The Third International Conference on
Automation, Robotics and Computer Vision
(ICARCV '94)
Singapore, November 9-11, 1994

A Functional Artificial Neural Network

Robert W. Newomb%
% Microsystems Laboratory
Electrical Engineering Department
University of Maryland
College Park, MD 20742 USA
Phone: [1]-(301)-405-3662
email: newcomb@eng.umd.edu

and

Rui J. P. de Figueiredo+
+ Department of Electrical and Computer Engineering
University of California
Irvine, CA 92717 USA
Phone: [1]=(714)-725-3408
email: rui@uci.edu

Abstract

An artificial neural network is proposed which processes functions rather than data point evaluations of functions. This functional neural network uses neurons which are functionals and is based upon the system identification techniques of Zyla and deFigueiredo. As such it uses Volterra functionals in a Fock space, which is a reproducing kernel Hilbert space, with weights which are themselves functions. The main advantage is that this functional neural network can identify systems as functional input-output maps rather than mappings of data points into data points.

1 Introduction

Present day artificial networks normally take input data, weights it and sums into a nonlinear output sigmoidal type of output function. Standardly the input data is a numerical data set, such as the pixel intensity of a discretized image. As such these artificial neural networks have proven to be effective classifiers and pattern recognizers in situations where closed form mathematical solutions are hard to obtain. But in many situations the data is a set of functions, rather than their values at specific arguments. Such would be the case for system identification where one desires the input output map as an operator rather than as the set of input-output data values. In the case of linear time-invariant systems such an operator could be specified by the transfer function or equivalently by the impulse response function. If we treat the latter in a functional way, for example as the kernel of the convolution input-output map, then we become interested in functional maps as descriptions of systems and significant problems of systems identification become those of identifying functional maps. Although the convolution functionals, represented by their kernels, characterize all linear time-invariant continuous systems, when we turn to the more prevalent nonlinear systems the situation is much more complex. Unfortunately the mathematics for general nonlinear systems is still rather primitive in terms of obtaining practical results. However, the theory of Volterra functionals is developed to the point that in the abstract one can obtain a Volterra functional representation of a system given its sufficient input and output function pairings. Here we review in section II the situation, as presented in [1], for system identification via Volterra functionals in using a Fock space. Then in section III we apply the results of section II by introducing a functional neural network which solves the minimum norm problem in a Bochner space related to the Fock space to which the Volterra functional belongs. In so doing the functional neural network is trained on input-output exemplar function pairings to set the weights, which themselves are functions. Then the functional neural network carries out the system identification by associating a Volterra functional input-output map.

2 Review of Fock Space Identification Theory

Because we are interested in characterizing dynamical systems described by nonlinear mappings V 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 interval I of time t over which identification is to be made. By definition we take $y(\cdot)=V_{\cdot}(u(\cdot))$ which when evaluated at time t is denoted $y(t)=V_t(u(\cdot))$; written as a Volterra series this is [2]

$$y(t) = V_t(u(\cdot)) = \sum_{k=0}^{\infty}\frac{1}{k!}\int_{t_1}\cdots\int_{t_k} h_k(t;t_1,\ldots,t_k)u(t_1)\ldots u(t_k)dt_1\ldots dt_k \qquad (2.1)$$

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 (all of which are taken over I here and in the following). We make all of the assumptions of [1] 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 real variable time, though extensions to multiple input multiple output cases are readily made. Also on physical grounds and in line with [1] we assume that u and y and sufficiently many, n, 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 [1] rests crucially upon the nonlinear Volterra functional $V_t(\cdot)$ belonging to a special reproducing kernel Hilbert space, called a Fock space and designated F_r 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, $h_k^{(i)} = \partial^i h_k / \partial t^i$, with respect to t of h_k exists everywhere on I for i=0,...,n-1 as a map from I into $L^2(I^k)$, where I^k is the k-dimensional cube of sides I, and the nth partial derivative with respect to t is a map from I into $L^2(I^{k+1})$.

b) There exists a real constant r such that

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

c) $h_k(t;t_1,\ldots t_k)$ is invariant to permutations (that is, symmetric) in the variables $t_1,\ldots, t_k$.

