The Fourth International Conference on
Control, Automation, Robotics and Vision
(ICARCV '96)
Singapore, 3-6 December, 1996

A Multivariable Functional Artificial Neural Network

Louiza Sellami[#]*, Robert W. Newcomb*[!], and Rui J. P. de Figueiredo[+]

Electrical Engineering Department, US Naval Academy, 105 Maryland Ave., Annapolis, MD 21402 USA
email: sellami@eng.umd.edu, URL; http://www.ee.umd.edu/~sellami
* Microsystems Laboratory, Electrical Engineering Department, University of Maryland, College Park, MD 20742 USA, email: newcomb@eng.umd.edu; URL; http://www.ee.umd.edu/newcomb/mslab.html
! Electrical Engineering Department, POSTECH, Pohang 790-784, Korea
+ Department of Electrical and Computer Engineering Department and Department of Mathematics, University of California at Irvine, Irvine, CA 92697 USA, email: rui@ece.uci.edu

Abstract

The previous functional artificial neural network is extended to handle functions of several variables, as may be of interest in two variable picture processing. The structure of a two-layer feedforward functional neural network is chosen with the processing being of multivariable functions rather than point evaluations. The neurons are represented by multivariable Volterra functionals in Fock space with synaptic weights being themselves multivariable functions. An example illustrates the theory.

I. Introduction

Recently there has arisen an interest in functional artificial neural networks [1] [2] [3] because of their increased capabilities over the more classical neural networks now to be found in standard textbooks [4]. For example functional neural networks allow the identification of a model for a model referenced system through its time-domain kernel, such as its impulse response, rather than through a discrete set of input-output data points. Previously, in [5], we introduced the basic functional neural network with its theory based upon that of nonlinear system identification as developed by Zyla and de Figueiredo [6]. Here we follow up by extending the same ideas to systems defined over many variables, such as the two dimensions of the plane used for picture processing.

In section II we present the basic theory of n variable functional neural networks which is close enough to the one variable theory that section II can also serve as a review of the one variable case. Then in section III we give the resulting feedforward n variable functional neural network. In section IV we consider an example and in section V we give a short discussion.

II. Basic n-Variable Theory

Because we are interested in characterizing dynamical systems described by nonlinear mappings $V(\cdot)$ of input functions $u=u(\cdot)$ into output functions $y=y(\cdot)$ we consider the Volterra series representation as it is a description of great generality. In order to carry out an identification we specify a real n dimensional "cube" $I=I_1xI_2...xI_n$ of n dimensional variable x with components x(1), ..., x(n) defined respectively over the intervals I_1, ..., I_n. I is the set over which identification is to be made. By definition we take $y(\cdot)=V_\cdot(u(\cdot))$ which when evaluated at position x is denoted $y(x)=V_x(u(\cdot))$; written as a Volterra series this is [2]

$$y(x) = V_x(u(\cdot)) = \sum_{k=0}^{\infty}\frac{1}{k!}\int_{x_1}\cdots\int_{x_k} h_k(x;x_1,\ldots,x_k)u(x_1)\ldots u(x_k)dx_1\ldots dx_k \quad (2.1)$$

in which the Volterra map $V_\cdot(\cdot)$ is characterized by the kernels h_k; these latter designate k-(multi)linear maps defined on the inputs as indicated by the integrals (the ith integration in (2.1) is an n-dimensional one over the n components $x_i(1),...x_i(n)$ of the ith instance of x). We make all of the assumptions of [6] on the spaces to which the various variables and operators belong, reviewing some of these as we proceed. For simplicity of the treatment we limit to the single-input single-output real-valued case, that is u and y are taken to be one-dimensional real valued functions of the n dimensional real variable x, though extensions to multiple input multiple output cases are readily made. Also on physical grounds and in line with [6] we assume that u and y and sufficiently many, K, of the outputs derivatives have finite energy by taking all such functions to be square integrable over I.

