RECURRENT NEURAL NETWORKS

Emilio Del-Moral-Hernandez
University of Sao Paulo, Sao Paulo, Brazil.

Magno T. M. Silva
University of Sao Paulo, Sao Paulo, Brazil.

Keywords: Recurrent Neural Architectures, Artificial Neural Networks, Neurocomputing, Hopfield Recurrent Neural Networks, Attractor Networks, Basins of Attraction, Auto-associative Memories, Williams-Zipser Neurodynamics, Real Time Processing with Neural Networks.

Contents

Summary

Recurrent Neural Networks, or simply RNNs, have the particular characteristic of having loops in their structures of interconnected model neurons. That brings interesting properties to this type of neural networks, such as the embedding, the sensing and the production of time varying signals, as well as the concept of attractor networks. All of these aspects make RNNs an important subject of study. The initial sections of this chapter address the basics on Artificial Neural Networks and the classical Multi Layer Perceptron neural architecture – the MLP. This relatively simple and well known non-recurrent neural network was chosen for the introduction of the key concepts in artificial neurocomputing, such as the elements of a model neuron, the learning from empirical examples, and the use of several model neurons to build neural networks that represent complex multidimensional nonlinear phenomena and produce complex computing

functionalities. After that, the chapter addresses the specific aspects of Recurrent Neural Networks, discusses their relevant phenomenology and their importance in neurocomputing and modeling, and it briefly presents several classes of recurrent neural networks. Then, the three final sections of the chapter are dedicated to addressing with more detail two recurrent architectures: the Hopfield neural network and the Williams-Zipser neural network. Together with the discussion of concepts such as attractor networks, associative memory, and real time adaptive signal processing, these two architectures are presented and explained, and two related application examples are explored: storage of images and their recovery under noisy conditions, using attractor networks; and continuous learning with recurrent neural networks, in the context of time varying signals and echo cancellation.

1. Introduction: General Concepts in Artificial Neural Networks, Properties, Their Power and Their Relevance

Artificial Neural Networks became a solid and mature area in the last four decades, offering at the present moment an important set of resources in the Computational Intelligence scenario. They are useful for the modeling of functional structures of the nervous system, for the modeling of nonlinear adaptive systems in general, and for the biologically inspired neurocomputing, particularly important for the solution of nonlinear problems, pattern recognition, data-mining, as well as many other relevant areas in information processing (Haykin, 1999).

The basic concept in ANNs (Artificial Neural Networks) is to bring to the computing environment the mathematical description of the processes of knowledge learning, storage, and representation of information observed in the nervous system, as well as some of its mechanisms for information processing. This is accomplished, from one side, with emphasis to the description of what happens at the level of a single neuron and the synaptic interconnections between neurons, with the definition of a model neuron described by mathematical expressions. From another side, the emphasis is on the architecture level, and how these model neurons can be collectively grouped and trained so to perform particular tasks.

Many of the interesting properties observed by biology in the nervous system of living beings are present in such Artificial Neural Networks. These properties include the ability to deal with nonlinearity, mechanisms of learning, adaptation and multi-functionality according to context and according to the training data (also named training samples), generalization capability, the ability to deal with multiple input variables, which are integrated in the performance of a task, the ability to produce multiple output variables, and the robustness to noisy data and robustness to moderate degradation of elements of the artificial neural network itself. All these properties are inspired in biology, and they bring to artificial neural networks powerful properties as information processing and computing platforms.

This chapter focuses in the class of neural networks named Recurrent Neural Networks, which have the particular characteristic of having loops in its structure of interconnections between model neurons, which brings interesting properties such as the embedding, sensing and production of time varying signals, as well as the concept of attractor networks, detailed ahead in the following sections. The fact that biology has such type of recurrent structures is one of the reasons that inspired the creation of Artificial Neural Networks with recurrence. Another reason to explore such class of neural architecture is that we know from mathematical analysis, as well as from several fields in engineering and quantitative sciences, that structures with feedback (recursion) have different properties when compared to systems without recursion, properties that can be explored for enhancing the functionalities of ANNs, for the benefit of better modeling and better solving of engineering and information processing problems.

