CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL INSTITUTO POLITÉCNICO NACIONAL

Departamento de Control Automático

Modelado de Series en Tiempo vía Redes Neuronales Wavelet Haar

Tesis que presenta:

M. en C. Juan Jose Cordova Zamorano[1]

Para obtener el grado de:
Doctor en Ciencias

En la Especialidad de
Control Automático

Director de Tesis:
Dr. Wen Yu Liu

México, D.F., **4 de Mayo del 2012**

[1]Becario del CONACyT

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL INSTITUTO POLITÉCNICO NACIONAL

Departamento de Control Automático

Modelado de Series en Tiempo vía Redes Neuronales Wavelet Haar

Tesis que presenta:

M. en C. Juan Jose Cordova Zamorano[1]

Para obtener el grado de:
Doctor en Ciencias

En la Especialidad de
Control Automático

Director de Tesis:
Dr. Wen Yu Liu

México, D.F., **4 de Mayo del 2012**

[1]Becario del CONACyT

Dedicatoria

Inicié este doctorando junto a la aventura de ser padre, gracias Miriam y gracias Emiliano por tomar el reto junto a mí, apoyarme y entender mis desvelos. Gracias por comprender mi aislamiento y facilitarme las horas de estudio.

Emiliano creciste a la par de esta tesis, afortunadamente alcanzo un meta, sin embargo se que solo es el inicio de nuevos retos, mi vida profesional, que será una nueva aventura y que recorreremos juntos, **Miriam, Emiliano** y coincidencias de la vida, inicié el doctorado con el nacimiento de Emiliano y lo culmino con otro nacimiento, el tuyo **Manzana Cruda**, a ustedes tres les dedico esta tesis.

Agradecimientos

A mis compañeros y amigos del Departamento de Control Automático por su apoyo y consejos

A mi director de tesis, Dr. Wen Yu Liu, por ser mi guía para el desarrollo de este trabajo de tesis, le agradezco su paciencia y tolerancia, que me permitió desarrollarme y descubrir nuevas áreas de conocimiento.

A mis asesores Dr. Ieroham Barouh, Dr, Moises Bonilla, Dr. Julio César Tovar Rodríguez y Dr. Floriberto Ortiz Rodríguez, por sus observaciones que me permitieron hacer un mejor trabajo.

Gracias a CONACYT por apoyarme con una beca, sin la cual no hubiera sido posible mi estadía como estudiante de doctorado.

Al departamento de Control Automático por proporcionarme los medio necesarios para desarrollarme académicamente durante mi estancia.

Índice general

Índice de figuras

Capítulo 1

Introducción

La teoría de aproximación de funciones esta orientada a resolver el problema de como aproximar una función difícil de analizar o representar por otra función mas simple. El problema de aproximación a una cantidad determinada es uno de los problemas más antiguos en las matemáticas, que data desde el descubrimiento de la irracionalidad, con la fórmula para la aproximación de la raíz cuadrada de un número, sin embargo, la teoría de aproximación matemática es una rama relativamente joven.

El primer enfoque a la definición de función basada en dependencia y hacer abstracción de fórmulas fue desarrollado por Euler, las fórmulas fueron desarrolladas para ayudar en la aproximación de funciones. En principio, estas representaciones se basaron en la fórmula de Taylor y algunos de interpolación basadas en las ideas de Newton. Aunque estas formulaciones dieron buenas aproximaciones en determinados casos especiales, en general, no pudo controlar el error de aproximación, donde la aproximación no ocurría de manera uniforme el error crecía.

Nuevas ideas surguieron en la materia, Laplace, Fourier, Chebyshev, Markov, Legendre dieron grandes aportes que sentaron la base de la teoría actual. La teoría moderna de aproximación ha sido un campo de mucho estudio, principalmente debido a la gran cantidad de aplicaciones y a la necesidad de aproximar funciones en tiempo continuo con funciones que utilizan un numero finito de parámetros, esta última esta relacionada con la forma en que

las computadoras digitales aproximan a funciones.

En esta teoría hay dos medidas importantes a estudiar entre la función original y la función aproximación:

- Que tanto es mas simple la función aproximación con respecto a la función original.
- Y que tanto es el error de aproximación entre las dos funciones.

Al día de hoy se pueden encontrar numerosos trabajo para aproximar funciones, en donde se elige el método de aproximación dependiendo a la clase que pertenezca la función a aproximar, por ejemplo, los polinomios trigonométricos para los problemas periódicos, polinomios de Legendre y polinomios de Chebyshev para problemas no periódicos.

Por otro lado, el análisis de series de tiempo se ha convertido en un área muy activa de investigación debido a sus múltiples aplicaciones, como lo es la predicción de eventos futuros en finanzas y al modelado e identificación de parámetros de sistemas dinámicos, en el modelado de sistemas dinámicos vía series de tiempo se encuentran varias técnicas como lo es representar los sistemas dinámicos en modelos como ARMA, ARIMA, ARMAX, [77], en términos prácticos ha sido muy utilizado el modelado vía series de tiempo de un sistema, debido a que las herramientas con las que se monitorean las plantas entregan datos digitales, o en caso contrarios se digitalizan para poder procesarse con una computadora. Sin embargo una planta escrita en serie de tiempo no resulta fácil de analizar, existen técnicas heurísticas para determinar la función de transferencias de una planta a partir de la serie de tiempo de su entrada y de su salida, por lo regular se considera el modelo ARMA como una representación de la función de transferencia y un modelo ARMAX como un modelo donde existe control, una alternativa ha sido representar las series de tiempo vía funciones ortogonales y con ellas hacer el modelado de un sistema dinámico [78]. Dos técnicas con funciones ortogonales son el uso de series de Fourier y de funciones Wavelets.

En la década de los ochentas se formalizo una nueva teoría en aproximación de funciones, el análisis Wavelet, que consiste en cortar una función en sus diferentes componentes de frecuencia y después cada componente de frecuencia estudiarla a distintas resoluciones, esta partición en frecuencia se realiza usando como ventana una función tipo wavelet, las wavelets

son un tipo de función con requerimientos especiales, que son usadas para aproximar otras funciones, la idea de aproximar una función mediante la super posición de otra no es nueva y se remontan a los estudios de Fourier.

La primera mención al análisis wavelets se remonta a 1910 cuando en su tesis doctoral Haar formulo la pregunta, de si existía algún otro sistema ortogonal $h_0, h_1, ..., hn, ...$ de funciones definidas en $[0, 1]$, tal que para cualquier función continua f definida en $[0, 1]$, la serie

$$< f, h_0 > h_0(x) + < f, h_1 > h_1(x) + ... + < f, h_n > h_n(x) + ...$$

converge a $f(x)$ uniformemente en $[0, 1]$.

La teoría de análisis utilizando funciones wavelets apareció como una alternativa al análisis de Fourier, dado que no requiere que las funciones a analizar sean periódicas [2], y además no se presenta en las discontinuidades la presencia del fenómeno de Gibbs [15]. La teoría Wavelet tiene mas de 40 años desarrollándose, se puede consultar una interesante reseña histórica del desarrollo de la teoría wavelet en [6].

$$\psi_{(m,n)}(t) = 2^{m/2}\psi\left(2^m t - n\right); m, n \in \mathbb{Z}$$

El análisis por ventanas de Fourier y el análisis wavelet tienen en esencia la misma forma de realizar el análisis de una función, que equivale a que una función ventana recorre a la función original y se calculan todas las correlaciones entre estas funcones y el tiempo-frecuencia que se están utilizando. Pero a diferencia del análisis de Fourier que tiene como objetivo medir la frecuencia contenida en una función, en el caso del análisis utilizando funciones wavelets se busca comparar múltiples magnitudes de una función con distintas resoluciones, el análisis de Fourier por ventanas se construye a través de senos y cósenos multiplicados por una ventana deslizante, en el análisis con funciones wavelets la ventana se encuentra oscilando y es llamada wavelet madre. Esta ventana wavelet madre ya no se multiplica por senos y cósenos, en lugar de ello se traslada y se ensancha, por traslaciones y ensanchamientos arbitrarios. Esa es la manera en que la wavelet madre forma nuevas wavelets las cuales son los bloques que construyen el análisis wavelet.

El interés por el estudio del análisis wavelet ha sido explosivo tanto para científicos como ingenieros, debido ha que es una herramienta matemática simple con una gran variedad de aplicaciones. El análisis wavelet ha sido utilizado para resolver problemas de análisis numérico [11], así como para resolver ecuaciones diferenciales ordinarias [4], [10].

En ingeniería ha sido ampliamente aceptada la teoría wavelet y una de sus mayores aplicaciones se encuentra en el tratamiento y análisis de señales, siendo un ejemplo de implementación el desarrollo de algoritmos para la compresión de imágenes como lo son los formatos MPEG y JPEG [3], en el área de control automático se han utilizado las propiedades de aproximar funciones para realizar identificación de parámetros de sistemas no lineales, [12],[13], también la transformada wavelet se ha utilizado como filtro para sistemas robustos para detección de fallas [14].

En la literatura sobre redes neuronales se puede encontrar algunos trabajos en donde se exponen las bondades y aplicaciones de las redes neuronales wavelets [12],[30], sin embargo, estas redes utilizan funciones wavelets continuas como la Morlet y Sombrero Mexicano, y no se ha puesto suficiente atención a las funciones wavelets Haar, las cuales son matemáticamente las mas simples, la principal razón por la que no se ha considerado a la wavelet Haar, se refiere a que es una función discontinua, sin embargo su aplicación en soluciones numéricas de ecuaciones diferencias han sido ampliamente documentadas, [26],[27].

La wavelet Haar es utilizada en el algoritmo de transformación rápida wavelet (FWT) [8], dicho algoritmo es muy difundido en el tratamiento de imágenes, [9], además se encuentran diversos trabajos que hacen la comparación entre las distintas bases wavelets, [51], [52], y muestran que el desempeño de la wavelet Haar es similar a otras bases y presenta además la ventaja de ser más sencilla de implementar.

1.1. Motivación

En el campo del control automático, las redes neuronales artificiales se utilizan principalmente como modelos para la identificación y el control de sistemas, [40],[22], [23]. La identificación de sistemas consiste crear un modelo matemático de un sistema dinámico basándose

en la observación del comportamiento del sistema [41], las redes neuronales artificiales tienen una aplicación directa en el campo de identificación de sistemas debido a su capacidad de ajustar los parámetros de un modelo propuesto, de tal forma que su comportamiento se aproxime al sistema o planta a ser estudiado.

Podríamos resumir en cuatro grandes áreas los problemas relacionados a la teoría de sistemas:

1. Modelado.
2. Análisis
3. Estimación
4. Control

La obtención de un modelo matemático más o menos preciso del sistema es fundamental ya que la mayoría de los métodos de diseño de controladores parten de la hipótesis de que un modelo parametrizado del proceso está disponible, la identificación de sistemas se puede definir como el área de la teoría de sistemas que estudia metodologías para la obtención de modelos matemáticos de sistemas dinámicos, a partir de datos de medición de las entradas y salidas del sistema.

La identificación de sistemas utilizando redes neuronales se puede clasificar en dos grupos:

- Identificación no paramétrica Input-Output.
- Representación Paramétrica usando redes neuronales recurrentes.

Las Redes Neuronales Wavelets han sido utilizadas para realizar los dos tipos de identificación y su efectividad ha sido probada, [36],[37], sin embargo se han utilizado funciones wavelets continuas, como lo es la wavelet Morlet y Sombrero Mexicano, en las cuales la determinación del coeficiente de ensanchamiento m, no tiene un procedimiento analítico para su determinación, sin embargo el algorito computacional para realizar la transformada wavelet

utiliza la función wavelet Haar y otorga un procedimiento para determinar el ensanchamiento de la función wavelet.

La función wavelet Haar es discontinua, sin embargo existen diversos trabajos donde se utilizan para obtener soluciones numéricas de ecuaciones diferencias y han sido ampliamente documentadas, [26],[27]. La wavelet Haar es utilizada en el algoritmo de transformación rápida wavelet (FWT) [8], y dicho algoritmo es ampliamente utilizado en el tratamiento de imágenes, [9].

1.2. Objetivos

1. Esta tesis pretende mostrar la efectividad de la red neuronal wavelet Haar para aproximar series de tiempo y funciones no lineales.

2. Presentar un algoritmo para optimizar la estructura de la red neuronal wavelet Haar.

3. Mostrar la función wavelet Haar en una estructura de red recurrente.

4. Probar la estabilidad en el aprendizaje de la Red Neuronal Wavelet Haar

5. Mostrar la efectividad de la red neuronal wavelet Haar para realizar identificación no paramétrica y paramétrica.

6. Presentar una estructura neuro difusa wavelet Haar.

1.3. Estructura

El análisis wavelet se puede ver como una variación del análisis de Fourier, que presenta mejor desempeño en los casos donde el análisis de Fourier tiene limitantes, en el Capítulo 2, se describen las propiedades del análisis de Fourier así como las limitantes que tiene, en el Capítulo 3 se estudia el análisis Wavelet y en particular se muestran las características de la wavelet Haar, así como se describe el algoritmo de la transformada wavelet Haar.

En el Capítulo 4 se presenta la estructura de las redes wavelets Haar Feedforward así como el algoritmo de aprendizaje y la prueba de estabilidad para ese algoritmo. En este capítulo se encuentra la propuesta para un algoritmo que ayuda a determinar la estructura óptima de la red neuronal wavelet Haar. Además se muestran ejemplos de aproximación a distintas series de tiempo y se muestra el efecto que produce el aumentar el número de neuronas y la resolución de la función wavelet Haar.

El Capítulo 5 muestra la estructura de la red neuronal wavelet Haar recurrente, y se propone como algoritmo de aprendizaje y ajuste de pesos utilizando el filtro de Kalman, además se hace una comparación con algoritmo de aprendizaje vía Gradiente descendente, se presenta la prueba de estabilidad de esos algoritmos. En el Capítulo 6 se presenta una estructura neuro difusa wavelet Haar, donde se muestran tres distintas situaciones de aprendizaje, conociendo la función de membresía y sin conocerla, y se muestra un ejemplo. Finalmente el Capítulo 7 presenta las conclusiones y el trabajo futuro.

1.4. Publicaciones

Revista

1. Juan Jose Cordova, Wen Yu, Two types of Haar wavelet neural networks for nonlinear system identification, *Neural Processing Letters,* DOI 10.1007/s11063-012-9218-0, 2012

Capitulo de libro

1. Juan Jose Cordova, Wen Yu, Wavelet Neural Networks for Nonlinear System Identification, *Recent Advances in Dynamics and Control of Neural Networks*, Eds. E.Kaslik, S.Sivasundaram, Vol.6, Cambridge Scientific Publishers, 2012

Congreso

1. Juan Jose Cordova and Wen Yu, Stable Fourier Neural Networks with Application to Modeling Lettuce Growth, *2009 International Joint Conference on Neural Networks, IJCNN'09*, Atlanta, USA, 591-596, 2009

2. Cruz Vega Israel, Wen Yu, Juan Jose Cordova, Multiple Fuzzy Neural Networks Modeling with Sparse Data, *2010 IEEE International Conference on Fuzzy Systems, Fuzzy'10*, Bacelona, Spain, 2010

3. Juan Jose Cordova, Wen Yu, Recurrent Wavelets Neural Networks Learning via Dead Zone Kalman Filter, *2010 International Joint Conference on Neural Networks, IJCNN'10*, Bacelona, Spain, 2010

4. Juan Jose Cordova, Wen Yu, Optimization of Fixed Wavelet Neural Networks, *2011 International Joint Conference on Neural Networks, IJCNN'11*, San Jose, USA, 2011

Capítulo 2

Preliminares

2.1. Análisis de Fourier

En el mundo real, muchos fenómenos físicos pueden describirse mediante funciones periódicas, Joseph Fourier fue el primero con la idea de representar funciones en el dominio de la frecuencia y genero muchos cambio en la matemática de su época, creó una nueva área de estudio llamada Análisis de Fourier. En ingeniería esta herramienta ha sido muy útil al ser las series de Fourier un método para resolver e interpretar sistemas dinámicos. El análisis de Fourier es un área muy extensa de la matemática clásica en la que teniendo una función f de periodo 2π, definida en $\mathbb{R}$

$$f(t+2\pi) = f(t), t \in \mathbb{R}, \tag{2.1}$$

periódica, se asume que f pertenece al espacio vectorial $L^2(-\pi, \pi)$

$$L^2(-\pi,\pi) := \left\{ f : \int_{-\pi}^{\pi} |f(t)|^2 \, dt < \infty \right\}. \tag{2.2}$$

Para tal función existe una familia de funciones $\varphi_0, \varphi_1, \varphi_2$, tal que puede expandir f

$$f(t) = \sum_{n=0}^{\infty} c_n \varphi_n \tag{2.3}$$

para algún coeficiente c_n, donde φ_n corresponde a una función polinomial o como es el caso de la serie de Fourier una función trigonometrica. De las funciones trigonométricas de periodo

2π,

$$\{1, \sin t, \cos t, \sin 2t, \cos 2t, ...\},$$

donde la combinación lineal de estas funciones representan funciones de período 2π, satisfacen la propiedad de ortogonalidad y las siguientes propiedades:

$$\begin{gathered}\int_{-\pi}^{\pi} \sin nt \cos mt \ dt = 0, \quad m \in \mathbb{Z}^+, \\ \frac{1}{\pi}\int_{-\pi}^{\pi} \sin nt \sin kt \ dt = \delta_{n,k}, \quad n, k \in \mathbb{N}^2 \\ \frac{1}{\pi}\int_{-\pi}^{\pi} \cos mt \cos lt \ dt = \delta_{m,l}, \quad m, l \in (\mathbb{Z}^+ \ x \ \mathbb{Z}^+) \backslash \{(0,0)\}\end{gathered}$$

donde $\delta_{n,k}$ es la delta de Kronecker:

$$\delta_{n,k} = \begin{cases} 1, & if \ n = k \\ 0, & \text{otro caso} \end{cases}$$

Esta propiedad de ortogonalidad nos permite representar una función periódica como una serie trigonométrica, como sigue;

$$f(x) = \frac{1}{2}a_0 + \sum_{k=1}^{\infty}(a_k \cos kx + b_k \sin kx), \tag{2.4}$$

donde la serie es uniformemente convergente en $[-\pi, \pi]$, entonces los coeficientes a_0, a_k y b_k $(k \geq 1)$ de la serie de Fourier (2.4) se calculan como:

$$\begin{gathered}a_0 = \frac{1}{2\pi}\int_0^{2\pi} f(t)dx \\ a_k = \frac{1}{\pi}\int_0^{2\pi} f(t)\cos(kt)dx, \ k = 1, 2, .. \\ b_k = \frac{1}{\pi}\int_0^{2\pi} f(t)\sin(kt)dx, \ k = 0, 1, 2, ...\end{gathered} \tag{2.5}$$

De la serie de Fourier 2.4, es posible escribir la serie trunca a N coeficientes como:

$$S_N(t) = \frac{1}{2}a_0 + \sum_{k=1}^{N}(a_k \cos kt + b_k \sin kt), \tag{2.6}$$

donde la desviación entre la función $f(t)$ y la serie trunca $S_N(t)$ se puede estimar por:

$$|f(x) - S_N(x)| \leq \frac{1}{\sqrt{N}}\frac{1}{\sqrt{\pi}}\sqrt{\int_{-\pi}^{\pi} |f'(t)| \, dt}. \tag{2.7}$$

Como se sabe, las Series de Fourier son una herramienta sumamente útil para el estudio de funciones periódicas, para el análisis de funciones que no son periódicas la herramienta clásica es la transformada de Fourier, sin embargo como veremos mas adelante la transformación wavelet se ha presentado como una herramienta con mayor capacidad de análisis. Cuando expandimos una función de período 2π en termino de funciones trigonométricas cosnt, sinnt, estas tienen período 2π, funciones que no tienen ese periodo no aparecen en la serie de Fourier, una serie de Fourier es la descomposición de una función f, en oscilaciones armónicas con frecuencia $\frac{n}{2\pi}$, $n = 0, 1, ..;$ donde la contribución de cada frecuencia esta dado por los coeficientes de Fourier. En una función no periódica todas las frecuencias pueden aparecer, y la suma discreta sobre el espacio de frecuencias $\frac{k}{2\pi}$ puede reemplazarse por una integral sobre todas las frecuencias. Se considera la función f en el espacio vectorial

$$L^1(\mathbb{R}) := \left\{ f : \mathbb{R} \to \mathbb{C} \,\middle|\, \int_{-\infty}^{\infty} |f(t)|\, dt < \infty \right\}.$$

La transformación de una función $f(t)$ se produce al multiplicarla por cierta funcion de análisis e integrarlas en el dominio del tiempo, como sigue:

$$f(t) = \int_{-\infty}^{\infty} f(u)\overline{g(u)}du. \tag{2.8}$$

La función de análisis caracteriza la transformación elegida, en el caso de la transformación de Fourier, la función de análisis se escribe como:

$$g_\omega(s) = e^{it\omega}$$

Para funciones $f \in L^1(\mathbb{R})$, la transformada de Fourier está definida como sigue:

$$\mathcal{F}(\omega) = \int_{-\pi}^{\pi} f(t)e^{-2\pi it\omega}dt \tag{2.9}$$

En la Figura 2.1 se muestran la funciones f y la parte real de $e^{-2\pi it\omega}$, necesarias para el cálculo de $\mathcal{F}(\omega)$.

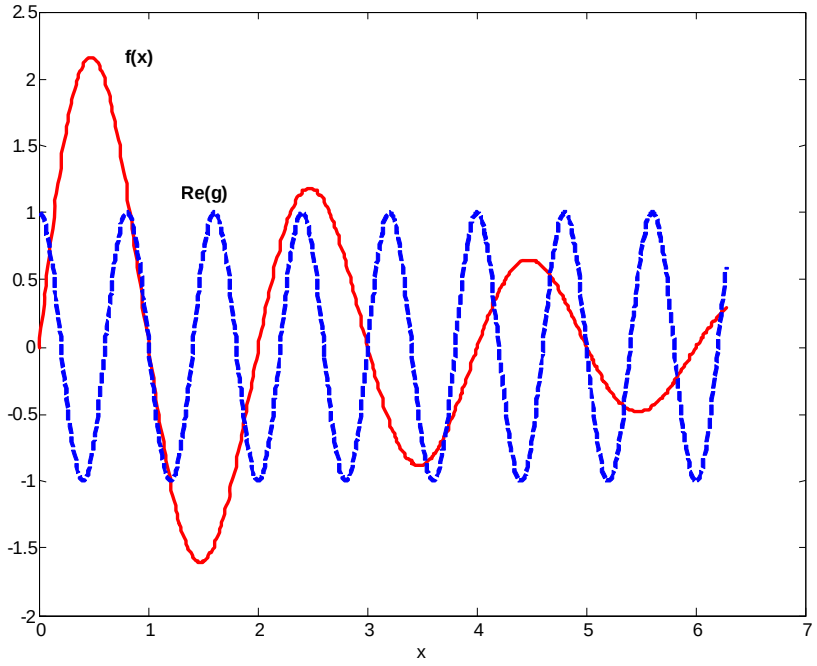


Figura 2.1: Función f y la parte real de $e^{-2\pi i\omega}$.

La teoría de Fourier señala que es posible recuperar la función original, es decir pasar del dominio de la frecuencia al dominio del tiempo,

$$f(t) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \mathcal{F}(\omega)e^{-2\pi it\omega}d\omega, \tag{2.10}$$

donde un proceso temporal puede considerarse como la superposición integral de una colección de ondas que oscilan con amplitud constante. Un tipo de representación tiempo-frecuencia que ha sido ampliamente estudiado se basa en el empleo de ventanas temporales. La ventana w selecciona una porción de la función f y permite aplicar localmente la transformada de Fourier, desplazando temporalmente la ventana w se cubre el dominio de función f obteniendo la completa información tiempo-frecuencia de la misma, de esta forma se formula la

Transformada de Fourier por Ventanas, (Windowed Fourier Transform, WFT).

$$\mathcal{F}w(t,\omega) = \int f(t)w(s-t)e^{-it\omega}dt \tag{2.11}$$

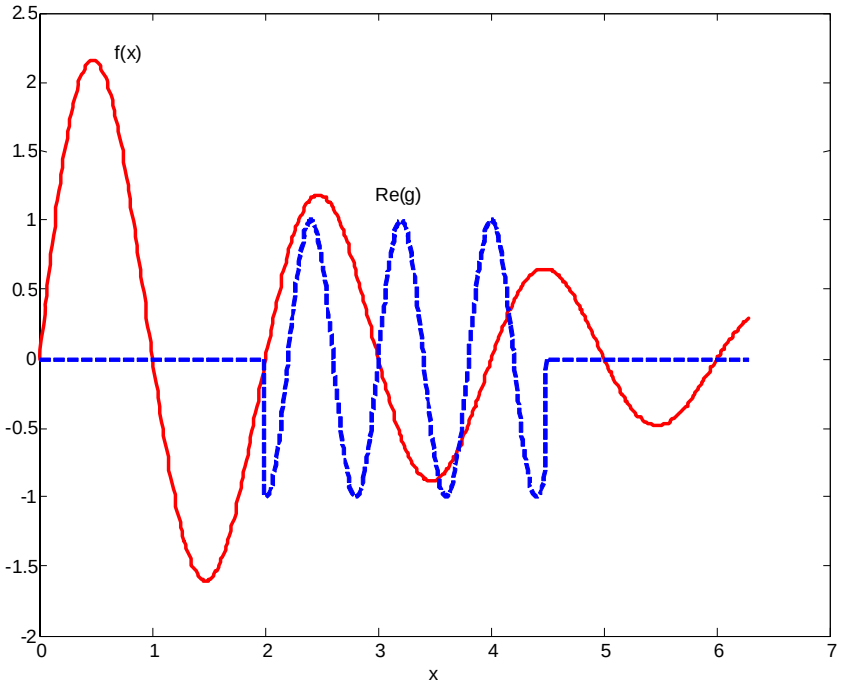


Figura 2.2: Ventana $w(s-t)$.

En la Figura 2.2 observamos la funciones f y la parte real de $e^{-it\omega}$ que se encuentra multiplicada por la ventana $w(s-t)$. La WFT puede entenderse como un tratamiento localizado de la función f mediante filtros (ventanas) deslizantes, de ancho constante. Es evidente que el ancho de la ventana determina las propiedades de la transformación y que presenta dificultades para caracterizar una función con distintas frecuencias. Las series de Fourier son capaces de aproximar una función periódica $f \in L^2(\mathbb{R})$, utilizando una base ortonormal $\{e^{-it\omega}\}$, cada elemento de esta base es una onda senoidal, que es global en

dominio x, sin embargo los coeficientes de la transformada de Fourier de una función no proveen un comportamiento local de la función,situación que se hace evidente y relevante cuando existen discontinuidades, al tratar funciones no continuas el fenómeno de Gibbs que aparece como se muestra en la Figura 2.3.

A estas limitantes surge la alternativa de utilizar ventanas moduladas, pero de dimensión variable, ajustada a la frecuencia de oscilación, que mantenga un mismo número de oscilaciones en el dominio de la ventana. Esto sugiere, contar con una única ventana modulada y generar una completa familia de funciones elementales mediante sus dilataciones o contracciones y traslaciones en el tiempo, lo que da origen a la teoría de análisis utilizando funciones wavelets, las cuales tienen la propiedades de modular su ensanchamiento y traslación.

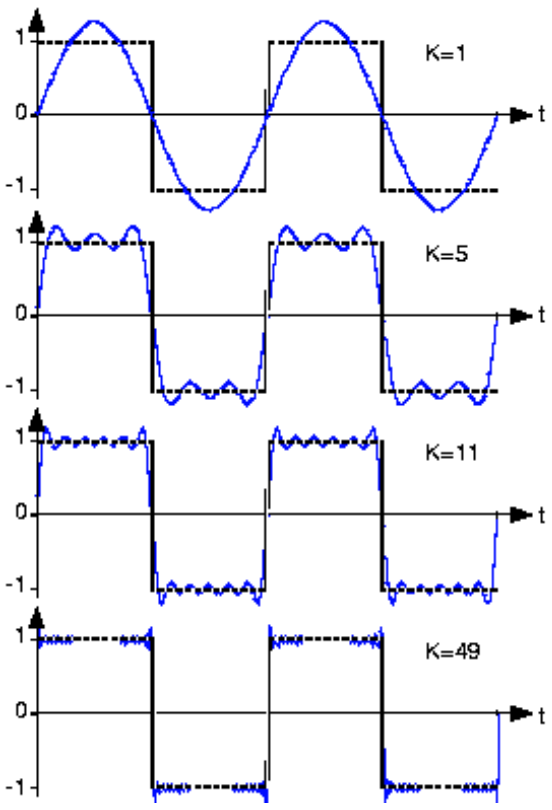


Figura 2.3: Fenómeno de Gibbs.