The identification of the system as carried out in [6] rests crucially upon the nonlinear Volterra functional $V_x(\cdot)$ belonging to a special reproducing kernel Hilbert space, called a Fock space and designated F_x and the associated Volterra operator $V.(\cdot)$ belonging to a Bochner space B_n^2, these assumptions being equivalent to assuming the following three physically reasonable conditions:

a) The ith partial derivative, $h_{j,k}^{(i)} = \partial^i h_k / \partial x(j)^i$, with respect to x(j) of h_k exists everywhere on I for i=0,...,K-1 as a map from I into $L^2(I^k)$, where I^k is the nk-dimensional cube of "sides" I, and the ith partial derivative with respect to x(j) is a map from I into $L^2(I^{k+1})$.

b) There exists a real constant r such that

$$\sum_{k=0}^{\infty} \frac{r^k}{k!} \left\| h_k^{(i)}(\cdot;\cdot,\cdots,\cdot) \right\|_{L^2(I^{k+1})}^2 < \infty \quad \text{for } i = 0,\ldots,K \tag{2.2}$$

c) $h_k(x;x_1,...x_k)$ is invariant to permutations (that is, symmetric) in the variables $x_1,..., x_k$.

Although not so important physically, condition a) is needed for guaranteeing the mathematical existence of the reproducing kernel to be introduced while b) is needed to guarantee convergence of the Volterra series. Condition c) is of secondary importance since the nonsymmetric parts cancel out in the integrals anyway.

Given r we introduce the scalar product of two elements $V_x(\cdot)$ and $W_x(\cdot)$ of Fock space F_r as follows. Let h_k be the kernels for V and g_k be those for W, then

$$\langle V_x(\cdot), W_x(\cdot) \rangle_{F_r} = \sum_{k=0}^{\infty} \frac{r^k}{k!} \langle h_k(x;.,\ldots,.) g_k(x;.,\ldots,.) \rangle_{L^2(I^k)} \tag{2.3a}$$

where the scalar product of any two (Lebesgue) square integrable functions g and h of nk variables is given by

$$\langle g,h \rangle_{L^2(I^k)} = \int_{x_1}\int_{x_2} \cdots \int_{x_k} g(x_1,x_2,\ldots,x_k) h(x_1,x_2,\ldots,x_k) dx_1 dx_2 \ldots dx_k \tag{2.3b}$$

With this latter scalar product the Fock space of the Volterra functionals $V_x(\cdot)$ becomes an Hilbert space. There is also an Hilbert space associated with the Volterra operator $V.(\cdot)$ which will be needed for the system identification; thus the Bochner space B_K^2 becomes an Hilbert space if we associate with it the scalar product

$$\langle V.(\cdot), W.(\cdot) \rangle_{B_n^2} = \sum_{i=0}^{K-1} a_i \int_x \langle V_x^{(i)}(\cdot), W_x^{(i)}(\cdot) \rangle_{F_r} dx \tag{2.4}$$

where the a_i are any positive constants. We note that the operator $V.(\cdot)$, which maps the full input function $u(\cdot)$ into the full output function $y(\cdot)$, represents the system as a Bochner space map taking x in the cube I into the Fock space Volterra map $V_x(\cdot)$ which maps full input functions $u(\cdot)$ into output functions evaluating them at the n dimensional point x, that is into y(x).

A reproducing kernel for F_r is the following functional $K(\cdot,\cdot)$ which maps $L^2(I) \times L^2(I)$ into the real numbers

$$K(u,v) = \exp\left(\frac{1}{r} \langle u,v \rangle_{L^2(I)}\right) \tag{2.5}$$

To see that this K(,) is a reproducing kernel for the Hilbert Fock space note that

$$K(u,\cdot) = \exp\left(\frac{1}{r} \langle u,\cdot \rangle_{L^2(I)}\right) \tag{2.6}$$

