Universidade Estadual de Maringá
Departamento de Estatística
Programa de Pós-graduação em Bioestatística

Matheus Henrique Cecilio Leme

Redes Neurais Wavelets Recorrentes para Previsão de Séries Temporais

Maringá - PR
2021

Abstract

Time series analysis allow to explain and predict a infinity of phenomena that occurs every day. Given its importance, this study discusses Recurrent Wavelet Neural Networks that are formed by the union of Recurrent Artificial Neural Networks and Wavelet Analysis theories. Hence, these two theories are presented throughout this work, whose purpose is to substantiate the proposed approach. In this dissertation a model for a new structure of Recurrent Neural Wavelet Network was proposed and three time series were used to evaluate the performance of the adjustments and predictions of the proposed model: AirPassengers, data on the distance of the asteroid Apophis, and bronchiolitis in the state of Paraná-BR (2000-2019). For this evaluation, the mean squared errors, root mean squared errors, mean absolute errors, and coefficients of determination R^2 of the training and testing phases of the network, were computed. The proposed network was developed and implemented in R programming language and will be incorporated into the WNN package of the same software.

Keywords: Artificial Neural Network, Recurrent Wavelet Neural Network, Time Series, Wavelet Analysis, Forecasting.

Resumo

A análise de séries temporais permite explicar e prever uma infinidade de fenômenos que ocorrem todos os dias. Dada a sua importância, este estudo discute Redes Neurais Wavelet Recorrentes que são formadas pela união das teorias de Redes Neurais Artificiais Recorrentes e Análise Wavelet. Assim, essas duas teorias são apresentadas ao longo deste trabalho, cujo objetivo é fundamentar a abordagem proposta. Nesta dissertação foi proposto um modelo para uma nova estrutura de Rede Wavelet Neural Recorrente e três séries temporais foram utilizadas para avaliar o desempenho dos ajustes e previsões do modelo proposto: AirPassengers, dados sobre a distância do asteróide Apophis e bronquiolite em Estado do Paraná-BR (2000-2019). Para esta avaliação, foram calculados os erros quadráticos médios, erros quadráticos médios de raiz, erros médios absolutos e coeficientes de determinação R^2 das fases de treinamento e teste da rede. A rede proposta foi desenvolvida e implementada em linguagem de programação R e será incorporada ao pacote WNN do mesmo software.

Palavras-chave: Redes Neurais Artificiais, Redes Neurais Wavelets Recorrentes, Séries Temporais, Análise Wavelet, Previsão.

List of Figures

List of Tables

List of abbreviations and acronyms

AF	Activation Functions
ANN	Artificial Neural Networks
AN	Artificial Neurons
BLRNN	Bilinear Recurrent Neural Network
DRWNN	Diagonal Recurrent Wavelet Neural Network
DWT	Discrete Wavelet Transform
DWT-SRWNN	Discrete Wavelet Transform - Stochastic Recurrent Wavelet Neural Network
FRWNN	Fully Connected Wavelet Neural Network
LSTM	Long-Short Term Memory
LSTM-RNN	Long-Short Term Memory - Recurrent Neural Network
MCRNN	Multi-Context Recurrent Neural Network
RNN	Recurrent Neural Network
SRN	Single Recurring Network
SRNN	Stochastic Recurrent Neural Network
TS	Time Series
WA	Wavelet Analysis
WNN	Wavelet Neural Network

WT	Wavelet Transform

Contents

Introduction

A Time Series (TS) is a set of sequential observations of a phenomenon over time, which usually have identical spacing in time. The analysis TS are of great importance in science, because they can translate real-world information into mathematical terms, allowing for actions biased by these studies, contributing to society. The goals about TS are to describe, model, explain or model and forecast data from a stochastic process (MORETTIN, 1981; CHATFIELD, 2000). By studying the behavior of a time series, it is possible to infer about basic characteristics of the stochastic process responsible for generating these observations.

In TS, besides de pontual estimation, the confidence interval (CI) is very important for predictions, because it makes clear how much is expected to be right and wrong through specific estimates. In addition, IC presentation adds a lot of information and utility, which generally are not provided by most current methodologies in TS. (MEDEIROS, 2018) in her work, estimates the uncertainty for the evolutionary mean in curves or TS using wavelet-based bootstrap methods.

Wavelet Analysis (WA) emerged and filled some gaps in classical models that are not always effective to deal with all problems around TS. WA has been show to be a very successful methodology in the literature because of its advantages to these models, especially regarding the adaptability to erratic fluctuations in the signal(PICARD; TRIBOULEY, 2000). Mentions (CAZELLES et al., 2008), that WA is especially relevant for the analysis of non-stationary systems, thus it is a suitable approach for many TS.

Classic approaches inevitably require feeling on the part of those who handle them, and in some cases, deep knowledge is needed to deal with TS. Artificial neural networks (ANN) are a great approach to these problems, since, the application of a neural network does not require *a priori* knowledge of the underlying process; one may not recognize all the complex relationships that exist between the various aspects of the process under investigation; neither restrictions nor an *a priori* solution structure is necessarily assumed or strictly applied in the

development of the neural network (RAMAN; SUNILKUMAR, 1995).

ANN are considered as a simplified model of the biological neural network structure, which consists of interconnected processing units (YEGNANARAYANA, 2009). Through interconnections, the network performs learning iterations between the provided input and output, without any external intervention. This makes it more accessible and contributes to the performance of several works, given its agility and independence.

In the literature, there are investigations of neural networks with WA to work with TS of different natures (ESEN et al., 2009; ADAMOWSKI; CHAN, 2011; SUBASI; YILMAZ; OZCALIK, 2006). The combination of the WA and NN approaches has shown good results in TS, and also proves to be promising and a facilitator for the analysis of TS by individuals from other areas. However, even though it is easier to use an NN to model or predict a TS without outside interference, the difficulty of implementation or availability of these networks arises.

Thus, motivated by these considerations, this research aims to build a Recurrent Wavelet Neural Network (RWNN) recurring able to include exogenous variables to study, model and make predictions of TS. To evaluate the performance of the proposed network, Airpassengers series, Apophis and Bronchiolitis data will be used.

As specific objectives, we are going to build, implement, and evaluate different structures of the proposed network for TS. In addition, it will be presented the confidence intervals using bootstrap methods for TS for both RWNN learning and forecasting estimates.

This dissertation is divided as follows. Chapter 1 discusses ANN that support the entire research. Chapter 2 presents a brief introduction about Wavelet Analysis to be used in combination with TS, which is also the pillar of this work. A literature review of Wavelet Recurrent Neural Networks, pointing out some advantages and limitations, is made in Chapter 3. Chapter 4 is used to present the entire methodology underlying this research. The results of the performance of the deployed network are mentioned in Chapter 5. In Chapter 6, there is an overview of the dissertation and discussions for future work.

Chapter 1

Artificial Neural Networks

The term Biomimicry emerged in literature in 1997 with the author Janime M (BENYUS, 1997), but its meaning has been present for a long time in humanity. Biomimicry is responsible for the process that copies nature's system and has been used in several areas. This is because many times, nature can offer solutions to unresolved problems. It is possible to cite simple examples, like velcro invented by George de Mestral (DE, 1955) inspired by the vegetation, as well as the new leds screens that are more efficient, from the eye structure of mosquitoes.

Thus, the ANN mimic the way the human brain performs a task. Therefore, ANN are formed by structure similar to biological neurons, called Artificial Neurons (AN). ANs, similar to biological neurons, when interconnected form networks with capacity to process, store, modify, and transmit information (HAYKIN, 2001).

The ANN were developed as generalization of mathematical models of the human nervous system (ABRAHAM, 2005). For that feat, it is necessary to convert all the biological process into mathematics. Of simple mode, ANN receive input information from the external environment, performs processing of this data and produces a response to the output. For example, an ANN can be built to receive (input) images of numbers made by hand, process that information and answer (output) which digits was contained in the image. At the same time, it is possible identify this same process in humans, because when looking at the image of a digit, this stimulus propagates through synapses and the brain converts them, getting a conclusion. With these intuitive ideas, the theory of ANN can be introduced.

1.1 Neural Networks Theory

ANN is a data-driven, nonlinear, self-adaptive, and nonparametric statistical method. In modeling, ANN demonstrates to be a useful tool, especially when the data relationship is not known (ANJOY; PAUL, 2019). Because of these characteristics, neural networks are options for tasks that involve binary data or even symbols, as long as they are properly coded (WANG, 2003).

In works like that of (ZHANG; PATUWO; HU, 1998) and (CRONE; NIKOLOPOULOS; HIBON, 2005) it is possible to observe that ANN turn out to be prominent. ANN can be used in a range of data, even when there is no specific understanding of the variables that compose it. ANN are able, through examples and training, to learn the relationships between the data, regardless of the difficulty of the obstacles. Moreover, ANN are highly refined to predict, because they understand the functioning of a sample of data and extend to the population. Furthermore, they are able to generalize nonlinear patterns. Figures 1.1.1 and 1.1.2 represent neural network architectures with single layer and multilayers, respectively.

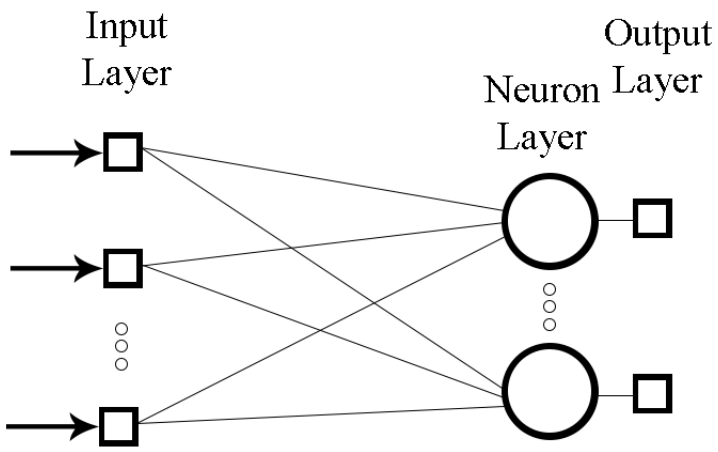

Figure 1.1.1 – Architecture of a Neural Network with a Single Layer

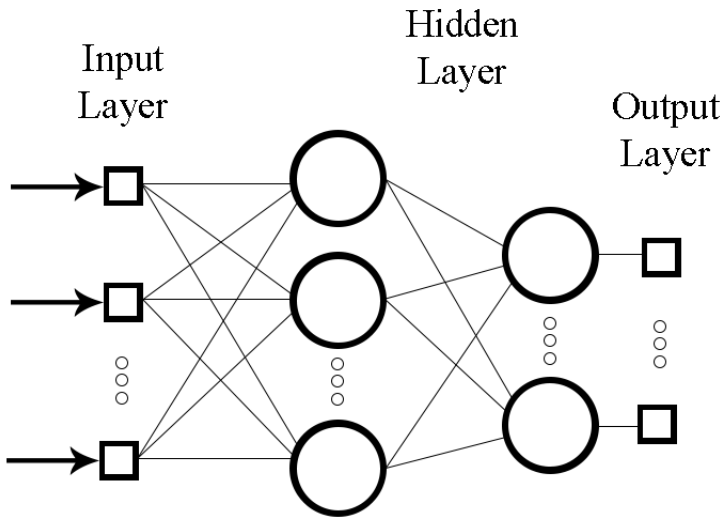

Figure 1.1.2 – Architecture of a Neural Network of Multilayers

Single-layer networks are characterized by a structure composed of an input layer, a hidden layer and an output layer. The hidden layer contains a single set of neurons that process the information received from the input layer and send it to the output layer. Multi-layer networks differ from the structure of single-layer networks because in their hidden layer there are more groups of neurons processing information. In other words, the information that enters the network through the input layer is transmitted to a group of neurons in the hidden layer, these transmit the processed information to other groups of neurons before sending it to the output layer.

Figure 1.1.3 illustrates the components of a single-layer network containing an artificial neuron operating in the hidden layer and the flow of operation.