2.2. Redes Neuronales Artificiales para Modelado de Series en Tiempo

Las Redes Neuronales Artificiales (RNA) son inspiradas en la organización de las neuronas biológicas, las cuales constituyen los bloques elementales que conforman el sistema nervioso y el cerebro humano [21], y este tipo de célula especializada se comunica con otras células mediante impulsos eléctricos a través de una membrana llamada sinapis, las neuronas están constituidas por tres regiones: Cuerpo celular - Soma, Dentrita, Axón. Las redes neuronales artificiales, son un modelo simplificado de las redes neuronales biológicas, y asemejan su estructura, las RNA se pueden clasificar dentro del grupo de sistemas inteligentes, entre los que se encuentran: sistemas adaptables, difusos, genéticos y todos aquellos que tratan de modelar el conocimiento y el aprendizaje. La calificación de inteligentes se debe a que los sistemas mencionados anteriormente tienen la capacidad de adaptarse a su medio ó aprender de él de forma autónoma. Las redes neuronales artificiales se definen como:

Definición 2.1 *[20]. Una Red Neuronal Artificial es una gran cantidad de procesadores paralelos distribuidos, formados por una única unidad de procesamiento, la cual tiene la cualidad natural de almacenar el conocimiento experimental y tener ese conocimiento disponible para usar.*

Las redes neuronales tiene un arreglo de elementos básicos de procesamiento con capacidad de entrenamiento, este entrenamiento consiste en el ajuste de algunos parámetros con el fin de que la red asimile con algún grado de precisión, la relación deseada entre las variables de entrada y de salida de la red neuronal.

El conocimiento adquirido por la red neuronal artificial proviene del medio ambiente a través de un proceso de aprendizaje, donde una conexión interna de pesos, conocidos como pesos sinapticos, son usados para almacenar el conocimiento adquirido. En modelo matemático de una neurona artificial se pueden observar tres elementos básicos:

1. Un conjunto de sinapsis o ligas, cada una caracterizada por un peso. Se tiene x_j que

es la entrada de la sinapsis j que esta conectada a la neurona k, x_j es multiplicado por el peso sinaptico w_{kj}.

2. Combinación lineal, es decir la suma de las señales de entrada multiplicadas por los pesos de las respectivas sinapsis.

3. Función de Activación para limitar la salida de la neuronal.

El tipo de estructura mas simple que existe en las redes neuronales es la llamada Feed-Forward, en la cual la salida de una red neuronal depende únicamente de sus entradas y no existe conexión de la salida de las neuronas con su entrada, el flujo de la información es hacia un solo sentido y las neuronas se actualizan siempre al mismo tiempo y no depende de señales en instantes anteriores de tiempo se dice que es una red neuronal estática porque la respuesta de la red neuronal es invariante en el tiempo. Existen tres tipos de redes neuronales estáticas: las redes de una sola capa, las redes neuronales múlti-capa y las redes neuronales de funciones radiales básicas. Las redes neuronales estáticas son muy útiles en los problemas de clasificación de patrones y aproximación de funciones porque construyen funciones no lineales entre el espacio de entrada al espacio de salida de las variables involucradas. Una red neuronal con conexiones hacia adelante con una capa oculta no lineal y una capa de salida lineal puede aproximar cualquier función con el grado de precisión que se desee. Es la forma mas simple de redes neuronales feedforward, tiene solo una capa de neuronas, el perceptrón es la red neuronal de una capa mas difundida, el cual consiste en una sola neurona.

Usualmente el umbral (w_0) es tratado como un peso sinaptico conectado al valor fijo -1, la salida del perceptrón es calculado como

$$y(k) = \varphi(\sum_{j=1}^{n} w_j x_j u + w_0), \tag{2.12}$$

donde w es el vector de los pesos sinapticos, para cada k, en el espacio $n-$dimensional, la ecuación $w^T u$, con coordenadas

$$u_1, u_2, .., u_n$$

Este algoritmo esta basado en el método de gradiente descendente, resultando la ley que actualiza los pesos de la red de la forma:

$$w(k+1) = w(k) + \eta e(k)u(k). \tag{2.13}$$

Las redes neuronales multicapa tienen tres características que las distinguen:

1. La función de activación de cada neurona es suave al contrario de las funciones usadas en las redes de una sola capa. Usualmente la función usada es el sigmoide el cual esta definido como:

$$\varphi_i(\nu_i) = 1/(1 + e^{-\nu_i})$$

2. La red neuronal tiene una o mas capas de neuronas ocultas.

3. Las redes tienen un alto grado de conectividad.

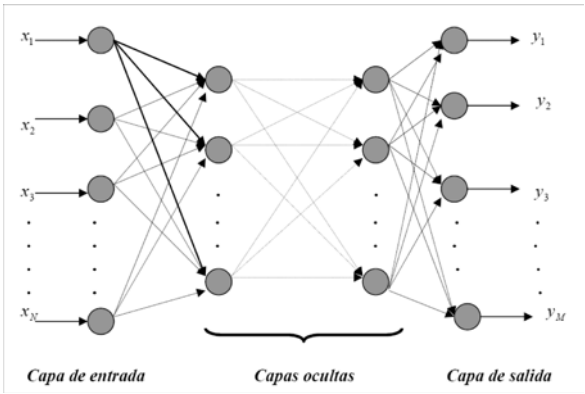


Figura 2.4: Red Neuronal Multi Capa

El algoritmo de aprendizaje utilizado para ajustar los pesos sinapticos de el perceptrón multi capa es el conocido como Back Propagation. El error a la salida de la neurona j, elemento de la capa de salida esta dada como:

$$e_j(k) = d_j(k) - y_j(k), \tag{2.14}$$

donde:

d_j es la salida deseada.

y_j es la salida de la red neuronal

k indica el elemento del arreglo de entrada.

El error cuadratico medio instantáneo se escribe como:

$$\mathcal{E}(k) = \frac{1}{2}\sum_{j=1}^{l} e_j(k) \tag{2.15}$$

donde l es el numero de neuronas de la capa de salida. El promedio del error es obtenido de la suma de $\mathcal{E}(k)$ y normalizado con respecto al el número de épocas.

$$\mathcal{E}_{av} = \frac{1}{N}\sum_{k=1}^{N} \mathcal{E}(k), \tag{2.16}$$

usando N ejemplos, que forman una epoca.$\mathcal{E}_{av}$ representa la función de costo a minimizar en el aprendizaje, el objetivo del proceso de aprendizaje es ajustar los parámetros libres de la red, para minimizar el valor de $\mathcal{E}_{av}$. Para hacer eso el algoritmo back propagation aplica una corrección $\Delta w_{ji}(k)$ al peso sinaptico $w_{ji}(k)$, el cual es proporcional al gradiente $\partial\mathcal{E}(k)/\partial w_{ij}(k)$. El calculo de ese gradiente tiene la forma:

$$\frac{\partial\mathcal{E}(k)}{\partial w_{ij}(k)} = \frac{\partial\mathcal{E}(k)}{\partial e_j(k)}\frac{\partial e_j(k)}{\partial y_j(k)}\frac{\partial y_j(k)}{\partial v_j(k)}\frac{\partial v_j(k)}{\partial w_{ji}(k)} \tag{2.17}$$

Diferenciando la ecuación (2.17), tenemos

$$\frac{\partial\mathcal{E}(k)}{\partial w_{ij}(k)} = -e_j(k)\varphi_j'(v_j(k))y_i(k) \tag{2.18}$$

El factor de corrección $\Delta w_{ji}(k)$ aplicado al peso sinapticos $w_{ji}(k)$ se define por la regla delta:

$$\Delta w_{ji}(k) = -\delta\frac{\partial\mathcal{E}(k)}{\partial w_{ij}(k)} \tag{2.19}$$

Un inconveniente de las redes neuronales MLP es que su entrenamiento es lento, la minimización del índice del error cuadratico de salida requiere de comparar en varias ocasiones el conjunto de datos de entrenamiento con la respuesta de la red neuronal. Las redes neuronales con funciones de base radial (RBR) son una alternativa a las redes neuronales MLP, en el contexto de que las RBF las capas ocultas están conformadas por un conjunto de funciones que constituyen una base para el problema de clasificación. La justificación matemática la establece el teorema de Cover (Cover, 1965), se basa en que un problema de clasificación es más probable que sea linealmente separable si se transforma en otro de dimensión mayor.

Las funciones de base radial fueron introducidas primero para la solución de problemas de interpolación multivariable. El trabajo pionero en este tema fue Powell, 1985. Broomhead y Lowe en 1988, exploraron por primera vez el uso de las redes neuronales con funciones de base radial para poder realizar una clasificación no lineal. A diferencia de la disposición que se tiene en las funciones de activación de la red MLP que permite construir modelos de entrenamiento mediante el algoritmo de back-propagation, las nuevas redes con funciones de base radial construyen sus modelos con funciones de activación que son diferente tanto en la capa oculta como en la capa de salida, esto es, una red RBR está diseñada con neuronas en la capa oculta activadas mediante funciones radiales de carácter no lineal con sus centros propios y en la capa de salida mediante funciones lineales.

La estructura de las redes de base radial presenta tres capas bien definidas:

1. La capa de nodos de entrada, completamente conectadas a las neuronas de la capa oculta.

2. La capa oculta de neuronas que proveen una transformación no lineal activada por las funciones de base radial.

3. La capa de salida, también completamente interconectada a la capa oculta y activada a través de una función lineal continua.

La construcción de una red RBR requiere de una mayor cantidad de neuronas en los nodos ocultos que en las redes MLP. Aunque las redes RBR no son comúnmente utilizadas

en aplicaciones que impliquen un alto volumen de patrones de entrenamiento, se le reconoce como una red con una alta eficiencia en la fase de entrenamiento. El entrenamiento a diferencia de la red MLP usando el algoritmo de aprendizaje de retro-propagación, es solamente hacia adelante, de este modo la salida de una red RBR en general, está influenciada por una transformación no lineal originada en la capa oculta a través de la función de base radial y una lineal en la capa de salida a través de la función lineal continúa. En la Figura 2.5 se presenta una red RBR, donde si, son las entradas a la red i = 1,2,...,n; la salida está dada por $y = w(s)$; en donde las i son las funciones de base radial, para este caso son funciones gaussianas. Si la red neuronal presenta n neuronas en la capa oculta, entonces la salida de la red neuronal de base radial se expresa de la forma:

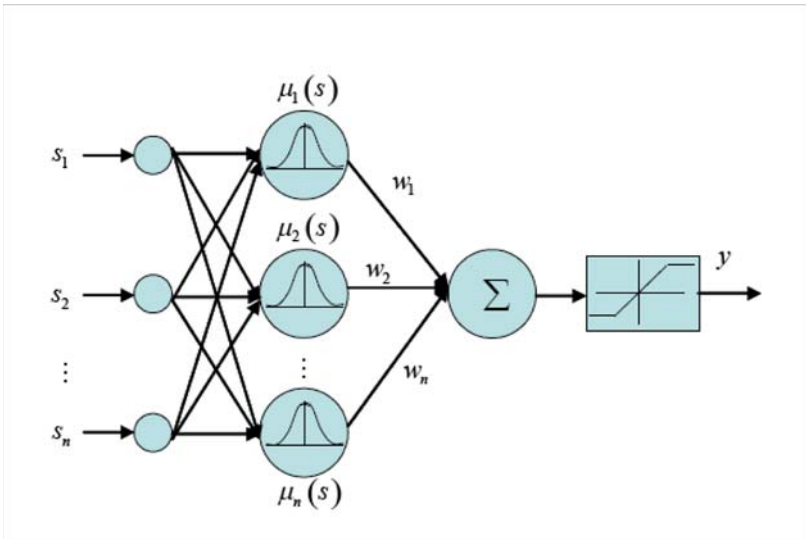


Figura 2.5: Estructura de una Red Neuronal Radial Básica.

En las RBR tienen la forma

$$F(\mathbf{x}) = \sum_{i=1}^{N} w_i \varphi \left(\|\mathbf{x} - \mathbf{x}_i\| \right) \tag{2.20}$$

donde $\{\varphi(\|\mathbf{x}-\mathbf{x}_i\|)\,|i=1,2,...,N\}$ es un conjunto de N funciones arbitrarias generalmente no lineales, conocidas como funciones radiales básicas, y $\|\cdot\|$ denota la norma que usualmente es la Euclideana, los datos conocidos $\mathbf{x}_i \in \mathbb{R}^{m_0}, i=1,2,...,N$ son tomadas para ser los centros de las funciones radiales básicas.

Algunas de las funciones radiales básicas mas utilizadas son:

Multi cuadratica: $\varphi(r)=(r^2+c^2)^{1/2}$, para algún $c>0$ y $r\in\mathbb{R}$

Multi cuadratica Inversa: $\varphi(r)=\frac{1}{(r^2+c^2)^{1/2}}$, para algún $c>0$ y $r\in\mathbb{R}$

Gausiana: $\varphi(r)=\exp\left(-\frac{r^2}{2\sigma^2}\right)$, para algún $\sigma>0$ y $r\in\mathbb{R}$

En general una red RBR tiene un mejor desempeño con un mayor volumen de datos de entrenamiento que su contraparte la red MLP, presentan una arquitectura simplificada con una capa oculta, su entrenamiento es rápido y se puede realizar una combinación de diferentes paradigma de aprendizaje.

Un problema en las Redes Neuronales Estáticas es el almacenar información termporalmente, esto se realiza incluyendo retrasos en las entradas y las salidas, sin embargo, es una representación limitada, ya que solo puede almacenar un numero finito de entradas previas, una alternativa ha sido utilizar Redes Neuronales en las cuales en su estructura existe al menos una retroalimentación. Para este tipo de redes neuronales retroalimentada en tiempo continuo se usa usualmente la terminología como Redes Neuronales Diferenciales (DNN), y para tiempo discreto utilizamos el nombre de Redes Neuronales Recurrentes (RNN). Algunas de las tareas para este tipo de redes son: La predicción de series, la identificación y el control de sistemas dinámicos. La Figura 2.6 muestra las diferentes arquitecturas de las redes dinámicas.

Una de las primeras estructuras de una red neuronal retroalimentada fue propuesta por Hopfield y usa su red para describir redes de circuitos eléctricos, ver [20]. Existen dos versiones de la red neuronal de Hopfield:

La red neuronal discreta de Hopfield la cual se representa por el modelo matemático siguiente

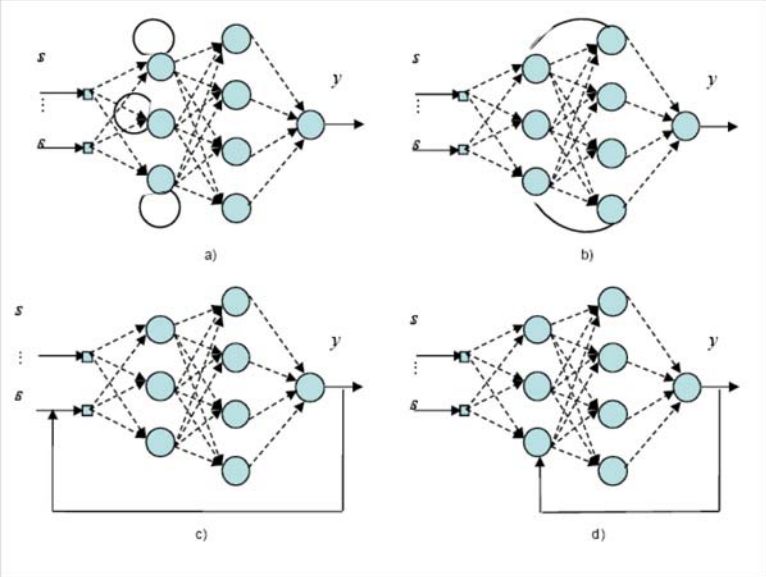


Figura 2.6: Diferentes Redes Neuronales Dinámicas.

$$x_i(k+1) = sgn\left[\sum_{j=1}^{n} w_{ij}x_{ij}(k) + u_i - \rho_i\right], \tag{2.21}$$

donde:

x_i es el estado de la $i-esima$ neurona,

n es el número de neuronas,

u_i es la entrada a la $i-esima$ neurona,

ρ_i es el umbral de la $i-esima$ neurona.

w_{ik} es el peso sinaptico que conecta la neurona j a la neurona i.

En tiempo continuo la Red Hoplfiel puede describir una red de circuito eléctricos,RC, conectados con amplificadores. El $i-esimo$ amplificador es caracterizado por las funciones de entrada salida: $x_i = \varphi_i(v_i)$, con x_i como la entrada y v_i la salida de voltaje.

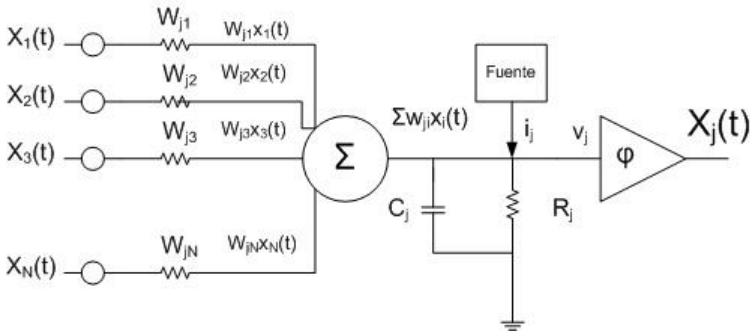


Figura 2.7: Circuito Eléctrico RC y un Amplificador.

De las leyes de corriente de Kirchoff se obtiene:

$$C_j \frac{dv_j(t)}{dt} = -\frac{v_j(t)}{R_j} + \sum_{i=1}^{N} w_{ji} x_i(t) + I_j, \quad j = 1, 2, .., N \tag{2.22}$$

normalizando $RC = 1$, se re escribe (2.22),

$$\frac{dv_j(t)}{dt} = -v_j(t) + \sum_{i=1}^{N} w_{ji} x_i(t) + I_j, \quad j = 1, 2, .., N. \tag{2.23}$$

Escribiendo en términos de redes neuronales dinámicas tenemos:

$$\frac{dx_j(t)}{dt} = -x_j(t) + \varphi\left(\sum_{i=1}^{N} w_{ji} x_i(t)\right) + K_j, \quad j = 1, 2, .., N \tag{2.24}$$

con $\varphi_i(v) = \frac{1-\exp(-a_i v)}{1+\exp(-a_i v)}$.

2.3. Redes Neuronales de Fourier

La forma más simple de una red neuronal es el perceptrón y es usado para la clasificación de un tipo especial de modelo linealmente separable. Consiste de una neurona única, estas redes se estudiaron desde la década de los cincuentas, el objetivo del perceptrón es clasificar

el conjunto de estímulos aplicados $x_1, x_2, \ldots, x_p$, dentro de una de las dos clases ξ_1 y ξ_2. La función de salida de un perceptrón se muestra en la ecuación (**??**).

$$Y = \varphi(v) = \varphi(\sum_{k=0}^{p} W_k X_k - \theta) \tag{2.25}$$

Las llamadas Redes Neuronales de Fourier (RNF), fueron introducidas por A. Silvescu en [57]. Y una nueva técnica para encontrar la serie de Fourier de una función fue presentada en [56], donde se hace uso de Redes Neuronales Artificiales (RNA) para modelar y encontrar los parámetros de una serie de Fourier.

La topología de una RNA y una RNF es similar. La función de salida de una RNF multicapa, se muestra en la ecuación (2.26).

$$Y = \varphi(\sum_{k=1}^{p} W_k \prod_{j=1}^{q} \cos(\omega_{kj} X_j + \phi_{kj})) \tag{2.26}$$

En [56] se muestra un procedimiento para encontrar los coeficientes de la serie de Fourier, que simplifica la estrucura al trabajar con una RNF de una sola capa. De esta forma la función de salida de las RNF sera como se muestra en (2.27).

$$Y = \varphi(\sum_{k=1}^{p} W_k \cos(\omega_k X_k + \phi_k) \tag{2.27}$$

Para poder simplificar el modelo y estructura de la RNF se requiere utilizar herramientas de Ingeniería en comunicaciones, como lo es la transformada de Fourier de una señal, con la cual se obtiene información en frecuencia del arreglo que define al coeficiente de transmitividad. Se observa de la ecuación de la RNF de una sola capa (2.27), en comparación con la ecuación de la RNF multi capa (2.26), que se tiene un solo peso ω, en lugar de $p * q$ distintos pesos ω, es condición necesaria encontrar este valor único ω que corresponde a la frecuencia fundamental del arreglo de valores interpolados linealmente que representan al coeficiente de transmitividad.

El procedimiento utilizado para encontrar la serie de Fourier se describe en los siguientes pasos:

1. Para facilitar la búsqueda de las frecuencias de mayor magnitud presentes en el espectro en frecuencia, es recomendable extraer la componente de frecuencia cero, es decir el valor constante en el tiempo del arreglo, por lo que se sustrae el valor medio del arreglo dado.

2. Una vez retirado el valor constante en el tiempo, se obtienen una función centrada en cero, a la cual se le aplica la transformada de Fourier para identificar las frecuencias principales de esta función.

3. Encontradas las frecuencias de mayor importancia, se buscan las condiciones iniciales para el funcionamiento de la RNF, ω, a_n y b_n, $(n = 1, .., p)$. En efecto estos valores serán el punto de partida con los cuales la RNF encontrará una red de Fourier que aproxime al coeficiente de transmitividad.

4. Finalmente dadas las condiciones iniciales se ejecuta la RNF hasta tener los parámetros de la red de Fourier que nos presenten un error menor al que damos como condición final.

Los pesos W_k escogidos para la RNA, determinan la función que representa la RNA a la salida, el proceso en el que los pesos cambian hasta llegar a ser los adecuados, se denomina aprendizaje. Para el aprendizaje existen varios algoritmos, en este trabajo se utiliza el algoritmo Back Propagation (BP). El BP es un algoritmo de aprendizaje de corrección por error, en el que se utiliza la técnica de gradiente de paso descendente para encontrar los pesos que mejor se ajustan para producir la salida deseada en la RNA. El error entre la función de aprendizaje $(\hat{f})$ y la salida de la RNA (Y), se muestra en la expresión (2.28).

$$E = \frac{1}{2}\sum_{t=0}^{n}(\hat{f}(t) - Y(t))^2 \tag{2.28}$$

El algoritmo busca determinar el vector de pesos W_k que minimice el error E. Esto se logra modificando los pesos en la dirección que produce el máximo descenso en la superficie del error.

La dirección de cambio se obtiene mediante el gradiente. El gradiente especifica la dirección que produce el máximo incremento, por lo que el mayor descenso es el negativo de la dirección. Para abundar mas sobre el algoritmo de Back Propagation consultar [20]. A continuación se muestra en (2.29), el cálculo del gradiente:

$$\begin{aligned}\frac{\partial E}{\partial W_k} &= \frac{\partial}{\partial W_k}\frac{1}{2}\sum_{t=0}^{n}(\hat{f}(t)-Y(t))^2 \\ &= \sum_{t=0}^{n}(\hat{f}(t)-Y(t))\frac{\partial}{\partial W_k}(\hat{f}(t)-W_kX(t)) \\ &= \sum_{t=0}^{n}(\hat{f}(t)-Y(t))(-W_kX(t)) \end{aligned} \tag{2.29}$$

La corrección de los pesos W_k, que se efectúa mediante el aprendizaje se produce como se muestra en (2.30)

$$W_k(t+1) = W_k(t) + \Delta W_k(t) \tag{2.30}$$

donde

$$\Delta W_k(t) = -\alpha \nabla E(t) \tag{2.31}$$

Sustituyendo el gradiente de (2.29) en (2.31), se tiene:

$$\Delta W_k(t) = \alpha(\hat{f}(t)-Y(t))(W_kX(t)) \tag{2.32}$$

Finalmente sustituyendo (2.32) en (2.30), queda la función de corrección de pesos W_k que se muestra en la siguiente ecuación:

$$W_k(t+1) = W_k(t) + \alpha(\hat{f}(t)-Y(t))(W_kX(t)) \tag{2.33}$$

La aplicación del algoritmo Back Propagation tiene los siguiente pasos:

1. Se necesita proponer el valor inicial de los pesos W_k.

2. Para el tiempo t, activar el Perceptrón aplicando un valor continuo del vector de entrada $X(t)$ y la respuesta deseada $\hat{f}(t)$.

3. Calcular la respuesta actual del Perceptrón, ver ecuación (2.27).

4. Actualizar el vector de los pesos del Perceptrón, ver ecuación (2.33).

5. Al llegar al tiempo final $t = t_f$, si E es menor a cierto ϵ predeterminado, se detiene el algoritmo. De lo contrario regresa al paso 2.

Hasta este momento se ha explicado el concepto de RNA, centrándose en su tipo específico Perceptrón, así como el algoritmo de aprendizaje que ajusta los pesos de la red para tener a la salida la función deseada. Haciendo énfasis en lo que se mencionó al principio de este capítulo, en esta tesis se utilizan RNF que hacen uso de una función de activación distinta a las listadas en esta sección. La función que caracteriza la RNF de una sola capa de neuronas, se muestra en la siguiente ecuación:

$$Y = \varphi\left(\sum_{k=1}^{p} W_k \cos(\omega_k X_k + \phi_k)\right) \tag{2.34}$$

La ecuación (2.34), se puede escribir como sigue:

$$Y = \varphi\left(\sum_{k=1}^{p} W_k \cos(\phi_k)\cos(\omega_k X_k) - W_k \sin(\phi_k)\sin(\omega_k X_k)\right) \tag{2.35}$$

Re definiendo los peso de la RNF de la siguiente forma

$$a_k = W_k \cos(\phi_k) \tag{2.36}$$

$$b_k = -W_k \sin(\phi_k) \tag{2.37}$$

Y considerando la función de activación como:

$$\varphi(\nu) = \nu$$

Podemos ahora escribir la RNF como:

$$Y = \sum_{k=1}^{p} a_k \cos(\omega_k X_k) + b_k \sin(\omega_k X_k) \tag{2.38}$$

Finalmente si para $k = n = [1, 2, ..., p]$, se tiene $\omega_n = n\omega$, y haciendo el vector de entrada $X = [t_1, .., t_p]$, con $t_1 = t_2 = .. = t_p$, entonces la salida de la RNF, ver (2.39), es similar a la expresión para la serie de Fourier mostrada en (2.4).

$$Y(t) = \sum_{n=1}^{p} a_n \cos(n\omega t) + b_n \sin(n\omega t) \tag{2.39}$$

Teniendo la expresión para la Red Neuronal y conociendo el valor de ω y $\frac{1}{2}a_o$, se procede ahora a obtener la condiciones iniciales para a_n y b_n y así poder simular el algoritmo BP que encuentre los parámetros a_n y b_n de la serie de Fourier

Capítulo 3

Análisis Wavelets

3.1. Introducción

La teoría de análisis utilizando funciones wavelets apareció como una alternativa al análisis de Fourier, dado que no requiere que las funciones a analizar sean periódicas, y no se presenta en las discontinuidades la presencia del fenómeno de Gibbs [15][2]. La teoría Wavelet tiene mas de 40 años desarrollándose, se puede consultar una interesante reseña histórica del desarrollo de la teoría wavelet en [6].

El análisis por ventanas de Fourier y el análisis wavelet tienen una misma forma de realizar el análisis de una función, que equivale a calcular todas las correlaciones entre esta función y el tiempo-frecuencia que se están utilizando. Pero a diferencia del análisis de Fourier que tiene como objetivo medir la frecuencia contenida en una función, en el caso del análisis utilizando funciones wavelets se busca comparar múltiples magnitudes de una función, con distintas resoluciones, el análisis de Fourier por ventanas se construye a través de senos y cósenos multiplicados por una ventana deslizante, en el análisis con funciones wavelets, la ventana se encuentra oscilando y es llamada wavelet madre.

Esta ventana wavelet madre ya no se multiplica por senos y cósenos, en lugar de ello se traslada y se ensancha, por traslaciones y ensanchamientos arbitrarios. Esa es la manera en que la wavelet madre forma nuevas wavelets las cuales son los bloques que construyen el

análisis wavelet.

3.2. Wavelets

Es posible escribir a una función f a partir de la suma de otras funciones,

$$f(x) = \sum_{n=0}^{\infty} a_n f_n(x) \tag{3.1}$$

eligiendo el valor adecuado para los coeficientes a_n. En el caso de las series de Fourier f_n corresponde a funciones trigonométricas, al aproximar una función queremos representar la función original por una estructura mas fácil de tratar, situación que las series de Fourier cumplen, sin embargo al representar funciones que no son periódicas, se tiene la alternativa de utilizar la transformada de Fourier, que es una representación usando una integral y no una sumatoria, que nos permita aproximar mediante una serie infinita. Otro inconveniente que presenta la aproximación mediante series de Fourier, es que el calculo de los coeficiente de Fourier depende de todos los valores de la función, es decir si la función f cambia su valor solo en un intervalo, todos los coeficientes de Fourier cambiaran.

Ante estos problemas aparece el análisis utilizando funciones wavelets, que tiene propiedades que la hacen muy útil para realizar aproximación y análisis donde no es posible hacerlo con series de Fourier. Con las series Wavelets es posible hacer aproximaciones con un mejor comportamiento local, donde al contrario de las series de Fourier, un cambio en un intervalo de la función afecta solo a unos cuantos coeficientes de la serie y deja sin cambio a las sumas parciales en otros lugares del dominio, dicha propiedad nos sera sumamente útil al construir una red neuronal wavelet y realizar el aprendizaje, como se vera en el siguiente capítulo.