If we expand this exponential in a power series indexed by k and if we set $K(u,\cdot)=W_x(\cdot)$ for (2.3) we see by observing (2.1) that the kernels for W are

$$g_k = \frac{1}{r^k} u \otimes u \otimes \ldots \otimes u = \frac{1}{r^k} u \otimes^k u \tag{2.7}$$

$\otimes^k$ = k-fold tensor product

In other words

$$\langle V_x(\cdot), K(u,\cdot) \rangle_{F_r} = \sum_{k=0}^{\infty} \frac{r^k}{k!} \langle h_k, (\frac{1}{r^k} u \otimes^k u) \rangle_{L^2(I)} = V_x(u) \tag{2.8}$$

That is, the scalar product in the Fock space of the Volterra kernel with the functional K(,) reproduces the Volterra kernel.

The beauty of using this reproducing kernel is that it reduces the estimation problem of nonlinear dynamical systems to that of linear operators. The details are carried out in [6] and next summarized here for use in our neural network.

We assume available m pairs of input-output test functions, $u_j(\cdot)$ and $y_j(\cdot)$ for $j=1,\dots,m$, with these functions (along with K derivatives of the output) being square integrable over I. We choose the m input functions to be linearly independent over I so that we have sufficient information to perform an identification. In preparation for the next section we note that these input-output function pairs serve like exemplars of artificial neural networks.

The desire is to identify a dynamical system characterized by $V_{\cdot}(\cdot)$such that

$$V_{\cdot}(u_j(\cdot))=y_j(\cdot) \qquad j=1,\dots,m \tag{2.9a}$$

subject to V describing the "best" such system, this latter being represented mathematically by V having the smallest norm, i.e.

$$\inf\|V_{\cdot}(\cdot)\|^2_{B^2_K} \quad \text{over all } V_{\cdot}(\cdot)\in B^2_K \tag{2.9b}$$

The number K of derivatives of interest plays an important mathematical role in solving this problem since this Bochner space norm is defined in terms of them by

$$\|V_{\cdot}(\cdot)\|^2_{B^2_K}=\sum_{i=0}^{K-1} a_i \int_{x\in I}\left\|V_x^{(i)}\right\|^2_{F_r} dx \tag{2.10}$$

The problem is actually tackled by solving the equivalent problem

$$\min \left\|V_x^{(i)}\right\|_{F_r} \quad \text{over all } x\in I \text{ and all } V_x^{(i)}\in F_r \tag{2.11a}$$

subject to

$$V_x^{(i)}(u_j(\cdot))=y_j^{(i)}(x) \quad \text{for } i=0,\dots,K,\ \ j=1,\dots,m \tag{2.11b}$$

The solution to the problem of equations (2.11) is rather easily phrased. First we form the m×m Grammian matrix

$$G=[G_{ij}]=[\exp(\frac{1}{r}\langle u_i(\cdot),u_j(\cdot)\rangle_{L^2(I)})] \tag{2.12a}$$

where for completeness we recall, see (2.3b), that the L^2 scalar product of functions of n variables is just the n-dimensional (Lebesgue) integral over the specified cube of the scalar product entries, that is

$$\langle u_i(\cdot),u_j(\cdot)\rangle_{L^2(I)}=\int_{x\in I} u_i(x)u_j(x)dx \tag{2.12b}$$

Note that G is nonsingular by virtue of the independence assumed for the input test functions. Forming the column m-vector of test outputs

$$y_{test}(\cdot)=[y_j(\cdot)] \tag{2.12c}$$

we obtain a column m-vector of coefficients

$$c(x)=G^{-1}y_{test}(x)=[c_j(x)] \tag{2.12d}$$

to place in the best estimation $V_x^{\wedge}(\cdot)$ of $V_x(\cdot)$. The key and end result is that this best estimate is given by [6, Eq.(20)]

$$V_x^{\wedge}(\cdot)=\sum_{j=1}^{m} c_j(x)\cdot\exp(\frac{1}{r}\langle u_j(\cdot),\cdot\rangle_{L^2(I)}) \tag{2.13}$$

It is upon equation (2.13) which we base our functional artificial neural network.