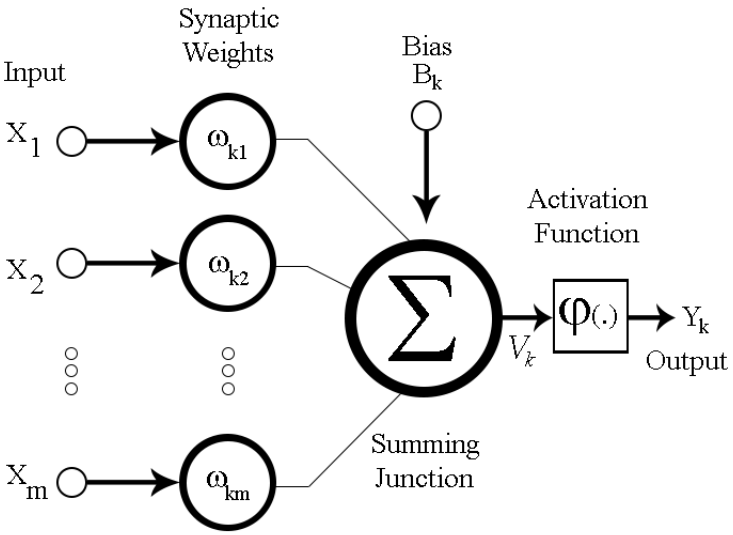

Figure 1.1.3 – Artificial Neural Network

Analyzing the figures 1.1.1, 1.1.2 and 1.1.3, it can be seen that there is a sequence of procedures. In RNN, the input layer is responsible for inputting the original information denoted by $x_1, x_2, \cdots, x_m$ and weighting it with the Synaptic Weights (SW)$\omega_{k1}, \omega_{k2}, \cdots, \omega_{km}$.Thus, an importance bias is attributed to each of this information. A Bias (B_k) operator can be included as an additional input to aid in tuning the enable function, so $B_k = x_0 = 1$. In sequence, the information is grouped by ν_k and sent to the activation function (φ) to designate the completion of the process, that is, submit a response using the initial information.

$$\nu_k = \sum_{i=0}^{m} x_i * \omega_{ki} \tag{1.1.1}$$

In a single-layer network containing k neurons, the nth output (response) of the kth neuron is defined by

$$y_k = \varphi(\nu_k) \tag{1.1.2}$$

The output $y_k(n)$ of the k-th neuron in relation to time n is defined by:

$$y_k(n) = \varphi_k(\nu_k(n)) \tag{1.1.3}$$

where ν_k can be called an activation potential. Equation 1.1.3 represents the analogy of converting biological factors into mathematical factors.

The Activation Functions (AF) of artificial neurons have being studied for a long time in the literature, and among the options it is possible to present the main functions used in neural networks today. The most known are, Logistic or Sigmoid Function, Hyperbolic Function, Step Function (or Heaviside Function) and Ramp Function, presented in the Figure 1.1.4.

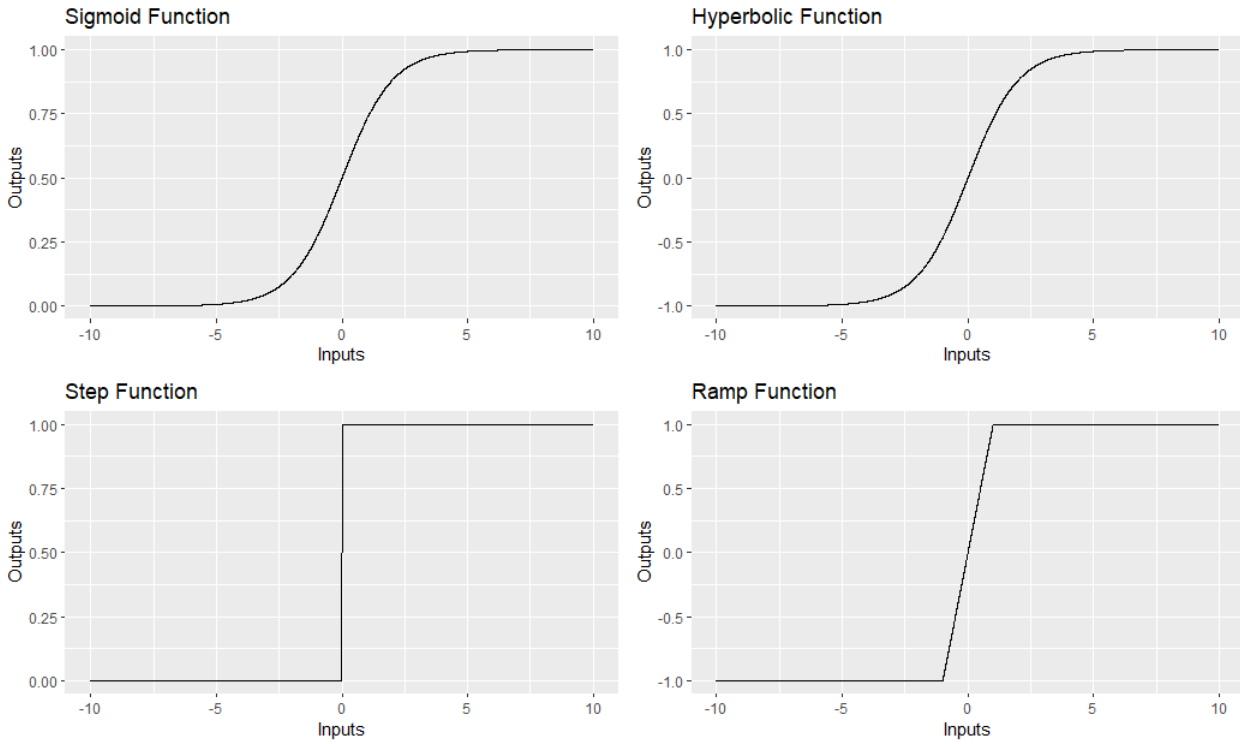

Figure 1.1.4 – Some activation functions

The choice between these functions depend on the purpose the neural network is built. If it is desirable for the network output values to be between $[0, 1]$, the Sigmoid function is recommended. However, if it is, preferable to be between $[-1, 1]$, the Hyperbolic Function is designated. The Step Function, in its turn, it is usually selected when the network output must provide a positive or negative response (yes or no). A special class of AF will be explored in the chapter 2.

The equations 1.1.4, 1.1.5, 1.1.6, and 1.1.7, present the mentioned AF.

$$\varphi(x|\beta) = \frac{1}{1 + e^{-(\beta * x)}} \tag{1.1.4}$$

$$\varphi(x|\beta) = \frac{1 - e^{(-\beta * x)}}{1 + e^{(-\beta * x)}} \tag{1.1.5}$$

$$\varphi(x) = \left\{ \begin{array}{c} 1, \text{if } x \geq 0 \\ 0, \text{if } x < 0 \end{array} \right\} \tag{1.1.6}$$

$$\varphi(x) = \left\{ \begin{array}{c} a, \text{if } x > a \\ x, \text{if } -a \leq x \leq a \\ -a, \text{if } x < -a \end{array} \right\} \tag{1.1.7}$$

In equations 1.1.4 and 1.1.5, it is clear the presence of β. The usefulness of this constant is to determine the slope of the curves generated from the AF. As beta increases, there is sharp inclination similar to the Step Function, on the other hand, the decrease in beta, causes a smoothing of the curve, similar to the Ramp Function, illustrated in Figure 1.1.5.

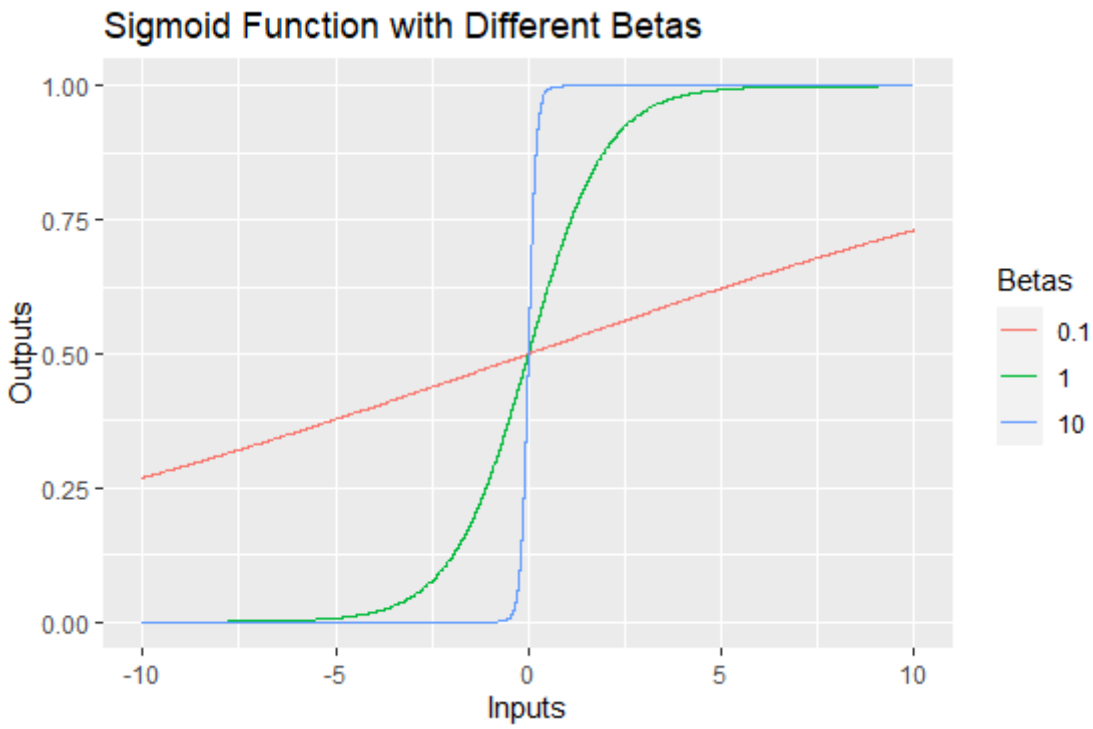

Figure 1.1.5 – Comparison of the Betas in Sigmoid Function

Neural networks are malleable mechanisms that can present structures with different configurations. In this sense, the literature is composed of many works that present different proposals regarding applications, architecture, AF or comparisons.

The architecture of a neural network is defined according to the scheme or composition of the layers present in the network and the way they are connected. Thus, changes in these schemes give rise to a distinct architecture, which has distinct characteristics and applications [5]. In ANN, there is a general division between recurring and non-recurring structures. The object of study of this work will be the Recurrent Neural Networks (RNN).

1.1.1 Recurrent Neural Network

Unlike the schemes shown in the figures 1.1.1 and 1.1.2, the RNN contains in its structure a mechanism that enables active information network in an instant t are not eliminated once the network reaches a conclusion. But that they are used to help the network in the next information. Naturally, there are some changes in the structure of the network, evidenced by the Figures 1.1.6 and 1.1.7.

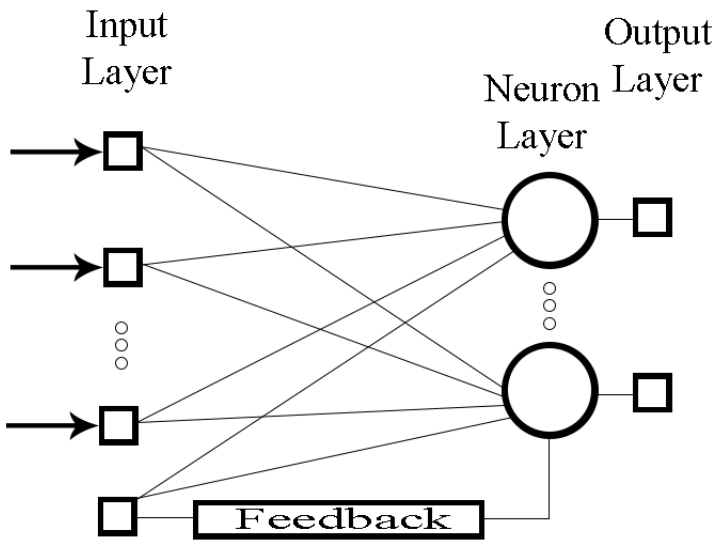

Figure 1.1.6 – Architecture of Recurrent Neural Network- single

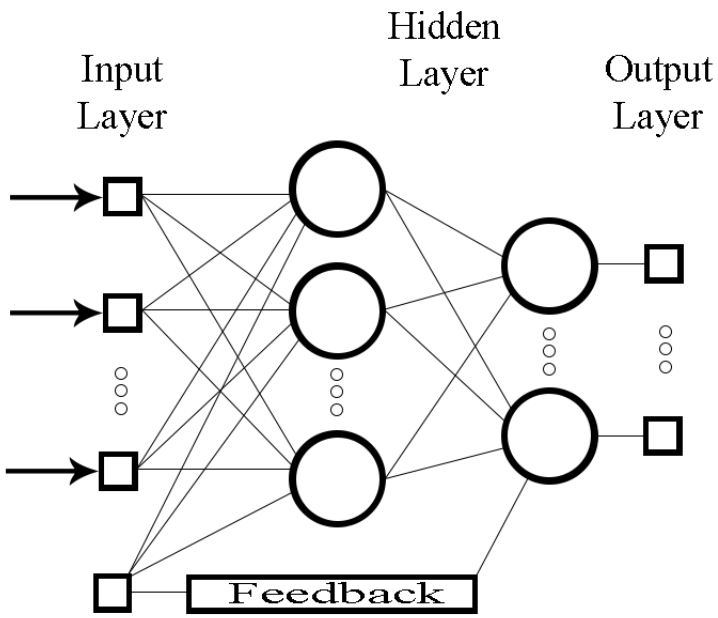

Figure 1.1.7 – Architecture of Recurrent Neural Network - multilayer

RNN is a neural network class that uses information produced in previous iterations to collaborate with future results. Thus, an RNN works through the inputs that are provided and also through the feedback produced by itself (KOLEN, 1994). RNN excels in learning time-varying sequences or patterns (MEDSKER; JAIN, 1999), so it is a great option to be used in TS, since time data are time variants.