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 $V_t(\cdot)$ and $W_t(\cdot)$ of Fock space F_r as follows. Let h_k be the kernels for V and g_k be those for W, then

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

where the scalar product of any two (Lebesgue) square integrable functions g and h of k variables, $t_1, \ldots, t_k$, is given by

$$\langle g,h\rangle_{L^2(I^k)} = \int_{t_1}\int_{t_2}\cdots\int_{t_k} g(t_1,t_2,\ldots,t_k) h(t_1,t_2,\ldots,t_k) dt_1 dt_2 \ldots dt_k \tag{2.3b}$$

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

$$\langle V.(\cdot), W.(\cdot)\rangle_{B_n^2} = \sum_{i=0}^{n-1} a_i \int_t \langle V_t^{(i)}(\cdot), W_t^{(i)}(\cdot)\rangle_{F_r} dt \tag{2.4}$$

where the a_i are any chosen 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 a time t in the time interval I into the Fock space Volterra map $V_t(\cdot)$ which maps full input functions $u(\cdot)$ into output functions evaluating them at time t, that is into y(t).

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 $K(u,\cdot)=W_t(\cdot)$ for (2.3) we see by observing (2.1) that the kernels for W are

$$g_k = \frac{1}{r^k} u \otimes u \otimes \ldots \otimes u = \frac{1}{r^k} u \otimes^k u \tag{2.7}$$
$$\text{where } \otimes^k = k\text{-fold tensor product}$$

In other words

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

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 [1] 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,..., m, with these functions (along with n 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 $V_.(\cdot)$ such that

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

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

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

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

$$\|V_.(\cdot)\|^2_{B^2_n}=\sum_{i=0}^{n-1} a_i \int_{t\in I} \left\|V_t^{(i)}\right\|^2_{F_r} dt \qquad (2.10)$$

The problem is actually tackled by solving the equivalent problem

$$\min \left\|V_t^{(i)}\right\|_{F_r} \quad \text{over all } t\in I \text{ and all } V_t^{(i)}\in F_r \qquad (2.11a)$$

subject to

$$V_t^{(i)}(u_j(\cdot))=y_j^{(i)}(t) \quad \text{for } i=0,\dots,n,\ j=1,\dots,m \qquad (2.11b)$$

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

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

where for completeness we recall, see (2.3b), that the L^2 scalar product of functions of one variable is just the (Lebesgue) integral over the specified interval of the scalar product entries, that is

$$\langle u_i(\cdot),u_j(\cdot)\rangle_{L^2(I)}=\int_{t\in I} u_i(t)u_j(t)dt \qquad (2.12b)$$

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

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

we obtain a column m-vector of coefficients

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

to place in the best estimation $V_t^{\wedge}(\cdot)$ of $V_t(\cdot)$. The key and end result is that this best estimate is given by [1, Eq.(20)]:

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

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

3 A 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 $c(t)=G^{-1}y_{test}(t)$ 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(t) act as time-varying synaptic weights while the exponentials of the inputs preceding these weights act as neuron nonliniarities 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.

Example:

Letting $1(\cdot)$ denote the unit step function, consider a system for which $u_1(t)=e^{-t}1(t)$ gives $y_1(t)=0.5(1-e^{-2t})1(t)$ and $u_2(t)=0.5e^{-t/2}1(t)$ yields $y_2(t)=0.25(1-e^{-t})1(t)$ all defined for time in the unit interval, I=[0,1]. We choose m=2 and find and $u_2(t)=0.5e^{-t/2}1(t)$ yields $y_2(t)=0.25(1-e^{-t})1(t)$ all defined for time in the unit interval, I=[0,1]. We choose m=2 and find

$$\|u_1\|^2_{L^2(I)} = \int_0^1 e^{-2t}dt = (1-e^{-2})/2$$

$$\|u_2\|^2_{L^2(I)} = \int_0^1 0.25e^{-t}dt = (1-e^{-1})/4$$

$$\langle u_1,u_2\rangle_{L^2(I)} = \langle u_2,u_1\rangle_{L^2(I)} = \int_0^1 0.5e^{-1.5t}dt = (1-e^{-1.5})/3$$

from which G is calculated according to equation (2.12a), after choosing r=0.9 and writing to three decimal places though carrying the calculation to eight:

$$G=\begin{bmatrix}1.617 & 1.333\\ 1.333 & 1.192\end{bmatrix}, G^{-1}=\begin{bmatrix}7.997 & -8.950\\ -8.950 & 10.847\end{bmatrix}$$