III. Two Layer n-Variable Functional Neural Network

The functional neural network of interest results from the key decomposition of equation (2.13) and is diagrammatically illustrated in Figure 1. Here m exemplar pairs $u_j(\cdot)$ $y_j(\cdot)$ are used to form the neural network coefficients $c(x)=G^{-1}y_{test}(x)$ according to equation (2.12d) where G is the Grammian matrix of the exponential of input scalar products, the scalar product being formed as the $L^2(I)$ integral so that G is a matrix of numbers, these numbers being the exponentials of these integrals as given in (2.12a). The entries of c(x) act as x-varying synaptic weights while the exponentials of the inputs preceding these weights act as neuron nonlinearities with the weighted neuron outputs summed to give the overall neural network output. Thus the exponentials act as input neurons feeding the synaptic weights which junction onto an output neuron which performs a linear summation.

Given an arbitrary input (of the class allowed by the system) this neural network gives an output which is an approximation to the output of the true system which the neural network is approximating. This approximation is based upon forcing the neural network to give the desired output on the exemplar inputs from which the system is approximated by the neural network. The approximation is in terms of functionals and as a consequence attempts to incorporate the nonlinearities and dynamics of the system which is being approximated. The key ideas are best illustrated by a simple example.

IV. A 2-D Example

Letting $1(\cdot)$ denote the unit step function, consider a system for which $u_1(x)=1$ gives $y_1(x)=1(0.5-[x(1)^2+x(2)^2])$ and $u_2(x)=0.5$ yields $y_2(x)=1(0.25-[x(1)^2+x(2)^2])$ all defined for $x(1)$ and $x(2)$ in the interval $[-1,1]$, that is $I=[-1,1]\times[-1,1]$. With these as test exemplars we have m=2 and find

$$\|u_1\|^2_{L^2(I)} = \int_{-1}^{1}\int_{-1}^{1} u_1(x)^2\,dx(1)dx(2) = 4$$

$$\|u_2\|^2_{L^2(I)} = \int_{-1}^{1}\int_{-1}^{1} u_2(x)^2\,dx(1)dx(2) = 2$$

$$\langle u_1, u_2\rangle_{L^2(I)} = \langle u_2, u_1\rangle_{L^2(I)} = \int_{-1}^{1}\int_{-1}^{1} u_1(x)u_2(x)\,dx(1)dx(2) = 2$$

from which G is calculated according to equation (2.12a), after choosing r=0.5 we have

$$G = \begin{bmatrix} e^8 & e^4 \\ e^4 & e^4 \end{bmatrix},\quad G^{-1} = \frac{1}{e^4-1}\begin{bmatrix} e^{-4} & -e^{-4} \\ -e^{-4} & 1 \end{bmatrix}$$

In turn the synaptic weights $c(x)=G^{-1}y_{test}(x)$ are given as

$$c(x) = \frac{1}{e^4-1}\begin{bmatrix} e^{-4} & -e^{-4} \\ -e^{-4} & e^8 \end{bmatrix}\begin{bmatrix} 1(0.5-[x(1)^2+x(2)^2]) \\ 1(0.25-[x(1)^2+x(2)^2]) \end{bmatrix}$$

For the two input neurons we calculate the exponentials of (2.13) to be

$$\exp(\frac{1}{r}\langle u_1, u\rangle_{L^2(I)}) = \exp(2\int_{-1}^{1}\int_{-1}^{1} u(x(1),x(2))dx(1)dx(2)$$

$$\exp(\frac{1}{r}\langle u_2, u\rangle_{L^2(I)}) = \exp(\int_{-1}^{1}\int_{-1}^{1} u(x(1),x(2))dx(1)dx(2)$$

We note that these terms, although nonlinear in the input, are independent of x, all of the x dependence having now been placed in c(x). For reference we state that the original system for which this neural network is an approximation processes pictures by cropping them over the unit circle with grey scale depending upon the amplitude of the input over the circle.