The chapter intends to be a useful material for the beginners in artificial neural networks and recursive artificial neural networks. At the same time, some of its sections bring challenging discussions for those already initiated in neural networks. The initial part of the chapter will discuss the basics on Artificial Neural Networks and will address briefly one of the most classical and well known neural networks, the Multi Layer Perceptron – MLP. In fact, the MLP has no recurrence at all, but it is for sure a good starting point for our discussion due to its simple structure, being a popular first step when getting contact with the area of artificial neural networks. After explaining the concepts of model neuron and MLP architectures, the chapter addresses structural aspects of Recurrent Neural Networks, their relevant phenomenology and potential exploration in neurocomputing and modeling, listing then several classes of recurrent architectures. Then it discusses in detail two recurrent architectures and illustrates related concepts and applications. The first one, the Hopfield architecture, is used to introduce and discuss several concepts such as attractors, state space landscape, and Attractor Networks. It deals with relatively simple calculations, being perfect for a beginner to start programming his/her own neural network. The second one, a Williams-Zipser architecture, for which we have more details in the mathematical formalization and in the related computer programming, is good for more advanced experiments, exploring time varying inputs and time varying outputs. Together with the exploration of concepts, architectures and illustrative applications, advanced topics on Recurrent Neural Networks are mentioned and references for extra reading are recommended.

2. Starting With the Basic Model Neuron and the Most Classical Non Recurrent Neural Network-The MLP

As it was mentioned in the introduction section, a basic concept in the building of neural networks is that of the "***model neurons***", also named "***artificial neurons***", which can be viewed as the building blocks that together compose the neural network architectures (Haykin, 1999). They usually have multiple inputs, representing the incoming stimuli arriving to a biological neuron through its multiple synaptic connections, and a single output, representing the rate of action potentials, i.e., the neurophysiologic electrical activity generated at the initial segment of the neuron axon. The relationship between the several inputs $x_i,\ i=1,2,\ldots,N$ and the output y in each neuron model that compose a classical neural architecture is given by

$$y=\varphi\left[\sum_{i=1}^{N} w_i\ x_i + w_{N+1}\ b\right], \qquad (1)$$

where N is the number of synaptic inputs, b represents the bias input which can be fixed at a constant value ($b=1$, for example), and $\varphi[\cdot]$ is usually a nonlinear function (hyperbolic tangent, for example). This function roughly represents the non linear relationship between the biological neuron incoming stimulation (integrated over all the synapses) and the rate of electrical pulses generated at the axon. The adaptable synaptic weights w_i, $i=1,2,\ldots,N+1$ provide the mechanism for learning: according to the desired storage of information, desired knowledge, and desired task for the ANN, the values of the several w_i are set, emulating thus the biological synaptic plasticity, a central element of biological adaptation, learning and information and knowledge storage in the nervous system. The neuron model is depicted in Figure 1.

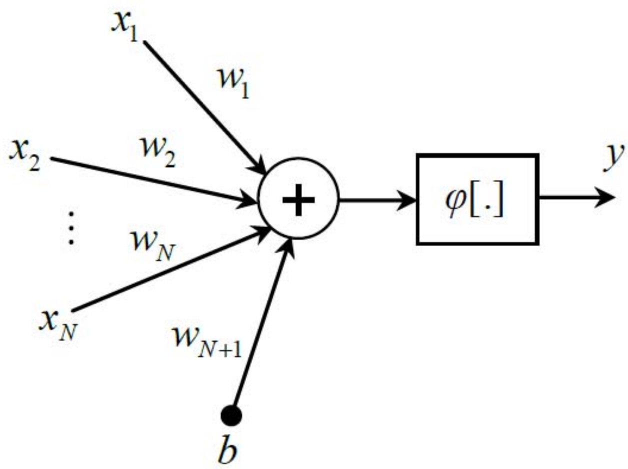
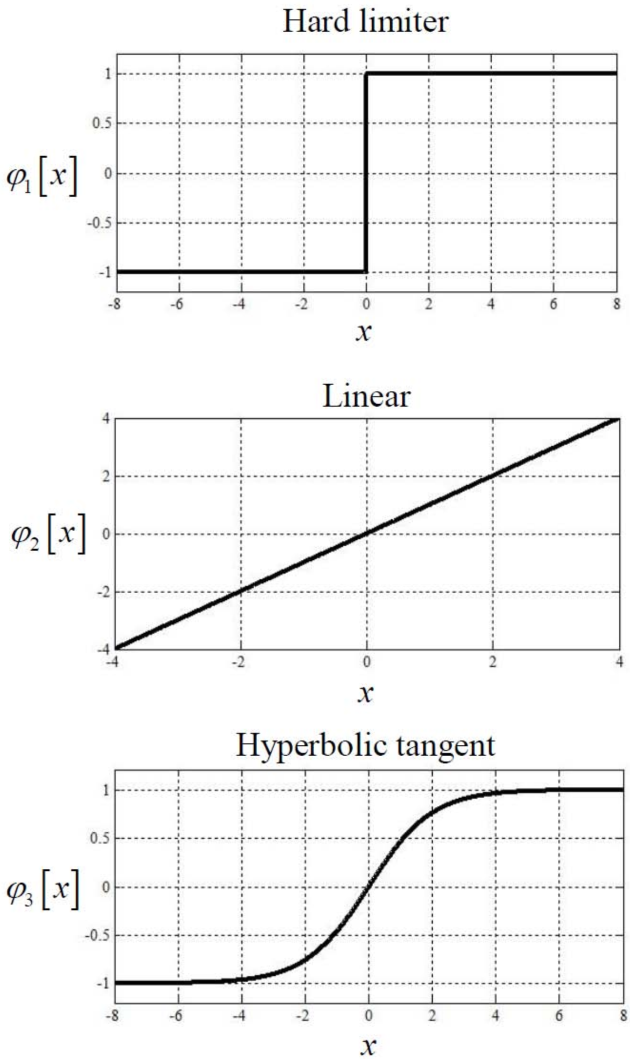