Self-feedback networks belong to the RNN class, as they make the outputs of each neuron return to itself, as illustrated in the Figure 1.1.8.

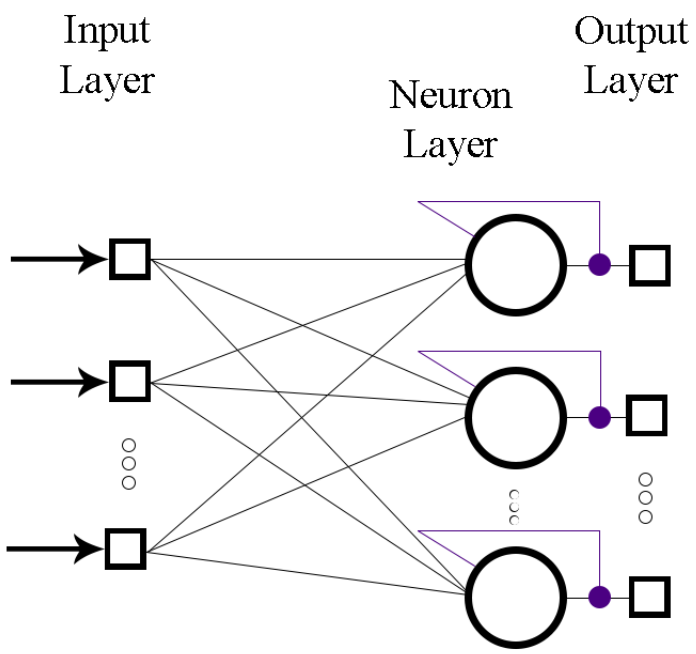

Figure 1.1.8 – Self-feedback network

The structure alternatives for these networks vary according to the types of inputs and their purpose, but the Time-Delay Neural Network (TDNN) has a structure that stands out for its simplicity and the way it incorporates information over time to influence in the present results. Thus, to estimate from $x(t)$ the results assumed in $x(t-1), x(t-2), \cdots, x(t-n)$ are incorporated in the network (SOARES, 2008). This process of feedback of information in past instants provides greater memory capacity in internal representations (MEDSKER; JAIN, 1999), as well as better modeling of temporal data, as it stores the values assumed by the variable over time in an indirect way.

1.1.2 Elman Network

The psycholinguist Jeffrey Elman in 1990 (ELMAN, 1990), proposed a single recurring network (SRN) containing an extra layer attached to the network architecture, called the Context Layer. Its function is to store temporal information from the hidden layer, and to reintegrate it to neuron inputs to contribute to future information processing.

Figures 1.1.9 and 1.1.10 present the scheme presented by Elman (1990) and a structure built to represent this scheme, respectively.

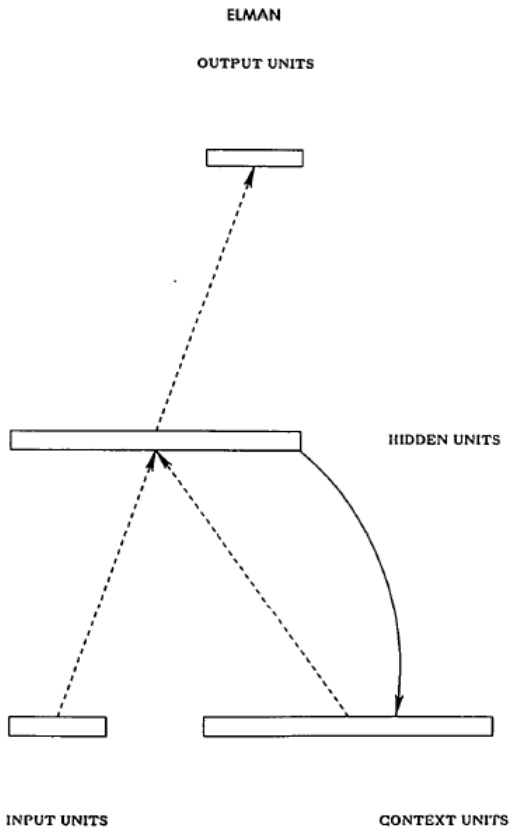

Figure 1.1.9 – Elman Network (ELMAN, 1990)

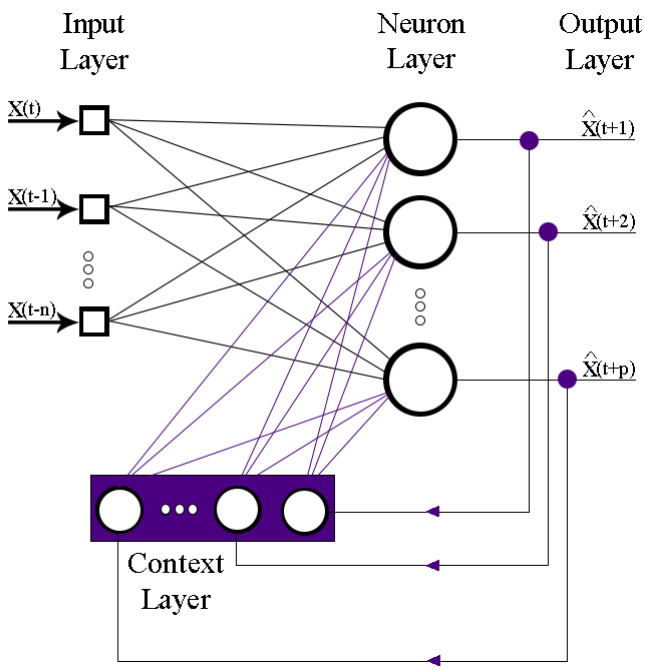

Figure 1.1.10 – Representation of Elman Network

The nomenclature "units" is used to represent layers in Elman's scheme. The Context Units provide dynamic memory to the network, that is, they are able to encode all the input information of the network, since the beginning of the represented sequence (MEDSKER; JAIN, 1999). Furthermore, Elman lattices are unique in that they can store information from past instants and aggregate into subsequent lattice predictions.

As for the influence of past information, it is understandable that more recent information has more influence, when compared to information from less recent moments. In fact, this because in this specific network, its structure provides a memory of great depth and little resolution, which means that the context layer keeps an exponentially decreasing record of the previous value of neurons outputs (PASQUOTTO, 2010). Or yet, Elman Networks do not contain restrictions on the number of memory lags and recent information loses resolution while distancing itself from the past (MEDSKER; JAIN, 1999).

In view of the architecture of Elman's network, shown in Figure 1.1.10, a single layer of neurons is seen. However, nothing is known about the amount of neurons contained in this layer. (WANG, 2003) states that there are divergences to conclude an expressive number of neurons, as well as the number of neuron layers in other networks. Some authors suggest that the number of hidden neurons be given per $\sqrt{(N * M)}$, where N and M represent the numbers of inputs and outputs, respectively. Others point out that this amount must be seventy-five percent of N. The truth is that it can be difficult to determine an exact value of neurons for a network, however that does not mean to say that there will be a bad network.

Neural Networks are structure capable of learning. But, for that feat, they need to carry out learning steps, which will be covered in Subsection 4. Intuitively, it is expected that change in the structure, further, the number of neurons affects the results and accuracy of an ANN. Therefore, the lack of consensus among the authors on the number of neurons, can be properly tested in practice. In other words, it is possible to perform test and check when Neural Network is optimized for certain application.

1.1.3 Neural Network Training

NN are attractive for having a strong learning capacity. Some authors such as (NILSSON, 1965) and (FU; MENDEL, 1970) contributed to disseminate works that were based on the process of learning networks.

Returning to the concepts of Biomimicry, it is interesting to reflect on the way humans learn, because in this way it becomes easier and more natural to understand the way of learning NN. With this in mind, one of the forms of learning is the student-teacher relationship, that is, when one person teaches another. In addition, observation is also a learning factor that brings

meaning to the individual, e.g. early in life a child observes horses and dogs, and through observations learns to distinguish them even if they both have similar characteristics, such as four legs, a tail, two ears among others. Trial and error is also part of the maturation and learning processes, e.g. consecutive attempts to take the first steps while being a baby.

In view of this, learning in the context of Neural Networks can be understood as a process of adaptation of free parameters resulting from environmental stimuli, where the learning classification is determined by the way the parameters are modified. A sequence of steps for learning Neural Networks can be formulated by the ideas mentioned, a certain stimulus from the environment incites the Neural Network, which makes changes in its free parameters in response to these stimuli, and subsequent to internal changes in its structure, neural network generates new conclusions for the environment arising from the stimuli (HAYKIN, 2001).

Below, a mathematical mechanism that reproduces the way humans learn, called Learning Algorithm (LA), will be presented. The LA are used to train the network through successive iterations that adjust the parameters (or Synaptic Weights) existing in the NN. Some LA are currently available, distinguished by the way in which the SW of neurons are adjusted.

Within the scope of LA (or training algorithms) there are so-called supervised and unsupervised learning. In few words, the difference between the two lies in the presence and absence of assertiveness indicators. At the end of each iteration where a response to the stimuli is generated, in training with supervision (Supervised Learning), the expected response is compared with the generated one, identifying when the network is obtaining correct or close to expected conclusions. From this, SW are adjusted to improve future predictions. On the other hand, networks with Unsupervised Learning do not have this validation step.

In particular, one of the approaches pertaining to supervised learning is error correction learning. The idea of this learning is to minimize prediction errors the cost function. In practice, the algorithm works as shown in Figure 1.1.11.

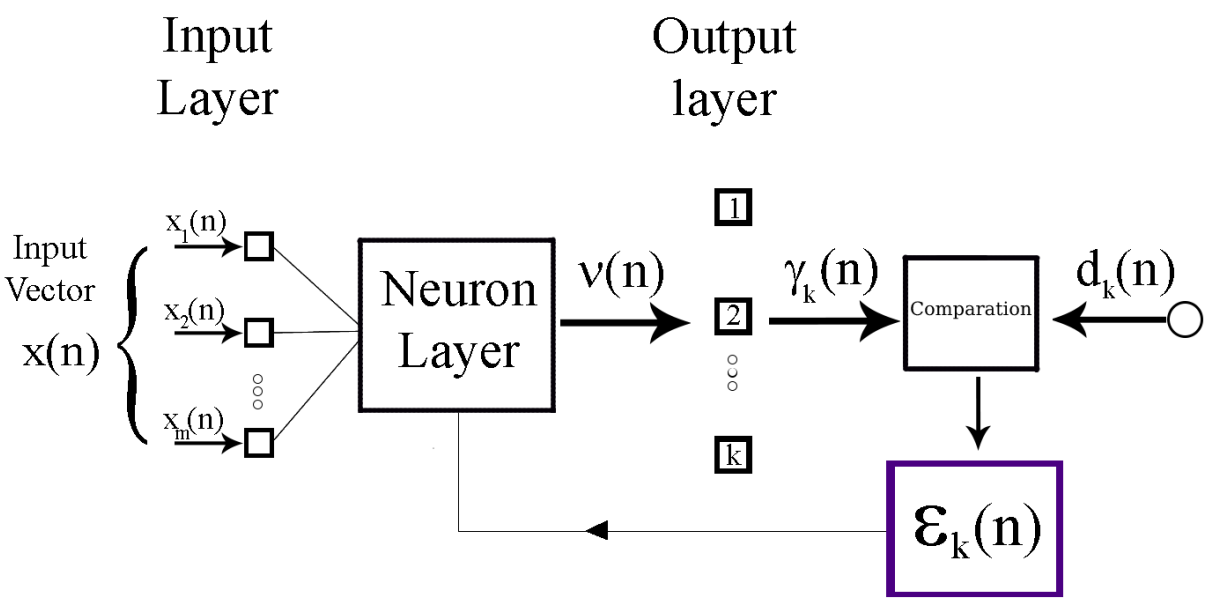

Figure 1.1.11 – learning process of a supervised network

A vector of information (stimuli) $x(n)$ is introduced into the input layer, where $x_m(n)$ represents the m information of $x(n)$. Information from the input layer is weighted and sent to the hidden layer, where are the neurons. The processing of this information is sent to the output layer by weighting the vector $\nu(n)$. Thus, the output layer generates a response to that stimulus.

Then, it is verified how good the network response was in view of the information provided in the input layer. For this, $\gamma_k(n)$ and $d_k(n)$ are compared, where $\gamma_k(n)$ represents the network response and $d_k(n)$ the desired response, generating an error ($\epsilon_k(n)$). Normally, the first iteration of the network is expected to generate an error, that is $\epsilon_k(n)$ neq 0, because initially the network is composed of random SW

$$\epsilon_k(n) = d_k(n) - \gamma_k(n).$$

The learning of the network continues through the SW adjustments, at each iteration of the network, $\epsilon_k(n)$ is generated and then the SW are adjusted to be tested in the next iteration. In general, at the beginning the SW randomly receive values between 0 and 1, which are modified at each iteration.