En análisis de señales es común considerar funciones que pertenecen al espacio vectorial

$$L^2(\mathbb{R}) = \left\{ f : \mathbb{R} \to \mathbb{C} | \int_{-\infty}^{\infty} |f(x)|^2 \, dx < \infty \right\}. \tag{3.2}$$

El análisis Wavelet se puede ver como una extensión del análisis de Fourier, que reemplaza las funciones trigonométricas seno y coseno, por una nueva familia de funciones con promedio cero llamadas wavelet,

$$\int_{-\infty}^{\infty} \psi(x)dx = 0, \tag{3.3}$$

las cuales están definidas como sigue.

Definición 3.1 *Una wavelet es una función $\psi \in L^2(\mathbb{R})$ tal que el conjunto*

$$\psi_{(m,n)}(x) = 2^{m/2}\psi\left(2^m x - n\right); m, n \in \mathbb{Z} \tag{3.4}$$

forma una base ortonormal para $L^2(\mathbb{R})$.

Algunas veces ψ es llamada **wavelet madre,** donde m, corresponde al coeficiente de ensanchamiento y n corresponde al coeficiente de traslación. Cada función $f \in L^2(\mathbb{R})$ tiene la representación

$$f = \sum_{m,n\in\mathbb{Z}} < f, \psi_{m,n} > \psi_{m,n,} \tag{3.5}$$

entendido en el sentido que

$$\left\| f - \sum_{|m|,|n|\leq N} < f, \psi_{m,n} > \psi_{m,n} \right\| \to 0 \text{ cuando } N \to \infty. \tag{3.6}$$

Es claro que se requieren condiciones especiales para pertenecer a la clase de funciones wavelets. Para ejemplificar la traslación y ensanchamiento de una función wavelet ψ, primero consideramos el valor de el ensanchamiento $m = 0$ y $traslaciónn = 0$, en la Figura 3.1, observamos la función:

$$\psi_{0,0}(x) = 2^0\psi\left(2^0 x - 0\right) = \psi(x). \tag{3.7}$$

En la Figura 3.2 se muestra como la función $\psi_{0,n}(x)$ es trasladada n unidades hacia la derecha, para mostrar el efecto del ensanchamiento

$$\psi_{0,n}(x) = 2^0\psi\left(2^0 x - 0\right) = \psi(x - n). \tag{3.8}$$

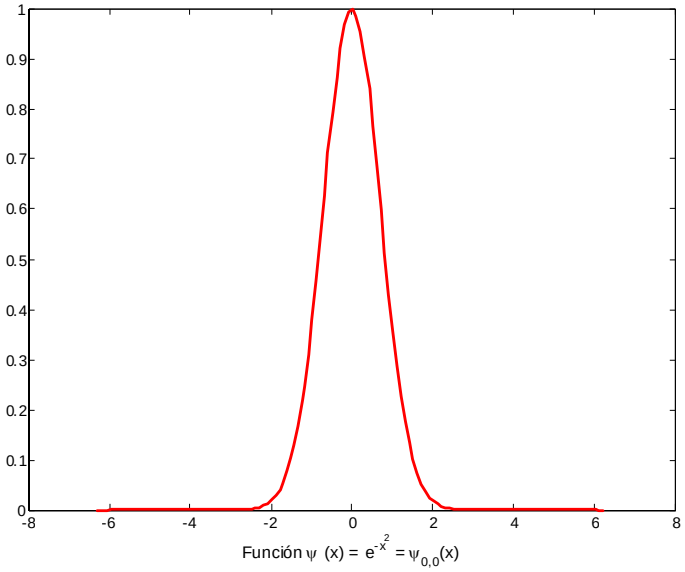


Figura 3.1: Función $\psi(x) = e^{-x^2} = \psi_{0,0}(x)$.

En las Figura 3.3 se observa la función $\psi_{m,0}(x)$ ensanchada, entre menor sea el valor de $m \in \mathbb{Z}$, la función se ensancha, se muestra una traslación $n = 0$.

$$\psi_{m,0}(x) = 2^{m/2}\psi\left(2^2 x\right). \tag{3.9}$$

Trasladando y ensanchando la función $\psi_{m,n}$ se forma un sistema wavelet asociado a la función ψ. Existen distintas funciones que constituyen la base ortonormal en L^2, y que forma un sistema wavelet, las mas utilizadas son:

$$\text{Morlet} = \psi_M(x) = \pi^{-\frac{1}{4}}\left(e^{-j\alpha x} - e^{-\frac{\alpha^2}{2}}\right)e^{-\frac{x^2}{2}}; \left(\alpha = \pi\sqrt{\frac{2}{\ln 2}}\right). \tag{3.10}$$

$$\text{Sombrero Mexicano} = \psi_{MH}(x) = \frac{2}{\sqrt{3\sigma\pi^{1/4}}}\left(1 - \frac{t^2}{\sigma^2}\right)e^{\frac{-t^2}{2\sigma^2}} \tag{3.11}$$

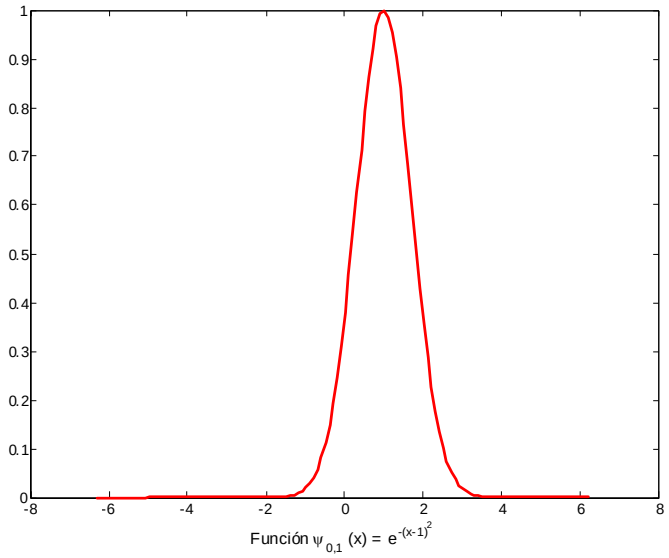


Figura 3.2: Función $\psi_{0,1}(x) = e^{-(x-1)^2}$

$$\text{Haar} = \psi_H(x) = \begin{cases} 1, \ 0 \leq x < \frac{1}{2} \\ -1, \ \frac{1}{2} \leq x < 1 \\ 0, \ \text{otro caso} \end{cases} \tag{3.12}$$

$$\text{Daubechies} = \psi_D(x) = -h_0\varphi(2x-1) + h_1\varphi(2x) - h_2\varphi(2x+1) + h_3\varphi(2x+2)$$

La función wavelet Daubenchies se construye a partir de bloques de escalamiento φ. Las funciones wavelet Haar y Daubechies son utilizadas para realizar la transformación discreta wavelet, sin embargo el algoritmo que utiliza Daubechies representa una mayor complejidad tanto teórica como computacional, la función wavelet Haar es utilizada en el algoritmo de transformación rápida wavelet (FWT), el cual representa una carga computacional menor y genera resultados similares en reconstrucción de funciones a la transformación hecha con funciones Daubechies.

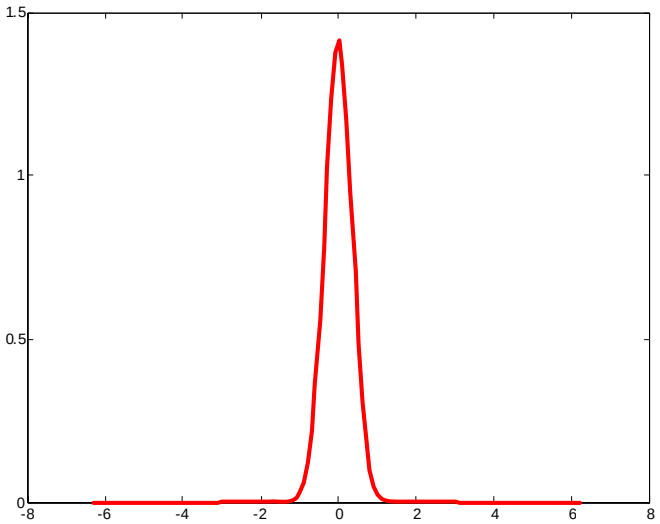


Figura 3.3: Función $\psi_{1,0}(x) = 2^{1/2}\psi(2x-0) = 2^{1/2}e^{-4x^2}$

3.3. Transformada Wavelet

Las series de Fourier son ideales para analizar señales periódicas, para señales no periódicas es comun utilizar la transformada de Fourier para el análisis, sin embargo no es una herramienta natural, dado que se usan funciones periódicas para analizar señales no periódicas, la transtormación de Fourier para alguna función $f(t)$ nos entrega su contenido en frecuencia, sin embargo información concerniente a localización tiempo-frecuencia, por ejemplo cambios repentinos de frecuencia no son facilmente derectados de la transformación $\mathcal{F}$, una alternativa es el uso de la transformada de Fourier por ventanas 3.14, que permite una localización tiempo - freciencia, en su versión discreta se tieneque t y ω estan muestreadas a intervalos regulares: $t = nt_0$, $\omega = m\omega_0$, donde $m, n \in \mathbb{Z}$, y ω_0, $t_0 > 0$, entonces 3.14 se re escribe como 3.15.

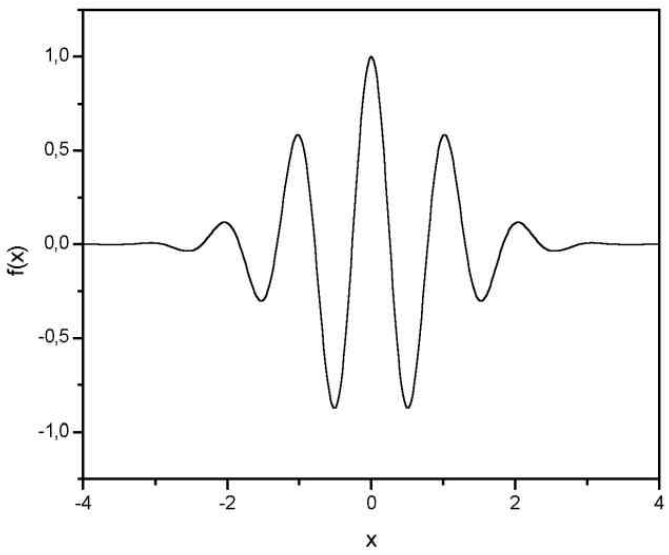


Figura 3.4: Wavelet Morlet

$$\mathcal{F}(\omega) = \frac{1}{\sqrt{2\pi}} \int e^{-i\omega t} f(t) dt \tag{3.13}$$

$$\mathcal{F}w(t,\omega) = \int f(s) w(s-\tau) e^{-2\pi i s\omega} ds. \tag{3.14}$$

$$\mathcal{F}w_{m,n}(t,\omega) = \int f(s) w(s-nt_0) e^{-ims\omega_0} ds. \tag{3.15}$$

La transformación wavelet provee una descripción tiempo-frecuencia similar a 3.15, con algunas diferencia importantes que le proporcionan caracteristicas especiales, la transformación wavelet es completa y mantiene la energia. Para la transformada continua wavelet es comun utilizar las wavelt Morlet.

$$W_{m,n} f(t) =< f, \psi_{m,n} >= \int_{-\infty}^{\infty} f(t) \frac{1}{\sqrt{m}} \psi(\frac{t-n}{m}) dt, \tag{3.16}$$

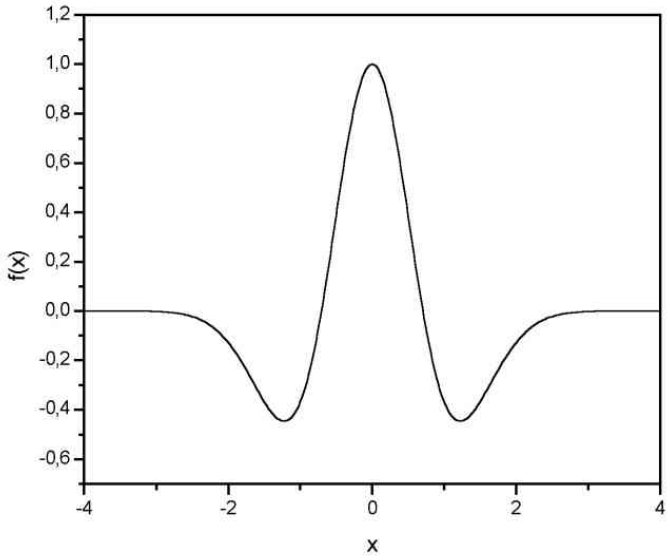


Figura 3.5: Wavelet Sombrero Mexicano

con $m, n \in \mathbb{R}$.

La transformada wavelt discreta (DWT) es una herramienta muy utilizada en el procesamiento de señales como lo es la compresión de video e imagenes, reconocimiento de objetos y análisis numérico. Los coeficientes de DWT son reales, pero el los valores de tiempo y escala son enteros, $m, n \in \mathbb{Z}$.

3.4. Multi resolución

El análisis por multi resolución es un pilar del análisis wavelet y que en esta tesis es fundamental para el desarrollo de una estructura neuronal y del aprendizaje de esta red. La multi resolución se define como sigue:

Definición 3.2 *La multi resolución de* $L^2(\mathbb{R})$ *mediante la función de escala* φ *consiste en*

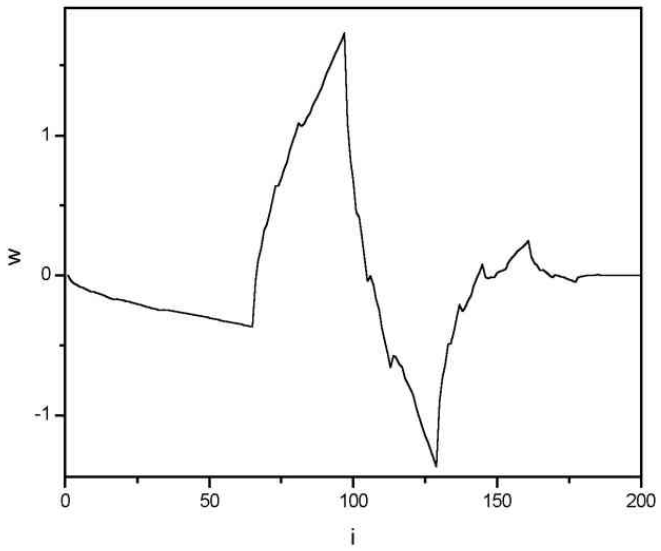


Figura 3.6: Wavelet Daubechies

la secuencia de sub espacios

$$V_n = span\left\{\varphi_{mn}(x) = 2^{m/2}\varphi(2^m x - n) : n \in \mathbb{Z}\right\}, \tag{3.17}$$

que satisfacen las siguientes condiciones:

1. *Ortogonalidad:*$\{\varphi(x-n) : n \in \mathbb{Z}\}$ *es una base ortonormal de* V_0
2. *Anidado:* $V_k \subset V_{k+1}$ *para todo* $k \in \mathbb{Z}$.
3. *Escalamiento:* $f(x) \in V_k$ *si y solo si* $f(2x) \in V_{k+1}$.
4. *Densidad:* $\overline{\cup_{k\in\mathbb{Z}} V_k} = L^2(\mathbb{R})$.
5. *Separación:* $\cap_{k\in\mathbb{Z}} V_k = \{0\}$.

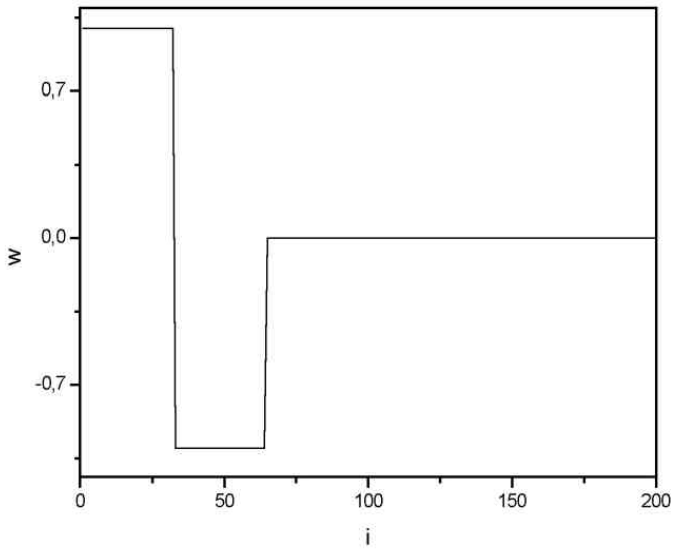


Figura 3.7: Wavelet Haar

La función φ es usualmente llamada wavelet padre. Note que:

$$< \varphi_{mi}, \varphi_{mj} >= \int_{-\infty}^{\infty} 2^m \varphi(2^m x - i)\varphi(2^m x - j)dx = \delta_{ij}. \tag{3.18}$$

entonces $\left\{\varphi_{mj} : j \in \mathbb{Z}\right\}$ forma una base ortonormal de V_m para cada $m \in \mathbb{Z}$, que implica la función φ llamada Wavelet padre genera la wavelet $\psi(x)$ como se muestra a continuación:

$$\varphi(x) = \sqrt{2}\sum_n h_m \varphi(2x - n) \tag{3.19}$$

$$\psi(x) = \sqrt{2}\sum_n g_m \varphi(2x - n) \tag{3.20}$$

donde h_m y g_m son filtros que están relacionados como sigue:

$$g_m = (-1)^{m-1} h_{-(m-1)} \tag{3.21}$$

Por otro lado teniendo la secuencia anidada V_m, se puede descomponer $L^2(\mathbb{R})$ por una suma directa de sub espacios, Sea el conjunto

$$W_m = \{f \in V_{m+1} : f \perp V_m\}, \tag{3.22}$$

donde es el complemento ortogonal de V_m en V_{m+1}, entonces

$$V_k = V_0 \bigoplus W_0 \bigoplus ... \bigoplus W_{k-1} \text{ y } V_0 = V_{-k} \bigoplus W_{-n} \bigoplus ... \bigoplus W_{-1}. \tag{3.23}$$

Lo que sugiere que la descomposición de $L^2(\mathbb{R})$ es una suma directa,

$$L^2(\mathbb{R}) = \oplus_m \boldsymbol{W}_m, \tag{3.24}$$

que implica que es posible descomponer la función $f(x) \in L^2(\mathbb{R})$ por la suma infinita

$$f = \sum_{m=-\infty}^{\infty} f_m, \text{ donde } f_m \in W_m. \tag{3.25}$$

Re escribiendo lo anterior, y definiendo la proyección

$$f_m = \sum_{n=-\infty}^{\infty} < f, \varphi_{mn} > \varphi_{mn}, \tag{3.26}$$

se genera el siguiente lema.

Lemma 3.1 *Suponiendo que $V_m \subset V_{m+1}$ para $m \in \mathbb{Z}$, es una secuencia anidada de sub espacios de la multi resolución de $L^2(\mathbb{R})$, Entonces*

$$\lim_{m \to \infty} \|f - f_m\|_2 = 0. \tag{3.27}$$

Entonces cualquier $f(x) \in L^2$ puede ser aproximada lo suficiente mente cerca en $\boldsymbol{V}_M$, para algún $M \in \mathbb{Z}$ y con $\epsilon > 0$, existe M lo suficientemente grande que

$$\left\| f(t) - \sum_{n=-\infty}^{\infty} \left\langle f, \varphi_{M,n} \right\rangle \varphi_{M,n}(t) \right\| < \epsilon \tag{3.28}$$

Las propiedades de multi resolución del análisis wavelet se resume en los dos siguientes teoremas.

Teorema 3.1 *[15] Suponiendo que $V_k \subset V_{k+1}$ para $k \in \mathbb{Z}$ es una secuencia anidada de sub espacios de la multi resolución de $L^2(\mathbb{R})$. Entonces $L^2(\mathbb{R})$ es descompuesta por la suma directa infinita $\oplus_m \boldsymbol{W}_m$.*

Teorema 3.2 *[15] Sea φ la función de escala generadora de la multi resolución $\{V_k\}$ de $L^2(\mathbb{R})$ con la relación de escalamiento $\varphi(x) = \sum_{n=-\infty}^{\infty} a_n \varphi(2x-n)$. Definiendo*

$$\psi(x) = \sum_{n=-\infty}^{\infty} (-1)^n a_{1-n} \varphi(2x-n). \tag{3.29}$$

Entonces ψ es la wavelet que genera la base wavelet $\{\psi_{mn} : m, n \in \mathbb{Z}\}$ tal que $W_m = span\{\psi_{mn} : n \in \mathbb{Z}\}$ para cada $m \in \mathbb{Z}$.

3.5. Función Wavelet Haar

La primera mención de una función Wavelets apareció en el apéndice de la tesis de A. Haar, cuando se pregunto si existía algún otro sistema ortogonal $h_0, h_1, ..., hn, ...$ de funciones definidas en $[0,1]$, tal que para cualquier función continua f definida en $[0,1]$, la serie

$$< f, h_0 > h_0(x) + < f, h_1 > h_1(x) + ... + < f, h_n > h_n(x) + ...$$

converge a $f(x)$ uniformemente en $[0,1]$?

En 1909 Haar descubrió una solución y de la misma forma abrió una ruta hacia las Funciones Wavelets, la función Wavelet de Haar es una función compacta soportada, que es una base ortonormal (también llamada base de Hilbert), que no es continuamente diferenciable. La expresión matemática de la Función Wavelet Haar se muestra a continuación:

$$H(x) = \begin{cases} 1, & 0 \leq x < \frac{1}{2}, \\ -1, & \frac{1}{2} \leq x < 1, \\ 0, & \text{otro caso}, \end{cases} \tag{3.30}$$

Esta es la función wavelet matemáticamente más sencilla que existe y que aproxima cualquier función continua $f \in L^2[0,1]$ mediante una serie,

En la contrucción de la Base Wavelet Haar iniciamos por describir la aproximación simple, la cual es muy utilizada para representar en valores discretos fenómenos que ocurren en tiempo continuo, y que debido a limitantes de computo se adquiere solo una secuencia finita de valores, llamada muestreo, ver Figura 3.8. El primer paso para el análisis de señales con Wavelets consiste en aproximar utilizando solamente la señal muestreada, un método simple de aproximación es usar un escalón rectangular extendido a través de cada punto del muestreo como se muestra en la Figura 3.8 (c). El resultado de estos escalones forma una nueva función llamada aproximación simple y denotada por $\tilde{f}$, la cual aproxima a la función f,

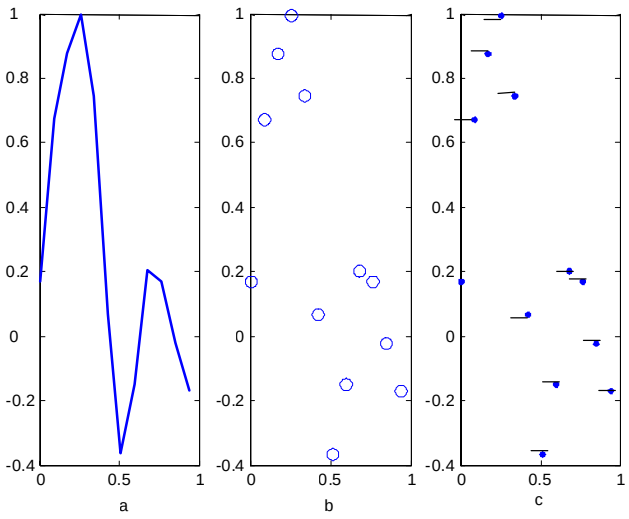


Figura 3.8: (a)Señal. (b)Muestro. (c)Aproximación.

Definición 3.3 *Para todos los números u y w, la notación $[u, w[$ representa el intervalo de todos los números de u incluido a w excluido:*

$$[u, w[= \{r : u \leq r < w\}. \tag{3.31}$$

El análisis de la función de aproximación $\tilde{f}$ en términos de Wavelets requiere una identificación precisa de cada paso, diferenciando de cada ensanchamiento y dilatación de la función de escalón base, denotada por $\chi_{[0,1[}$ la cual se muestra en la Figura 3.9. La función escalón se escribe como:

$$\chi_{[0,1[}(r) = \begin{cases} 1 \text{ si } 0 \leq r < 1, \\ 0 \text{ otro caso.} \end{cases} \tag{3.32}$$

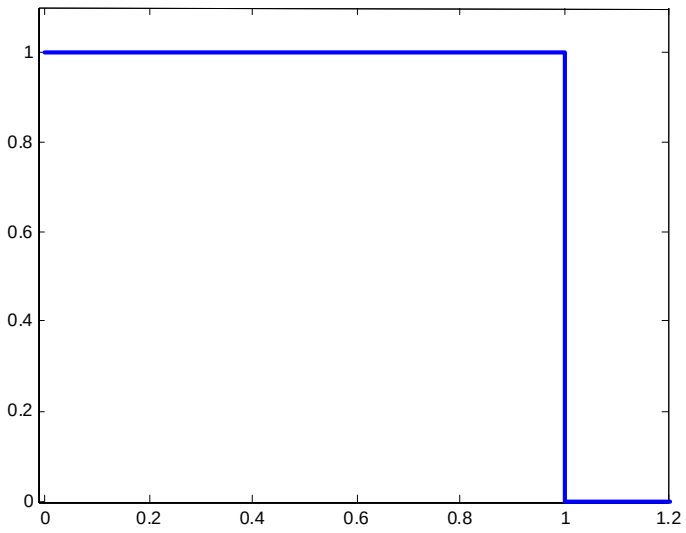


Figura 3.9: Función Escalón

De manera similar se construye la función escalón de tamaño c, que inicia en u, y termina en w, la Figura 3.10 muestra la función $c * \chi_{[u,w[}$, la cual de define como sigue:

$$c * \chi_{[u,w[}(r) = \begin{cases} c \text{ si } u \leq r < w, \\ 0 \text{ otro caso.} \end{cases} \tag{3.33}$$

Entonces si se tiene un conjunto de puntos (r_j, s_j), de una función de N muestras f, donde $s_j = f(r_j)$,

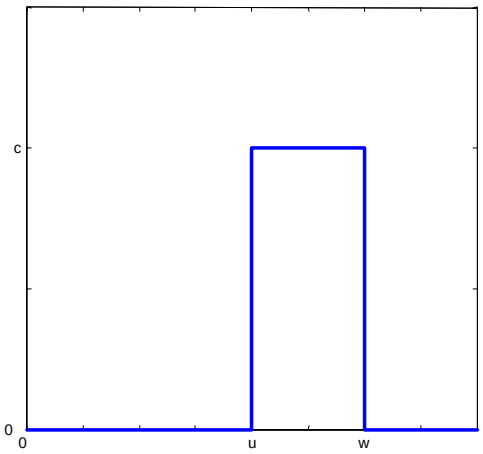


Figura 3.10: Función Escalón Compuesta

$$\tilde{f}_j = s_j * \chi_{[r_j, r_{j+1}[},$$

la cual aproxima a la función f a el valor s_j en el intervalo $[r_j, r_{j+1}[$, entonces la aproximación a f se escribe como:

$$\tilde{f} = \sum_{j=0}^{N-1} s_j * \chi_{[r_j, r_{j+1}[} \tag{3.34}$$

Generalizando la función escalón (3.33), y re escribiendo en torno a ensanchamiento y traslación, se tiene que:

$$\chi_{m,n}(x) = \begin{cases} 1, & 2^{-m}n \leq x < 2^{-m}(n+1), \\ 0, & \text{otro caso}, \end{cases} \tag{3.35}$$

Entonces para cualquier función $f \in L^2$, existe la función $\tilde{f}$ que la aproxime con una resolución m

$$\tilde{f}_m(x) = \sum_{n \in \mathbb{Z}} a_{m,n} \chi_{m,n}(x) \ , \ m \in \mathbb{N}, \tag{3.36}$$

tal que

$$\lim_{m\to\infty} ||f - f_m|| = 0.$$

3.5.1. Base Wavelet Haar

Utilizando la estructura anidada (3.23), podemos construir una base ortonormal de L^2. Siendo W_n un complemento ortonormal de V_m con respecto a V_{m+1}:

$$W_m \oplus V_m = V_{m+1},\ W_m \perp V_m. \tag{3.37}$$

Por lo que se tiene

$$L^2 = \oplus_{m\in\mathbb{Z}} W_m,\ W_m \perp W_{m'} \tag{3.38}$$

Como cada sub espacio V_m es un ensanchamiento de 2^m de V_0, W_m es también un ensanchamiento de 2^m de W_0, recalcando que el ensanchamiento preserva la otorgonalidad. De esta manera, si $\{e_k\}_{k\in\mathbb{Z}}$ es una base ortogonal de W_0, entonces esta dilatación 2^m es una base ortogonal de W_m. Por lo que la tarea se reduce a encontrar una base ortonormal de W_0. Para hacer esto se define

$$H(x) = \begin{cases} 1,\ 0 \le x < \frac{1}{2} \\ -1,\ \frac{1}{2} \le x < 1 \\ 0,\ \text{otro caso} \end{cases} \tag{3.39}$$

que es llamada la función Haar. Lo cual nos lleva al siguiente lema.