Figure 1. Neuron model used to compose a classical neural architecture, such as the popular MLP architecture, as well as many of the Recurrent Neural Networks addressed in this chapter. The elements of the model neuron are based on the main elements of a biological neuron: synaptic connections *w*, integrative soma, and nonlinear mapping between stimulus and level of axonal activity (Haykin, 1999).

It is imporant to notice that new model neurons are under intense and evolving research. These new models have more complex functionalities such as elements of neurodynamics (this is the case, for example, in the novel models so called spiking model neurons (Gerstner and Kistler, 2002)), nonlinear synaptic effects, and higher order composition of inputs (products among inputs). However, the simple and classical model neuron represented in Figure 1 is still the prevalent one in most of the artificial neural networks currently in use in a large number of different applications, and it is still being a basic building block in the composition of many novel neural architectures which are object of research.

The transfer function $\varphi[\cdot]$ (also known in the literature as *activation function*) in Eq. (1) can appear in different neural architectures with several variations, according to the properties needed for a particular strategy for representation of information, for storage of knowledge and for information processing, or even according to the need for mathematical convenience. For example, the fact that the hyperbolic tangent is smooth and differentiable is explored in classical algorithms employed in the learning of the

MLPs architectures (Haykin, 1999). In certain architectures, we can have more simplified versions of the transfer function $\varphi[\cdot]$, with respect to the standard hyperbolic tangent: we can have a linear function, or even a function mapping analog input values to only two binary values, also known as hard limiter (we will see this type of binary output used in Section 5 for the Hopfield architecture, where the two output values of each node are +1 or -1). Figure 2 shows the graphics and analytical expressions of three types of $\varphi[\cdot]$: hard limiter, linear, and hyperbolic tangent.



$$\varphi_1[x] = \begin{cases} -1, & x < 0 \\ +1, & x \geq 0 \end{cases}$$

$$\varphi_2[x] = x$$

$$\varphi_3[x] = \frac{1 - e^{-x}}{1 + e^{-x}}$$

Figure 2. Three types of activation function $\varphi[\cdot]$: graphs and analytical expressions.

2.1. The Multi Layer Perceptron - A Classical and Popular Non Recursive Architecture

The next step in the building of a neural architecture is nothing but the use of a number of model neurons, with the synaptic connections between them: the output of some nodes can feed one or more of the synaptic inputs of other nodes. With the replication of the model neuron needed for the formation of the network, Eq. (1) has to be adapted to reflect the existence of multiple neurons, each one with its particular output y_j, where j identifies the neuron.

