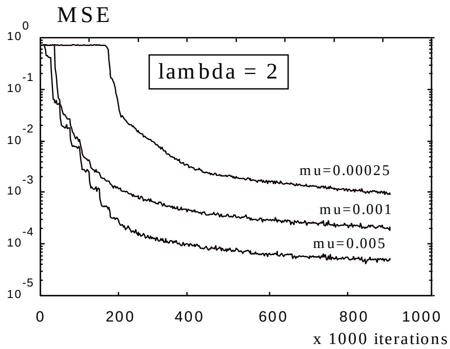

Figure 6: Learning curves, $\lambda = 2$.