The cost function defined by equation 1.1.8, quantifies the errors made in all n iterations

performed in the learning . In practice $\epsilon_k(n)$ hardly becomes absolutely zero, but if the SW change at each iteration is not significant (too small), it means that the system is stable and the learning training ends (HAYKIN, 2001)

$$\xi(n) = \frac{1}{2} * \epsilon_k^2(n). \tag{1.1.8}$$

So far, the need to update the SW to generate satisfactory answers is known, but the way to do this has not yet been presented. Widrow and Hoff (1960) created the Delta rule to minimize the cost function, making adjustments in each SW. The rule is defined by the equation 1.1.9,

$$\Delta\omega_{kj}(n) = \eta * \epsilon_k(n) * x_j(n), \tag{1.1.9}$$

where $\Delta\omega_{kj}(n)$ represents the SW adjustment in time step n, $\omega_{kj}(n)$ determines the value of the Synaptic Weight (ω_{kj}) of neuron k which is excited by an element $x_j(n)$ belonging to the input vector ($x(n)$) and η represents the learning rate, being a positive constant. From this, the SW update is defined by the equation 1.1.10,

$$w_{kj}(n+1) = w_{kj}(n) + \Delta\omega_{kj}(n). \tag{1.1.10}$$

Chapter 2

Wavelets

The Wavelet Transform is historically presented as the generalization of Haar's transform functions, which have been used since 1910 (STANKOVI; BOGDAN, 2003). Throughout the century, some works about wavelets emerged, however, only in the mid-nineties its use was widely diffused and accepted by the scientific academy. Some works such as Daubechies (1990), Mallat (1989) e Morlet et al. (1989) allied to the growing computational technology, contributed to the expansion of Wavelet studies. From this, the effort of the scientific community to investigate the potential of Wavelets in coding applications, representation, compression and signal filtering was observed.

The Wavelet Transform (WT) or WA, considered as a "microscope" in mathematics (CAO et al., 1995), by performing a local analysis among other properties, is a valuable tool for analyzing TS. Its use is efficient in reducing signal noise, image processing, signal and image compression, density estimate and time scale decomposition (ALEXANDRIDIS; ZAPRANIS, 2013).

When talking about a WT, the concepts about wavelets are intrinsic. In a way, it is said that wavelets are the nuclei of wavelets transform. Briefly, it can be said that Wavelets are small waveforms or reduced waveforms located in frequency (space) or time (KENDERDINE, 2010). WA contrast with other approaches in TS, such as Fourier Analysis or ARIMA models which comprise exclusively the domain in space (frequency) or time, respectively. Thus, with WA it is possible to analyze the local behavior of an ST as well as its singularities (MORETTIN, 1999).

Figure 2.0.1 presents the graph of two wavelet functions well know in the literature, as well as the sine function. It is easily noticeable that, different from the sine function, wavelet functions are dampened and null to t values that do not belong to a certain interval $[a, b]$.

The wavelet functions Mexican Hat and the real part of the Morlet[1], shown in Figure 2.0.1 are described by the Equations 2.0.1 and 2.0.2,

$$f(t) = \left(1 - t^2\right) * e^{\left(\frac{-t^2}{2}\right)}, \tag{2.0.1}$$

$$f(t) = cos({\omega_0}^2 * t) * e^{\left(\frac{-t^2}{2}\right)}. \tag{2.0.2}$$

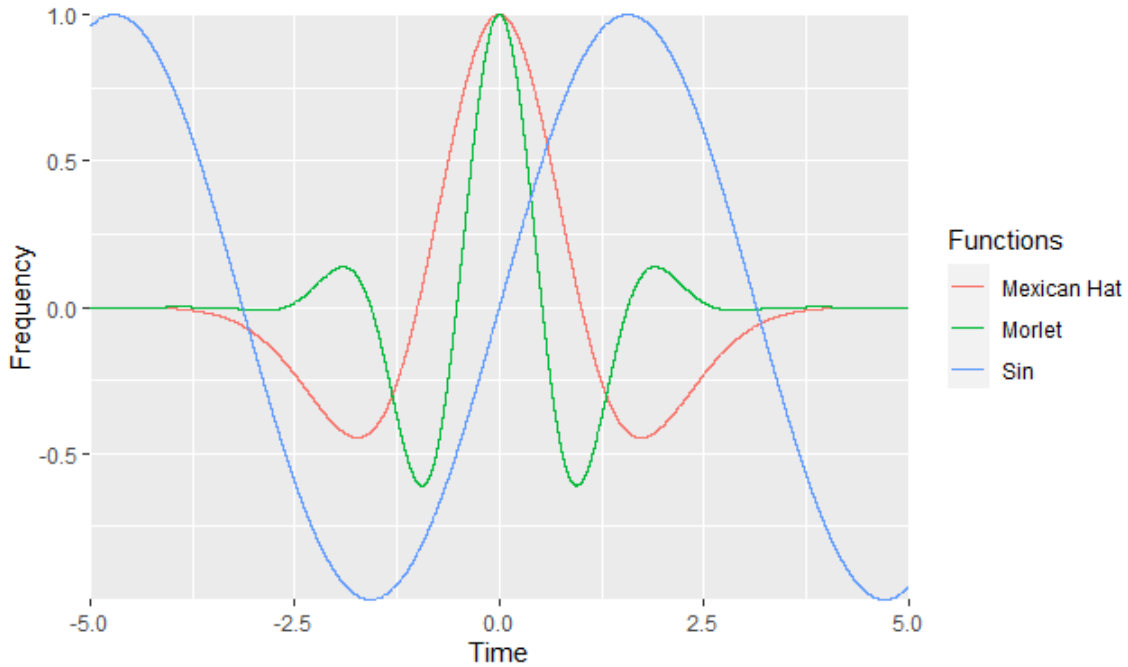

Figure 2.0.1 – Frequency of some functions

It can be thought that wavelet is the decomposition or representation of functions or data series, which is used to study the behavior of functions from signals or cleaning signals through WT. In practice, given a signal or even a TS, a pseudo fragmentation of the signal can be made in regions, ie, with a extensive signal it is possible to obtain several other more compact pseudo signals that in sequence form the original signal. This rupture allows wavelets to be used to more accurately analyze each region of the original signal.

From this point of view, wavelets can also enjoy the possibility of a more detailed analysis in certain regions of the original signal, for this, the frequency scale is increased. However, this

[1] Known as cosine-gaussian wavelet

leads to a decrease in the time scale, since both are inversely proportional. Therefore, the more details is desired, the smaller the pseudo signal that will be analyzed. Figure 2.0.2, present a TS as well as the Morlet wavelet, in different regions of the series and frequency scale.

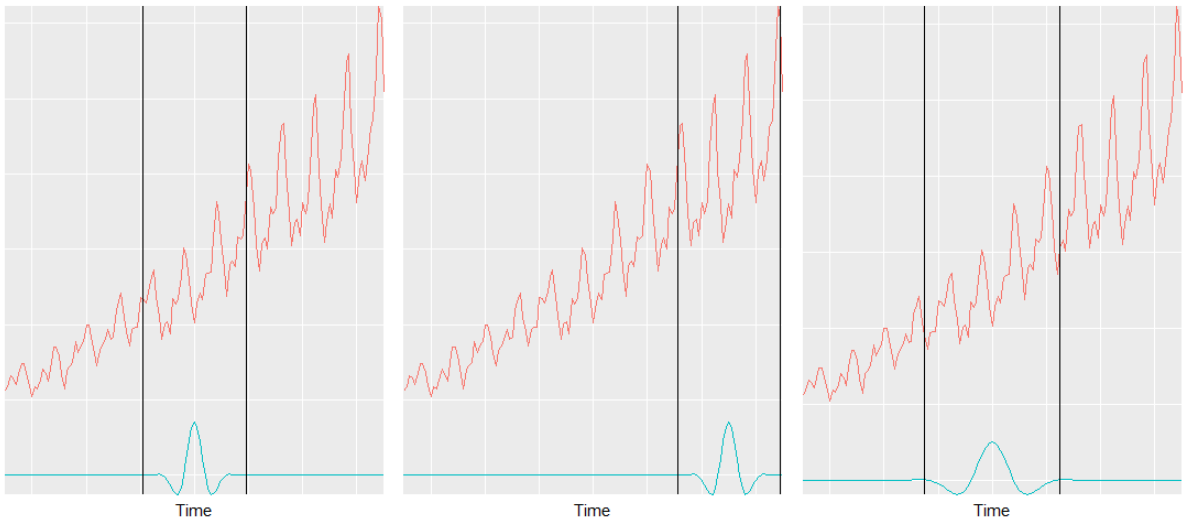

Figure 2.0.2 – Translations and dilations of the Morlet wavelet

In summary, the ideas from Figure 2.0.2 help to understand the concept of wavelet families. A wavelet function, like the Mexican Hat 2.0.1 or Morlet 2.0.2 that have been seen so far, is commonly called the Mother Wavelet (MW). Naturally, this function describes a graph that is centred on the origin $(0, 0)$, and behind the regions chosen for analysis, a horizontal translation is responsible for such action. Furthermore, compression or dilation resulting in an increase or reduction in the frequency scale, is also achieved by modifying the MW.

According to the changes in MW, a "new" wavelet is formed and is called Daughter Wavelet. In other words, for each MW countless Daughter Wavelet can be generated with the expansion or compression and translation of the mother. Frequently, the $\psi_{a.b}$ notation represents the Daughter Wavelet, where a is the expansion/compression parameter and b represents the translation, whose ψ function is the chosen MW itself. Equation 2.0.3 presents a Daughter mathematically,

$$\psi_{a,b} = \frac{1}{\sqrt{|\ a\ |}} * \psi\left(\frac{t-b}{a}\right), \tag{2.0.3}$$

such that, constant $\frac{1}{\sqrt{|a|}}$ is a weighing used to ensure that the same energy is assigned to the wavelets at all scales (KENDERDINE, 2010).

It is not difficult to perceive that for values of a and b equal to one and zero, respectively, the result is a MW without changes in the time and frequency scales, because

$$\psi_{1,0} = \frac{1}{\sqrt{|\,1\,|}} * \psi\left(\frac{1-0}{1}\right) = \psi(t).$$

Hwever, for other values of a and b the time and frequency scales are changed. Figures 2.0.3 and 2.0.4 present different daughter wavelets for mothers Mexican Hat e Morlet, respectively.

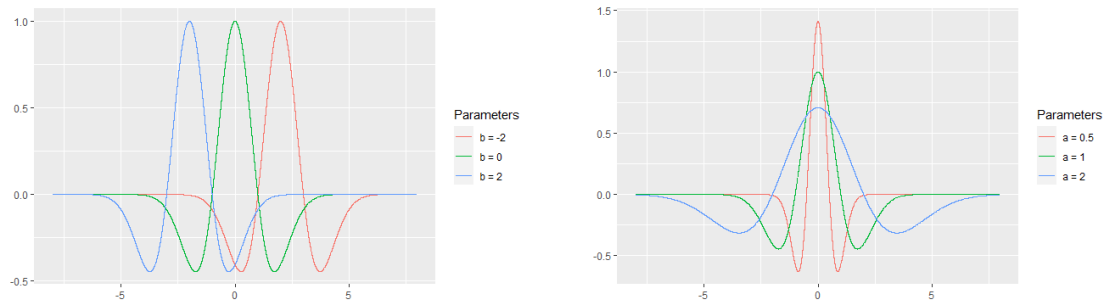

Figure 2.0.3 – Daughter wavelets of Mexican Hat

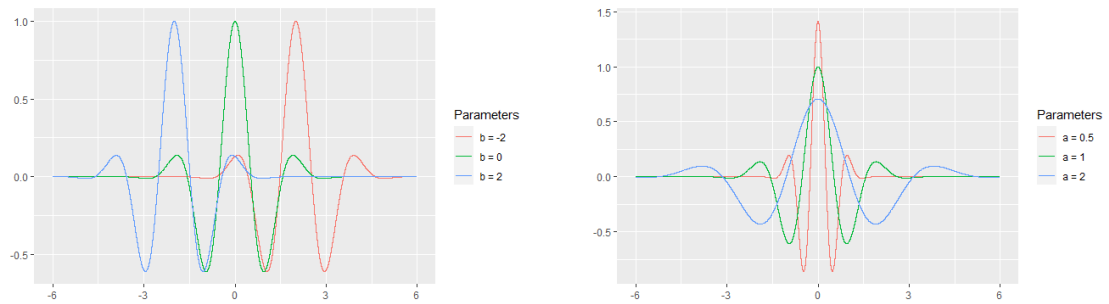

Figure 2.0.4 – Daughter wavelets of Morlet

The first definitions about Wavelets can be attributed to the work of (MORLET et al., 1982) and (GROSSMANN; MORLET, 1984) that defined Wavelets with broad meaning. However, the term Wavelet has had refinements and currently, it is usually assigned to a ψ function belonging to $\mathbb{L}_2(\mathbb{R})$ such that ψ translations and expansions,

$$\psi_{jk}(x) = 2^{\frac{j}{2}}\psi(2^j x - k), \text{ for } j, k \in \mathbb{Z}, \tag{2.0.4}$$

constitute an orthogonal base of $\mathbb{L}_2(\mathbb{R})$.