Lemma 3.2 *[15] Dado*

$$H_m(x) = H(x-m), m \in \mathbb{Z}.$$

Entonces el sistema $\{H_m\}_{m\in\mathbb{Z}}$ *es una base ortonormal de* W_0*. Por consecuencia el sistema* $\{H_{m,n}(x)\}_{m,n\in\mathbb{Z}}$ *es una base ortonormal de el espacio* W_m*.*

Del lema, se obtiene el siguiente teorema.

Teorema 3.3 *[15] El sistema* $\{H_{m,n}(x)\}_{m,n\in\mathbb{Z}}$ *forma una base ortonormal de* L^2

Usualmente se nombra a el espacio W_m, como espacio Haar y se llama a $\{H_{m,n}(x)\}_{n\in\mathbb{Z}}$ como la base Haar de L^2. El significado de la descomposición Haar de una función puede explicarse como sigue. Se observa que cada elemento $H_{m,n}$ en la base Haar representa una onda cuadrada centrada en $\frac{2n+1}{2^{m+1}}$ teniendo una anchura de $\frac{1}{2^m}$, Entonces, se dice que $H_{m,n}$ tiene una frecuencia llamada Haar de 2^m, esta función es la mas pequeña onda local. Entonces, el ancho de la onda provee la información espacial, esto es, el índice m de $H_{m,n}$ indica la frecuencia de la función mientras que el índice n, indica la localización espacial.

Una base ortonormal con la estructura como la base Haar es muy útil para el análisis tiempo-frecuencia, en la Figura 3.11 se muestra las distintas formas de la función Haar dependiente de su anchura y traslación. Siguiendo esta idea se da la siguiente definición.

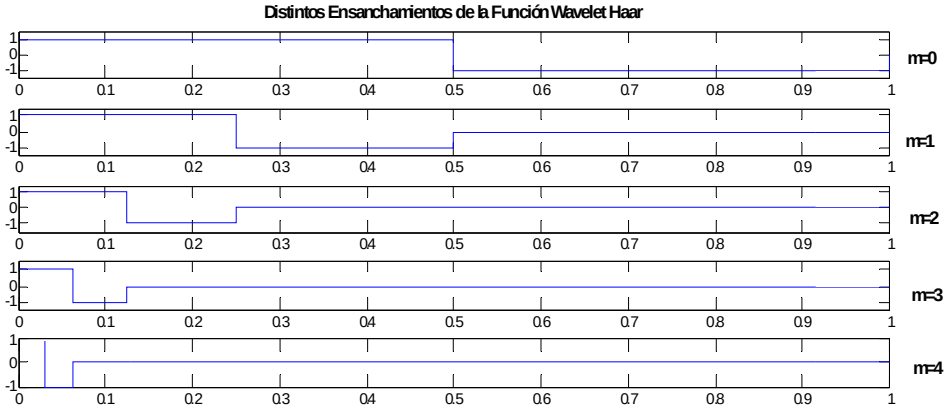


Figura 3.11: Distintas resolución de la función Wavelet Haar.

Definición 3.4 *Una función $\psi \in L^2$ es llamada una wavelet ortonormal si $\{\psi_{m,n}\}_{m,n\in\mathbb{Z}}$ forma una base ortonormal en L^2.La base $\{\psi_{m,n}\}_{m,n\in\mathbb{Z}}$ generada por ψ es llamada base ortonormal wavelet de L^2*

$$H_{n,m}(x) = \begin{cases} 1 \ si \ m2^n \leq x < (m+\frac{1}{2})2^n, \\ -1 \ si \ (m+\frac{1}{2})2^n \leq x < (m+1)2^n \\ 0 \ otro \ caso. \end{cases} \tag{3.40}$$

3.5.2. Descomposición de Funciones en Series Wavelet Haar

La manera natural de aproximar una función $f : [0,1] \to \mathbb{R}$, es partir el intervalos $[0,1]$ en cierto número de sub intervalos y después cada uno de esos sub intervalos aproximará $f(x)$ por el promedio de f, en ese intervalo. El intervalo $[0,1]$, se parte en 2^m intervalos para $m = 0, 1, ...$; cada intervalo debe tener el mismo tamaño. Los intervalos son:

$$I_n = \left[n2^{-m}, (n+1)2^{-m}\right[, \ n = 0, 1, ..., 2^m - 1 \tag{3.41}$$

Esos intervalos tienen una longitud de 2^{-m}, el promedio de f en el intervalo I_n es

$$a_n = 2^m \int_{n2^{-m}}^{(n+1)2^{-m}} f(x)dx. \tag{3.42}$$

El intervalo I_n así como el coeficiente a_n, dependen del valor elegido de m. Por lo que los escribimos como I_{mn}, a_{mn}.

Si el valor de m es grande, i.e. el intervalo I_n es pequeño, el valor de a_n es una buena aproximación para $f(x)$ para $x \in I_n$; Entonces la aproximación a f, se puede escribir como:

$$f_m := \sum_{n=0}^{2^m-1} a_n \mathcal{X}\left[n2^{-m}, (n+1)\,2^{-m}\right[(x) \tag{3.43}$$

El parámetro m indica el nivel de la resolución o escala de la aproximación; un valor grande de m provee una aproximación fina, mientras mas pequeño sea el valor de m, mas gruesa es la aproximación.

Lemma 3.3 *[15] Asumiendo que $f : [0,1] \to \mathbb{R}$ es continua. Entonces, para cualquier $\epsilon > 0$ existe $K \in \mathbb{N}$ tal que*

$$|f(x) - f_m(x)| \leq \epsilon, \forall x \in [0,1[\,,$$

para todo $k \geq K$.

Escribimos explícitamente las primeras aproximaciones para (3.43)

$$\begin{array}{c} f_0(x) = a_0 \mathcal{X}[0,1[(x), \text{ donde } a_0 = \int_0^1 f(x)\,dx \\ f_1(x) = \sum_{n=0}^{1} b_n \mathcal{X}\left[\frac{n}{2}, \frac{(n+1)}{2}\right[(x), \text{donde } b_n = 2\int_{n/2}^{(n+1)/2} f(x)\,dx \\ f_2(x) = \sum_{n=0}^{3} c_n \mathcal{X}\left[\frac{n}{4}, \frac{(n+1)}{4}\right[(x), \text{ donde } c_n = 4\int_{n/4}^{(n+1)/4} f(x)\,dx \\ f_3(x) = \sum_{n=0}^{7} d_n \mathcal{X}\left[\frac{n}{8}, \frac{(n+1)}{8}\right[(x), \text{donde } d_n = 8\int_{n/8}^{(n+1)/8} f(x)\,dx \end{array}, \tag{3.44}$$

No importa que valor de m se considere, sin perder generalidad y considerando f_{m-1} una aproximación a f_m, se puede escribir

$$f_m(x) = f_{m-1} + [f_m(x) - f_{m-1}(x)]. \tag{3.45}$$

Para valores grande de m, tanto f_m como f_{m-1}, son buenas aproximaciones a f, y la diferencia entre $f_m - f_{m-1}$ consistirá solo en detalles finos, sin embargo si el valor de m es pequeño, la resolución sera gruesa y la aproximación sera mala, y la diferencia $f_m - f_{m-1}$ sera grande.

Teorema 3.4 *[15] Sea $f : [0,1] \to \mathbb{R}$ una función continua. Entonces, para cualquier valor de $m \in \mathbb{N}$, la relación entre la aproximación f_m y f_{m-1} esta dada como*

$$\begin{aligned} f_m(x) &= \sum_{n=0}^{2^m-1} a_{k,n} \mathcal{X}\left[n2^{-m}, (n+1)2^{-m}\right[(x) \\ &= f_{k-1}(x) + \textit{detalles} \\ &= \sum_{n=0}^{2^{m-1}-1} a_{k-1,n} \mathcal{X}\left[n2^{-m+1}, (n+1)2^{-m+1}\right[(x) + \\ &+ \sum_{n=0}^{2^{m-1}-1} d_{k-1,n} \psi\left(2^{m-1}x - n\right), \end{aligned} \tag{3.46}$$

donde los coeficientes están relacionados como

$$a_{m-1,n} = \frac{a_{m,2n} + a_{m,2n+1}}{2}, \tag{3.47}$$

$$d_{m-1,n} = \frac{a_{m,2n} - a_{m,2n+1}}{2}, \tag{3.48}$$

teniendo f_m la representación multi escala

$$f_m = a_0 \mathcal{X}\left[0,1\right[(x) + \sum_{j=0}^{m-1}\sum_{n=0}^{2^j-1} d_{j,n}\psi\left(2^j x - n\right). \tag{3.49}$$

Se observa que la sumatoria de la expresión (3.49) que aproxima f_k es finita, sin embargo el número de términos en la serie incrementa al incrementar el valor de m, es decir aumentar la resolución de la aproximación, para lograr una exacta representación de una función dada f, se realiza mediante una serie infinita.

Teorema 3.5 *[15] Se asume que $f : [0,1] \to \mathbb{R}$ es continua. Entonces*

$$f(x) = \langle f, \mathcal{X}\left[0,1\right[(x)\rangle\, \mathcal{X}\left[0,1\right[(x) + \sum_{j=0}^{\infty}\sum_{n=0}^{2^j-1} \left\langle f, \psi_{j,n}\right\rangle \psi_{j,n}(x). \tag{3.50}$$

Para la mayoría de las aplicaciones practicas una función es representada por una serie trunca, permitiendo un error de aproximación, similar a las series truncas de Fourier. El algoritmo para encontrar los coeficientes a_0, d_m, (3.47), (3.48), es conocido como transformada Wavelet.

3.6. Transformada Wavelet Haar

La transformación wavelet Haar expresa la función aproximada $\tilde{f}$ con wavelets, reemplazando un par de escalones adyacentes por un escalón amplio y por una wavelet. El escalón amplio mide el promedio de los dos escalones iniciales, mientras que la wavelet formada por dos escalones alternados, mide la diferencia entre el par inicial de escalones. Por ejemplo, la suma de dos escalones adyacentes de anchura $\frac{1}{2}$ produce el escalón unitario $\varphi_{[0,1]}$,

$$\varphi_{[0,1[} = \varphi_{\left[0,\frac{1}{2}\right[} + \varphi_{\left[\frac{1}{2},1\right[} \tag{3.51}$$

de manera similar, la diferencia de dos escalones adyacentes produce la wavelet básica, denotada por $\psi_{[0,1[}$ y definida como

$$\psi_{[0,1[} = \varphi_{\left[0,\frac{1}{2}\right[} - \varphi_{\left[\frac{1}{2},1\right[} \tag{3.52}$$

La wavelet $\psi_{[0,1[}$ es una función escalón simple, con un primer escalón de valor 1, seguido de un segundo escalón de valor -1, así, de este primer escalón al segundo escalón, el valor de la wavelet $\psi_{[0,1[}$ sometido a un salto de tamaño -2, en términos del escalón unitario $\varphi_{[0,1[}$ y de la wavelet $\psi_{[0,1[}$ se tienen las operaciones

$$\begin{cases} \frac{1}{2}\left(\varphi_{[0,1[} + \psi_{[0,1[}\right) = \varphi_{[0,\frac{1}{2}[}, \\ \frac{1}{2}\left(\varphi_{[0,1[} - \psi_{[0,1[}\right) = \varphi_{[\frac{1}{2},1[}. \end{cases}$$

Para dos escalones adyacentes de valores $[s_0, s_1]$, la ecuación derivada representada por un escalón y una wavelet

$$\begin{aligned} \tilde{f} &= s_0 * \varphi_{[0,\frac{1}{2}[} + s_1 * \varphi_{[\frac{1}{2}=,1[} \\ &= s_0 * \frac{1}{2}\left(\varphi_{[0,1[} + \psi_{[0,1[}\right) + s_1 * \frac{1}{2}\left(\varphi_{[0,1[} - \psi_{[0,1[}\right) \\ &= \frac{s_0 + s_1}{2}\varphi_{[0,1[} + \frac{s_0 - s_1}{2}\psi_{[0,1[}. \end{aligned} \tag{3.53}$$

La transformación básica wavelet Haar, preserva toda la información de la función muestreada, $\frac{s_0+s_1}{2}$ mide el promedio de la función $\tilde{f}$ y $\frac{s_0-s_1}{2}$ mide el cambio en la función $\tilde{f}$.

Para aplicar la transformación básica wavelet Haar sobre un intervalo $[u, w]$, definimos la traslación y el ensanchamiento de la wavelet $\psi_{[u,w[}$ a partir del punto medio $v = \frac{(u+w)}{2}$:

$$\psi_{[u,w[}(r) = \begin{cases} 1 \text{ si } u \leq r < v, \\ -1 \text{ si } v \leq r < w. \end{cases} \tag{3.54}$$

Generalizando esta transformada básica se llega a el algoritmo conocido como la transformada rápida wavelet, que se presenta a continuación.

Para analizar una señal o función en términos de wavelets, la Transformada Rápida Wavelet Haar inicia con la inicialización de un arreglo con 2^n entradas, y después se realiza un procedimiento con n iteracciones de la transformada básica explicada anteriormente. Para cada índice $l \in \{1, ..., n\}$, antes de la iteración número l, el arreglo consistirá de $2^{n-(l-1)}$ coeficientes de $2^{n-(l-1)}$ funciones escalón $\varphi_k^{(n-[l-1])}$, después de la iteración número l, el arreglo consistirá de la mitad de su tamaño inicial, 2^{n-l} coeficientes de 2^{n-l} funciones escalón $\varphi_k^{(n-l)}$, y 2^{n-l} coeficientes de la wavelet $\psi_k^{(n-l)}$.

Definición 3.5 *Para cada $n \in \mathbb{Z}^+$, y cada índice $l \in \{0,..,n\}$, se define la función escalón $\varphi_k^{(n-l)}$, y la wavelet $\psi_k^{(n-l)}$ por*

$$\varphi_k^{(n-l)}(r) = \varphi_{[0,1[}(2^{n-l}[r - k2^{n-l}]) \tag{3.55}$$
$$= \begin{cases} 1 \text{ si } k2^{l-n} \leq r < (k+1)2^{l-n}, \\ 0 \text{ otro caso.} \end{cases}$$

$$\psi_k^{(n-l)}(r) = \psi_{[0,1[}(2^{n-l}[r - k2^{n-l}]) \tag{3.56}$$
$$= \begin{cases} 1 \text{ si } k2^{l-n} \leq r < (k+\frac{1}{2})2^{l-n}, \\ -1 \text{ si } (k+\frac{1}{2})2^{l-n} \leq r < (k+1)2^{l-n} \\ 0 \text{ otro caso.} \end{cases}$$

El algoritmo tiene como condición de aplicación que la función muestreada este formada por un arreglo de 2^n muestras

$$\tilde{f}^{(n)} = \sum_{j=0}^{2^n-1} a_j^{(n)} \varphi_j^{(n)} \tag{3.57}$$

En general, el $l - esimo$ paso de la transformación básica inicia con un arreglo de $2^{n-(l-1)}$ valores

$$\vec{a}^{(n-[l-1])} = \left(a_0^{(n-[l-1])}, ..., a_{2^{n-(l-1)}-1}^{(n-[l-1])}\right), \tag{3.58}$$

y aplica la transformación básica a cada par $\left(a_{2k}^{(n-[l-1])}, a_{2k+1}^{(n-[l-1])}\right)$, los cuales generan dos coeficientes wavelets

$$a_k^{(n-l)} := \frac{a_{2k}^{(n-[l-1])} + a_{2k+1}^{(n-[l-1])}}{2}, \tag{3.59}$$
$$c_k^{(n-l)} := \frac{a_{2k}^{(n-[l-1])} - a_{2k+1}^{(n-[l-1])}}{2} \tag{3.60}$$

Esos $2^{(n-l)}$ nuevos coeficientes representan el resultado de $l - esima$ iteración, el resultado son dos arreglos

$$\vec{a}^{(n-l)} = \left(a_0^{(n-l)}, a_1^{(n-l)}, ..., a_k^{(n-l)}, ..., a_{2^{n-l}-1}^{(n-l)}\right)$$
$$\vec{c}^{(n-l)} = \left(c_0^{(n-l)}, c_1^{(n-l)}, ..., c_k^{(n-l)}, ..., c_{2^{n-l}-1}^{(n-l)}\right)$$

En el arreglo para la $l-esimam$ iteración. $\vec{a}^{(n-[l-1])}$: El arreglo inicial.

$$\vec{a}^{(n-[l-1])} = \left(a_0^{(n-[l-1])}, ..., a_{2^{n-(l-1)}-1}^{(n-[l-1])}\right), \tag{3.61}$$

representa la lista de valores $a_k^{(n-[l-1])}$ de una función $\tilde{f}^{(n-[l-1])}$ que aproxima a la función f con $2^{n-(l-1)}$ escalones de muestreo de ancho $2^{(l-1)-n}$:

$$\tilde{f}^{(n-[l-1])} = \sum_{j}^{2^{n-(l-1)}-1} a_j^{(n-[l-1])}\varphi_j^{(n-[l-1])}. \tag{3.62}$$

$\vec{a}^{(n-l)}$: Primer arreglo producto de la iteración $l-esima$.

$$\vec{a}^{(n-l)} = \left(a_0^{(n-l)}, ..., a_{2^{n-(l-1)}-1}^{(n-l)}\right), \tag{3.63}$$

lista de valores $a_k^{(n-l)}$ de la función compuesta por escalones $\tilde{f}^{(n-l)}$ que aproxima a la función f con 2^{n-l} escalones de ancho 2^{l-n}

$$\tilde{f}^{(n-l)} = \sum_{j}^{2^{n-l}-1} a_j^{(n-l)}\varphi_j^{(n-l)}. \tag{3.64}$$

$\vec{c}^{(n-l)}$: Segundo arreglo producto de la iteración $l-esima$.

$$\vec{c}^{(n-l)} = \left(c_0^{(n-l)}, ..., c_{2^{n-(l-1)}-1}^{(n-l)}\right), \tag{3.65}$$

lista de valores $c_k^{(n-l)}$ de la función wavelet $\psi_j^{(n-l)}$ con escalones de ancho 2^{l-n}

$$\dot{f}^{(n-l)} = \sum_{j}^{2^{n-l}-1} c_j^{(n-l)}\psi_j^{(n-l)}.$$

Las wavelets que entrega el segundo nuevo arreglo $\vec{c}^{(n-l)}$, representa las diferencias entre el paso fino de la aproximación inicial $\tilde{f}^{(n-[l-1])}$, y el paso mas grueso de $\tilde{f}^{(n-l)}$. Entonces, cada iteración de transformada básica expresa la aproximación fina previa como la suma de una nueva, aproximación mas gruesa y un nuevo conjunto de wavelets de menor frecuencia. Sin embargo, la trasformada básica Haar no altera la función muestreada, simplemente la expresa con wavelets, esto es que la aproximación inicial $\tilde{f}^{(n-[l-1])}$ es igual a la suma de sus nuevas aproximaciones, $\tilde{f}^{(n-l)}$ y $\dot{f}^{(n-l)}$

$$\tilde{f}^{(n-[l-1])} = \tilde{f}^{(n-l)} + \dot{f}^{(n-l)}. \tag{3.66}$$

Capítulo 4

Modelado de Series de Tiempo via Redes Neuronales Wavelets Haar

4.1. Introducción

A partir de la aparición de la teoría moderna de análisis Wavelets, se desarollaron muchas aplicaciones en diversas áreas de ingeniería, sobre todo en el tratamiento de imágenes y datos [50],[49],[37]. Múltiples investigadores en campo de las redes neuronales artificiales vieron las propiedades de las funciones wavelets y estas fueron bien recibidas, las Redes Neuronales Wavelets (RNW) fueron propuestas por primera vez por Zhang y Benveniste [28], en donde se presentaron las RNW como un nuevo tipo de red neuronal feed forward que además presenta ventajas sobre las MLP, mas tarde en [25] las RNW utilizan funciones Wavelets ortogonales y son mostradas como un caso especial las redes neuronales radiales básicas, estas redes dejaron de lado la propiedad de multi resolución del análisis wavelet y solo trabajan como una función radial básica con un ensanchamiento determinado.

En la literatura sobre redes neuronales wavelet se presenta una nueva generación de RNW que utilizan funciones wavelets continuas como la Morlet y Sombrero Mexicano [12],[30], se exponen las bondades y las múltiples aplicaciones de las redes neuronales wavelets, sin embargo se mantiene el problema de establecer el valor de los coeficientes de traslación (n)

y ensanchamiento (m) que tendrán las funciones wavelets, así como tampoco se trata el problema de cuantas funciones wavelets se utilizaran.

En redes neuronales no se ha puesto suficiente atención a la funcion wavelet Haar, la cuale es matemáticamente la mas simple, la principal razón para este situación es que la wavelet Haar es discontinua, aun cuando sus aplicaciones en soluciones numéricas de ecuaciones diferencias y han sido ampliamente documentadas, [26],[27].

La wavelet Haar es utilizada en el algoritmo de transformación rápida wavelet (FWT) [8], y dicho algoritmo es muy difundido en el tratamiento de imágenes, [9], además encontramos diversos trabajos que hacen la comparación entre las distintas bases wavelets, [51], [52], y que muestran que el desempeño de la wavelet Haar es similar a otras bases y que presenta además la ventaja de ser mas sencilla de implementar.

En esté trabajo consideramos la ventaja del desarrollo de algoritmos para sistemas discretos con redes neuronales [16],[17], así como las ventajas ya expuestas de la implementación de la función wavelet Haar, la cual nos permite tener algoritmos discretos que pueden ser programados sin la necesidad de recurrir a librerías de funciones trigonométricas [53].

En las redes neuronales wavelets Haar feed forward podemos distinguir dos tipo, las redes neuronales wavelets de una sola capa y RNW multicapa, en las RNW de una sola capa se conserva la estructura de una serie wavelet, donde los coeficientes de traslación y ensanchamiento se eligen por algún método determinado, y el caso de mas RNW multicapa, donde los coeficientes de traslación y ensanchamiento se encuentra en la red como capas ocultas, en ambos casos el problema de optimizar el tamaño de la red se encuentra abierto, dado que se requiere determinar por algún método el número de neuronas que se utilizaran, el número de neuronas esta relacionado con el valor de los coeficientes de traslación y ensanchamiento, en el caso de las redes neuronales wavelets utilizan funciones wavelets continuas como la Sombrero Mexicano y Morlet, el valor de traslación y ensanchamiento puede ser infinito, el usar una función no continua y aprovechar la teoría que hay para la transformada rápida wavelet que utiliza una wavelet Haar, nos permite desarrollar un método para optimizar el tamaño de la red neuronal wavelet Haar.

4.2. Redes Neuronales Wavelets Haar

La propiedad de la series wavelet de aproximar cualquier función $f \in L^2(\mathbb{R})$, ha sido expuesta en el capitulo anterior, las redes neuronales wavelets están inspiradas en la forma en que la serie wavelet aproxima a una función, la cual tiene la forma siguiente:

$$f(k) = \sum_{m=0}^{\infty} \sum_{n=0}^{2^m-1} w_{mn}\psi_{mn}(k), \tag{4.1}$$

donde el coeficiente $w_{m,n}$ corresponde al coeficiente de la transformada wavelet, que está definido como el producto interno $w_{m,n} =< f, \psi_{mn} >$, con $m, n \in \mathbb{Z}$.y ψ es una función wavelet. Es posible hacer una aproximación a la función f utilizando una serie wavelet trunca, es decir fijando el valor de traslación y ensanchamiento, de la siguiente forma:

$$f(k) = \sum_{m=0}^{M} \sum_{n=0}^{2^m-1} w_{mn}\psi_{mn}(k) + \mu(k), \tag{4.2}$$

donde $\mu(k)$ es el error de truncamiento de la serie, el cual es acotado, $\mu(k) < \mu_T$. De forma inmediata se puede escribir la serie (4.2), como una red neuronal wavelet,

$$\hat{f}(k) = \sum_{m=0}^{M} \sum_{n=0}^{2^m-1} w^*_{mn}\psi_{mn}(k), \tag{4.3}$$

donde w^*_{mn} corresponde al coeficiente óptimo de la red neuronal wavelet, ψ corresponde a la función wavelet, la Figura 4.1 se observa la estructura de una neurona de la red.

La estructura de la red neuronal wavelet se muestra en el Figura 4.2, como se ha expuesto en esta tesis la función wavelet utilizada es la wavelet Haar, $H_{mn}(k)$ (3.39).

Plantear la estructura de la red neuronal presenta el problema de establecer un método para determinar el valor de m y n, diversos métodos han sido mostrados en la literatura, dos principales caminos se han tomado, los cuales consisten en:

1. La Red Neuronal Wavelet como un caso especial de una red radial básica.

2. La Red Neuronal Wavelet aprovechando la propiedad de multi resolución del análisis wavelet.

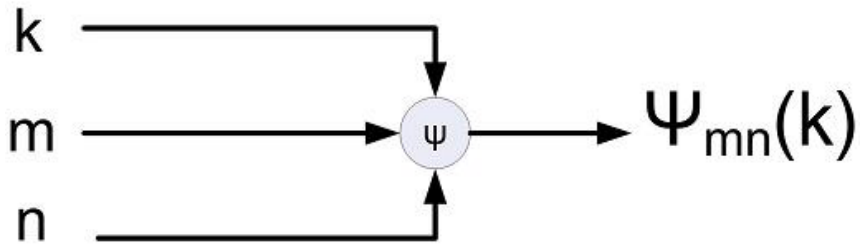


Figura 4.1: Neurona Wavelet

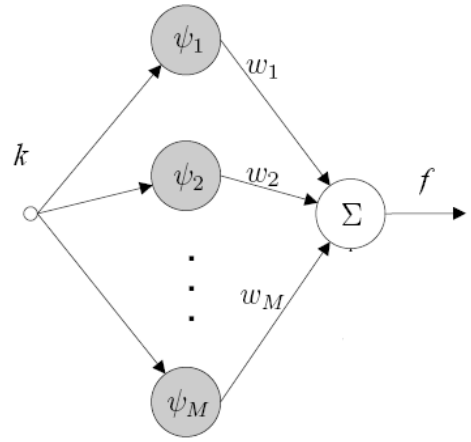


Figura 4.2: Estrucutra de la Red Neuronal Wavelet

El primer caso es la manera mas simple de diseñar una RNW, dado que solo en lugar de utilizar una función radial, se utiliza una función wavelet, y se utiliza valor único de ensanchamiento el cual puede aprenderse, este tipo de red la podemos ver en [25],[28],[29] sin embargo esta estructura se limita al no tener distintas resoluciones en la red. El segundo método ha sido presentado en [36], [37], [39], sin embargo en estos trabajos no determinan un procedimiento para obtener el tamaño de la red neuronal y el valor de el coeficiente de ensanchamiento m.

En la literatura se encuentra como problema abierto como decidir que valores son los

adecuado a utilizar para el ensanchamiento m y traslación n, debido a que entre mayor sea el valor de estos, la red neuronal sera mas grande, existen principalmente tres formas de solucionar el problema,

- Establecer el valor del ensanchamiento y traslación, m, n, como capas ocultas de una red neuronal wavelet multicapa.
- Fijar por algún criterio el valor de m, n, en una red neuronal wavelet de una sola capa.
- Encontrar el valor de m, n, mediante una red neuro difusa.

Esas tres técnicas se presenta aun el inconveniente de fijar el numero de neuronas de la red, en la primera técnica aun cuando se tome como capas ocultas a m, n, aun se debe establecer por algún criterio el tamaño de la red neuronal en sus distintas capas, así como determinar el número de capas ocultas. En las redes neuro difusas se tiene el mismo problema, que en las redes multicapa wavelet, es necesario pre establecer el tamaño de la red.

Resulta entonces evidente la ventaja de usar wavelets Haar, que tomando como referencia el algoritmo de la transformada rápida wavelt, en donde si conocemos el tamaño del arreglo podemos determinar el número de ensanchamientos m, necesarios para aproximar una función sin perdida de información.

4.2.1. Red Neuronal Wavelet de una sola capa

La estructura de la RNWH esta basada en la serie wavelet

$$f(x) = a_0 \mathcal{X}(x) + \sum_{m=0}^{\infty} \sum_{n=0}^{2^m-1} d_j H_{m,n}(x). \tag{4.4}$$

donde H es la función wavelet Haar,

$$H_{n,m}(x) = \begin{cases} 1 \text{ si } m2^n \leq x < (m+\frac{1}{2})2^n, \\ -1 \text{ si } (m+\frac{1}{2})2^n \leq x < (m+1)2^n \\ 0 \text{ otro caso.} \end{cases} \tag{4.5}$$

Y para obtener los coeficientes nos basamos en el algoritmo FWT (3.57), que tiene como condición que el tamaño del arreglo a transformar debe tener 2^M valores, por lo que el valor máximo que toman los coeficientes de traslación n, y ensanchamiento m, depende del tamaño del arreglo, el valor máximo del ensanchamiento m igual a $M-1$.

Implementar una red neuronal wavelet Haar de una sola capa tiene la ventaja de conservar la estructura de una serie wavelet, sin embargo conservar esta estructura implica conservar los valores redundantes de la serie y tener una red neuronal con un número igual de neuronas que el tamaño del arreglo del que se aprende, no resulta práctico, sin embargo como se vera mas adelante, es posible establecer un algoritmo que reduzca el tamaño de la red sin perder calidad en la aproximación de la función f.