The patterns of synaptic connections, the mechanism for the calculation of the w_{ij} values linking a generic pair of nodes i and j, and the form of interaction of the nodes with the environment, are all central elements that define what is named a ***neural architecture***. Many different neural architectures were conceived along the last four or five decades, what is attested by the vast literature available on neural network architectures. In this chapter, we will mention several of them, with emphasis to the recurrent ones, and we will illustrate different forms of interconnections as well as different forms of interaction of the network nodes with the environment, in terms of inputs received from it and outputs produced by the neural network. In this respect (inputs and outputs), for the moment we can say that, in the general case, a part of the model neurons composing an arbitrary artificial neural network architecture can have dedicated synaptic inputs being fed by external variables coming from the external environment (similar to what happens in the sensory inputs in the biological nervous system). Similarly, another part of the model neurons of the network can have their outputs acting on the environment (similar to what happens in the neuromuscular structures in the biological nervous system). We could also add something particularly important for some of the recursive neural networks addressed in this chapter, and specifically central for the Hopfield architecture discussed ahead: the fact that some artificial neural networks also consider as a possible channel for information input the prompting of the initial state of the nodes composing the network.

In the specific case of the MLP, which is the most classical neural architecture, and the one most employed in current practice, we have a very particular and organized way to define the connections between inputs and outputs of the model neurons that compose the whole architecture. As some of the readers might already know (those with some previous contact with neural networks), the organization in layers of an MLP means that the flow of information from one neuron to another happens in a unidirectional (feed-forward) way, as it is illustrated in Figure 3, where we represent an MLP with 4 layers, **A**, **B**, **C** and **D**, with a total of 9 (3+3+2+1) model neurons: each numbered node in the figure (1 though 9) represents a model neuron mathematically described by Eq. (1), and each arrow represents a synaptic connection. The 3 neurons of layer **A** produce outputs that only can feed the inputs of the nodes of layer **B**; by their turn, these nodes of layer **B** produce outputs that only can feed the inputs of the nodes of layer **C**, and so on. In this architecture organized by layers, we do not have any connection going from an output of a given node to any input of the other nodes in the same layer; we also do not have connections to inputs of nodes in previous layers, nor connections to layers ahead of the next layer. The outputs of layer **B**, for example, do not affect the inputs of nodes in layer **A**, **B** or even **D**, they only can affect the inputs of node in layer **C**. We have only a forward flow of information in the classical MLPs, and we even forbid the presence of forward connections to any forward layer different from the one immediately ahead.

Equation (1) and Figures 1 and 2 already presented the input-output relationship of a generic node that composes the MLP network. That equation, valid for a single neuron, is replicated for each node of the network, i.e., it is replicated 9 times in the particular MLP represented in Figure 3. Notice though that each of the 9 nodes can have different values for its synaptic weights (the *w's* in the generic equation), and each of them can have a distinct nonlinearity function $\varphi[\cdot]$.

It is important to state here clearly what is the expressive power of the MLP: it is proven that the MLP can do universal approximation with arbitrarily small error, for arbitrary functions of multiple variables, which makes it very powerful for nonlinear regression as well as for automatic pattern recognition (Cybenko, 1989). Notice that the structure represented in Figure 3 can implement a multidimensional mapping of variables, linking the input variables in vector $\mathbf{X}$ to the output variables in vector $\mathbf{Y}$: $\mathbf{Y} = \mathbf{F}(\mathbf{X})$.

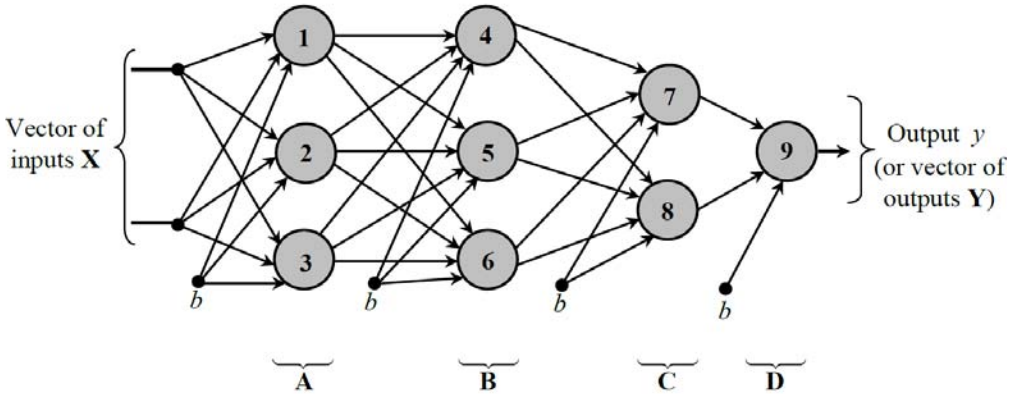


Figure 3. A Multi Layer Perceptron (MLP) architecture, with 9 neurons (numbered 1 to 9) arranged in 4 layers: A, B, C and D.