In turn, the notation $\mathbb{L}_2(\mathbb{R})$ represents the space of all the integrable square functions. Given a function f, f is said to belong to $\mathbb{L}_2(\mathbb{R})$ if:

- $\int |f|^2 < \infty$
- $\|f\| = \sqrt{\int f^2}$
- $\langle f, g \rangle = \int fg$

Now, a Wavelet is a ψ function belonging to $\mathbb{L}_2(\mathbb{R})$ that satisfies the following conditions:

1. $\int_{-\infty}^{+\infty} \psi(x)dx = 0$
2. $\int_{-\infty}^{+\infty} \psi^2(x)dx = 1$
3. If $\hat{\psi}(f)$ is the Fourier transform of $\psi(x)$ then

$$C_\psi = \int_0^{+\infty} \frac{\mid \hat{\psi}(f) \mid}{f} \, df$$

C_ψ is said admissibility constant and the first condition motivates the name small wave (KENDERDINE, 2010; VIDAKOVIC, 2009).

Generally, the wavelet transform is categorized as continuous and discrete, giving rise to Discrete Wavelet Transform (DWT) and Continuous Wavelet Transform (CWT). As in TS, there are many continuous[3] signs of some variable as a function of time, denoted by $x(t)$ and the graph can give the sensation of continuity, intuitively, the idea that can arise is to use approaches with CWT. But, this is not necessarily true, because DWT has been applied in many studies focusing on TS analysis, Therefore, the need for each context results in the choice of the transform (MEDEIROS, 2018).

The CWT, denoted by $W(a, b)$, is represented by the Equation 2.0.5,

$$W(a, b) = \int_{-\infty}^{\infty} x(t) * \frac{1}{\sqrt{\mid a \mid}} * \psi^* \left(\frac{t - b}{a} \right) dt, \tag{2.0.5}$$

where * indicates the complex conjugate of the wavelet. With some manipulations and properties mentioned earlier, the transform can be rewritten as the internal product between the signal $x(t)$ and Daughter Wavelet $\psi_{a,b}$, resulting in the following expression:

$$W(a, b) = \langle x(t), \psi_{a,b}(t) \rangle = \int_{-\infty}^{\infty} x(t) * \psi^*_{a,b}(t)dt. \tag{2.0.6}$$

[3] Time series are said to be continuous if observations are collected continuously over time

On the other hand, it is also possible to obtain the original signal $x(t)$ through the inverse wavelet transform, that is given by Equation 2.0.7 (KENDERDINE, 2010),

$$x(t) = \frac{1}{C_\psi a^2} \int_{-\infty}^{\infty} \int_{0}^{\infty} W(a,b) * \psi_{a,b} \ da \ db, \tag{2.0.7}$$

where C_ψ is show in condition 3 (2).

Chapter 3

State of Art

In the literature it is possible to find countless works approaching and exploring the theme of ANN. Such researches are driven by benefits and attractiveness of this powerful tool, as well as the wide applications capacity in images processing (PARISI et al., 1998), language processing (GOLDBERG, 2017), robotics (BEKEY; GOLDBERG, 2012), finance (WONG; SELVI, 1998), forecasting (ZHANG; QI, 2005) among other possibilities. In a more elementary way, some books present the basis for the mentioned ANN applications (HAYKIN, 2001; MEDSKER; JAIN, 1999).

The applications of neural networks in TS predictions with strong impact on the scientific community is consequence of a wide collaboration (CASDAGLI, 1989; WHITE, 1988; CONNOR; MARTIN; ATLAS, 1994). According to the theory, combined with practice, the NN developments have been consolidated or even amplified, allowing researchers to investigate more and more possibilities to incorporate new approaches in this branch of science. Some approaches that have presented to be very powerful for several kind of data are those regarding Wavelet Neural Networks (WNN) (GANJEFAR; TOFIGHI, 2015; HUANG; WANG, 2018b; CAZELLES et al., 2008). Thus, we will go through some works and researches, whose core on their construction are wavelets, that substantiate this dissertation.

A part of the collection of publications on neural network are from the changes that are made in the network structure, originating a new architecture and, consequently, a network still unknown. In this sense, some authors have proposed different network structures in order to study mainly the advantages and improvements in TS predictions, because traditional forecasting models often do not show satisfactory results.

In a general way, the appropriate ANNs for the TS approach are Recurrent Neural Network (RNN) or their variations, because they are able of to incorporate information about the

previous time instants in order to infer future predictions, e.g. (DORFFNER, 1996; ZHANG; CHAI; FU, 2012). However, authors such as (CHEN; YANG; DONG, 2006; WANG; LIU, 2008; KRISHNA; SATYAJI; PC, 2011) proposed some modifications in the functioning of networks to predict TS using non-recurring networks, certain works present satisfactory results.

In this perspective, in some works as (RAO; KUMTHEKAR, 1994), the network is fully connected (FRWNN), this means that inputs and neurons are connected to all existing neurons in the network. This neural network gives rise to simpler ones, as the Diagonal Recurrent Wavelet Neural Network (DRWNN), where neurons in the hidden layer do not transmit information to each other, seen in (LEE et al., 1992; KU; LEE, 1995; CAO; LIN, 2008).

In the work of (SHARMA et al., 2016), it was proposed a neural network in which the entries propagate information to the neurons of the hidden layer, but are also connected directly to the output layer, though linear weights. In other researches, as in (LIN; LEE; CHIN, 2006; YOO; PARK; CHOI, 2007), a more singular recurrent network is found, where the recurrence occurs in hidden layers neurons and they transmit such information from the past moment(s) to themselves, originating the Self-Recurrent Wavelet Neural Network.

The Stochastic Recurrent Neural Network (SRNN) model is built by integrating an effective random-time function. Due to the stochastic time force, the model ends up assigning different weights to the historical data at different times (LI; YAN, 2019). By combining with wavelet approaches, the Stochastic Recurrent Wavelet Neural Network (SRWNN) can be created. Another model that combines SRWNN is seen in (LI; WANG, 2020). In this hybrid models, the Ensemble Empirical Mode Decomposition (EEMD) approach is used to decompose the original TS into others with different frequencies and then apply the SRWNN. Posteriorly, combines the results to make predictions, this network is denoted by E-SRWNN. Another similar work is approached in (HUANG; WANG, 2018a), in which it uses the combination of Discrete Wavelet Transform (DWT), to make decomposition in subseries of different frequencies, and the SRWNN, called DWT-SRWNN.

There are networks that take into account the dynamics of the TS which is being studied. In paper of (JIANG; ADELI, 2005), a Dynamic Recurrent Wavelet Neural Network is observed, being used to incorporate time and days of the week due to the specificity of the series that addresses traffic flow forecasting. Similarly, in (SHI; CHU; CHEN, 2011) a dynamic network is implemented to the predict future copper price returns on the Shanghai Futures Exchange, this network incorporates useful information such as exogenous factors that aid in prediction.

A network model that bases its algorithm on bilinear polynomial is seen in (PARK; ZHU, 1994), called of Bilinear Recurrent Neural Network (BLRNN). The structure of this model is similar to that of Elman, since the information that is obtained by the hidden layer area assigned as a vector in the inputs layer for the next future instants. Also, inspired by Elman

and Jordan (JORDAN, 1990), the work of (HUANG; RASHID; KECHADI, 2006) addresses a multi-context recurrent neural network (MCRNN) that presents recurrence with a multi-context layer, in which it receives the synaptic signals and include the hidden neurons in the next inputs of the network. Furthermore, this network includes the information from the multi-context layer in the output layer, that is, in MCRNN the multi-context layer is linked directly to the hidden and output layers.

The work (KIM et al., 2016) shows an application of a Long-Short Term Memory (LSTM) architecture to a RNN giving rise to the LSTM-RNN hybrid model. However, the recurrent hidden layer is replaced by an LSTM cell, which takes into account the previous instant to obtain the result of the current instant. In (TIAN; PAN, 2015) there is a very similar structure, where it is said that there is a totally recurrent internal connection Another LSTM-RNN is presented in (ZEN; SAK, 2015), but unlike the previous two, it also has recurrence in the output layer. The recurrent output layer is a simple extension of the conventional RNN.

In another way, the structure of a neural network can be modified according to the composition of its phases. Therefore, the way a network groups the information resulting from the hidden layer can be modified, giving different results. In (YOO; PARK; CHOI, 2007; LIN; TAI; CHUNG, 2014; YOO; PARK; CHOI, 2005), the result of the hidden layer are joined by means of producers and then the sum of these signals is carried out, along with weights, to generate the network output. On the other hand, in (CAO; LIN, 2008; WANG; LIU, 2008; KRISHNA; SATYAJI; PC, 2011), summations are used for both phases of network signal processing.

In line with the possibilities discussed above, the work developed in this dissertation contemplates some characteristics of these networks, incorporated into a new structure. In this way, the start point was the recurrence that has long been widespread with Elman (ELMAN, 1990). Besides, supported by (HUANG; RASHID; KECHADI, 2006) aiming to optimize the learning speed and accuracy that are bestowed when context information is also transmitted to the output layer. Furthermore, the linear layer presented by (SHARMA et al., 2016) was incorporated into this neural network, because the redundancy provided with the weighted direct connection is beneficial to the process as it makes training simpler.

Chapter 4

Methodology

The RWNN model proposed to study TS in this work is an extension and fusion of the networks addressed by (ELMAN, 1990; HUANG; RASHID; KECHADI, 2006; SHARMA et al., 2016) that were explained throughout the text. The interior of the network is divided into five layers of operations, namely input layer, hidden layer, context layer, exogenous layer and output layer, which develop exclusive functionalities.

The network proposed in this work will be incorporated into the WNN (SOUZA, 2021) package that is under development. The creation of this package aims to present WNN as tools for statistical problems, e.g. explain a response variable as a function of other variables, whether categorical or numerical.

4.1 Network structure

The structure of the implemented network is presented in this paragraph and illustrated with the Figure 4.1.1. The input layer is responsible for weightingly sending the input data to the hidden layer and also for making a direct and weighted connection with the output layer. When activated, the exogenous layer stores and sends the observations of the exogenous variable with weights to the hidden layer. The hidden layer is where the wavelons (nodes) are found, which receive information from the input, context and exogenous layers and process it with wavelet functions, then transmit the weighted results to the output layer and a copy of the results is stored in the context layer. The context layer, in turn, directs weighted information to the hidden and output layer.

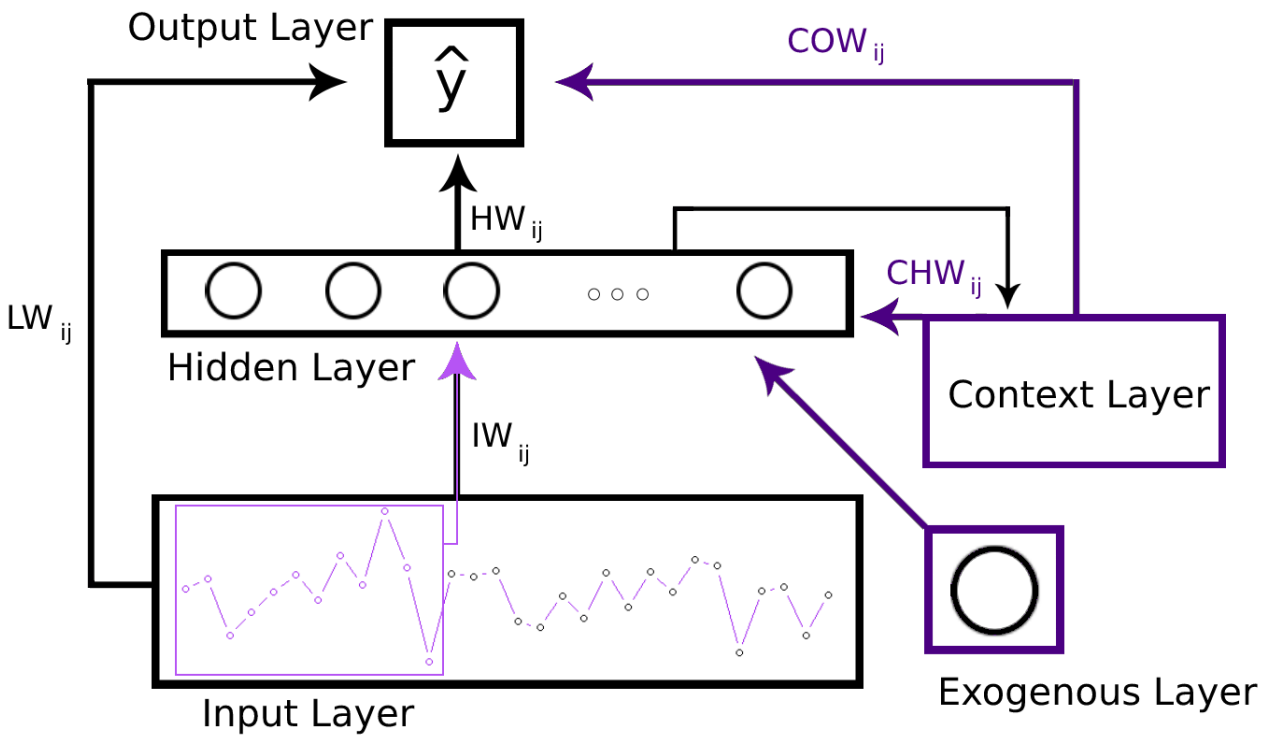

Figure 4.1.1 – Proposed structure of a recurrent wavelet neural network

4.2 Network operation

This network is prepared to learn the behavior of a TS. Therefore, the network is designed to accept a main TS (study object). In addition, when necessary, exogenous variables or other time series of interest can be included in the network, since they are properly adjusted according their cross-correlation lag. In addition to these external elements, it is necessary to provide the following arguments to the network boot:

- The info window, which is responsible for establishing how many previous observations in time will be used as predictors of the future observation. E.g. being the observation window corresponding to twelve, means that the first twelve observations will be used to predict the thirteenth, and so on;
- The forecasting horizon, that must be reserved at the end of the series, limiting the size of the series used in the training phase and determining how many predictions will be performed in the testing phase;
- The data normalization, being possible to choose one of the five of the alternatives presented in Table 1;

- The choice of the mother wavelet, being possible to choose between the Morlet wavelet, Mexican hat and second derivative of the sigmoid function;
- The number of wavelons, which will be operating internally in the hidden layer;
- The error correction rate or learning rate (η);
- The maximum accepted epoch, that is, a limit on how many iterations the network can perform in the training phase;
- The choice between forecasting one step ahead or several steps ahead. While in the first, the original observations are included one by one after the one-step-ahead prediction, in the second, the predictions from the test phase are incorporated into the network and are used in the next estimates.

Table 1 – Data normalization mode, considering $m = \frac{max(x)-min(x)}{8}$

Normalization mode	Equation
One	$\frac{x}{max(x)}$
Two	$\frac{x-mean(x)}{sd(x)}$
Three	$\frac{x-min(x)}{max(x)-min(x)}$
Four	$0.1 + \frac{0.8*(x-min(x))}{max(x)-min(x)}$
Five	$\frac{(x-(min(x)-m))}{(max(x)-m)-(min(x)-m)}$,

In order to keep the reading flowing, the observations destined for the test phase are named by test data and the original TS subtracted from the test data is named training data.

4.2.1 The feeding of the network

The feeding of the proposed network is made through the TS itself. More precisely, the input layer is prepared to receive the input vector of size k, where k is the observation window size defined initially. The input vector is responsible for identifying the observations that will be active at each instant of time. For the first instant of learning time, the first k-th observations of the series are used. In the next instant, removes the first and includes the k-th + 1 observation, keeping the default input vector size. Likewise, this translation process is carried out until the training data are contemplated and included in the network.

Visually, the Figure 4.1.1 illustrates this mechanism, where the purple rectangle represents the information window that contains the input vector of the first training time instant. The translations of a unit to the right of this window, carry the input vector information of the next instants of time. When the exogenous layer is activated, the predictions are also influenced by this extra feeding.

The Table 2 presents the observations pertaining to the input vector for each training time instant

Table 2 – Network input vectors proposed in this work

Observations of the training data	Instants
$y_1, y_2, \cdots, y_k$	one
$y_2, y_3, \cdots, y_{k+1}$	two
$y_3, y_4, \cdots, y_{k+2}$	three
...	...

4.3 Network learning

In section 4.1, the structure of the proposed network and its working flow are discussed. In this section, the learning mechanisms will be explained and the numerical results that are used in the neural network developed in this work will be presented.

As presented in subsection 4.2.1, network input is provided through subsequences of the original series. This means that each one of the inputs influences in the learning. When an input vector is established and enters the network, the result of operations between the layers (hidden, linear and context) is a prediction of the next observation of the subsequence of the series. The network is said to be able to make predictions when all the weights present inside the NN are precisely adjusted, given that they are the ones who do the hard work of inferring answers about the network inputs. In this sense, the first input subsequences result in a very bad result, that is, estimates far from the real expected value, because the weights are initially randomized. However, at each input of the network, the weights are adjusted by the gradient method, which is a numerical method used to find the minimum of a function, which in this case is the error function or cost function 1.1.8.

For each error made, the network tries to correct it through weight adjustments, learning the TS behavior at each iteration. Corrections follow from the equations below where e represents the error committed, ψ is the mother wavelet, parameters a and b are responsible for dilation and translation and $aux = \frac{(input \ * IW)+(context \ * CHW)-b}{a}$, being IW the input layer weights and CHW the the context layer weights for the hidden layer.

$$
\begin{aligned}
HW_{ij}(t+1) &= HW_{ij}(t) + \eta * \frac{\partial E}{\partial HW_{ij}} \\
&= HW_{ij}(t) + \eta * e * \psi\left(aux\right); \\
IW_{ij}(t+1) &= IW_{ij}(t) + \eta * \frac{\partial E}{\partial IW_{ij}} \\
&= IW_{ij}(t) + \eta * e * input * HW * a^{-1} * \psi'\left(aux\right); \\
CHW_{ij}(t+1) &= CHW_{ij}(t) + \eta * \frac{\partial E}{\partial CHW_{ij}} \\
&= CHW_{ij}(t) + \eta * e * context * HW * a^{-1} * \psi'\left(aux\right); \\
COW_{ij}(t+1) &= COW_{ij}(t) + \eta * \frac{\partial E}{\partial COW_{ij}} \\
&= COW_{ij}(t) + \eta * e * context; \\
LW_{ij}(t+1) &= LW_{ij}(t) + \eta * \frac{\partial E}{\partial LW_{ij}}; \\
&= LW_{ij}(t) + \eta * e * input; \\
a_i(t+1) &= a_i(t) + \eta * \frac{\partial E}{\partial a_i} \\
&= a_i(t) - \eta * e * HW * \psi'\left(aux\right) * a^{-1} * \left(aux\right); \\
b_i(t+1) &= b_i(t) + \eta * \frac{\partial E}{\partial b_i} \\
&= b_i(t) - \eta * e * HW * a^{-1} * \psi'\left(aux\right).
\end{aligned}
$$

When the last error is generated, the iterative process comes to an end and the learning ending conditions are checked. If the error is greater than the maximum acceptable error and the maximum epoch has not been exceeded either, the network starts the next iteration doing the same process successively until one of the two closure conditions is met. After learning of the network with training data, the validation phase begins.

4.4 Network validation

At this moment, everything that was possible to learn from the series is stored and will be used to validate the results of the NN adjustment. In other words, the weights responsible for processing the information that arrives through the input layer and is transmitted throughout the NN are tested, in order to verify that the estimates meet expectations.

Validations are performed using test data reserved in advance, this means that the network will have to predict them. For this, the two possible ways of validating the adjustment are

taken into account, being they one step ahead and multi-step ahead. However, regardless of which option is chosen, the test is started in such a way that the input layer uses the last sequence of training data of size k, as these are the observations prior to the first element of the test data.

If the one step ahead option is chosen, the forecasts are made with the data from the series, i.e, the input vector groups the real information of the series and sends it inside the NN that uses the stored weights of the training phase to generate all the predictions. Basically, it works the same as the learning step without weight correction, until you reach the end of the series.

With the several steps ahead option, the results are different because the predictions are being incorporated in the network. This means that instead of the input vector contain only the original data from the series to make the predictions, this data is replaced by the predictions. Or yet, as the values are predicted, they are used to predict the next elements, until there is an estimate for all test data. Finally, it is enough to validate the results obtained by comparing the original series with the estimates made by the network.

4.5 Structure alternatives for the network

The motivation of the structure proposed in this work was to create a new recurrent wavenet neural network structure for TS in view of the possibilities of neural network structures present in the literature. With this, we sought to improve the learning and forecasting results, which are detailed in Chapter 5.

The default structure of the network is formed by input layer, hidden layer, context layer, exogenous layer, output layer and also by a direct connection from the input layer to the output layer. Since the input, hidden and output layer are basic units for the functioning of any network and the exogenous layer is optional depending on the TS, the removal of the other units generates different structures that can also be used. However, not including the context layer mischaracterize a recurrent neural network structure.

Table 3 presents the possible combinations of networks in view of the main structure that was proposed in the work.

Table 3 – Different structures considered in this research

Structure	Author
Input, hidden, output, and linear connection	(SHARMA et al., 2016)
Input, hidden, output, and context for hidden and output	(HUANG; RASHID; KECHADI, 2006)
Input, hidden, output, and context for hidden	(ELMAN, 1990)
Input, hidden, output, and context for output	Proposed
Input, hidden, output, linear connection and context for hidden	Proposed
Input, hidden, output, linear connection and context for output	Proposed
Structures above with exogenous variables and/or TS included	Proposed

4.6 Data set

To evaluate the adjustment and prediction capacity of the proposed RWNN, TS of different natures were used. The classic Box & Jenkins airline data, with monthly totals of international airline passengers, between 1949 and 1960. TS of monthly distances[1] between the asteroid Apophis and Planet Earth, from 2004 to 2021. And also the TS of the hospitalization rate for bronchiolitis in the state of Paraná-BR, between 2000 and 2019.

4.7 Implementation

The implemented RWNN is formed by the elementary layers (input, hidden and output), by a context layer that stores information from the hidden layer and uses them to add information to the hidden and output layers at the next instant of time, and a linear connection which transmits information from the input layer to the output layer.

The R programming language (R Core Team, 2020) was used as the basis for implementing the developed network. As well as the data mentioned in section 4.6 were selected to evaluate the performance of the main RWNN (which is illustrated in figure 4.1.1) in TS. The computational resources used in the implementation were core i5-4440 3.10 GHz processor and 8 GB DDR3 RAM memory.

As discussed in section 4.2, for the network to start the learning process of a TS it is necessary to provide some arguments. Their choice directly impacts the performance of tuning and predicting the network. With this in mind, simulations were performed using combinations of RWNN initialization arguments for each of the series. In the simulations of the AirPassengers Series, the mother wavelets of Morlet, Mexican Hat and the second derivative of the sigmoid function were used, input windows of sizes 12, 24, 36, and 48, the five normalization modes

[1] The scale of distances is given in Astronomical Unit (AU), where 1 AU is equivalent to 149,597,870,700 meters.

(shown in Table 1), a amount of 7, 8, 9, 10, and 12 wavelons and 12 observations to be predicted. In the simulations for the Apophis asteroid data, the three mother wavelets, the input windows of size 10, 20, and 30, both normalization modes, the amount of 7,8,9,10, and 12 wavelons and 12 observations to be predicted. For the bronchiolitis data, the three mother wavelets were used, input windows of sizes 48, 60, 72, and 84, all five normalization modes, the amount of 10, 20, and 30 wavelons and 24 observations to be predicted. In all simulations 1000 maximum training times and learning rate with values of 0.1 and 0.01 were used.

In addition, both tests were done using several steps ahead predictions to assess the predictive capability of the network, even though better results are expected in one step ahead predictions. Two simulations are performed to present the performance of RWNN under the conditions of 1 and 24 steps ahead on the bronchiolitis data. For this, the second derivative sigmoid, info window equal to 24, forecast horizon of 24 observations, normalization five, learning rate equal to 0.01, and 9 wavelons were used.

In order to evaluate the models proposed by RWNN, the Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and coefficient of determination R2 were presented for both training and forecasting.

Chapter 5

Results

In this chapter the results of the proposed main structure of the proposed RWNN are presented, that is, the network with layers input, hidden, output and context and linear connection between input and output layer.

Among the RWNN initialization possibilities, the arguments selected for presenting the best predictive results through the simulations are presented in Table 4, according to each TS.

Table 4 – Arguments selected through simulations for each TS

	Time series		
Arguments	AirPassengers	Apophis	Bronchiolitis
Info window	48	20	84
Forecast horizon	12	12	24
Mother wavelet	Morlet	Morlet	Sigmoid second derivative
Number of wavelons	8	8	20
Normalization	Four	Five	Five
Learning rate	0.1	0.01	0.01
Maximum epoch	1000	1000	1000

The results of adjustment and forecasting performance and the time spent on learning are seen in Table 5. Furthermore, the visual results are illustrated in Figures 5.0.1, 5.0.2, and 5.0.3, where the gray, purple and black colors represent the original series, the training series adjustment and the prediction of the model adjusted by the network, respectively.