Cuadro 4.1;Valores de m y n, dependiendo el tamaño del arreglo.

Elementos del Arreglo	Ensanchamientos m	Total número de Tralaciones n	Neuronas
256	7	256	256
512	8	512	512
1024	9	1024	1024
2048	10	2048	2048
32768	14	32768	32768
262144	17	262144	262144

De la serie (4.4), podemos escribibir esta serie dividiendo por bloques de ensanchamiento,

$$\begin{array}{c}\hat{f}=\sum_{n=0}^{2^{M-1}-1}a_0\mathcal{X}(x)+\sum_{m=0}^{M_0}\sum_{n=0}^{2m-1}d_{m,n}\psi\left(2^m x-n\right)\\ +\sum_{m=0}^{M_1}\sum_{n=0}^{2^m-1}d_{m,n}\psi\left(2^m x-n\right)+...+\sum_{m=0}^{M_M}\sum_{n=0}^{2^m-1}d_{m,n}\psi\left(2^m x-n\right)\end{array} \tag{4.6}$$

cada bloque de la sumatoria representa un bloque de neuronas de nuestra red neuronal wavelet Haar, como se muestra en la Figura 4.3.

No resulta práctico tener una red neuronal donde el número de neuronas es igual al tamaño del arreglo a aproximar, por lo que en la mayoría de las aplicaciones una función es representada por una serie de Haar trunca, las características de la transformada Wavelet ha

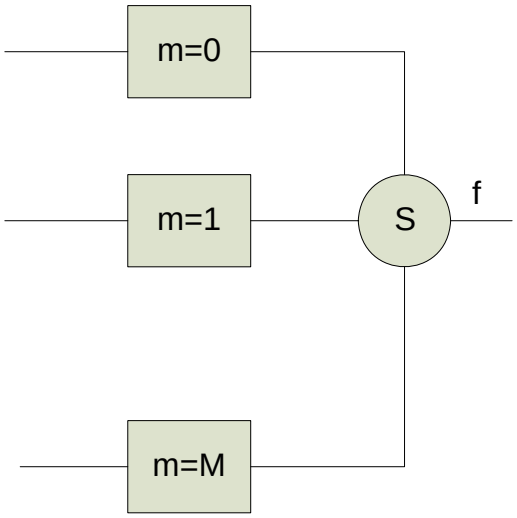


Figura 4.3: Bloques de Neuronas de la RNWH

sido ampliamente documentada, por lo que tener una red neuronal con un truncamiento adecuado nos permitirá eliminar ruido de nuestros datos de entrada y es muy útil en aplicaciones con datos dispersos. Escribiendo la serie wavelet Haar en forma matricial donde:

$$\hat{f}(k) = W^{*}\Psi(k), \tag{4.7}$$

resulta inmediato el implementar dicha aproximación como una red neuronal y ajustar los coeficientes wavelets de la serie como pesos sinapticos de una red neuronal, se escribe entonces la donde $\widehat{f}\,(k)$ es el vector de salida, con $W^{*}(k) \in R^{2^{M}\times 2^{M}}$, la matriz de pesos óptima y Ψ es el vector de funciones wavelets Haar.

Es necesario pre establecer el numero de neuronas que se utilizaran, lo que determinará el tamaño del error de truncamiento, este tipo de red neuronal posee la propiedad de determinar el error de truncamiento en base al número de neuronas, recordando que el número de neuronas está determinado por el valor de los coeficientes de traslación y ensanchamiento. Para obtener el valor de los coeficientes de la RNWH utilizamos el algoritmo Back Propagation.

4.2.2. Algoritmo de Aprendizaje Back Propagation

El Algoritmo de aprendizaje Back Propagation ha sido ampliamente utilizado para ajustar los pesos sinapticos de una red neuronal, su estabilidad en el aprendizaje ha sido probada, ver [20], para el caso de las RNWH este algoritmo funciona me manera eficiente, corrige el peso sinaptico W_k con el gradiente ΔW_k.

La ley de aprendizaje se escribe como:

$$W_{k+1} = W_k + \Delta W_k, \tag{4.8}$$

donde error a la salida de la RNWH se define como la diferencia con respecto a la función a seguir,

$$e(k) = \widehat{f}(k) - f(k). \tag{4.9}$$

y error cuadratico medio ε, se escribe como sigue:

$$\varepsilon = \frac{1}{2}\sum_{t=0}^{n} e(k)^2 \tag{4.10}$$

Re escribimos la ley de aprendizaje (4.8),

$$W_{k+1} = W_k - \eta \frac{\partial \varepsilon\,(k)}{\partial W_k}, \tag{4.11}$$

donde η es la razón de aprendizaje y $1 > \eta > 0$. El termino $\frac{\partial \varepsilon(k)}{\partial w_j(k)}$ se calcula como

$$\frac{\partial \varepsilon\,(k)}{\partial w_j\,(k)} = \frac{\partial \varepsilon\,(k)}{\partial e\,(k)} \frac{\partial e\,(k)}{\partial f\,(k)} \frac{\partial f\,(k)}{\partial w_j\,(k)}. \tag{4.12}$$

Las derivadas parciales de la ecuación (4.12), se calculan como sigue,

$$\begin{aligned} \frac{\partial \varepsilon}{\partial e} &= e\,(k)\,, \\ \frac{\partial e\,(k)}{\partial f\,(k)} &= -1, \\ \frac{\partial f\,(k)}{\partial W_j\,(k)} &= \psi\,(k)\,, \end{aligned}$$

por que finalmente escribimos (4.11) de la siguiente forma,

$$W_{k+1} = W_k - \eta\psi(k)e\,(k)\,. \tag{4.13}$$

El entrenamiento de una red neuronal wavelet Haar requiere considerar varios elementos, como lo es el valor inicial que tendrán los pesos de la red, en la implementación de la RNWH se tomaron valores generados aleatoriamente entre $[0, 1]$, de igual forma se requiere establecer el valor de la razón de aprendizaje, es bien conocido que el valor de la razón de aprendizaje η toma valores $0 < \eta < 1$. En las simulaciones presentadas en esta tesis se utilizo $\eta = 0{,}1$ que proporciona una aproximación suave, existen diversos estudios sobre que valor es el adecuado, sin embargo depende de cada conjunto de entrenamiento.

Otro punto importante a señalar es como decidir cuando detener el algoritmo de aprendizaje, existen varios criterios, decidir en torno a un valor mínimo tolerado del error cuadrático medio de aproximación ε, otro criterio es determinar que el algoritmo se detiene después de ser entrenada la RNWH por un número determinado de valores de entrenamiento, como se explicara de una manera detallada en la posterior sección, Optimización de la Red Neuronal Wavelet Haar, en esta tesis se propone un criterio para detener el algoritmo de aprendizaje basado en los cambios en el gradiente de aprendizaje para cada coeficiente wavelet.

4.3. Modelado de Series en Tiempo usando Redes Neuronales Wavelet Haar

Una red multicapa feed forward con una función de activación sigmoide es capaz de funcionar como un aproximador universal a funciones continuas, el primero en demostrar esta habilidad fue Cybenño [18], de igual forma las redes neuronales con funciones radiales básicas (RNB) son consideradas como aproximadores universales de funciones [24], y tiene la forma siguiente

$$F(u) = \sum_{i=1}^{m} w_i G\left(\|u_i - c_i\|_{r_i}^2\right),$$

donde $G(r)$ es una función base, la mas usada es la del tipo Gausiana $\exp(-r^2/2)$.

Otro tipo de red neuronal que ha sido presentada como aproximador son las redes de Fourier [56], sin embargo este tipo de red tienen las limitantes intrínsecas de las series de Fourier, las cuales ya se enlistan en el capítulo anterior (2.3), en [56] se hace la aproximación a la serie de tiempo que representa la radiación solar haciendo 5601 muestras, usando una red neuronal de Fourier de 24 neuronas que corresponden a 12 coeficientes a_n,y 12 coeficientes b_n, esta serie presenta un error cuadratico medio igual a 0,005039, en la Figura 4.4 se muestra la aproximación realizada.

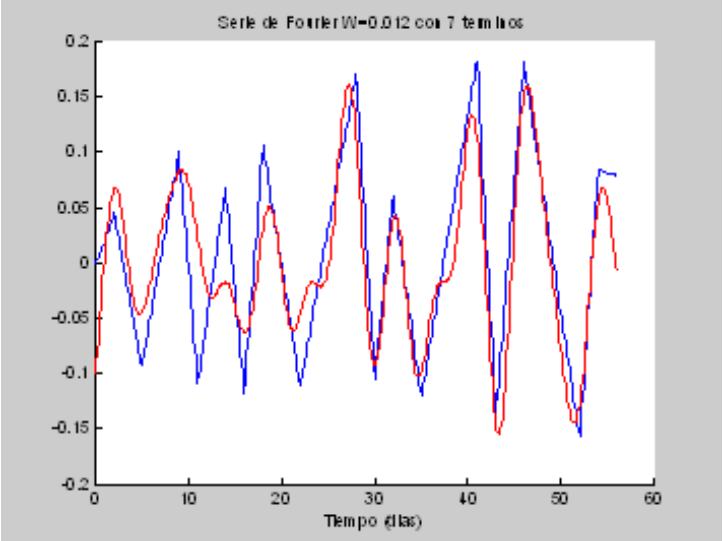


Figura 4.4: Serie de Fourier de 7 terminos.

Desde que apareció la teoría de Wavelets, fue adoptada por los investigadores en el área de redes neuronales, y se ha mostrado atractiva para aproximar funciones tanto periódicas así como no periódicas y son muchas sus ventajas sobre otros métodos de aproximación. En las RNW el problema de aproximación esta dirigido a encontrar el valor óptimo de los coeficientes wavelets de la red y determinar la resolución con la que se realizará la aproximación, en la literatura se encuentran ejemplos del uso de las RNW como aproximadores universales [25], sin embargo en las redes neuronales wavelets se ha puesto poca atención el uso de la función

wavelet Haar, en el campo de las redes neuronales existen pocos trabajos que utilicen la wavelet Haar se puede ver [39], donde hace la comparación del desempeño en aproximación de funciones tanto de la red neuronal wavelet Haar (RNWH) y las RNB, en ese trabajo se utiliza un algoritmo secuencial de aprendizaje sin embargo no se presenta una prueba analítica en la estabilidad de ese algoritmo.

Comparando la aproximación hecha por la red de Fourier en [56], con una aproximación realizada por una red neuronal wavelet Haar, se muestran tres simulaciones, en la primera se fija el tamaño de la red a 16 neuronas, ver la Figura 4.5, que utiliza un valor de ensanchamiento $m = 3$, lo que significa la red neuronal esta formada por neuronas que representan los coeficientes $c_0, m_{00}, m_{1\frac{1}{2}}, m_{11}, m_{2\frac{1}{4}}, m_{2\frac{1}{2}}, m_{2\frac{3}{4}}, m_{21}, m_{3\frac{1}{8}}, m_{3\frac{2}{8}}, m_{3\frac{3}{8}}, m_{3\frac{1}{2}}, m_{3\frac{5}{8}}, m_{3\frac{6}{8}}, m_{3\frac{7}{8}}, m_{31}$. Se hace la aproximación utilizando un valor de ensanchamiento $m = 4$, ver la Figura 4.6), la cual esta esta formada por 32 neuronas, En la Figura 4.7, se muestra la aproximación realizada por una RNWH con un coeficiente de ensanchamiento $m = 5$, dicha red esta formada por 32 neuronas

Como se observa en la Figuras 4.5, 4.6 y 4.7, y como se ha explicado anteriormente, el error de aproximación de una red neuronal wavelet disminuye si se aumenta el número de neuronas, es decir aumenta el valor del ensanchamiento m, y hacer aproximación con una resolución mas fina. Sin embargo aun cuando el algoritmo de aprendizaje y la estructura de esta red neuronal wavelet Haar es simple, aumentar el número de neuronas hace una mayor carga de computo y cantidad de operaciones que se realizan, la determinación del tamaño de la red es un problema abierto, así como el establecer el momento en el que se debe detener el entrenamiento, en esta tesis se propone un algoritmo para determinar una RNWH con un número óptimo de neuronas.

Sea un sistema no lineal con la siguiente forma:

$$x(k+1) = f\left[x(k), u(k)\right] \tag{4.14}$$

$$y(k) = h\left[x(k)\right] \tag{4.15}$$

donde $u(k) \in \mathbb{R}^m$ es el vector de entrada, $x(k) \in \mathbb{R}^n$ es el vector de estados, y $y(k) \in \mathbb{R}^l$ es el

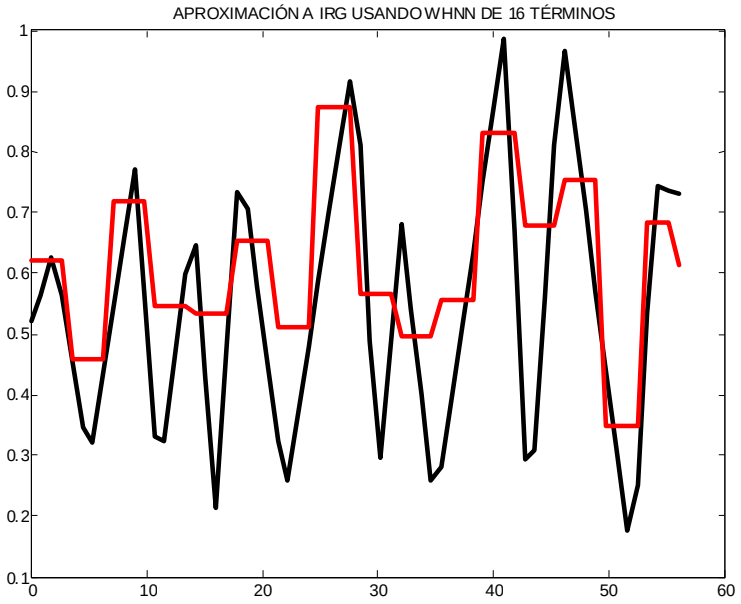


Figura 4.5: Aproximación a IRG utilizando una red neuronal wavelet Haar con ensanchamiento $m = 3$.

vector de salida. f y h son en general funciones suaves no lineales, $f, h \in \mathbb{C}^{\infty}$, $k = 0, 1, ...N-1$.

Un sistema no lineal se puede representar en función de observaciones pasadas de su entrada y salida, cuatro modelos para modelar sistemas discretos son propuestos en [40]

ModeloI:

$$y_p(k+1) = \sum_{i=0}^{n-1} \alpha_i y_p(k-i) + \\ + g\left[u(k), u(k-1), ..., u\left(k-m+1\right)\right] \tag{4.16}$$

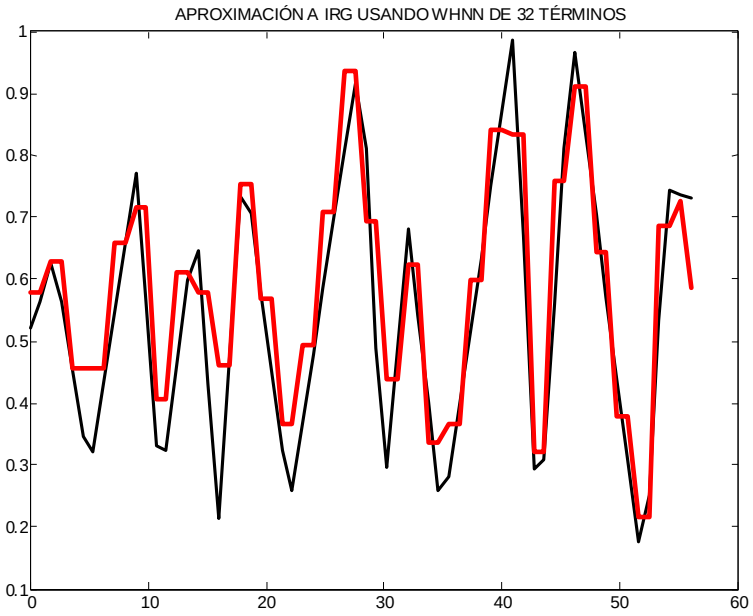


Figura 4.6: Aproximación a IRG utilizando una red neuronal wavelet Haar con ensanchamiento $m = 4$.

Modelo II:

$$y_p(k+1) = f\left[y_p(k), y_p(k-1), ..., y_p\left(k-n+1\right)\right] + \\ + \sum_{i=0}^{m-1} \beta_i u\left(k-i\right) \tag{4.17}$$

Modelo III:

$$y_p(k+1) = f\left[y_p(k), y_p(k-1), ..., y_p\left(k-n+1\right)\right] + \\ + g\left[u(k), u(k-1), ..., u\left(k-m+1\right)\right] \tag{4.18}$$

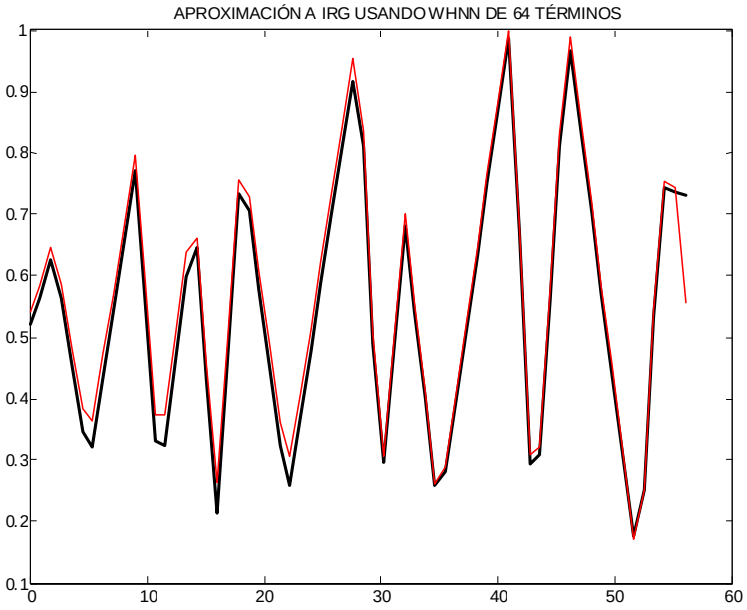


Figura 4.7: Aproximación a IRG utilizando una red neuronal wavelet Haar con ensanchamiento $m = 5$.

Modelo IV:

$$y_p(k+1) = f\left[y_p(k), y_p(k-1), ..., y_p\left(k-n+1\right); \right. \\ \left. u(k), u(k-1), ..., u\left(k-m+1\right)\right] \tag{4.19}$$

Cada uno de estos cuatro modelos (4.16), (4.17), (4.18) y (4.19), tienen el mismo comportamiento de entrada - salida que el sistema original (4.15), y esos modelos son fácilmente representados por una red neuronal del tipo feedforward, tanto la entrada como la salida de la red neuronal esta disponible para todo tiempo [40].

Una red Neuronal Wavelet es representada como:

$$\hat{y}(k) = W(k)\Psi(k) \tag{4.20}$$

donde $\hat{y}(k)$ es la salida escalar de la red, $W(k) \in \mathbb{R}^{1xn}$ es la matriz de coeficientes wavelets, $\Psi(k) \in \mathbb{R}^{nx1}$ es la matriz de funciones wavelets. Del Modelo IV (4.19), un sistema no lineal se puede escribir como:

$$y(k) = \Phi\left[\mathbf{X}\left(k\right)\right], \tag{4.21}$$

donde

$$\begin{aligned}\mathbf{X}\left(k\right) = [&y_p(k), y_p(k-1), ..., y_p\left(k-n+1\right); \\ &u(k), u(k-1), ..., u\left(k-m+1\right)].\end{aligned} \tag{4.22}$$

$\Phi\left(\cdot\right)$ es una ecuación no lineal desconocida que representa la dinámica de la planta, como hemos visto, es posible aproximar cualquier función en un intervalo de tiempo determinado mediante una serie wavelet,

$$y(k) = W^*(k)\Psi(k) - \mu(k),$$

donde W^* corresponde a los coeficientes de la serie y Ψ las funciones wavelets, $\mu(k)$ es el error de modelado, que sabemos tiende a cero si el intervalo de tiempo es amplio y la resolución de la serie es la adecuada. Estableciendo el error de neuro identificación se tiene:

$$e(k) = \widehat{y}(k) - y(k)$$

Expandiendo las series Wavelets,

$$e(k) = W(k)\Psi(k) - W^*(k)\Psi(k) + \mu(k),$$

para finalmente escribir el error como:

$$e(k) = \widetilde{W}\left(k\right)\psi\left(k\right) + \zeta\left(k\right), \tag{4.23}$$

donde

$$\widetilde{W}(k) = W(k) - W^*(k).$$

con $\zeta\left(k\right) = \varepsilon\left(k\right) + \mu\left(k\right)$, $\varepsilon\left(k\right)$ es el error de aproximación de la RNW, $\varepsilon\left(k\right)$ es acotada $\varepsilon^2\left(k\right) \leq \overline{\varepsilon}$, $\overline{\varepsilon}$ es una constante positiva desconocida.

4.4. Optimización de la Red Neuronal Wavelet Haar

Un problema de gran importancia en el campo de las redes neuronales artificiales (RNA), consiste en el desarrollo de algoritmos que ayuden a la determinación del número óptimo de neuronas necesarias en su estructura para lograr una tarea dada. Para determinar el número de neuronas en una red neuronal existen varias técnicas heurísticas [32], que sin embargo no son generalizadas y dependiendo la función a aproximar será la técnica utilizada [33], en las RNA generalmente el que se tenga un número grande de neuronas no garantiza que la red efectuara una mejor aproximación.

Uno de los algoritmos para determinar el número de neuronas es la técnica Forward Select, la cual consiste en añadir nodos a la red, uno a la vez en base a un criterio. Los nodos se añaden a la red y se verifica el criterio de selección, es muy frecuente utilizar como criterio la validación cruzada.

Otro método es Optimal Pruned ELM (OP-ELM) [34], el cual establece un número inicial muy elevado de neuronas ocultas y mediante el algoritmo Least Angle Regression (LARS) [35] elimina aquellas neuronas que no son útiles para resolver el problema de mínimos cuadrados. Para ello se ordena el conjunto posible de neuronas, conforme a su importancia. La eliminación de neuronas se realiza mediante la técnica de validación cruzada.

Las Redes Neuronales Wavelets (RNW) ofrecen una estructura que comporta como un aproximador universal, el número de neuronas en este tipo de red esta determinado por el número de traslaciones y ensanchamientos de la función wavelet madre. Se tienen dos formas de tratar el problema de establecer el valor de los parámetros de traslación y ensanchamiento:

1. RNW Multicapa [28],[25],[37]. En esté planteamiento se tratan como desconocidos los parámetros de traslación, ensanchamiento y los coeficientes de la RNW, para esté tipo de planteamiento se requiere el uso de redes neuronales multicapa y se consideran los parámetros de traslación y el ensanchamiento como una capa oculta.

2. WNN Fixed [36], [38]. En este planteamiento se utiliza una red neuronal de una sola capa donde se predetermina el valor de los parámetros de traslación y ensanchamiento.

Sin embargo no está resuelto el problema que implica cuantas neuronas utilizar, dado que con las redes multi capa aun es necesario pre determinar el tamaño de las capas ocultas y cuantas se utilizaran. Los trabajos existentes utilizan funciones wavelets continuas, y plantean el problema de aproximar una función con una serie wavelet continua, necesitando para ello obtener el valor de los la transformación wavelet continua. La red RNW Fixed es una red neuronal de una sola capa y considera una serie trunca wavelet, donde se determina bajo un criterio el valor de los coeficiente de ensanchamiento y traslación. Al ser una red de una sola capa hace que la formación de la RNW Fixed sea mucho más fácil en comparación con RNW de múltiples capas.

En una serie wavelet los coeficientes necesarios para la reconstrucción de cualquier función son fijos y únicos, por la propiedad de multi resolución cada elemento de la serie es independiente de los demás. Este postulado no permite establecer un algoritmo que determine cuando detener el aprendizaje en cada neurona de manera independiente a las demás neuronas sin importar el proceso de actualización de las otras. De la misma forma cada segmento de la serie con el mismo valor de ensanchamiento de la wavelet se puede ver como una red neuronal independiente de otras redes que tienen funciones Haar con otro ensanchamiento, así, podemos crear un algoritmo evolutivo en el que la RNWH vaya creciendo en tamaño aumentando un bloque de ensanchamiento, hasta llegar a un valor del error deseado. El procedimiento de análisis con funciones wavelets es implementado mediante la traslación y ensanchamiento de la wavelet madre, la transformada wavelet permite analizar una componente de frecuencia a distintas escalas, al reconstruir una transformada wavelet a la función original se tiene una serie con mayor numero de datos que el arreglo original, situación producto de las redundancias necesarias para el análisis multi escala.

Sin embargo en redes neuronales estas redundancias no resultan útiles, en la construcción de una red neuronal wavelet el número de neuronas esta determinado por la traslación y el valor del coeficiente de ensanchamiento de la función wavelet, hay dos formas de establecer el valor de los coeficientes de traslación y ensanchamiento, uno es considerarlos como una capa oculta de la red y otra es establecer valores fijos para el ensanchamiento y traslación, sin embargo aun queda el problema de establecer el numero y que de valores fijos que se

tomaran. Como se ha visto en el algoritmo de la transformada rápida wavelet, la resolución mas pequeña que se utiliza depende del tamaño del arreglo, un arreglo de 2^γ datos tendrá $\gamma - 1$ distintos ensanchamientos. Podemos ver de la teoría wavelet que la construcción de una base $V \in L^2$, ocurre con espacios ortonormales W, (3.23), i.e. existe independencia lineal entre los vectores que forman la base W, que significa que podemos calcular cada coeficiente w_{mn} de la red neuronal wavelet Haar de manera independiente, sin que exista interferencia en el algoritmo de aprendizaje. Proponemos un algoritmo evolutivo para determinar el tamaño óptimo de la red bajo un índice de error determinado. Este algoritmo tiene dos condiciones para determinar el crecimiento y el aprendizaje de la red neuronal. Una condición de error de aproximación y una condición para detener el aprendizaje en cada neurona. Iniciamos el algoritmo determinando el error cuadratico medio mas grande tolerado E, para la aproximación realizada por la red neuronal y partimos con una red neuronal wavelet Haar de solo dos neuronas es decir con la resolución mas gruesa, $m = 0$. Para cada época de entrenamiento i, se tiene el índice J_i que depende del error de aproximación de la red neuronal.

$$J_i = \left(\frac{\sum e_i(k)^2}{K} \right)^{\frac{1}{2}}. \tag{4.24}$$

y el gradiente de aprendizaje para cada coeficiente wavelet se define como:

$$I_{i,w_{n,m}} = \frac{w_{n,m_i}(k+1) - w_{n,m_i}(k)}{\Delta k}. \tag{4.25}$$

1. Se establece el error máximo tolerado E.

2. Inicia el entrenamiento para un ensanchamiento $m = 0$.

3. Para cada paso k del entrenamiento, se verifica el valor de I_{iw_n} para cada neurona, si es cercano a cero se fija el valor de ese coeficiente w^*_{mn}.

4. Al finalizar una epoca de entrenamiento i, si no se alcanza el valor esperado y si no se ha llegado a un mínimo de error J_i, se realiza nuevamente el entrenamiento con la misma estructura de la red pero sin entrenar los valores w^*_{mn} ya fijados.

5. Cuando todos los coeficientes w^*_{mn} se han fijado y se alcanza un mínimo en el error J_i, la red crecerá en su estructura, con 2^m nuevas neuronas, con $m = \{0, 1, ..., \gamma - 1\}$ y se regresa al paso 3.

6. Si se ha alcanzado el error mínimo tolerado E, el algoritmo se detiene.

Al finalizar el algoritmo, es posible reducir el tamaño de la red eliminando las neuronas con poco peso. Para mostrar la efectividad de este algoritmo se muestran dos ejemplos.

4.5. Estabilidad de Redes Neuronales Wavelet Haar

Se tiene el siguiente sistema no lineal en tiempo discreto

$$x(k+1) = f\left[x\left(k\right), u\left(k\right)\right], \quad y(k) = h\left[x\left(k\right)\right] \tag{4.26}$$

donde $u\left(k\right) \in \mathbb{R}^m$ es el vector de entrada, $x\left(k\right) \in \Re^n$ es el vector de estados, y $y\left(k\right) \in \Re^l$ Es el vector de salida. f y h son funciones suaves no lineales $f, h \in C^\infty$. Vamos a ver las siguientes definiciones.