In turn the synaptic weights $c(t)=G^{-1}y(t)$ are given as

$$c(t)=\begin{bmatrix}(1.761+2.238e^{-t}-3.999e^{-2t})1(t)\\ (-1.763-2.711e^{-t}+4.475e^{-2t})1(t)\end{bmatrix}$$

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

$$\exp(\frac{1}{r}\langle u_1,u\rangle_{L^2(I)}) = \exp(1.111\int_0^1 e^{-t}u(t)dt)$$

$$\exp(\frac{1}{r}\langle u_2,u\rangle_{L^2(I)}) = \exp(0.556\int_0^1 e^{-2t}u(t)dt)$$

We note that these terms, although nonlinear in the input, are independent of time, all of the time dependence having now been placed in c(t). For reference we state that the original system for which this neural network is an approximation is a squaring device followed by an integrator, both with unity gain.

4 Discussion

Because of the inherent importance of using neural networks for predicting the performance of nonlinear dynamical systems, here we have generalized the theory of artificial neural networks to incorporate that of functional input output pairing of nonlinear dynamical systems. The theory is based upon the theory of Volterra kernels but using a very important difference of viewpoint than one finds in much of the literature on Volterra series outside those related to reference [1]. The key idea is the reproducing kernel within the mathematical framework of Fock Hilbert space concepts, although the reader should not let the technical details of the mathematical spaces involved get in the way of the fundamental ideas. The use of the reproducing kernel allows the estimation of nonlinear systems to revert back to that of linear dynamical systems while still incorporating the nonlinearites for which the Volterra series is tailored. The theory was previously developed and expounded in Zyla and deFigueiredo [1] and adapted here to fit within the framework of neural network theory. However, we have only given the rudiments necessary to make the ideas available to the neural networks community while it remains to develop operational details as well as to coordinate the ideas with others, such as the OMNI neural network [3].

References

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

[2]. J. N. Holtzman, "Nonlinear system Theory: A Functional analysis Approach," Prentice-Hall, Englewood Cliffs, NJ, 1970.

[3]. R. J. P. deFigueiredo, "An Optimal Multilayer Neural Interpolating (OMNI) Net in a Generalized Fock Space Setting," Proceedings of the International Joint Conference on Neural Networks, Baltimore, MD, June 7-11, 1992, pp. I-111 - I-120.

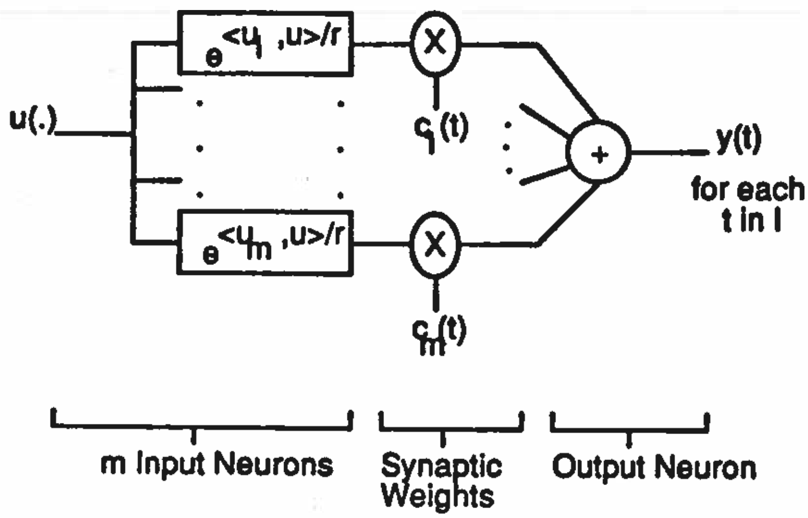

Figure 1
Functional Artificial Neural Network

ICARCV '94

THE THIRD INTERNATIONAL CONFERENCE ON AUTOMATION, ROBOTICS AND COMPUTER VISION

9-11 November 1994
Shangri-La, Singapore

PROCEEDINGS
(VOLUME 1 OF 3)

ORGANISED BY

School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore

The Institution of Engineers, Singapore