Table 5 – Results of adjustment and prediction of the network for each of the series

	Time series		
Results	AirPassengers	Apophis	Bronchiolitis
Training MAE	0.01281	0.01098	0.03930
Training RMSE	0.01679	0.01730	0.05099
Training MSE	0.00014	0.00014	0.00130
Training R2	0.99333	0.99510	0.93413
Test MAE	0.02147	0.03150	0.06852
Test RMSE	0.02549	0.03741	0.08791
Test MSE	0.00032	0.00069	0.00386
Test R2	0.95502	0.84971	0.88752
Training time	17.31 seconds	24.14 seconds	25.68 seconds

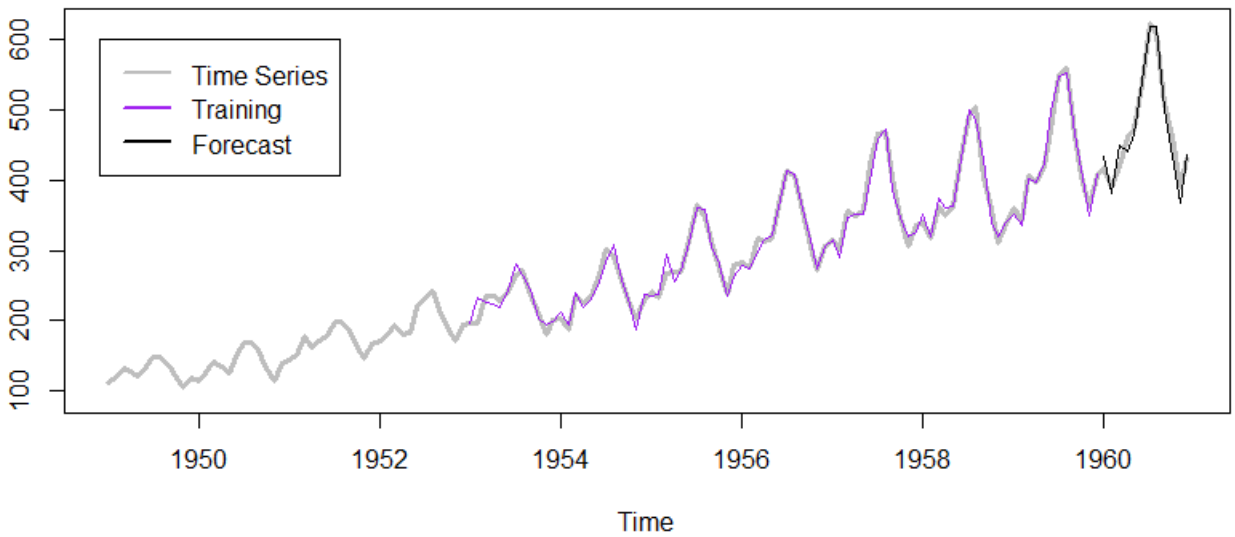

Figure 5.0.1 – Adjustment and prediction of Time Series AirPassengers between the years 1949 and 1960, using RWNN with Morlet wavelet, info window equal to 48, forecast horizon of 12 observations, 8 wavelons, normalization four and learning rate equal to 0.1 as presented in Table 4

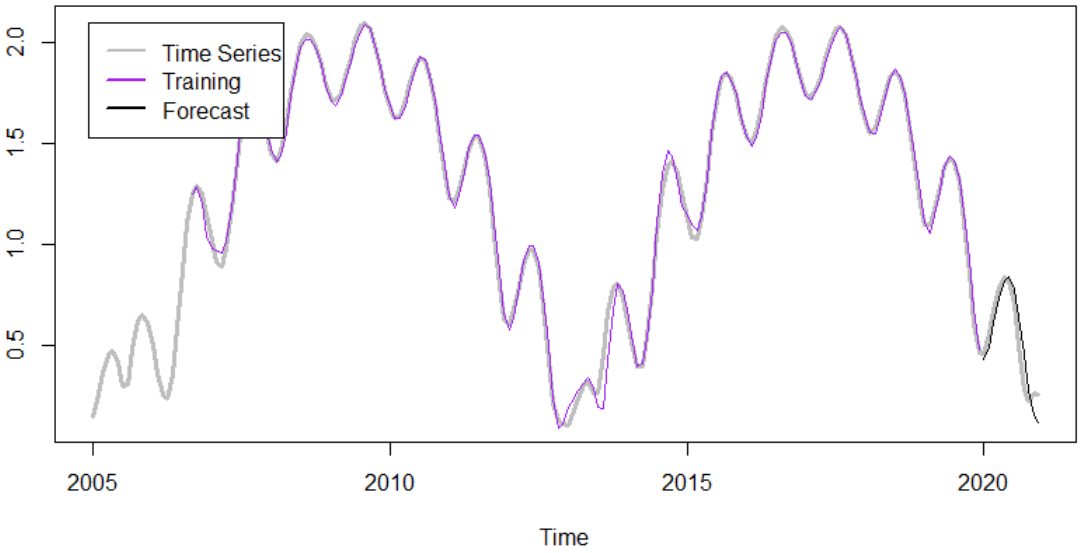

Figure 5.0.2 – Adjustment and prediction of Apophis asteroid distance data between the years 2005 to 2020, using RWNN with Morlet wavelet, info window equal to 20, forecast horizon of 12 observations, 8 wavelons, normalization five and learning rate equal to 0.01 as presented in Table 4

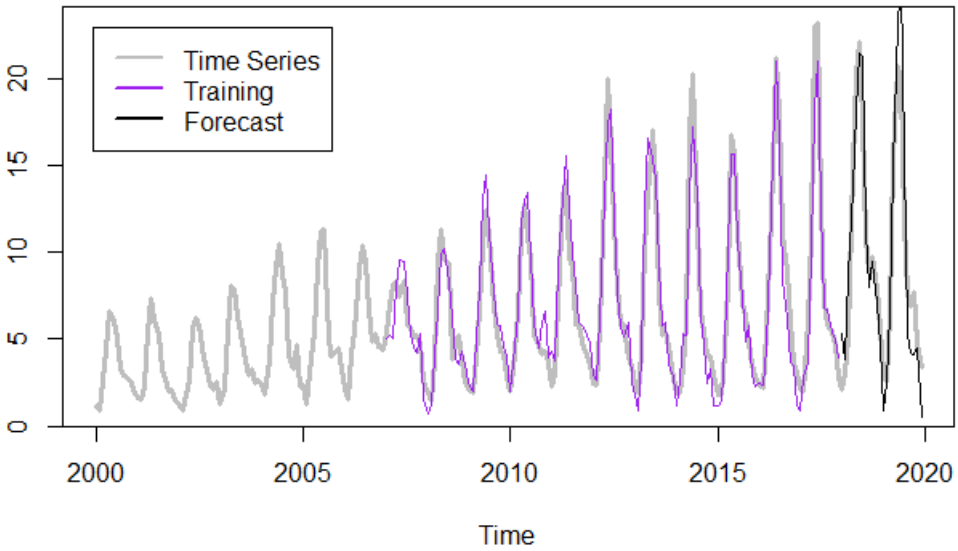

Figure 5.0.3 – Adjustment and prediction of bronchiolitis data between the years 2000 and 2019, using RWNN with Sigmoid second derivative wavelet, info window equal to 84, forecast horizon of 24 observations, 20 wavelons, normalization five and learning rate equal to 0.01 as presented in Table 4

The combinations of arguments presented in Table 4 were chosen because they presented the best results in the simulations according to the maximum epoch training allowed. However, using other combinations of arguments yield similar results. As an illustration, in Tables 6 and 7, another combination used in RWNN and the results of its performance in the series AirPassengers are presented. Figure 5.0.4 shows the graphical results.

Table 6 – Selected arguments for training of the network for AirPassengers serie

Arguments	AirPassengers
Info window	24
Forecast horizon	10
Mother wavelet	Sigmoid second derivative
Number of wavelons	8
Normalization	Five
Learning rate	0.1
Maximum epoch	1000

Table 7 – Results of adjustment and prediction of the network with another combination for AirPassengers serie, using RWNN with Sigmoid second derivative wavelet, info window equal to 24, forecast horizon of 10 observations, 8 wavelons, normalization five and learning rate equal to 0.1 as presented in Table 6

Results	AirPassengers
Training MAE	0.01023
Training RMSE	0.01496
Training MSE	0.00011
Training R2	0.99324
Test MAE	0.01936
Test RMSE	0.02577
Test MSE	0.00033
Test R2	0.95704
Training time	17.31 seconds

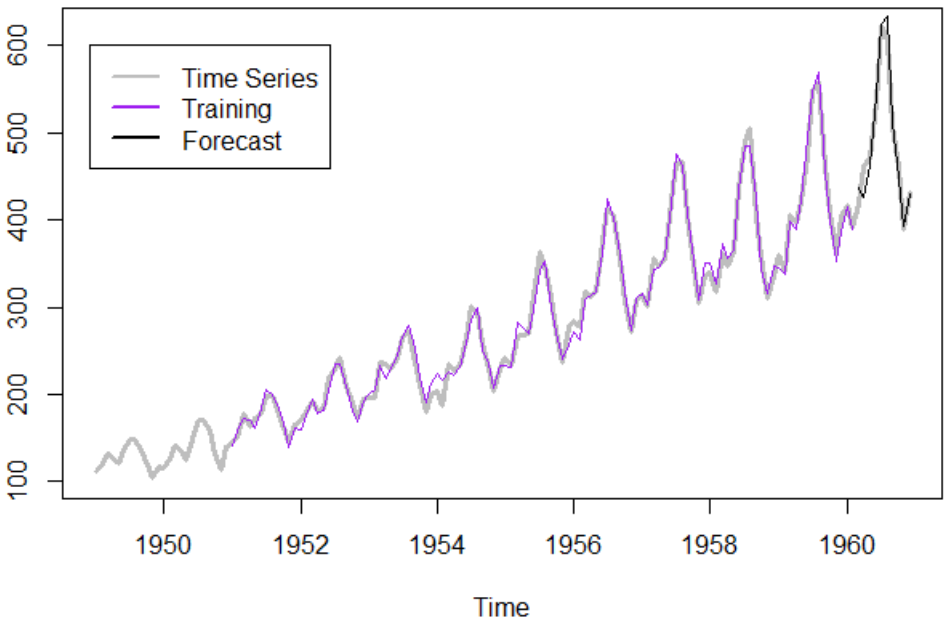

Figure 5.0.4 – Adjustment and prediction of other model of Time Series AirPassengers between the years 1949 and 1960, using RWNN with Sigmoid second derivative wavelet, info window equal to 24, forecast horizon of 10 observations, 8 wavelons, normalization five and learning rate equal to 0.1 as presented in Table 6

It is worth noting that both training and tests are carried out after the normalization of the series among the alternatives available on the network, as shown in Table 4. This means that

the statistics presented in Table 5 are under the influence of normalization, however in the future it is intended to present them in the data unit. But, it is notorious that the implemented RWNN presented good results, allied to a low processing time, taking into account factors that influence this, that is, the number of observations of each series, the input observation windows, the number of wavelons and the maximum epochs.

The results of the simulations of Chapter 4 to evaluate the performance of the network in the predictions one and h-steps ahead (h = 24), where an info window equal to 24, forecast horizon equal to 10, a sigmoid second derivative wavelet were used, normalization five, 9 wavelons, learning rate equal to 0.01 in bronchiolitis data were compared. While in the 24 steps ahead forecast the network presents an MSE equal to 0.003348, in the 1 step ahead forecast this value decreases to 0.001636.

Chapter 6

Final considerations

This work addresses Recurrent Wavelets Neural Networks which is still a recent theme and has fields that can be further explored. For this, it was necessary to study primitive subjects related to RWNN, that is, ANN and Wavelet Analysis that are present in this dissertation and are detailed in Chapters 1 and 2.

Through this, it was possible to understand the mechanisms to start the construction of a recurrent neural network combined with WA. In this sense, Chapter 3 seeks to present some possibilities of structures and different strategies that have been added to the current state of knowledge about this category of ANN.

In order to corroborate with the progress of this branch of science, this work brings a RWNN composed by other networks, i.e, the proposed architecture combines a extends some ideas of sight approaches in the work of (SHARMA et al., 2016; HUANG; RASHID; KECHADI, 2006; ELMAN, 1990). Briefly, it was proposed to investigate a network whose architecture is formed by input layer, hidden layer, context layer, output layer and a linear connection between input and output layers. Arising from this, Chapter 5 contains the results from the network developed in this work, which proved to be efficient and promising when applied to the three real TS, including a usual TS in the literature as well as current TS in Health Sciences.

The programming language R was chosen to implement the developed RWNN. However, it is still possible to take advantage of the network created to scrutinize the proposal of this study. In view of this, it is intended to generate another more complex and even more complete network. So, it is up to future work to add the ability to include exogenous variables to improve the learning process of this network.Furthermore, it is up to the next work, to take advantage of the methods developed by (MEDEIROS, 2018) at RWNN, and thus build the confidence intervals of the point estimates in the training data and forecast of time series.

Bibliography

ABRAHAM, A. Artificial neural networks. *Handbook of measuring system design*, Wiley Online Library, 2005.

ADAMOWSKI, J.; CHAN, H. F. A wavelet neural network conjunction model for groundwater level forecasting. *Journal of Hydrology*, Elsevier, v. 407, n. 1-4, p. 28–40, 2011.