Definición 4.1 *Un sistema (4.26) se dice que es globalmente estable de la entrada a los estados, si existe una $\mathcal{K}$-función $\gamma(\cdot)$ (continua y estrictamente creciente $\gamma(0) = 0$) y $\mathcal{KL}$ -función $\beta\left(\cdot\right)$ ($\mathcal{K}$-función y $\lim_{s_k \to \infty} \beta\left(s_k\right) = 0$) , tal que, para cada $u \in L_\infty$ $(\sup\{\|u(k)\|\} < \infty)$ y para cada estado inicial $x^0 \in R^n$, cumpla con*

$$\left\|x\left(k, x^0, u\left(k\right)\right)\right\| \leq \beta\left(\left\|x^0\right\|, k\right) + \gamma\left(\left\|u\left(k\right)\right\|\right)$$

Definición 4.2 *Una función suave $V : \Re^n \to \Re \geq 0$ es llamada una función suave ISS-Lyapunov del sistema (4.26) si: (a) Existe $\mathcal{K}_\infty$-función $\alpha_1(\cdot)$ y $\alpha_2(\cdot)$ ($\mathcal{K}$-función y $\lim_{s_k \to \infty} \beta\left(s_k\right) = \infty$) tal que*

$$\alpha_1(s) \leq V\left(s\right) \leq \alpha_2(s), \quad \forall s \in \Re^n$$

(b) Existe una $\mathcal{K}_\infty$-función $\alpha_3(\cdot)$ y una $\mathcal{K}$-función $\alpha_4(\cdot)$ tal que

$$V_{k+1} - V_k \leq -\alpha_3(\|x\left(k\right)\|) + \alpha_4(\|u\left(k\right)\|), \quad \textit{para todo } x\left(k\right) \in \Re^n, u\left(k\right) \in \Re^m$$

Teorema 4.1 *Para un sistema discreto no lineal, los siguiente enunciados son equivalentes [19]*

- *Estabilidad de entrada a los estados (input-to-state stability, ISS).*
- *Es robustamente estable.*
- *Admite una función suave de Lyapunov (ISS).*

Comentario 4.1 *Si un sistema no lineal estable de la entrada a los estados (IIS), el comportamiento del sistema es acotado cuando sus entradas son acotadas.*

De (4.26) se tiene

$$\begin{aligned} y(k) = h\left[x\left(k\right)\right] := F_1\left[x\left(k\right)\right], \quad y(k+1) = h\left[f\left[x\left(k\right), u\left(k\right)\right]\right] := F_2\left[x\left(k\right), u\left(k\right)\right] \\ y(k+n-1) := F_n\left[x\left(k\right), u\left(k\right), u(k+1)\cdots u(k+n-2)\right] \end{aligned} \tag{4.27}$$

mostrando que

$$\begin{aligned} Y(k) = \left[y\left(k\right), y\left(k+1\right), \cdots y\left(k+n-1\right)\right]^T, \\ U(k) = \left[u\left(k\right), u\left(k+1\right), \cdots u\left(k+n-2\right)\right]^T, \end{aligned}$$

entonces $Y(k) = F\left[x\left(k\right), U(k)\right], F = \left[F_1 \cdots F_n\right]^T$. Si $\frac{\partial Y}{\partial x}$ es no singular a $x = 0, U = 0$, (4.27)

puede expresarse como $x(k+1) = g\left[Y(k+1), U(k+1)\right]$. Que nos lleva a el modelo NARMAX

$$y(k) = h\left[x\left(k\right)\right] = \Phi\left[y\left(k-1\right), y\left(k-2\right), \cdots u\left(k-1\right), u\left(k-2\right), \cdots\right] = \Phi\left[X\left(k\right)\right] \tag{4.28}$$

donde

$$X\left(k\right) = \left[y\left(k-1\right), y\left(k-2\right), \cdots u\left(k-d\right), u\left(k-d-1\right), \cdots\right]^T \tag{4.29}$$

$\Phi\left(\cdot\right)$ es una ecuación no lineal desconocida que representa la dinámica de una planta, $u\left(k\right)$ y $y\left(k\right)$ son escalares medibles de la entrada y la salida, d es el retardo

El siguiente teorema demuestra la estabilidad del algoritmo de aprendizaje de un Red Wavelet de una sola capa en tiempo discreto.

Teorema 4.2 *Si usamos una Red Neuronal Wavelet de una sola capa (4.20), para identificar una planta no lineal (4.26), la siguiente ley de aprendizaje puede hacer identificación del error* $e(k)$ *en el sentido de estabilidad* L_∞.

$$W(k+1) = W(k) - \eta_k e(k)\Psi(k) \tag{4.30}$$

donde

$$\eta_k = \frac{\eta}{1 + \|\Psi(k)\|^2}; 0 < \eta \leq 1$$

Demostración. Seleccionamos una función candidata de Lyapunov de la forma:

$$V_k = \left\|\widetilde{W}_k\right\|^2 = \sum_{i=1}^{n} \widetilde{w}_i^2 = tr\left\{\widetilde{W}^T\widetilde{W}\right\} \tag{4.31}$$

De la ley de aprendizaje (4.30)

$$\begin{aligned} \widetilde{W}_{k+1} &= \widetilde{W}_k - \eta_k e(k)\Psi(k) \\ \Delta V_k &= V_{k+1} - V_k \\ &= \left\|\widetilde{W}_k - \eta_k e(k)\Psi_k\right\|^2 - \left\|\widetilde{W}_k\right\|^2 \\ &= \eta_k^2 e^2(k)\left\|\Psi(k)\right\|^2 - 2\eta_k e(k)\left\|\widetilde{W}_k\Psi(k)\right\| \end{aligned} \tag{4.32}$$

De (4.23) y considerando que $0 < \eta \leq 1, 0 \leq \eta_k < \eta \leq 1$

$$\begin{aligned} \Delta V_k &= \eta_k^2 e^2(k)\left\|\Psi(k)\right\|^2 - 2\eta_k e(k)\left\|e(k) - \zeta(k)\right\| \\ &\leq \eta_k^2 e^2(k)\left\|\Psi(k)\right\|^2 - 2\eta_k e^2(k) + \eta_k e^2(k) + \eta_k \zeta^2(k) \\ &= -\eta_k\left[1 - \frac{\eta\left\|\Psi(k)\right\|^2}{1 + \left\|\Psi(k)\right\|^2}\right] e^2(k) + \eta_k \zeta^2(k) \\ &\leq -\pi e^2(k) + \eta\zeta^2(k) \end{aligned}$$

donde

$$\pi = \frac{\eta}{1+\kappa}\left[1 - \frac{\kappa}{1+\kappa}\right] > 0$$
$$\kappa = \max_k \|\Psi(k)\|^2$$

dado que Ψ es una matriz de funciones Wavelet Ortonormales

$$\Delta V_k = -\eta(1-\eta)e^2(k) + \eta\zeta^2(k)$$

entonces

$$n \min\left(\widetilde{w}_i^2\right) \leq V_k \leq n \max\left(\widetilde{w}_i^2\right)$$

donde $n \times \min(\widetilde{w}_i^2)$ y $n \times \max(\widetilde{w}_i^2)$ son funciones $\mathcal{K}_\infty$, y $\eta(1-\eta)e^2(k)$ es una función $\mathcal{K}_\infty$, $\eta\zeta^2(k)$ es una función $\mathcal{K}$. Por (4.23) y (4.31) sabemos que V_k es función de $e(k)$ y $\zeta(k)$, entonces V_k cumple con ser una función ISS-Lyapunov de la *Definición 4.2*. Del *Teorema* 4.1, la dinámica del error de identificación es input-to-state estable. La "Entrada " corresponde al modelo del error $\zeta(k) = \varepsilon(k) + \mu(k)$, el "Estado" corresponde al error de identificación $e(k)$. Como la "Entrada" $\zeta(k)$ es acotado y la dinámica es ISS, el "Estado" $e(k)$ es también acotado. ■

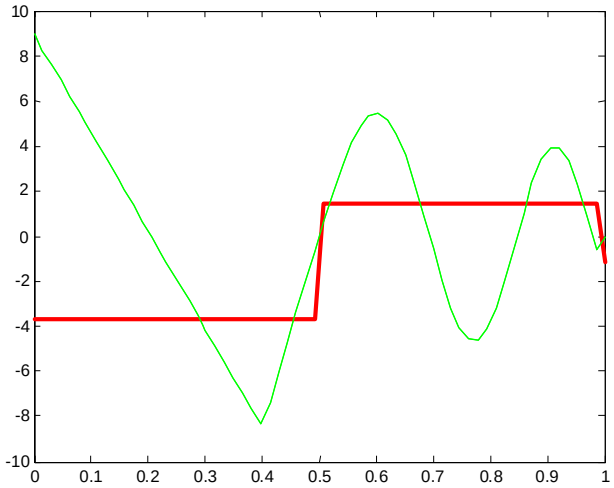
4.6. Simulaciones

4.6.1. Redes red neuronal wavelet Haar con la red neuronal wavelet

En [28] se muestra el ejemplo de una función a aproximar (4.33), que en varios artículos se toma como referencia para compara resultados de aproximación entre distintas RNW, [39],[54].

$$F(k) = \begin{cases} -2{,}186k - 12{,}864 \quad if \quad -10 \leq k < -2 \\ 4{,}246k \quad if \quad -2 \leq k < 0 \\ 10e^{-0{,}05k-0{,}5}\sin\left[(0{,}03k + 0{,}7)\,k\right] \quad if \quad 0 \leq k \leq 10 \end{cases} \tag{4.33}$$

En nuestro caso realizamos la aproximación a la función F (4.33), para ejemplificar la implementación de la RNWH como aproximador en distintas resoluciones, las cuales son dependientes del valor del coeficiente de ensanchamiento m. La primera aproximación a F se realiza con el ensanchamiento mas grueso, $m = 0$ lo que implica una red neuronal con solo dos neurona, una neurona que se adapta al valor del coeficiente c_o y otra que aparte al valor del coeficiente m_0, la aproximación se muestra en la Figura **??**.



Aproximación a f con una resolución $m = 0$.

Una segunda aproximación se realiza con la resolución $m = 1$, que significa se utiliza una red neuronal con cuatro neuronas, que representan los coeficientes $c_0, m_{00}, m_{1\frac{1}{2}}, m_{11}$, está aproximación se muestra en Figura 4.9,.

La tercera aproximación se realiza con un truncamiento al ensanchamiento $m = 2$, lo que significa que la red neuronal hará una aproximación utilizando ocho neuronas, que corresponde a los coeficientes $c_0, m_{00}, m_{1\frac{1}{2}}, m_{11}, m_{2\frac{1}{4}}, m_{2\frac{1}{2}}, m_{2\frac{3}{4}}, m_{21}$, la aproximación se muestra en la Figura 4.10,

La cuarta aproximación tiene un truncamiento al valor del coeficiente de ensanchamien-

to $m = 3$, que representa una red neuronal con 16 neuronas, que corresponde a los coeficientes $c_0, m_{00}, m_{1\frac{1}{2}}, m_{11}, m_{2\frac{1}{4}}, m_{2\frac{1}{2}}, m_{2\frac{3}{4}}, m_{21}, m_{3\frac{1}{8}}, m_{3\frac{2}{8}}, m_{3\frac{3}{8}}, m_{3\frac{1}{2}}, m_{3\frac{5}{8}}, m_{3\frac{6}{8}}, m_{3\frac{7}{8}}, m_{31}$, la aproximación que hace esta RNWH se muestra en la Figura 4.11),

En una quinta aproximación el truncamiento se realiza con el coeficiente de ensanchamiento $m = 4$, por lo que la red neuronal estará compuesta de 32 neuronas, la aproximación se muestra en la Figura 4.12),

Hacemos una sexta aproximación truncando la serie a un valor del coeficiente de ensanchamiento $m = 5$, con lo que se tendrá una RNWH con 64 neuronas, que corresponde a los coeficientes $c_0, m_{00}, m_{1\frac{1}{2}}, m_{11}, m_{2\frac{1}{4}}, m_{2\frac{1}{2}}, m_{2\frac{3}{4}}, m_{21}, ..., m_{5\frac{1}{64}}, m_{5\frac{2}{64}}, m_{5\frac{3}{64}}, ..., m_{5\frac{62}{64}}, m_{5\frac{63}{64}}, m_{51}$, la aproximación se muestra en la Figura 4.11),

Para esté ejemplo se utilizaron 2^{13} pares de entrenamiento, si quisiéramos hacer una aproximación sin error por truncamiento, tendríamos una RNWH de 16383 neuronas, situación que no es práctica, en la tabla mostramos un comparativo entre las distintas aproximaciones realizadas por las distintas RNWH, comparando los distintos errores cuadratico medios de cada aproximación y el número de épocas de entrenamiento necesarias para obtener el primer mínimo, todas las redes usaron $\eta = 0{,}1$.

Cuadro 4.2: Comparación de distintas RNWH.

m	Neurónas	Error	Épocas
0	2	2.332554	2
1	4	1.893018	19
2	8	0.7970679	5
3	16	0.2943906	10
4	32	0.197417	10
5	64	0.01861065	8

4.6.2. Modelado a la paridad Peso Mexicano y Dolar Estadounidense

Para ejemplificar el algoritmo utilizamos los datos de la paridad peso Mexicano contra el Dolar Estadounidense durante 14364 días, estos datos se tomaron de la página web del

Banco de México ver [31].

Iniciamos el algoritmo estableciendo el error máximo tolerado $E = 0{,}5$, el algoritmo inicia con la resolución mas gruesa $m = 0$, el menor error obtenido de esta red wavelet Haar de dos neuronas es $J = 4$, en la Figura 4.14 observamos la aproximación.

Al alcanzar un mínimo local el algoritmo aumenta la resolución ahora con $m = 1$, que representa una red de cuatro neuronas, y la mejor aproximación alcanzado produce una error $J = 3{,}628578$. La aproximación podemos observarla en la Figura 4.15.

El siguiente ciclo incrementa la resolución a $m = 2$, lo cual significa que se construye una RNWH de ocho neuronas y esta red obtiene un error mínimo de aproximación de 2,058947, la aproximación puede observarse en la Figura 4.16.

La resolución $m = 5$, significa que la RNWH es construida con la resolución $m = 4$, que significa que la red contiene 32 neuronas, y ofrece una aproximación con un error acumulado de $7{,}663324e - 001$.La resolución $m = 8$, entrega una red de 256 neuronas, y presenta un error acumulado de $4{,}843117e - 001$.

Con el valor de $m = 7$, se alcanza el valor esperado para el índice J, el algoritmo podía continuar y encontrar un índice mas pequeño, pero implicaría que la red aumentará su tamaño.

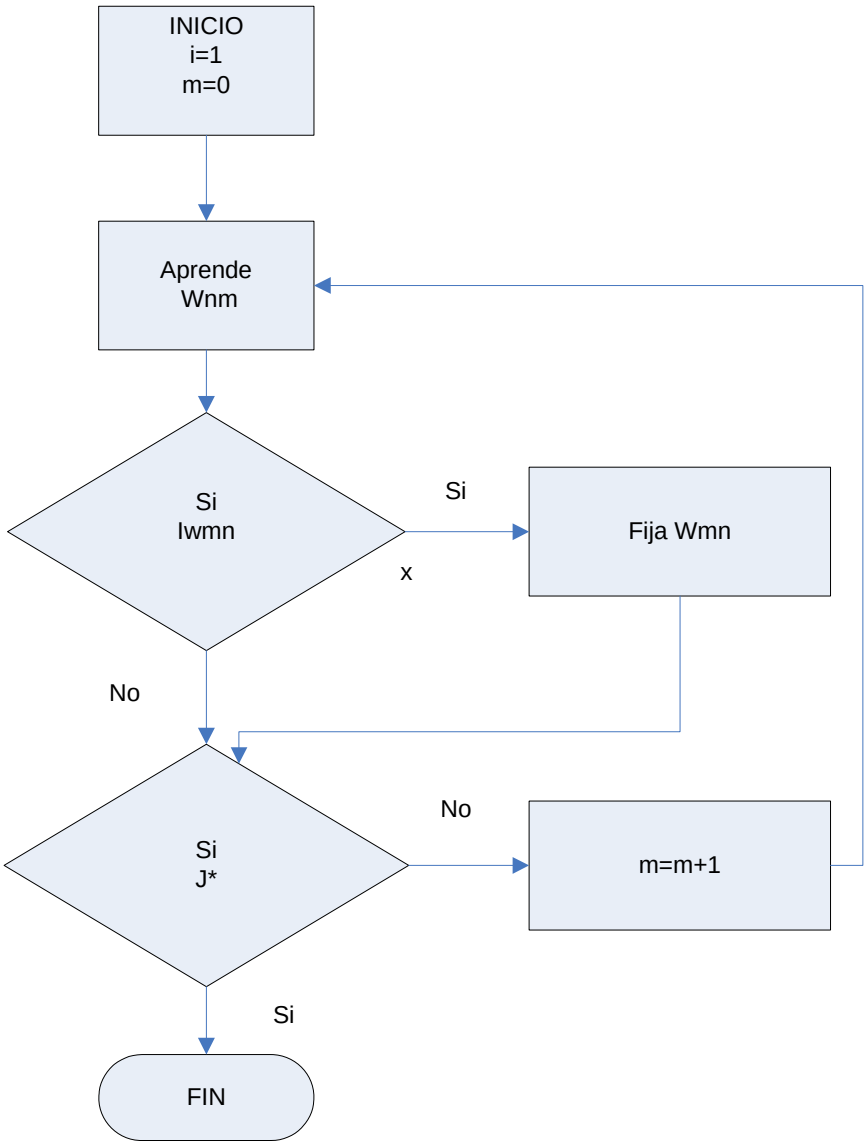


Figura 4.8: Optimization Algorithm

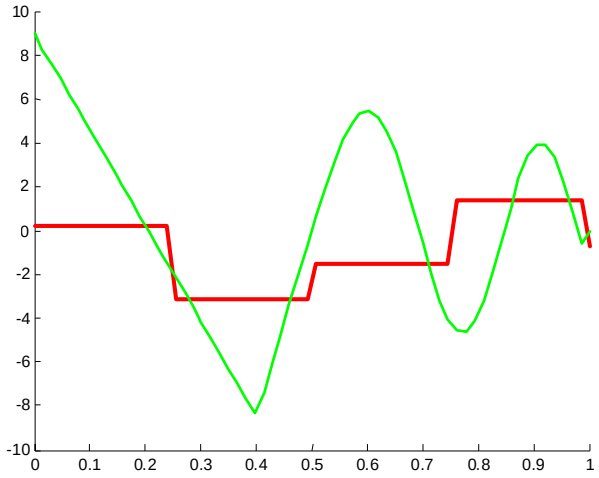


Figura 4.9: Aproximación a f con una resolución $m = 1$.

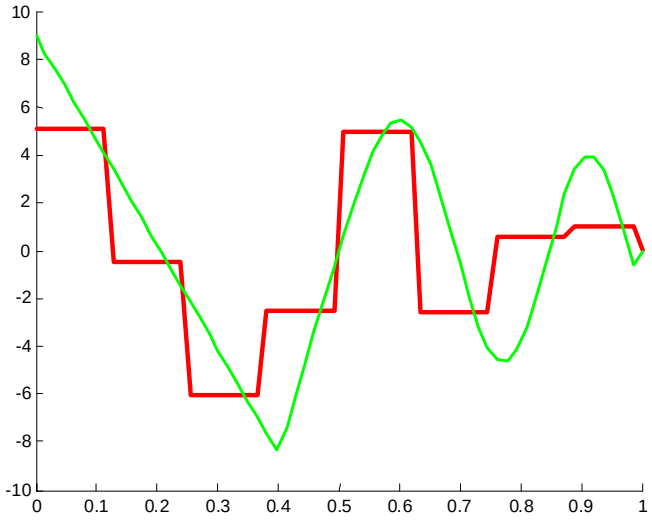


Figura 4.10: Aproximación a f con una resolución $m = 2$.

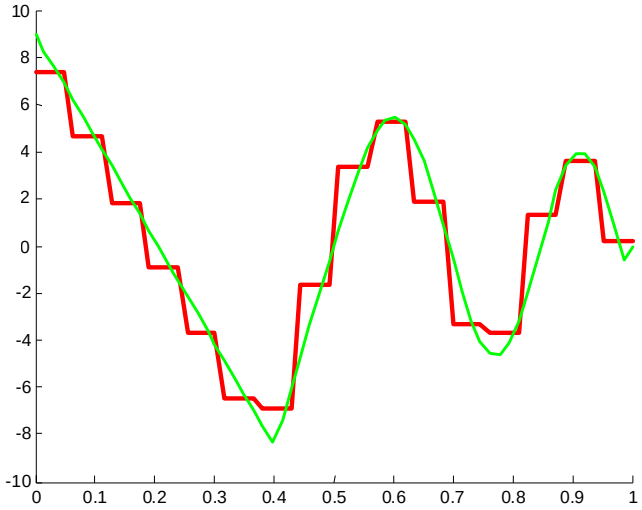


Figura 4.11: Aproximación a f con una resolución $m = 3$.

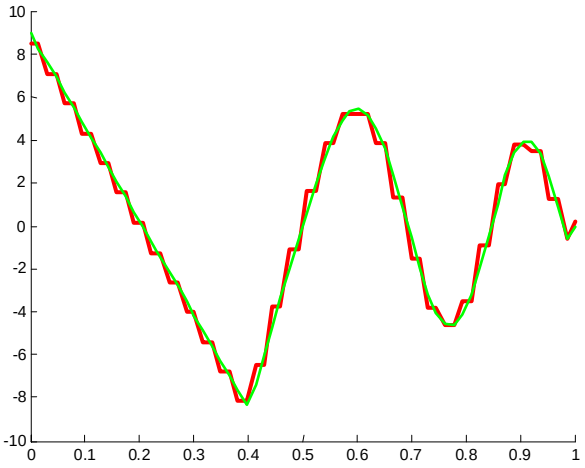


Figura 4.12: Aproximación a f con una resolución $m = 4$.

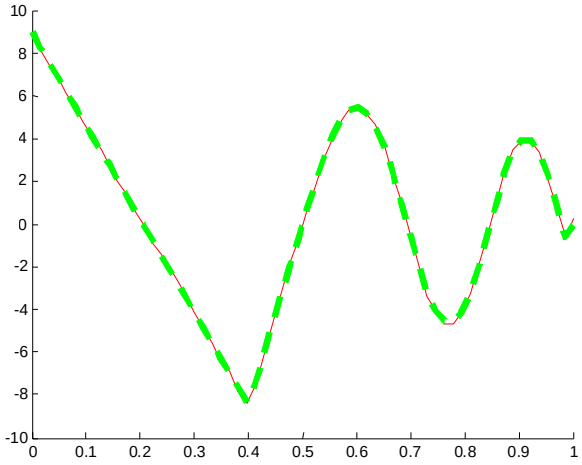


Figura 4.13: Aproximación a f con una resolución $m = 5$.

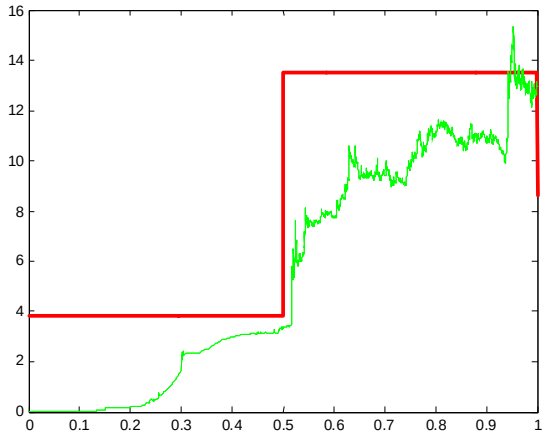


Figura 4.14: Aproximación con una resolución $m = 0$.

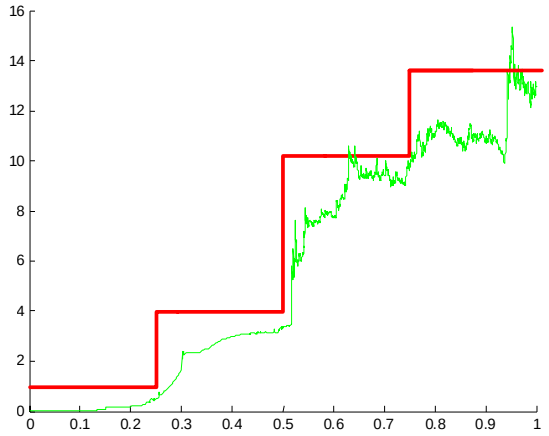


Figura 4.15: Aproximación con una resolución $m = 1$.

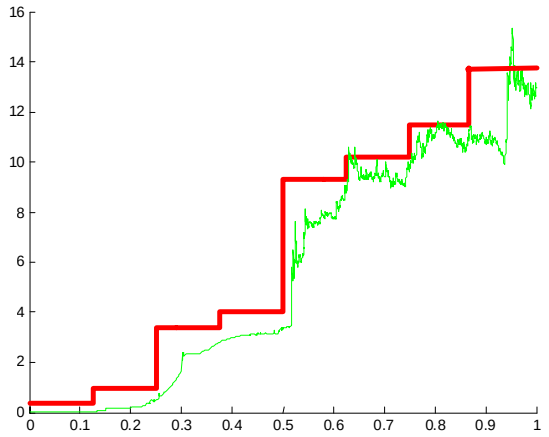


Figura 4.16: Aproximación con una resolución $m = 2$.

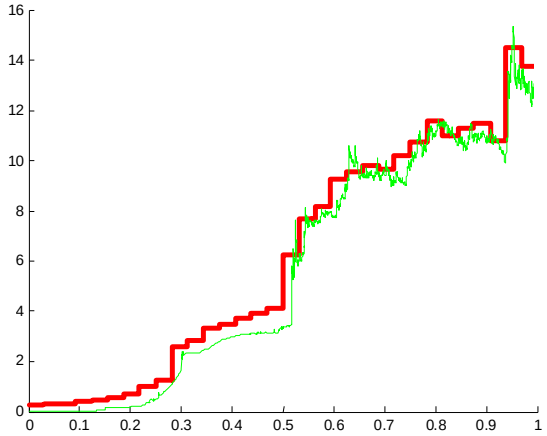


Figura 4.17: Approximation with a resolution $m = 5$.

Figura 4.18: Approximation with a resolution $m = 8$

Capítulo 5

Modelado de Series de Tiempo vía Redes Neuronales Wavelets Recurrentes

5.1. Introducción.

La identificación de sistemas se ha convertido en una herramienta fundamental en muchas ramas de la ingeniería y otras áreas tan diversas como bio-tecnología y economía, ciencias que requieren la existencia de modelos precisos de sistemas que posibiliten el análisis, la simulación y el diseño e implementación de estrategias de control..Existen numerosos procesos y métodos de identificación tanto paramétrica [41] como no paramétrica [44], que ofrecen información variada sobre el sistema en estudio,

En [40] se introducen métodos de identificación de parámetros utilizando redes neuronales, este tipo de técnica ha tenido mucho éxito, debido a la propiedad de las redes neuronales de adaptarse al medio ambiente.

Son bien conocidas las propiedades de las redes neuronales multi capa para funcionar como aproximador universal [46], esto es, sea $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ una función suave, entonces dado un conjunto compacto $S \in \mathbb{R}^n$ y ε_N un numero positivo, entonces existe una RNM de

dos capas, tal que

$$f(x) = W^T \sigma(V^T x) + \varepsilon \tag{5.1}$$

con $\|\varepsilon\| = \varepsilon_N$ para toda $x \in S$, el parámetro ε es la función de error de aproximación de la red neuronal. Nos interesa identificar sistemas del tipo discreto, que tienen la forma:

$$x(k+1) = f\left[x(k), u(k)\right] \tag{5.2}$$

$$y(k) = h\left[x(k)\right] \tag{5.3}$$

donde $u(k) \in \mathbb{R}^m$ es el vector de entrada, $x(k) \in \mathbb{R}^n$ es el vector de estados, y $y(k) \in \mathbb{R}^l$ es el vector de salida.f y h son en general funciones suaves no lineales, $f, h \in \mathbb{C}^\infty$, $k = 0, 1, ...N-1$.

La funciones wavelets han sido utilizadas para realizar identificación de sistemas de forma analítica, podemos ver [45], [12] y [55]. Sin embargo la implementación de esas técnicas resulta pesado computacionalmente, ya que se requiere la implementación de la transformada rápida wavelet en varias ocasiones, la alternativas es utilizar redes neuronales wavelet, ver [29] y [36]. Sin embargo en [29] se realiza la identificación del la planta utilizando una función wavelet Moorlet con una resolución fija la cual se elige mediante un criterio que compara el resultado de la identificación realizada con distintas resoluciones, en [36] se aprovecha la propiedad de multi resolución de las funciones wavelets, propone un estructura multi dimensional y se utiliza la función Sombrero Mexicano, sin embargo es necesario establecer la condición inicial para el ensanchamiento mas pequeño a utilizar.

En [16] se demostró que se puede tener un neuro identificador estable sin hacer modificación alguna en el algoritmo de aprendizaje ante la presencia de perturbaciones, el uso de la teoría de pasividad para sistemas continuos y sistemas discretos es una manera elegante de analizar la estabilidad en el sistema, usamos estabilidad de entrada a los estados para obtener las leyes de aprendizaje sin hacer modificaciones robustas. Con la red neuronal wavelet Haar que se propone en esta tesis es posible realizar identificación no paramétrica para sistemas discretos, sin modificar robustamente el algoritmo de aprendizaje back propagation.