In addition, we also add here that the widely popular learning algorithm known as Error Back Propagation (EBP), for the definition of the values of the MLP's weights, is essentially based on the gradient descent numerical method for the minimization of the quadratic error of the neural network with respect to the targeted values for its outputs: repeated adaptations are made on the weights of the network (the w_{ij} values), moving in the opposite direction of the local gradient of the function $\sum_j (d_j - y_j)^2$, where d is the target training value (or desired value) for the network output, y is the output produced by the MLP under training, and the index j runs over all the samples of the empirical data available for that training.

3. Recurrent Neural Networks, In Artificial Neurocomputing and In Biology - Structures with Cyclic Paths in the Flow of Information

In order to characterize and understand the nature of Recurrent Neural Networks -or simply **RNNs**-, let us first discuss the elements related to the structure of interconnections that characterize the recurrent neural networks. After doing that, we can discuss, in the following sections, the phenomenology that arises in RNNs and how it can be explored in information storage and information processing.

As we could see in the previous section, in the classical MLP architecture, we have a feedforward structure, where information flows in one direction only. We can say that we have a strictly non recurrent neural network, because we do not have any output of a given node feeding any synaptic input of a node which is back in the path that produces information to that same given node. We do not have cyclic paths in the flow of information in an MLP. Different from the organized feedforward processing, in Recurrent Neural Networks (both in biology and in Artificial Neural Networks), we have one or more of these loops, breaking the unidirectional flow of information. In other words, in RNNs the output of at least one neuron is fed back, either through a synaptic input that belongs to the neuron itself (self feedback), or through feeding another neuron, whose output impacts the input of that neuron (either directly or through a chain of neurons). This is the formal definition for RNNs, i.e., having at least one loop in the flow of information, but in fact, many of the recurrent neural networks proposals that have gained attention for their effective functionalities, including the Hopfield RNN discussed in Section 5 and the Williams-Zipser RNN discussed in Section 7, present a large number of recursive connections, much beyond a single loop, producing a scenario of multidimensional feedback.

In biology, we observe that feedback structures are present in several parts of the nervous system and they play important roles in the processing of nervous information. The pattern and the degree of the feedback can vary significantly, as well as the presumed function of such feedback in the nervous system. We can say though that the extreme case of full connectivity among neurons, i.e., all neurons of a population are linked to all the other neurons, is not observed in biology. Nevertheless, when we go to artificial neural networks, we can have much more stereotyped loops, involving sometimes highly symmetric structures and full connectivity between all nodes.

MLPs and other feedforward networks used for processing of dynamical systems tend to capture the dynamics by including past inputs in the input vector. However, they might not be powerful enough to capture the dynamics of complex systems. In many cases, it is essential to employ feedback, i.e., to use RNNs (Haykin, 1999; Mandic and Chambers, 2001). Furthermore, as the feedforward MLP networks, some classes of RNNs may also satisfy the Kolmogorov universal approximation theorem (Mandic and Chambers, 2001). This makes RNNs suitable for universal approximation and prediction.

We should mention here some of the most important recurrent neural architectures, including the pioneer ones and the more recently proposed ones: the classic John Hopfield neural network for auto-association of images and binary strings in general (depicted in Figure 6), the extension architecture conceived by Bart Kosko for the bidirectional hetero-association of binary strings of different lengths, the Jordan Recurrent Networks, the Elman Recurrent Networks, the Williams-Zipser Recurrent Networks, the Echo-state Machines, the Liquid State Machines, as well as more complex architectures for the exploration of rich bifurcation and chaotic dynamics in neural oscillators, such as Aihara's, Adachi's, and RPEs (Recursive Processing Elements) architectures (Hopfield, 1982; Kosko, 1988; Williams and Zipser, 1989; Mandic and Chambers, 2001; Lukosevicius and Jaeger, 2009; Emilio Del-Moral-Hernandez, 2009). All these networks explore, in one way or another, some of the

-

-

-

Bibliography

[1] S. Haykin, (1999) Neural Networks, Prentice Hall, Upper Saddle River. [This is an excellent and comprehensive book on neural networks, with a good mix on concepts, theory and application. While being an excellent textbook, it also includes topics describing advanced subjects in neural networks.]