ALEXANDRIDIS, A. K.; ZAPRANIS, A. D. Wavelet neural networks: A practical guide. *Neural Networks*, Elsevier, v. 42, p. 1–27, 2013.

ANJOY, P.; PAUL, R. K. Comparative performance of wavelet-based neural network approaches. *Neural Computing and Applications*, Springer, v. 31, n. 8, p. 3443–3453, 2019.

BEKEY, G. A.; GOLDBERG, K. Y. *Neural networks in robotics*. [S.l.]: Springer Science & Business Media, 2012. v. 202.

BENYUS, J. M. *Biomimicry: Innovation inspired by nature*. [S.l.]: Morrow New York, 1997.

CAO, J.; LIN, X. Study of hourly and daily solar irradiation forecast using diagonal recurrent wavelet neural networks. *Energy Conversion and Management*, Elsevier, v. 49, n. 6, p. 1396–1406, 2008.

CAO, L.; HONG, Y.; FANG, H.; HE, G. Predicting chaotic time series with wavelet networks. *Physica D: Nonlinear Phenomena*, Elsevier, v. 85, n. 1-2, p. 225–238, 1995.

CASDAGLI, M. Nonlinear prediction of chaotic time series. *Physica D: Nonlinear Phenomena*, Elsevier, v. 35, n. 3, p. 335–356, 1989.

CAZELLES, B.; CHAVEZ, M.; BERTEAUX, D.; MÉNARD, F.; VIK, J. O.; JENOUVRIER, S.; STENSETH, N. C. Wavelet analysis of ecological time series. *Oecologia*, Springer, v. 156, n. 2, p. 287–304, 2008.

CHATFIELD, C. *Time-series forecasting*. [S.l.]: CRC press, 2000.

CHEN, Y.; YANG, B.; DONG, J. Time-series prediction using a local linear wavelet neural network. *Neurocomputing*, Elsevier, v. 69, n. 4-6, p. 449–465, 2006.

CONNOR, J. T.; MARTIN, R. D.; ATLAS, L. E. Recurrent neural networks and robust time series prediction. *IEEE transactions on neural networks*, IEEE, v. 5, n. 2, p. 240–254, 1994.

CRONE, S. F.; NIKOLOPOULOS, K.; HIBON, M. Automatic modelling and forecasting with artificial neural networks–a forecasting competition evaluation. *Final report for the IIF/SAS Grant*, v. 6, p. 2008, 2005.

DE, M. G. *Velvet type fabric and method of producing same*. [S.l.]: Google Patents, 1955. US Patent 2,717,437.

DORFFNER, G. Neural networks for time series processing. In: CITESEER. *Neural network world*. [S.l.], 1996.

ELMAN, J. L. Finding structure in time. *Cognitive science*, Wiley Online Library, v. 14, n. 2, p. 179–211, 1990.

ESEN, H.; OZGEN, F.; ESEN, M.; SENGUR, A. Artificial neural network and wavelet neural network approaches for modelling of a solar air heater. *Expert systems with applications*, Elsevier, v. 36, n. 8, p. 11240–11248, 2009.

FU, K. S.; MENDEL, J. M. *Adaptive, Learning, and Pattern Recognition Systems: Theory and Applications*. [S.l.]: Academic Press, 1970.

GANJEFAR, S.; TOFIGHI, M. Single-hidden-layer fuzzy recurrent wavelet neural network: Applications to function approximation and system identification. *Information Sciences*, Elsevier, v. 294, p. 269–285, 2015.

GOLDBERG, Y. Neural network methods for natural language processing. *Synthesis lectures on human language technologies*, Morgan & Claypool Publishers, v. 10, n. 1, p. 1–309, 2017.

GROSSMANN, A.; MORLET, J. Decomposition of hardy functions into square integrable wavelets of constant shape. *SIAM journal on mathematical analysis*, SIAM, v. 15, n. 4, p. 723–736, 1984.

HAYKIN, S. *Neural networks: principles and practices*. [S.l.]: Bookman Publisher, 2001.

HUANG, B.; RASHID, T.; KECHADI, M. Multi-context recurrent neural network for time series applications. *International Journal of Computational Intelligence*, v. 3, n. 1, p. 45–54, 2006.

HUANG, L.; WANG, J. Forecasting energy fluctuation model by wavelet decomposition and stochastic recurrent wavelet neural network. *Neurocomputing*, Elsevier, v. 309, p. 70–82, 2018.

HUANG, L.; WANG, J. Global crude oil price prediction and synchronization based accuracy evaluation using random wavelet neural network. *Energy*, Elsevier, v. 151, p. 875–888, 2018.

JIANG, X.; ADELI, H. Dynamic wavelet neural network model for traffic flow forecasting. *Journal of transportation engineering*, American Society of Civil Engineers, v. 131, n. 10, p. 771–779, 2005.

JORDAN, M. I. Attractor dynamics and parallelism in a connectionist sequential machine. In: *Artificial neural networks: concept learning*. [S.l.: s.n.], 1990. p. 112–127.

KENDERDINE, R. D. Wavelet-based resampling techniques. 2010.

KIM, J.; KIM, J.; THU, H. L. T.; KIM, H. Long short term memory recurrent neural network classifier for intrusion detection. In: IEEE. *2016 International Conference on Platform Technology and Service (PlatCon)*. [S.l.], 2016. p. 1–5.

KOLEN, J. F. *Exploring the computational capabilities of recurrent neural networks*. Tese (Doutorado) — The Ohio State University, 1994.

KRISHNA, B.; SATYAJI, R. Y.; PC, N. Time series modeling of river flow using wavelet neural networks. *Journal of Water Resource and Protection*, Scientific Research Publishing, v. 2011, 2011.

KU, C.-C.; LEE, K. Y. Diagonal recurrent neural networks for dynamic systems control. *IEEE transactions on neural networks*, IEEE, v. 6, n. 1, p. 144–156, 1995.

LEE, K.; CHOI, T.; KU, C.; PARK, J. Short-term load forecasting using diagonal recurrent neural network. In: IEEE. *[1993] Proceedings of the Second International Forum on Applications of Neural Networks to Power Systems*. [S.l.], 1992. p. 227–232.

LI, J.; WANG, J. Stochastic recurrent wavelet neural network with eemd method on energy price prediction. *Soft Computing*, Springer, v. 24, p. 17133–17151, 2020.

LI, Z.; YAN, L. Stochastic averaging for two-time-scale stochastic partial differential equations with fractional brownian motion. *Nonlinear Analysis: Hybrid Systems*, Elsevier, v. 31, p. 317–333, 2019.

LIN, C.-J.; LEE, C.-Y.; CHIN, C.-C. Dynamic recurrent wavelet network controllers for nonlinear system control. *Journal of the Chinese Institute of Engineers*, Taylor & Francis, v. 29, n. 4, p. 747–751, 2006.

LIN, C.-M.; TAI, C.-F.; CHUNG, C.-C. Intelligent control system design for uav using a recurrent wavelet neural network. *Neural Computing and Applications*, Springer, v. 24, n. 2, p. 487–496, 2014.

MEDEIROS, A. E. *Métodos bootstrap para séries temporais baseados em wavelets*. Dissertação (Mestrado) — Maringa State University, 2018.

MEDSKER, L.; JAIN, L. C. *Recurrent neural networks: design and applications*. [S.l.]: CRC press, 1999.

MORETTIN, P. A. *Modelos para previsão de séries temporais*. [S.l.]: Instituto de Matemática Pura e Aplicada, 1981. v. 1.

MORETTIN, P. A. *Ondas e Ondaletas Vol. 23*. [S.l.]: Edusp, 1999.

MORLET, J.; ARENS, G.; FOURGEAU, E.; GLARD, D. Wave propagation and sampling theory—part i: Complex signal and scattering in multilayered media. *Geophysics*, Society of Exploration Geophysicists, v. 47, n. 2, p. 203–221, 1982.

NILSSON, N. J. Learning machines. New York, 1965.

PARISI, R.; CLAUDIO, E. D. D.; LUCARELLI, G.; ORLANDI, G. Car plate recognition by neural networks and image processing. In: IEEE. *1998 IEEE International Symposium on Circuits and Systems (ISCAS)*. [S.l.], 1998. v. 3, p. 195–198.

PARK, D. C.; ZHU, Y. Bilinear recurrent neural network. In: IEEE. *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*. [S.l.], 1994. v. 3, p. 1459–1464.

PASQUOTTO, J. Previsão de séries temporais no varejo brasileiro: uma investigação comparativa da aplicação de redes neurais recorrentes de elman. 2010. 191f. *São Paulo*, 2010.

PICARD, D.; TRIBOULEY, K. Adaptive confidence interval for pointwise curve estimation. *The Annals of Statistics*, Institute of Mathematical Statistics, v. 28, n. 1, p. 298–335, 2000.

R Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria, 2020. Disponível em: <https://www.R-project.org/>.

RAMAN, H.; SUNILKUMAR, N. Multivariate modelling of water resources time series using artificial neural networks. *Hydrological Sciences Journal*, Taylor & Francis, v. 40, n. 2, p. 145–163, 1995.

RAO, S. S.; KUMTHEKAR, B. Recurrent wavelet networks. In: IEEE. *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*. [S.l.], 1994. v. 5, p. 3143–3147.

SHARMA, V.; YANG, D.; WALSH, W.; REINDL, T. Short term solar irradiance forecasting using a mixed wavelet neural network. *Renewable Energy*, Elsevier, v. 90, p. 481–492, 2016.

SHI, L.; CHU, L.; CHEN, Y. Dynamic wavelet neural network model for forecasting returns of shfe copper futures price. In: *Proceedings of the 7th International Conference on Digital Enterprise Technology, DET 2011*. [S.l.: s.n.], 2011.

SOARES, A. d. S. *Predição de séries temporais econômicas por meio de redes neurais artificiais e transformada Wavelet: combinando modelo técnico e fundamentalista*. Tese (Doutorado) — Universidade de São Paulo, 2008.

SOUZA, E. M. de. *WNN: Wavelet Neural Network*. [S.l.], 2021. R package version 0.0.0.9000.

STANKOVI, R. S.; BOGDAN, J. F. The haar wavelet its status and achievement. *Computer and Electronic Engineering*, 2003.

SUBASI, A.; YILMAZ, M.; OZCALIK, H. R. Classification of emg signals using wavelet neural network. *Journal of neuroscience methods*, Elsevier, v. 156, n. 1-2, p. 360–367, 2006.

TIAN, Y.; PAN, L. Predicting short-term traffic flow by long short-term memory recurrent neural network. In: IEEE. *2015 IEEE international conference on smart city/SocialCom/SustainCom (SmartCity)*. [S.l.], 2015. p. 153–158.

VIDAKOVIC, B. *Statistical modeling by wavelets*. [S.l.]: John Wiley & Sons, 2009. v. 503.

WANG, P.; LIU, Y. Network traffic prediction based on improved bp wavelet neural network. In: IEEE. *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*. [S.l.], 2008. p. 1–5.

WANG, S.-C. Artificial neural network. In: *Interdisciplinary computing in java programming*. [S.l.]: Springer, 2003. p. 81–100.

WHITE, H. Economic prediction using neural networks: The case of ibm daily stock returns. In: *ICNN*. [S.l.: s.n.], 1988. v. 2, p. 451–458.

WONG, B. K.; SELVI, Y. Neural network applications in finance: A review and analysis of literature (1990–1996). *Information & Management*, Elsevier, v. 34, n. 3, p. 129–139, 1998.

YEGNANARAYANA, B. *Artificial neural networks*. [S.l.]: PHI Learning Pvt. Ltd., 2009.

YOO, S. J.; PARK, J. B.; CHOI, Y. H. Direct adaptive control using self recurrent wavelet neural network via adaptive learning rates for stable path tracking of mobile robots. In: IEEE. *Proceedings of the 2005, American Control Conference, 2005.* [S.l.], 2005. p. 288–293.

YOO, S. J.; PARK, J. B.; CHOI, Y. H. Indirect adaptive control of nonlinear dynamic systems using self recurrent wavelet neural networks via adaptive learning rates. *Information Sciences*, Elsevier, v. 177, n. 15, p. 3074–3098, 2007.

ZEN, H.; SAK, H. Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis. In: IEEE. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.], 2015. p. 4470–4474.

ZHANG, G.; PATUWO, B. E.; HU, M. Y. Forecasting with artificial neural networks:: The state of the art. *International journal of forecasting*, Elsevier, v. 14, n. 1, p. 35–62, 1998.

ZHANG, G. P.; QI, M. Neural network forecasting for seasonal and trend time series. *European journal of operational research*, Elsevier, v. 160, n. 2, p. 501–514, 2005.

ZHANG, K.; CHAI, Y.; FU, X.-a. A network traffic prediction model based on recurrent wavelet neural network. In: IEEE. *Proceedings of 2012 2nd International Conference on Computer Science and Network Technology*. [S.l.], 2012. p. 1630–1633.