5.2. Modelado de Sistemas No Lineales vía Redes Neuronales Wavelets Recurrentes

Almacenar información termporalmente es un problema en las Redes Neuronales Estáticas, y se realiza incluyendo retrasos en las entradas y las salidas, (4.16), (4.17), (4.18) y (4.19). Sin embargo, es una representación limitada, ya que solo puede almacenar un numero finito de entradas previas, una alternativa ha sido utilizar Redes Neuronales en las cuales en su estructura existe al menos una retroalimentación. Para este tipo de redes neuronales retroalimentada en tiempo continuo se usa usualmente la terminología como Redes Neuronales Diferenciales (DNN), y para tiempo discreto utilizamos el nombre de Redes Neuronales Recurrentes (RNN). Algunas de las tareas para este tipo de redes son: La predicción de series, la identificación y el control de sistemas dinámicos. La Figura 2.13 muestra las diferentes arquitecturas de las redes dinámicas. Las Figuras (a) y (b) muestran redes con recurrencia local, la primera una recurrencia en la misma capa y la segunda muestra una recurrencia entre las capas ocultas. Las Figuras (c) y (d) muestran una recurrencia global, donde la retroalimentación va de la salida de la red a la entrada y en el segundo caso la retroalimentación solo llega a la segunda capa de la re .

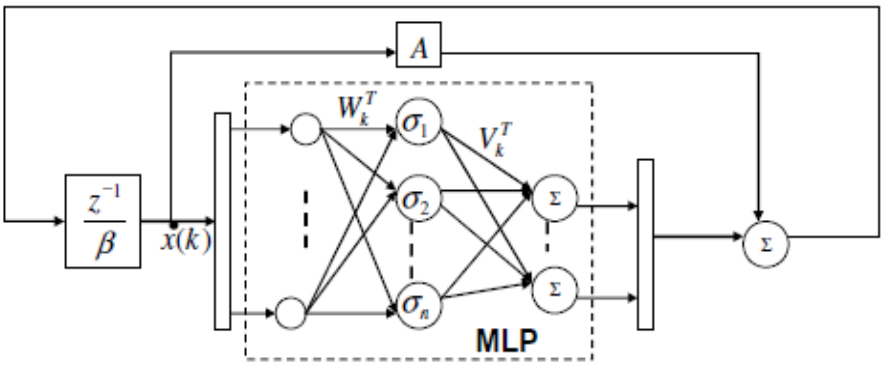


Figura 5.1: Estrucutura de la Red Neuronal Recurrente

Considerando la forma canónica de un sistema no lineal

$$x_{k+1} = f(x_k, u_k) \tag{5.4}$$

$$y_k = Cx_k \tag{5.5}$$

$$f(\cdot,\cdot) : \mathbb{R}^{n+p+1} \rightarrow \mathbb{R}^n \tag{5.6}$$

Donde $x_k \in \mathbb{R}^n$ es el vector de estado del sistema no lineal original a el tiempo de muestra k, $u_k \in \mathbb{R}^m$ es la acción de control, $C \in \mathbb{R}^{lxn}$ es la matriz de salida, $y_k \in \mathbb{R}^p$ es el vector de salida. Las siguientes condiciones se consideran satisfechas por el sistema:

1. La función $f(\cdot,\cdot)$ es lo suficientemente suave y satisface las condiciones de Lipschitz con respecto a sus argumentos, esto es:

$$\begin{aligned} \|f(x_k, u_k) - f(y_k, u_k)\| &\leq L_1 \|x_k - y_k\| + L_2 \|x_k - y_k\| \\ x, y &\in \mathbb{R}^n; \quad u_k, v_k \in \mathbb{R}^m \end{aligned} \tag{5.7}$$

1. Existe una función de Lyapunov V_k, tal que la diferencia entre las funciones al tiempo $k+1$ y k es acotado por:

$$V_{k+1} - V_k \leq -\lambda \|x_k\|^2 \tag{5.8}$$

2. El sistema original (5.4) puede representarse por:

$$x_{k+1} = Ax_k + W_1^* \Psi_1(x_k) + W_2^* \Psi_2(x_k) u(k) + \tilde{f}_k \tag{5.9}$$

donde $A \in \mathbb{R}^{nxn}$; $x_k \in \mathbb{R}^n$; $u_k \in \mathbb{R}^m, W_1^* \in \mathbb{R}^{nxk}$; $W_2^* \in \mathbb{R}^{nxr}, \Psi_1 \in \mathbb{R}^{kx1}, \Psi_2 \in \mathbb{R}^{rxm}, .\tilde{f}_k$ representa el error de modelado y es acotado como: $\left\|\tilde{f}_k\right\|^2_{\Lambda_{\tilde{f}_k}} \leq n_1 + n_2 \|x_k\|^2_{\Lambda_{\tilde{f}_k}}$, $n_1, n_2 = n_1, n_2 \in \mathbb{R}^+$, $\Lambda_{\tilde{f}_k} \in \mathbb{R}^{nxn}, 0 < \Lambda_{\tilde{f}_k}$.

1. El error de modelado esta dado por

$$\Delta f(x_k, u_k) := f(x_k) + \xi_{1,k} - [Ax_k + W_1^* \Psi_1 x_k + W_2^* \Psi_2 x_k u_k]$$

Donde $\left\|\xi_{1,k}\right\|^2_{\Lambda_{\xi_j}} \leq \gamma_j$, la siguiente Y considerando la aproximación al sistema discreto mediante una serie Wavelet Haar se tiene:

$$f^*(.) = \sum_{m,n=0}^{N} w1^*_{mn} H_{mn}(k) + \mu_1(k) \tag{5.10}$$

$$g^*(.) = \sum_{m,n=0}^{N} w2^*_{mn} H_{mn}(k) + \mu_2(k) \tag{5.11}$$

Utilizando una Red Neuronal Wavelet Recurrente (RNWR), del tipo Haar se tiene

$$f(.) = \sum_{m,n=0}^{N} w1_{mn} H_{mn}(k) + \varepsilon_1(k) \tag{5.12}$$

$$g(.) = \sum_{m,n=0}^{N} w2_{mn} H_{mn}(k) + \varepsilon_2(k) \tag{5.13}$$

se escribe el error de neuro identificación como:

$$e(k) = x^*(.) - x(.) + \zeta(k) \tag{5.14}$$

$$e(k) = f^*(.) + g^*(.) - f(.) - g(.) + \zeta(k) \tag{5.15}$$

donde $\zeta(k) = \varepsilon(k) + \mu(k)$, $\varepsilon(k)$ es el error de aproximación de la RNWR, $\varepsilon(k)$ es acotada $\varepsilon^2(k) \leq \overline{\varepsilon}$, $\overline{\varepsilon}$ es una constante positiva desconocida. Se tiene que $\widetilde{W}(k) = W(k) - W^*(k)$, Escribimos el error de neuro identificación como:

$$e(k) = \widetilde{f}(.) + \widetilde{g}(.) + \zeta(k) \tag{5.16}$$

Escribiendo nuevamente el error en términos de series Wavelets Haar se tiene

$$e(k) = \widetilde{W_1}(k)\,\psi_1(.) + \widetilde{W_2}(k)\,\psi_2(.) + \zeta(k) \tag{5.17}$$

La identificación de sistemas es la determinación del modelo dinámico de un sistema desconocido, que puede ser usado sub secuentemente en una retroalimentación con propósito de controlar el sistema.

Por otro lado la estimación de estados involucra la determinación de los estados internos desconocidos de un sistema dinámico. Existen diversas técnicas para la identificación de los estados, y ha sido un campo muy activo con muchos trabajos publicados, y gran parte de estas técnicas utilizan redes neuronales las cuales han demostrado su eficiencia al aproximar funciones no lineales.

5.3. Algoritmo Estable de Entrenamiento para Redes Neuronales Wavelets Recurrentes.

En este trabajo consideramos la clase de sistemas no lineales discretos SISO gobernados por n ecuaciones de diferencias no lineales, la salida del sistema y los estados algebraicos están dados como:

$$x(k+1) = f(x(k), u(k)) + \zeta_{1,k};\ x_o = x(0) \tag{5.18}$$

$$y(k) = Cx(k) + \zeta_{2,k} \tag{5.19}$$

donde $x(k) \in \mathbb{R}^n$ es el vector de estados del sistema no lineal original a el tiempo de muestreo k, $u(k) \in \mathbb{R}^m$ es la acción de control, $C \in \mathbb{R}^{1xn}$ es el conocimiento a priori de la matriz de salida, $y(k) \in \mathbb{R}^p$ es el vector de salida. De acuerdo al teorema de Stone-Weierstrass [18], el sistema no lineal puede escribirse como

$$\beta x(k+1) = Ax(k) + \sigma(M_1^* x(k)) + \phi(M_2^* x(k))U(k) + \xi(k) \tag{5.20}$$

donde M_1^* y M_2^* son pesos constantes que pueden minimizar el error de modelado $\xi(k)$. Con σ y ϕ son funciones acotadas

Usando la teoría Wavelet como aproximador universal, podemos re escribir la ecuación (5.20) como:

$$\beta x(k+1) = Ax(k) + W_1^* \Psi_1(x(k)) + W_2^* \Psi_2(x(k))U(k) + \mu(k) \tag{5.21}$$

donde $A \in \mathbb{R}^{nxn}$; $x(k) \in \mathbb{R}^n$; $U(k) \in \mathbb{R}^m, W_1^* \in \mathbb{R}^{nxk}$; $W_2^* \in \mathbb{R}^{nxr}, \Psi_1 \in \mathbb{R}^{kx1}, \Psi_2 \in \mathbb{R}^{rxm}$.

Un sistema no lineal (5.18), el cual tienen entradas estables acotadas y salida acotadas (BIBO), *i.e.*, x_k y U_k son acotados. $U_k = [u_1, u_2 \cdots u_m, 0, \cdots 0]^T \in \Re^n$. $|U(k)|^2 \leq \overline{u}$. $\mu(k)$ es acotado con $\mu^2(k) \leq \overline{\mu}$, $\overline{\mu}$ es una constante positiva no conocida. $\mu^2(k)$ es la norma del vector la cual esta definida como $\mu^2(k) = \mu_1^2(k) + \cdots + \mu_n^2(k)$, $\mu(k) = [\mu_1(k), \cdots, \mu_n(k)]^T$. En el horizonte finito, $k < K$, cuando la identificación del sistema se ha realizado, $\mu^2(k)$ es acotado. Este sistema no lineal puede escribirse en forma de una serie de tiempo discreta utilizando redes neuronales recurrentes wavelet Haar como:

$$\beta\widehat{x}(k+1) = A\widehat{x}(k) + \hat{W}_1\Psi_1(x_k) + \hat{W}_2\Psi_2(x_k)U_k + \varepsilon_1(k) \tag{5.22}$$

donde $\widehat{x}(k) \in \Re^n$ representa el estado interno de la red neuronal. La matiz $A \in \Re^{n\times n}$ es una matriz estable que sera después especificada. Las matrices $W_1(k), W_2(k) \in \Re^{n\times n}$ son los pesos de la Red Neuronal Wavelet Haar. $\varepsilon_1(k)$ es acotado de la forma $\|\varepsilon_1(k)\|^2 \leq \overline{\varepsilon}_1$, $\overline{\varepsilon}_1$ es una constante positiva no conocida. El error de neuro identificación esta definido como:

$$e(k) = \widehat{x}(k) - x(k)$$

De (5.21) y (5.22)

$$\begin{aligned} \beta e(k+1) = Ae(k) + (W_1^* - \hat{W}_1)\Psi_1(x_k) + \\ + (W_2^* - \hat{W}_2)\Psi_2(x_k)U_k - \mu(k) \end{aligned} \tag{5.23}$$

escribiendo $\widetilde{W_1}(k) = W_1^* - \hat{W}_1(k)$, entonces

$$\begin{aligned} \beta e(k+1) = Ae(k) + \widetilde{W_1}(k)\Psi_1(x_k) + \\ + \widetilde{W_1}(k)\Psi_2(x_k)u(k) + \zeta(k) \end{aligned} \tag{5.24}$$

donde $\zeta(k) = \varepsilon_1(k) - \mu(k)$. El siguiente teorema ilustra un algoritmo de aprendizaje estable para una red neuronal wavelet Haar de una sola capa.

Teorema 5.1 *Si la Red Neuronal Wavelet Haar (5.22) es usada para identificar una planta no lineal (5.18) y los eigenvalores de A son seleccionados como* $-1 < \lambda(A) < 0$, *la siguiente*

ley de actualización gradiente sin modificación robusta puede hacer el error de identificación $e(k)$ *acotado (estable en el sentido* L_∞*)*

$$W_1(k+1) = W_1(k) - \eta(k)\,\Psi_1(x_k)e^T(k) \tag{5.25}$$

$$W_2(k+1) = W_2(k) - \eta(k)\,u(k)\Psi_2(x_k)e^T(k) \tag{5.26}$$

donde $\eta(k)$ *satisface*

$$\eta(k) = \begin{cases} \dfrac{\eta}{1 + \|\Psi_1(x_k)\|^2 + \|u(k)\Psi_2(x_k)\|^2} & \text{if } \beta\|e(k+1)\| \geq \|e(k)\| \\ 0 & \text{if } \beta\|e(k+1)\| < \|e(k)\| \end{cases} \tag{5.27}$$

con $0 < \eta \leq 1$.

Demostración. Se selecciona la función candidata de Lyapunov de la forma

$$V(k) = \left\|\widetilde{W}_1(k)\right\|^2 + \left\|\widetilde{W}_2(k)\right\|^2 \tag{5.28}$$

donde $\left\|\widetilde{W}_1(k)\right\|^2 = \sum_{i=1}^{n} \widetilde{w}_1(k)^2 = tr\left\{\widetilde{W}_1^T(k)\,\widetilde{W}_1(k)\right\}$. De la ley de actualización (5.25)

$$\widetilde{W}_1(k+1) = \widetilde{W}_1(k) - \eta(k)\,\Psi_1(x_k)e^T(k) \tag{5.29}$$

entonces

$$\begin{aligned}
\Delta V(k) &= V(k+1) - V(k) \\
&= \left\|\widetilde{W}_1(k) - \eta(k)\,\Psi_1(x_k)e^T(k)\right\|^2 - \left\|\widetilde{W}_1(k)\right\|^2 \\
&+ \left\|\widetilde{W}_2(k) - \eta(k)\,u(k)\Psi_2(x_k)e^T(k)\right\|^2 \\
&- \left\|\widetilde{W}_2(k)\right\|^2 \\
&= \eta^2(k)\,\|e(k)\|^2\,\|\Psi_1(x_k)\|^2 \\
&- 2\eta(k)\left\|\Psi_1(x_k)e^T(k)\right\| \\
&+ \eta^2(k)\,\|e(k)\|^2\,\|u(k)\Psi_2(x_k)\|^2 \\
&- 2\eta(k)\left\|u(k)\Psi_2(x_k)e^T(k)\right\|
\end{aligned}$$

Existe una constante $\beta > 0$, tal que si $\|\beta e(k+1)\| \geq \|e(k)\|$, usando (5.24) y $\eta(k) \geq 0$,

$$\begin{aligned}
& -2\eta(k)\left\|\Psi_1(x_k)e^T(k)\right\| - 2\eta(k)\left\|u(k)\Psi_2(x_k)e^T(k)\right\| \\
\leq\ & -2\eta(k)\left\|e^T(k)\right\|\|\beta e(k+1) - Ae(k) - \zeta(k)\| \\
=\ & -2\eta(k)\left\|e^T(k)\beta e(k+1) - e^T(k)Ae(k)\right. \\
& \left. -e^T(k)\zeta(k)\right\| \\
\leq\ & -2\eta(k)\left\|e^T(k)\beta e(k+1)\right\| + 2\eta(k)e^T(k)Ae(k) \\
& +2\eta(k)\left\|e^T(k)\zeta(k)\right\| \\
\leq\ & -2\eta(k)\|e(k)\|^2 + 2\eta(k)\lambda_{\text{máx}}(A)\|e(k)\|^2 \\
& +\eta(k)\|e(k)\|^2 + \eta(k)\|\zeta(k)\|^2
\end{aligned}$$

de $0 < \eta \leq 1$, ,

$$\begin{aligned}
\Delta V(k) \leq\ & \eta^2(k)\|e(k)\|^2\|\Psi_1(x_k)\|^2 \\
& + \eta^2(k)\|e(k)\|^2\|u(k)\Psi_2(x_k)\|^2 \\
& - \eta(k)\|e(k)\|^2 + 2\eta(k)\lambda_{\text{máx}}(A)\|e(k)\|^2 \\
& + \eta(k)\|\zeta(k)\|^2 \qquad (5.30) \\
& = -\eta(k)\left[(1 - 2\lambda_{\text{máx}}(A)) - \right. \\
& \left. -\eta\frac{\|\Psi_1(x_k)\|^2 + \|u(k)\Psi_2(x_k)\|^2}{1 + \|\Psi_1(x_k)\|^2 + \|u(k)\Psi_2(x_k)\|^2}\right]e^2(k) \\
& + \eta_k\zeta^2(k) \qquad (5.31) \\
& \leq -\pi e^2(k) + \eta\zeta^2(k) \qquad (5.32)
\end{aligned}$$

donde

$$\begin{aligned}
\pi &= \frac{\eta}{1+\kappa}\left[1 - 2\lambda_{\text{máx}}(A) - \frac{\kappa}{1+\kappa}\right] \\
\kappa &= \max_k\left(\|\Psi_1(x_k)\|^2 + \|u(k)\Psi_2(x_k)\|^2\right)
\end{aligned}$$

de $-1 < \lambda(A) < 0$, $\pi > 0$

$$n \min\left(\widetilde{w}_i^2\right) \leq V_k \leq n \max\left(\widetilde{w}_i^2\right)$$

donde $n \times \min(\widetilde{w}_i^2)$ y $n \times \max(\widetilde{w}_i^2)$ son $\mathcal{K}_\infty$-funciones, y $\pi e^2(k)$ es un $\mathcal{K}_\infty$-función, $\eta\zeta^2(k)$ es $\mathcal{K}$-función, entonces V_k admite la función suave ISS-Lyapunov, la dinámica de el error de identificación es estable de entrada a la salida, (input-to-state estable). La "ENTRADA" corresponde a el segundo término de la última línea en (5.32), *i.e.*, el error de modelado $\zeta(k) = \varepsilon_1(k) + \varepsilon_2(k) - \mu(k)$, el "ESTADO" corresponde al el primer termino de la última línea en (5.32), *i.e.*, el error de identificación $e(k)$. Debido a que la "ENTRADA" $\zeta(k)$ es acotada y la dinámica es ISS, el .ESTADO" $e(k)$ es acotado. Si $\beta \|e(k+1)\| < \|e(k)\|$, $\Delta V(k) = 0$. $V(k)$ es constante, $W_1(k)$ es constante. Entonces $\|e(k+1)\| < \frac{1}{\beta}\|e(k)\|$, $\frac{1}{\beta} < 1$, $e(k)$ es acotado. ■

Comentario 5.1 *La condición $\beta \|e(k+1)\| \geq \|e(k)\|$ es dead-zone. Si β es seleccionada lo suficientemente grande, la dead-zone se vuelve pequeña.*

5.4. Filtro Extendido de Kalman para el Entrenamiento de Redes Neuronales Wavelets Recurrentes.

Considerando la planta no lineal siguiente:

$$x(k+1) = f(x(k), u(k)) \tag{5.33}$$

$$y(k) = Cx(k) \tag{5.34}$$

Se propone la siguiente Red Neuronal Wavelet Haar para identificar (5.33).

$$\widehat{x}(k+1) = A\widehat{x}(k) + V_{1,k}\sigma\left[W_{1,k}x(k)\right] + V_{2,k}\phi\left[W_{2,k}x(k)\right]u(k) \tag{5.35}$$

donde $\widehat{x}(k) \in \Re^n$ represente el estado interno de la red neuronal. La matriz $A \in \Re^{n\times n}$ es una matriz estable. Los pesos en la capa de salida son $V_{1,k}$, $V_{2,k} \in R^{n\times m}$, los pesos $W_{1,k}$,

$W_{2,k} \in R^{m\times n}$, σ es una función wavelet Haar $m-$dimensional, $\sigma = [\sigma_1 \cdots \sigma_m]^T$, $\phi(\cdot)$ es $\Re^{m\times m}$ una matriz diagonal wavelet Haar. De acuerdo con el teorema de Stone-Weierstrass y las propiedades de densidad de las redes neuronales recurrentes [70], el sistema no lineal (5.33) puede escribirse de la siguiente forma

$$\begin{aligned} x(k+1) = Ax(k) + V_{1,k}\sigma\left[W_{1,k}x(k)\right] \\ +V_{2,k}\phi\left[W_{2,k}x(k)\right]u(k) - \eta(k) \end{aligned} \tag{5.36}$$

donde $\eta(k) = f\left[x(k), u(k)\right] - Ax(k) - V_{1,k}\sigma\left[W_{1,k}x(k)\right] - V_{2,k}\phi\left[W_{2,k}x(k)\right]u(k)$ es el error de modelado con respecto a $V_{1,k}$, $V_{2,k}$, $W_{2,k}$ y $W_{2,k}$, y son pesos variantes en el tiempo los cuales serán actualizado por un gradiente que minimice el error de identificación. Por [70] nosotros sabemos que para el termino $\eta(k)$ puede ser arbitrariamente pequeño para simplificar la selección del número adecuado de neuronas en las capas ocultas (en este caso es m). En el caso de dos variables independientes, una función suave f tiene la siguiente expansión en series de Taylor

$$f = \sum_{k=0}^{l-1}\frac{1}{k!}\left[\left(x_1 - x_1^0\right)\frac{\partial}{\partial x_1} + \left(x_2 - x_2^0\right)\frac{\partial}{\partial x_2}\right]_0^k f + \varepsilon \tag{5.37}$$

el coeficiente ε es el residuo de la formula de Taylor. Si tomamos x_1 y $x_2$correspondientes a $W_{1,k}x\left(k\right)$ y $V_{1,k}$, x_1^0, x_2^0 corresponde $W_1^0x\left(k\right)$ y V_1^0, y define $\widetilde{W}_{1,k} = W_{1,k} - W_1^0$, $\widetilde{V}_{1,k} = V_{1,k} - V_1^0$, entonces nosotros tenemos

$$V_{1,k}\sigma\left[W_{1,k}x(k)\right] = V_1^0\sigma\left[W_1^0x\left(k\right)\right] + \Theta_{1,k}B_{1,k} + \varepsilon_1$$

donde V_1^0, V_2^0, W_1^0 y W_2^0 son un conjunto de condiciones iniciales conocidas para los pesos, $B_{1,k} = \left[\sigma, \sigma' V_{1,k}^T x\right]^T \in R^{2m\times 1}$, $\Theta_{1,k} = \left[V_{1,k}, W_{1,k}^T\right]^T \in R^{n\times 2m}$, σ' es la derivada de la función no lineal de activación $\sigma\left(\cdot\right)$ con respecto a $W_{1,k}x(k)$. De forma similar

$$V_{2,k}\phi\left[W_{2,k}x(k)\right]u(k) = V_2^0\phi\left[W_2^0x\left(k\right)\right]u(k) + \Theta_{2,k}B_{2,k} + \varepsilon_2$$

en donde $B_{2,k} = \left[\phi u, \phi'\mathrm{diag}(u)V_{2,k}^T x\left(k\right)\right]^T$, $\Theta_{2,k} = \left[V_{2,k}, W_{2,k}^T\right]^T$. Nosotros definimos el error de modelado $\zeta(k) = \varepsilon_1 + \varepsilon_2 - \eta(k)$, nosotros tenemos

$$y(k) = B_k^T\Theta_k + \zeta(k) \tag{5.38}$$

con

$$\begin{aligned} \Theta_k &= \begin{bmatrix} \Theta_{1,k} \\ \Theta_{2,k} \end{bmatrix} = \left[V_{1,k}, W_{1,k}^T, V_{2,k}, W_{2,k}^T\right]^T \\ B_k &= \begin{bmatrix} B_{1,k} \\ B_{2,k} \end{bmatrix} = \left[\sigma, \sigma' V_{1,k}^T x, \phi u, \phi' \mathrm{diag}(u) V_{2,k}^T x\left(k\right)\right]^T \end{aligned} \tag{5.39}$$

la salida $y(k)$ es

$$y(k) = x(k+1) - Ax(k) - V_1^0 \sigma\left[W_1^0 x\left(k\right)\right] - V_2^0 \phi\left[W_2^0 x\left(k\right)\right] u(k)$$

Ahora nosotros usamos la técnica del filtro de Kalman para entrenar la red neuronal wavelet recurrente (5.35) tal que el error de identificación $\epsilon_i\left(k\right)$ entre la planta (5.33) y la red neuronal (5.35), *i.e.*,

$$\epsilon_i\left(k\right) = \widehat{x}_i(k) - x_i(k)$$

es acotada. El parámetro de la matriz Θ_k se supone una constante desconocida mas una pequeño componente independiente aleatorio, en el proceso artificial el ruido Ω_k es definido para servir el cambio, *i.e.*, $\Theta_{k+1} = \Theta_k + \Omega_k$, $\Omega_k = \left[\omega_1(k) \cdots \omega_n(k)\right]^T$. Re escribimos (5.38) en espacio de estados con una sola salida.

$$\begin{cases} \theta_i\left(k+1\right) = \theta_i\left(k\right) + \omega_i(k) \\ y_i\left(k\right) = B_k^T \theta_i\left(k\right) + \zeta_{i,k} \end{cases} \tag{5.40}$$

donde $i = 1 \cdots n$, $\theta_i\left(k\right) \in R^{4m \times 1}$, $\Theta_k = \left[\theta_1\left(k\right), \cdots \theta_n\left(k\right)\right]$, $y\left(k\right) = \left[y_1\left(k\right), \cdots y_n\left(k\right)\right]$, $\zeta = \left[\zeta_1\left(k\right) \cdots \zeta_n\left(k\right)\right]^T$, $\omega_i(k)$ es independiente de $\theta_i\left(k\right)$.

$$\begin{aligned} y_i(k) = x_i(k+1) - a_i x_i(k) \\ + V_1^0 \sigma\left[W_1^0 x_i\left(k\right)\right] + V_2^0 \phi\left[W_2^0 x_i\left(k\right)\right] u(k) \end{aligned}$$

Porque $\zeta_{i,k}$ y $\omega_i(k)$ son no correlacionados ($\omega_i(k)$ es independiente de $\theta_i\left(k\right)$), $E\left\{\omega_i(k)\zeta_{i,k}^T\right\} = 0$.

Comentario 5.2 *El vector de estados $\theta_i\left(k\right)$ se asume como una constante desconocida con un pequeña componente aleatorio para permitir una estimación adaptativa. Este paso aleatorio sirve como un ruido artificial en el proceso $\omega_i(k)$. Asumimos que se satisface*

$$E\left\{\omega_i(k)\right\} = 0, \quad E\left\{\omega_i(k)\omega_i(k)^T\right\} = R_1 \tag{5.41}$$

R_1 puede elegirse como αI, donde α es pequeña y positiva. De hecho, si no tenemos un cambio durante el intervalo de tiempo, R_1 tiende a cero, y llega a el algoritmo de mínimos cuadrados. No se esperan cambios en la planta durante todo el tiempo de interés, la covarianza de el proceso de ruidos $\zeta_{i,k}$ puede asumirse como

$$E\left\{\zeta_{i,k}\zeta_{i,k}^{T}\right\} = R_2$$

Esas ideas pueden encontrarse en [71] y [72], pero [72] aplica el filtro de Kalman para una red del tipo feedforward. Nosotros usamos el filtro de Kalman para redes wavelet Haar recurrentes.