[2] W. Gerstner and W. M. Kistler, (2002) Spiking Neuron Models: Single Neurons, Populations, Plasticity, Cambridge University Press, Cambridge, UK. [A classic reference on spiking neurons and several models with elements of neurodynamics.]

[3] G. Cybenko, (1989) "Approximation by superpositions of a sigmoidal functions", Mathematics of Control, Signals, and Systems, vol. 2, pp. 303-314. [This paper presents a key result on the MLPs networks, by proving that they are universal approximators.]

[4] J. J. Hopfield, (1982) "Neural networks and physical systems with emergent collective computational abilities", Proceedings of the National Academy of Sciences of the USA, vol. 79 no. 8 pp. 2554-2558, April 1982. [A seminal pioneer paper, which inspired many other developments on recurrent neural networks operating with the paradigm of attractor networks.]

[5] B. Kosko, (1988) "Bidirectional Associative Memories", IEEE Transactions on Systems, Man, and Cybernetics, vol. 18, pp.49-60. [A proposal for hetero-association with recurrent neural networks, extending the auto-associative memory conceived by Hopfield]

[6] R. J. Williams and D. Zipser, (1989) "A learning algorithm for continually running fully recurrent neural networks", Neural Computation, vol. 1, pp. 270–280. [This paper proposes the real-time recurrent learning algorithm for adaptation of the weights of a fully connected RNN.]

[7] D. P. Mandic, J. A. Chambers, (2001) Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability, Wiley, West Sussex. [This book presents a tutorial on real–time recurrent neural networks and their application for prediction of time series.]

[8] M. Lukosevicius and H. Jaeger, (2009) "Reservoir computing approaches to recurrent neural network training", Computer Science Review, vol. 3 no.3, pp. 127-149. [An interesting review paper addressing Echo State Machines and Liquid State Machines, two important RNNs.]

[9] Emílio Del Moral Hernandez, (2009) "Chaotic Neural Networks", chapter in: Encyclopedia of Artificial Intelligence, Juan R. Rabuñal, Julian Dorado, Alejandro Pazos (book editors), Hershey, Pennsylvania: Idea Group Publishing, p. 275-281. [A book chapter on neural networks based on model neurons with properties of rich dynamics, including model neurons with local recurrence]

[10] R. J. McEliece, E. C. Posner, E. R. Rodemich, and S. S. Venkatesh, (1987) "The capacity of the Hopfield associative memory", IEEE Transactions on Information Theory, vol. IT-33, pp.461-482. [A pioneer paper addressing the statistical quantification of the storage capacity on Hopfield memories.]

[11] J. J. Hopfield, (1984) "Neurons with graded response have collective computational properties like those of two-sate neurons", in Proceedings of the National Academy of Sciences of the USA, vol. 81, pp. 3088-3092. [A paper addressing the continuous time and continuous valued version of the Hopfield architecture.]

[12] Emilio Del-Moral-Hernandez, (2007) "Recursive Nodes with Rich Dynamics as Modeling Tools for Cognitive Functions", chapter in: Neurodynamics of Higher Level Cognition and Consciousness, Leonid Perlovsky e Robert Kozma. (book editors), New York - Berlin - Heidelberg: Springer, p. 279-304. [A book chapter addressing Hopfield-like associative neural networks composed by model neurons with rich dynamics, local recurrence and bifurcating dynamics.]

[13] G. Kechriotis, E. Zervas, and E. S. Manolakos, (1994) "Using recurrent neural networks for adaptive communication channel equalization", IEEE Transactions on Neural Networks, vol. 5, pp. 267–278, Mar. 1994. [This paper proposes the use of the Williams-Zipser RNN adapted with the RTRL algorithm for adaptive equalization of linear and nonlinear channels.]

[14] M. T. M. Silva and M. Gerken, (2002) "A RNN-LC hybrid equalizer", in Proc. of EUSIPCO'2002, Sept. 2002, vol. 1, pp. 341–344. [This paper proposes a hybrid equalizer composed by a linear combiner and the Williams-Zipser RNN for equalization of communication channels.]