V. Discussion

In this paper we have extended our previous functional neural network from one variable to n variables, essentially by replacing the one variable, formerly designated as t for time, by a vector variable x representing an n dimensional quantity, such as two or three dimensional space (or four dimensional space time). It should be noted that this FANN can be generalized to use more general Hilbert spaces than L^2 and is to be considered as an optimal interpolative (OI) neural network [7].

References

[1]. E. Alcorta and E. N. Sanchez, "Non-Linear Identification via Neural Networks" ACASP/92 (preprints of the Fourth IFAC International Symposium on Adaptive Systems in Control and Signal Processing) Grenoble, France, July 1992, pp. 675 - 680.

[2]. E. N. Sanchez, "Dynamic Neural Networks for Nonlinear Systems Identification," Proceedings of the 33rd Conference on Decision and Control, Lake Buena Vista, Florida, December 1994, pp. 2480 - 2481.

[3]. R. W. Newcomb and R. J. P. de Figueiredo, "A Multi-Input Multi-Output Functional Artificial Neural Network," Journal of Intelligent and Fuzzy Systems, to appear.

[4]. J. Zurada, "Introduction to Artificial Neural Systems," West Publishing Co., St. Paul, MN, 1992.

[5]. R. W. Newcmb and R. J. P. de Figueiredo, "A Functional Artificial Neural Network," The Third International Conference on Automation, Robotics and Computer Vision (ICARV'94), Singapore, November 1994, pp. 566 - 570.

[6]. L. V. Zyla and R. J. P. de Figueiredo, "Nonlinear System Identification Based on a Fock Space Framework," SIAM Journal on Control and Optimization, Vol. 21, No. 6, November 1983, pp. 931 - 939.

[7]. R. J. P. de Figueiredo, "The OI, OS, OMNI, and OSMAN Networks as Best Approximations of Nonlinear Systems Under Training Data Constraints," Proceedings of the 1996 IEEE ISCAS, Atlanta, May 1996, Vol. 3, pp. 349 - 352.

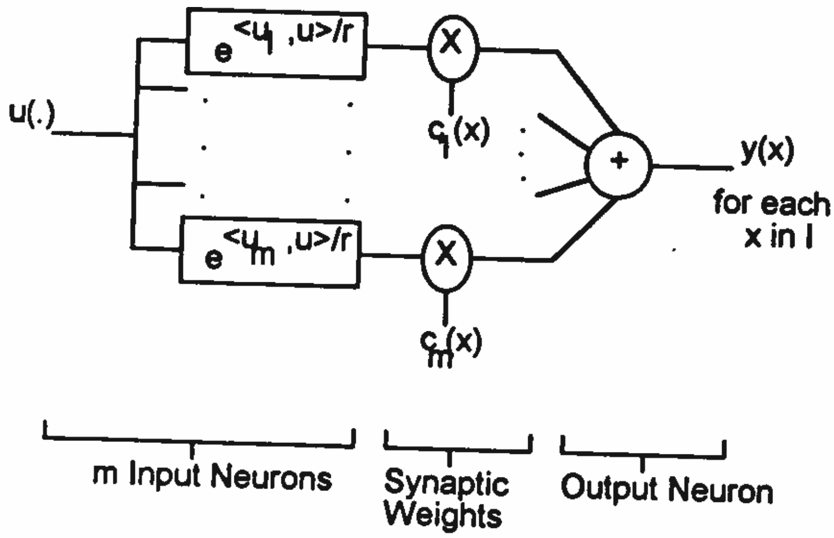

Figure 1
n Dimensional Functional Artificial Neural Network

ICARCV'96

FOURTH INTERNATIONAL CONFERENCE ON CONTROL, AUTOMATION, ROBOTICS AND VISION

4-6 December 1996
Westin Stamford, Singapore

PROCEEDINGS

VOLUME 2 OF 3

ORGANISED BY

School of Electrcial and Electronic Engineering
Nanyang Technological University, Singapore