[15] V. H. Nascimento and M. T. M. Silva, (2014) E Adaptive filters. In: R. Chellappa, S. Theodoridis, editors: Academic Press Library in Signal Processing: Signal Processing Theory and Machine Learning (Vol. 1), Chennai, Academic Press, pp. 619-761, ISSN 978-0-12-396502-8., [This chapter provides an introduction on adaptive signal processing, covering basic principles through the most important recent developments.]

[16] S. Haykin, (2002) Adaptive Filter Theory, Prentice Hall, Upper Saddle River, 4th edition. [This is the most classical textbook on adaptive filtering. It examines both the mathematical theory behind various linear adaptive filters and the elements of supervised Multi Layer Perceptrons.]

[17] A. H. Sayed, (2008) Adaptive Filters, John Wiley & Sons, NJ. [This book enables readers to gain a gradual and solid introduction to the adaptive filters, its applications to a variety of topical problems, existing limitations, and extensions of current theories.]

[18] A. Borys, (2001) Nonlinear Aspects of Telecommunications: Discrete Volterra Series and Nonlinear Echo Cancellation, CRC Press, Inc., [This book contains the fundamentals of the discrete Volterra series and presents some results on acoustic echo cancellation.]

[19] A. Ben Rabaa and R. Tourki, (1998) "Acoustic echo cancellation based on a recurrent neural network and a fast affine projection algorithm", in Proceedings of the 24th Annual Conference of the IEEE, vol. 3, pp. 1754–1757. [This paper proposes a cascade of a recurrent neural network with the fast affine projection algorithm for acoustic echo cancelation.]

[20] L. A. Azpicueta-Ruiz, M. Zeller, A. R. Figueiras-Vidal, J. Arenas-Garcia, and W. Kellermann, (2011) "Adaptive combination of Volterra kernels and its application to nonlinear acoustic echo cancellation", IEEE Trans. Acoust., Speech, Signal Process., vol. 19, pp. 97–110, Jan. 2011. [This paper proposes a combination of adaptive Volterra filters as the most versatile nonlinear models with memory for acoustic echo cancelation.]

[21] L. A. Azpicueta-Ruiz, M. Zeller, A. R. Figueiras-Vidal,W. Kellermann, and J. Arenas-García, (2013) "Enhanced adaptive Volterra filtering by automatic attenuation of memory regions and its application to

acoustic echo cancellation", IEEE Trans. Signal Process., vol. 61, no. 11, pp. 2745–2750. [This paper presents a scheme for nonlinear acoustic echo cancellation based on adaptive Volterra Filters with linear and quadratic kernels.]

Biographical Sketches

Emilio Del-Moral-Hernandez received his BSc and MSc degrees in electrical engineering from the Polytechnic School of the University of Sao Paulo, Brazil, and his PhD degree in electrical engineering from the University of Pennsylvania, Philadelphia. He is currently with the Department of Electronic Systems Engineering, University of Sao Paulo, where he is Associate Professor and Head of the Group of Computational Intelligence, Modeling and Electronic Neurocomputing. His research activities include the fields of pattern recognition, neuro-like computation and biologically inspired information processing, application of computational intelligence techniques to several areas, complex systems and nonlinear dynamics, modeling of nonlinear phenomena, data mining, the development of novel neurocomputing architectures that explore bifurcation and chaos, pulsed model neurons, and electronic implementation of neural architectures. He is member of the INNS (International Neural Network Society) and the IEEE-CIS (IEEE Computational Intelligence Society), having served in several of its committees.

Magno T. M. Silva received the B.S., M.S., and Ph.D. degrees, all in electrical engineering, from Polytechnic School, University of São Paulo, São Paulo, Brazil, in 1999, 2001, and 2005, respectively. From February 2005 to July 2006, he was an Assistant Professor at Mackenzie Presbyterian University, São Paulo, Brazil. Since August 2006, he has been with the Department of Electronic Systems Engineering at Polytechnic School, University of São Paulo, where he is currently an Associate Professor. From January to July 2012, he worked as a Postdoctoral Researcher at Universidad Carlos III de Madrid, Leganés, Spain. His research interests include linear and nonlinear adaptive filtering.