Usamos n filtros de Kalman para estimar los pesos. Para el $i-esimo$ sub sistema el observador es

$$\begin{cases} \widehat{\theta}_i\left(k+1\right) = \widehat{\theta}_i\left(k\right) + K\left[y_i\left(k\right) - \widehat{y}_i\left(k\right)\right] \\ \widehat{y}_i\left(k\right) = B_k^T\widehat{\theta}_i\left(k\right) \end{cases} \tag{5.42}$$

donde $\widehat{y}_i\left(k\right)$ es el estado estimado de $y_i\left(k\right)$, K es la ganancia del observador. El error de estimación del estado está definido como

$$\widetilde{\theta}_i\left(k\right) = \theta_i\left(k\right) - \widehat{\theta}_i\left(k\right) \tag{5.43}$$

Sustituyendo (5.40) y (5.42) en (5.43)

$$\widetilde{\theta}_i\left(k+1\right) = \left[I - KB_k^T\right]\widetilde{\theta}_i\left(k\right) + \omega_i(k) - K\zeta_{i,k} \tag{5.44}$$

Definimos el índice de referencias $P_k \in R^{4m\times 4m}$ el cual usa la covarianza

$$P_k = cov(\widetilde{\theta}_i) = E\left\{\left[\widetilde{\theta}_i - E\left(\widetilde{\theta}_i\right)\right]\left[\widetilde{\theta}_i - E\left(\widetilde{\theta}_i\right)\right]^T\right\}$$

Por (5.40) y $E\left[\omega_i(k)\right] = 0$, nosotros tenemos $E\left[\theta_i\left(k+1\right)\right] = E\left[\theta_i\left(k\right)\right] + E\left[\omega_i(k)\right] = 0$, $E\left[\theta_i\left(k+1\right)\right] = E\left[\theta_i\left(k\right)\right] = E\left[\theta_i\left(1\right)\right] = 0$. Porque $E(\widehat{\theta}_i\left(k\right)) = 0$, $E(\widetilde{\theta}_i\left(k\right)) = E(\theta_i\left(k\right)) - E(\widehat{\theta}_i\left(k\right)) = 0$. So

$$P_k = E\left\{\widetilde{\theta}_i\left(k\right)\widetilde{\theta}_i^T\left(k\right)\right\} \tag{5.45}$$

Sustituimos (5.44) en (5.45)

$$
\begin{aligned}
P_{k+1} &= E\left\{\widetilde{\theta}_i\left(k+1\right)\widetilde{\theta}_i^T\left(k+1\right)\right\} \\
&= E\left\{\begin{array}{l}\left[\left(I-KB_k^T\right)\widetilde{\theta}_i\left(k\right)+\omega_i(k)-K\zeta_{i,k}\right] \\ \times\left[\left(I-KB_k^T\right)\widetilde{\theta}_i\left(k\right)+\omega_i(k)-K\zeta_{i,k}\right]^T\end{array}\right\} \\
&= \left(I-KB_k^T\right)E\left\{\widetilde{\theta}_i\left(k\right)\widetilde{\theta}_i^T\left(k\right)\right\}\left(I-KB_k^T\right)^T + E\left\{\omega_i(k)\omega_i(k)^T\right\} + KE\left\{\zeta_{i,k}\zeta_{i,k}^T\right\}K^T \\
&+\text{cross-terms}(\widetilde{\theta}_i, \omega_i(k), \zeta_{i,k})
\end{aligned}
$$

Debido a que $\widetilde{\theta}_i\left(k\right)$, $\omega_i(k)$ y $\zeta_{i,k}$ son independientes, los términos cruzados de $(\widetilde{\theta}_i, \omega_i(k), \zeta_{i,k})$ son cero. Entonces P_{k+1} es

$$
\begin{aligned}
P_{k+1} &= P_k + R_1 - P_kB_k\left(R_2+B_k^TP_kB_k\right)^{-1}B_k^TP_k \\
&+\left[K-P_kB_k\left(R_2+B_k^TP_kB_k\right)^{-1}\right]\left(R_2+B_k^TP_kB_k\right)\times\left[K-P_kB_k\left(R_2+B_k^TP_kB_k\right)^{-1}\right]^T
\end{aligned}
$$

El último término $[\cdot]\left(R_2+B_k^TP_kB_k\right)[\cdot]^T \geq 0$, y $P_k \geq 0$, $R_1 > 0$. Si queremos minimizar P_{k+1}, tenemos que hacer el último término cero, *i.e.*, $K - P_kB_k\left(R_2+B_k^TP_kB_k\right)^{-1} = 0$. Entonces la ganancia del observador es seleccionada como:

$$
K = P_kB_k\left(R_2+B_k^TP_kB_k\right)^{-1} \tag{5.46}
$$

Por (5.46), P_{k+1} se escribe como:

$$
\begin{aligned}
P_{k+1} &= P_k + R_1 - P_kB_k\left(R_2+B_k^TP_kB_k\right)^{-1}B_k^TP_k \\
&= R_1 + \left[I-KB_k^T\right]P_k
\end{aligned} \tag{5.47}
$$

(5.42), (5.46) y (5.47) son la extensión del filtro de Kalman para el aprendizaje de los pesos $\widehat{\theta}_i\left(k\right)$ de las redes neuronales wavelet Haar recurrentes

$$
\begin{aligned}
&\widehat{\theta}_i\left(k+1\right) = \widehat{\theta}_i\left(k\right) - K_ke_i\left(k\right) \\
&\widehat{y}_i\left(k\right) = B_k^T\widehat{\theta}_i\left(k\right) \\
&P_{k+1} = R_1 + \left[I-K_kB_k^T\right]P_k \\
&K_k = P_kB_k\left(R_2+B_k^TP_kB_k\right)^{-1}
\end{aligned} \tag{5.48}
$$

donde

$$
e_i\left(k\right) = \widehat{y}_i\left(k\right) - y_i\left(k\right)
$$

Comentario 5.3 *El error de Kalman $e_i(k)$ no es igual al error de identificación $\epsilon_i(k) = \widehat{x}_i(k) - x_i(k)$, sin embargo ambos son minimizados al mismo tiempo. De (5.35) y (5.36), tenemos*

$$\epsilon_i(k+1) = a_i\epsilon_i(k) + e_i(k) \tag{5.49}$$

Teniendo la relación

$$\begin{aligned}
&\epsilon_i(1) = a_i\epsilon_i(0) + e_i(0)\\
&\epsilon_i(2) = a_i\epsilon_i(1) + e_i(1) = a_i^2\epsilon_i(0) + a_ie_i(0) + e_i(1)\\
&\vdots\\
&\epsilon_i(k) = a_i^k\epsilon_i(0) + \sum_{j=0}^{k-1} a_i^{k-j-1}e_i(j)
\end{aligned}$$

Debido a que $|a_i| < 1$

$$|\epsilon_i(k)| \leq |\epsilon_i(0)| + \sum_{j=0}^{k-1} |e_i(j)|$$

Por lo que $\epsilon_i(0)$ es una constante, la minimización del error de Kalman $e_i(j)$ implica que el límite superior del error de identificación $\epsilon_i(k)$ es minimizado.

Comentario 5.4 *El observador (5.42) es para cada subsistema. Este método puede reducir la carga computacional cuando se estima el peso de la red neuronal recurrente, ver [20] y [73]. Sabemos que la matriz B_k depende de los parámetros $V_{1,k}^T$ y $V_{2,k}^T$, y estos no afectaran el algoritmo de actualización de los parámetros (5.48), porque el parámetro desconocido $\widehat{\theta}_i(k+1)$ es calculado por el parámetro conocido $\widehat{\theta}_i(k)$ y B_k . Para cada elemento Θ_k^T y B_k en (5.48), tenemos*

$$\begin{aligned}
V_{1,k+1} &= V_{1,k} - \tfrac{P_k}{R_2+B_k^TP_kB_k}\sigma\left[W_{1,k}x(k)\right]e^T(k)\\
W_{1,k+1} &= W_{1,k} - \tfrac{P_k}{R_2+B_k^TP_kB_k}\sigma'\left[W_{1,k}x(k)\right]V_{1,k}^Tx(k)e^T(k)
\end{aligned} \tag{5.50}$$

que tiene la misma forma que el algoritmo BackPropagation [20], pero el índice de aprendizaje no es una constante positiva, es la matriz $\frac{P_k}{R_2+B_k^TP_kB_k}$ que cambia a través del tiempo. Lo cual es la principal razón de porque el aprendizaje con el filtro de Kalman tiene una mayor velocidad de convergencia. Como [73] como se señala, $R_2 > 0$ puede incrementar la velocidad de convergencia.

A continuación usaremos la teoría de estabilidad de Lyapunov para probar el algoritmo de aprendizaje usando el filtro de Kalman (5.48) con zona muerta es estable para la identificación de sistemas. Por (5.40), (5.42) se convierte en

$$e_i(k) = B_k^T \widetilde{\theta}_i(k) + \zeta_{i,k} \tag{5.51}$$

donde $\widetilde{\theta}_i(k) = \theta_i(k) - \widehat{\theta}_i(k)$, $i = 1 \cdots n$. Nosotros modificamos el filtro extendido de Kalman (5.48) como filtro de zona muerta de Kalman

$$\begin{aligned} \widehat{\theta}_i(k+1) &= \widehat{\theta}_i(k) - R_k^{-1} P_k B_k s_k \\ \widehat{y}_i(k) &= B_k^T \widehat{\theta}_i(k) \end{aligned} \tag{5.52}$$

donde $R_k = R_2 + B_k^T P_k B_k$ es escalar,

$$s_k = \begin{cases} e_i(k) & \frac{1}{R_k} e_i^2(k) \geq \frac{3}{R_2} \overline{\zeta}_i^2 \\ 0 & \frac{1}{R_k} e_i^2(k) < \frac{3}{R_2} \overline{\zeta}_i^2 \end{cases} \tag{5.53}$$

$\overline{\zeta}_i$ es el límite superior de la incertidumbre $\zeta_{i,k}$, $\left|\zeta_{i,k}\right| < \overline{\zeta}_i$.

Teorema 5.2 *El filtro de Kalman de zona muerta (5.52) puede asegurar el error de identificación $\epsilon_i(k)$ y acotar los pesos de la red. Existe $T \in (0, \infty)$ el error a la salida $e_i(k)$ converge a el conjunto residuo*

$$D_{e_i} = \left\{ e_i(k) \mid \frac{1}{R_k} e_i^2(k) \leq \frac{3}{R_2} \overline{\zeta}_i^2 \right\} \tag{5.54}$$

Demostración. Definimos la siguiente función de Lyapunov

$$V(k) = \widetilde{\theta}_i^T(k) P_k^{-1} \widetilde{\theta}_i(k) \tag{5.55}$$

definimos $\overline{P}_{k+1} = \left[I - K_k B_k^T\right] P_k$, dado que $P_{k+1} = \overline{P}_{k+1} + R_1$, $R_1 > 0$, entonces $x^T P_{k+1} x > x^T \overline{P}_{k+1} x > 0$ y $x^T \overline{P}_{k+1}^{-1} x \geq x^T P_{k+1}^{-1} x > 0$. Aplicado a (5.55)

$$\begin{aligned} \Delta V(k) &= \widetilde{\theta}_i^T(k+1) P_{k+1}^{-1} \widetilde{\theta}_i(k+1) - \widetilde{\theta}_i^T(k) P_k^{-1} \widetilde{\theta}_i(k) \\ &\leq \widetilde{\theta}_i^T(k+1) \overline{P}_{k+1}^{-1} \widetilde{\theta}_i(k+1) - \widetilde{\theta}_i^T(k) P_k^{-1} \widetilde{\theta}_i(k) \end{aligned} \tag{5.56}$$

Primero, consideramos el caso $e_i^2(k) \geq 3\overline{\zeta}_i^2$ y $\frac{1}{R_k}P_kB_kB_k^TP_k \geq R_1$. Entonces $s_k = e_i(k)$. Substituyendo (5.51) en (5.52) tenemos

$$\widetilde{\theta}_i(k+1) = \widetilde{\theta}_i(k) - \frac{1}{R_k}P_kB_kB_k^T\widetilde{\theta}_i(k) - \frac{1}{R_k}P_kB_k\zeta_i(k) \tag{5.57}$$

De (5.52), (5.51) y (5.57), tenemos

$$\begin{array}{l} \widetilde{\theta}_i(k+1) = \widetilde{\theta}_i(k) - \frac{1}{R_k}P_kB_ke_i(k) \\ \widetilde{\theta}_i(k+1) = \left[I - \frac{1}{R_k}P_kB_kB_k^T\right]\widetilde{\theta}_i(k) - \frac{1}{R_k}P_kB_k\zeta_{i,k} \end{array} \tag{5.58}$$

De (5.47) sabemos que $P_{k+1} = \overline{P}_{k+1} + R_1$. Entonces $\overline{P}_{k+1} = P_k - \frac{1}{R_k}P_kB_kB_k^TP_k$, $\overline{P}_{k+1}P_k^{-1} = I - \frac{1}{R_k}P_kB_kB_k^T$, (5.58) tenemos que

$$\overline{P}_{k+1}^{-1}\widetilde{\theta}_i(k+1) = P_k^{-1}\widetilde{\theta}_i(k) - \frac{1}{R_k}\overline{P}_{k+1}^{-1}P_kB_k\zeta_{i,k} \tag{5.59}$$

substituyendo (5.59) en (5.56) obtenemos

$$\begin{array}{c} \Delta V(k) \leq \left[\widetilde{\theta}_i(k+1) - \widetilde{\theta}_i(k)\right]^T P_k^{-1}\widetilde{\theta}_i(k) \\ -\frac{1}{R_k}\widetilde{\theta}_i^T(k+1)\overline{P}_{k+1}^{-1}P_kB_k\zeta_{i,k} \end{array} \tag{5.60}$$

ahora $\left[\widetilde{\theta}_i(k+1) - \widetilde{\theta}_i(k)\right]^T$ es substituido por (5.57), (5.60) es

$$\begin{array}{c} \Delta V(k) \leq -\frac{1}{R_k}\widetilde{\theta}_i^T(k)B_kB_k^T\widetilde{\theta}_i(k) - \frac{1}{R_k}\zeta_{i,k}B_k^T\widetilde{\theta}_i(k) \\ -\frac{1}{R_k}\widetilde{\theta}_i^T(k+1)\overline{P}_{k+1}^{-1}P_kB_k\zeta_{i,k} \end{array} \tag{5.61}$$

El último término de (5.61) es

$$\begin{array}{l} \frac{\zeta_{i,k}}{R_k}\widetilde{\theta}_i^T(k+1)\overline{P}_{k+1}^{-1}P_kB_k = \frac{\zeta_{i,k}}{R_k}B_k^TP_k\overline{P}_{k+1}^{-1}\widetilde{\theta}_i(k+1) \\ = \frac{\zeta_{i,k}}{R_k}B_k^TP_k\left[P_k^{-1}\widetilde{\theta}_i(k) - \frac{1}{R_k}\overline{P}_{k+1}^{-1}P_kB_k\zeta_{i,k}\right] \\ = \frac{\zeta_{i,k}}{R_k}B_k^T\widetilde{\theta}_i(k) - \frac{\zeta_{i,k}^2}{R_k^2}B_k^TP_k\overline{P}_{k+1}^{-1}P_kB_k \end{array}$$

Ahora aplicando el lema de inversa de matriz para $\overline{P}_{k+1} = P_k - \frac{1}{R_k}P_kB_kB_k^TP_k$

$$(A+BCD)^{-1} = A^{-1} - A^{-1}B\left(DA^{-1}B + C^{-1}\right)^{-1}DA^{-1} \tag{5.62}$$

con $A^{-1} = P_k$, $B = B_k$, $C^{-1} = R_2$, $D = B_k^T$, $R_k = R_2 + B_k^T P_k B_k$, so $\overline{P}_{k+1}^{-1} = P_k^{-1} + \frac{1}{R_2} B_k B_k^T$, $\frac{1}{R_k} \leq \frac{1}{R_2}$ (5.61) se llega a

$$\begin{aligned}
&\Delta V(k) \leq -\tfrac{1}{R_k}\widetilde{\theta}_i^T(k) B_k B_k^T \widetilde{\theta}_i(k) - \tfrac{1}{R_k}\zeta_{i,k} B_k^T \widetilde{\theta}_i(k) \\
&-\tfrac{\zeta_{i,k}}{R_k} B_k^T \widetilde{\theta}_i(k) + \tfrac{\zeta_{i,k}^2}{R_k^2} B_k^T P_k \overline{P}_{k+1}^{-1} P_k B_k \\
&= -\tfrac{1}{R_k}\left[\widetilde{\theta}_i^T(k) B_k B_k^T \widetilde{\theta}_i(k) + 2\zeta_i(k) B_k^T \widetilde{\theta}_i(k) + \zeta_i^2(k)\right] \\
&+ \left(\tfrac{1}{R_k} + \tfrac{1}{R_k^2} B_k^T P_k \left(P_k^{-1} + \tfrac{1}{R_2} B_k B_k^T\right) P_k B_k\right) \zeta_{i,k}^2 \\
&= -\tfrac{1}{R_k}\left[B_k^T \widetilde{\theta}_i(k) + \zeta_{i,k}\right]^2 \\
&+ \tfrac{1}{R_2}\left(1 + \tfrac{1}{R_k} B_k^T P_k B_k + \tfrac{1}{R_k^2}\left(B_k^T P_k B_k\right)^2\right) \zeta_{i,k}^2
\end{aligned} \tag{5.63}$$

Usando (5.51), (5.63) nos da

$$\begin{aligned}
&\Delta V(k) \leq -\tfrac{1}{R_k} e_i^2(k) \\
&+ \tfrac{1}{R_2}\left(1 + \tfrac{B_k^T P_k B_k}{R_2 + B_k^T P_k B_k} + \tfrac{\left(B_k^T P_k B_k\right)^2}{\left(R_2 + B_k^T P_k B_k\right)^2}\right) \zeta_{i,k}^2 \\
&\Delta V(k) \leq -\tfrac{1}{R_k} e_i^2(k) + \tfrac{1}{R_2}(1 + 1 + 1)\,\zeta_{i,k}^2
\end{aligned}$$

Entonces

$$\Delta V(k) \leq -\frac{1}{R_k} e_i^2(k) + \frac{3}{R_2}\overline{\zeta}_i^2 \tag{5.64}$$

donde $\left|\zeta_{i,k}\right| < \overline{\zeta}_i$. Debido que $\frac{1}{R_k} e_i^2(k) \geq \frac{3}{R_2}\overline{\zeta}_i^2$ y con $R_k > 0$, entonces $\Delta V(k) \leq 0$. También se tiene que

$$\widetilde{\theta}_i^T(k) P_k^{-1} \widetilde{\theta}_i(k) \leq \widetilde{\theta}_i^T(1)\, P_1^{-1} \widetilde{\theta}_i(1) \leq \lambda_{\text{máx}}\left(P_1^{-1}\right) \left\|\widetilde{\theta}_i(1)\right\|^2$$

De (5.48) tenemos $P_{k+1} = P_k - \frac{1}{R_k} P_k B_k B_k^T P_k + R_1$,porque $\frac{1}{R_k}\left\|P_k B_k B_k^T P_k\right\| \geq \|R_1\|$, $x^T P_{k+1}^{-1} x \geq x^T P_k^{-1} x$

$$\begin{aligned}
\lambda_{\text{mín}}\left(P_1^{-1}\right) \left\|\widetilde{\theta}_i(k)\right\|^2 &\leq \widetilde{\theta}_i^T(k) P_1^{-1} \widetilde{\theta}_i(k) \\
&\leq \widetilde{\theta}_i^T(k) P_k^{-1} \widetilde{\theta}_i(k)
\end{aligned}$$

elegimos $\lambda_{\text{mín}}\left(P_1^{-1}\right) \neq 0$, entonces

$$\left\|\widetilde{\theta}_i(k)\right\|^2 \leq \frac{\lambda_{\text{máx}}\left(P_1^{-1}\right)}{\lambda_{\text{mín}}\left(P_1^{-1}\right)} \left\|\widetilde{\theta}_i(1)\right\|^2$$

$\widehat{\theta}_i(k)$ es acotada. Por otro lado, si $\frac{1}{R_k}e_i^2(k) < \frac{3}{R_2}\overline{\zeta}_i^2$ o $\frac{1}{R_k}\left\|P_kB_kB_k^TP_k\right\| < \|R_1\|$ $s_k = 0$, $\widehat{\theta}_i(k+1) = \widehat{\theta}_i(k)$, $\widehat{\theta}_i(k)$ es acotado. Para todos los casos, los pesos $\widehat{\theta}_i(k)$ son acotados. De (5.51) sabemos que la acotación de B_k^T, $\widetilde{\theta}_i(k)$ y $\zeta_{i,k}$ implicando que $e_i(k)$ es acotado, P_k es también acotado (5.45). De (5.35) el error de identificación $\epsilon_i(k)$ es acotado.

Debido a que $e_i^2(k)$ es acotado, es acotado para todo el tiempo $e_i^2(k) \geq 3\overline{\zeta}_i^2$. Teniendo T_j denota el intervalo del tiempo durante el cual $e_i^2(k) \geq 3\overline{\zeta}_i^2$. (a) Si solo se tiene tiempo finito $e_i^2(k)$ que permanece fuera del circulo de radio $3\overline{\zeta}_i^2$ (y entonces re ingresa), $e_i^2(k)$ eventualmente se colocara dentro del circulo. (b) Si $e_i^2(k)$ deja el circulo un numero infinito de veces, entonces el numero de veces que $e_i^2(k)$ sale del circulo es finito,

$$\sum_{j=1}^{\infty} T_j < \infty, \quad \lim_{j\to\infty} T_j = 0 \tag{5.65}$$

Entonces $e_i^2(k)$ es acotado vía un conjunto invariante. Dado $e_{i,j}^2(k)$ denota el error de seguimiento mas grande durante el intervalos T_j. Entonces (5.65) y que $e_{i,j}^2(k)$ es acotado, implica que

$$\lim_{k\to\infty}\left[-\frac{1}{R_k}e_i^2(k) + \frac{3}{R_2}\overline{\zeta}_i^2\right] = 0$$

Teniendo $\frac{1}{R_k}e_{i,j}^2(k)$ convergerá a $\frac{3}{R_2}\overline{\zeta}_i^2$, y se cumple (5.54). ■

Comentario 5.5 *Este resultado puede considerarse como una extensión de [?] para redes neuronales recurrentes. La principal diferencia teórica es que nosotros aplicamos la zona muerta para el aprendizaje de redes neuronales. La zona muerta s_k depende de dos sistemas de ruido (5.40), $\omega_i(k)$ (corresponde a R_1) y $\zeta_{i,k}$ (corresponde a $\overline{\zeta}_i^2$). Si los pesos convergen a el mismo valor (no el peso óptimo), (5.41) puede expresarse como*

$$E\left\{\omega_i(k)\right\} = 0, \quad E\left\{\omega_i(k)\omega_i(k)^T\right\} = R_1 = \frac{1}{k}\overline{R}_1$$

Podemos escribir la zona muerta como $s_k = \begin{cases} e_i(k) & \frac{1}{R_k}e_i^2(k) \geq \frac{3}{R_2}\overline{\zeta}_i^2 \\ 0 & \frac{1}{R_k}e_i^2(k) < \frac{3}{R_2}\overline{\zeta}_i^2 \end{cases}$ *que solo depende de $\zeta_{i,k}$. En la prueba del Teorema 1, después de (5.64) nosotros tenemos*

$$\begin{aligned}\widetilde{\theta}_i^T(k+1)P_{k+1}^{-1}\widetilde{\theta}_i(k+1) &\leq \widetilde{\theta}_i^T(1)P_1^{-1}\widetilde{\theta}_i(1) \\ &\leq \lambda_{\text{máx}}\left(P_1^{-1}\right)\left\|\widetilde{\theta}_i(1)\right\|^2\end{aligned}$$

De (5.48) sabemos que

$$\begin{aligned} P_{k+1} &= P_k + R_1 - \tfrac{1}{R_k} P_k B_k B_k^T P_k \\ &\leq P_k + R_1 \leq P_1 + kR_1 = P_1 + \overline{R}_1 \end{aligned}$$

Entonces $P_{k+1}^{-1} \geq \left(P_1 + \overline{R}_1\right)^{-1}$,

$$\begin{aligned} &\widetilde{\theta}_i^T(k+1) P_{k+1}^{-1} \widetilde{\theta}_i(k+1) \geq \widetilde{\theta}_i^T(k+1) \left(P_1 + \overline{R}_1\right)^{-1} \widetilde{\theta}_i(k+1) \\ &\geq \lambda_{\text{mín}} \left(\left(P_1 + \overline{R}_1\right)^{-1}\right) \left\|\widetilde{\theta}_i(k+1)\right\|^2 \end{aligned}$$

elegimos P_1 *tal que* $\lambda_{\text{mín}} \left(\left(P_1 + \overline{R}_1\right)^{-1}\right) \neq 0$, *entonces*

$$\left\|\widetilde{\theta}_i(k+1)\right\|^2 \leq \frac{\lambda_{\text{máx}}\left(P_1^{-1}\right)}{\lambda_{\text{mín}} \left(\left(P_1 + \overline{R}_1\right)^{-1}\right)} \left\|\widetilde{\theta}_i(1)\right\|^2$$

Y se llega al mismo resultado que el Teorema 1

Los siguiente pasos muestran como entrenar los pesos de una red neuronal recurrente con el algoritmo del filtro de Kalman en zona muerta:

1. Construir la red neuronal recurrente (5.35) para identificar el sistema no lineal desconocido (5.33). La matriz A es seleccionada para que sea estable.

2. Re escribimos la red neuronal en su forma lineal $y(k) = B_k^T \Theta_k + \zeta(k)$, $\Theta_k = \left[V_{1,k}, W_{1,k}^T, V_{2,k}, W_{2,k}^T\right]^T = \left[\theta_1(k), \cdots \theta_n(k)\right]$, $B_k = \left[\sigma, \sigma' V_{1,k}^T x, \phi u, \phi' \text{diag}(u) V_{2,k}^T x(k)\right]^T$

3. Los pesos se entrenan como

$$\begin{aligned} \widehat{\theta}_i(k+1) &= \widehat{\theta}_i(k) - s_k \left(R_2 + \widehat{B}_k^T P_k \widehat{B}_k\right)^{-1} P_k \widehat{B}_k, \widehat{B}_k \\ &= \left[\sigma, \sigma' \widehat{V}_{1,k}^T x, \phi u, \phi' \text{diag}(u) \widehat{V}_{2,k}^T x(k)\right]^T \end{aligned}$$

   La zona muerta s_k es definida en (5.53), R_1 es elegido como una constante positiva pequeña. R_2 y $\overline{\zeta}_i$ son elegidas constantes positivas.

4. P_k cambia con el algoritmo del filtro de Kalman:

$$\begin{aligned} P_{k+1} &= R_1 + \left[I - K_k B_k^T\right] P_k, K_k \\ &= P_k B_k \left(R_2 + B_k^T P_k B_k\right)^{-1} \end{aligned}$$

Con condiciones iniciales para los pesos $\widehat{\theta}_i(0)$ y $P_0 > 0$, podemos inicializar la identificación de los sistemas con la red neuronal recurrente.

Comentario 5.6 *Se han implementado dinstintos algoritmos de aprendizaje para determinar los coeficientes de la red neuronal wavelet Haar, el algortimo del filtro extendido de Kalman presenta mejores resultados en convergencia al resultado al evitar mínimos locales, sin embargo representa computacionalmente una carga mayor, por otro lado el algortimo Back Propagation es muy sencillo de implementar sin embargo puede caer en mínimos locales.*

5.5. Simulaciones

5.5.1. Modelado del sistema Mackey-Glass

La ecuación Mackey-Glass es una ecuación diferencias con retraso con la forma

$$\frac{dx}{dt} = \beta \frac{x_\tau}{1 + x_\tau^n} - \gamma x, \text{ donde } \gamma, \beta, n < 0,$$

con β, γ, τ, n números reales, y x_τ representa el valor de la variable x al tiempo $(t - \tau)$. Dependiendo el valor de los parámetros, esta ecuación muestra un comportamiento periódico y un comportamiento caótico. Usando $\beta = 0{,}2, \gamma = 0{,}1, \tau = 17, n = 10$.

Hacemos la identificación usando el algoritmo evolutivo visto en el capítulo anterior y tenemos.

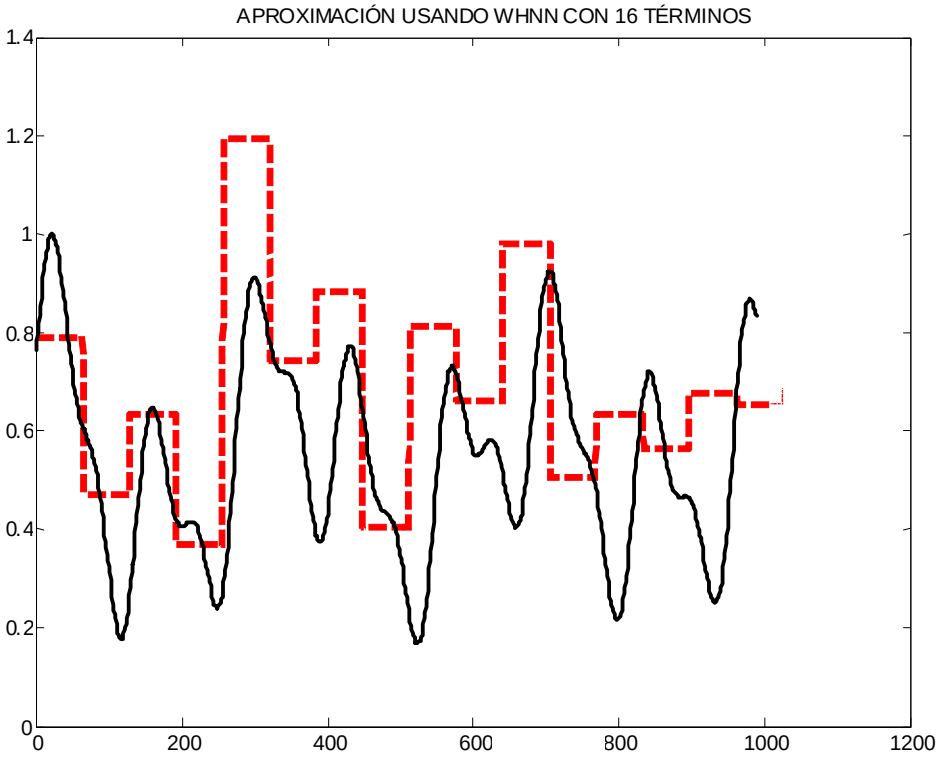


Figura 5.2: Aproximación usando WHNN con 16 términos.

5.5.2. Modelo Matemático de la infección de VIH

El modelo básico de la dinámica de la infección por el virus del VIH es descrita por un modelo 3-D de tiempo discreto, dicho modelo se desarrolla en [79]. Este modelo esta dado como:

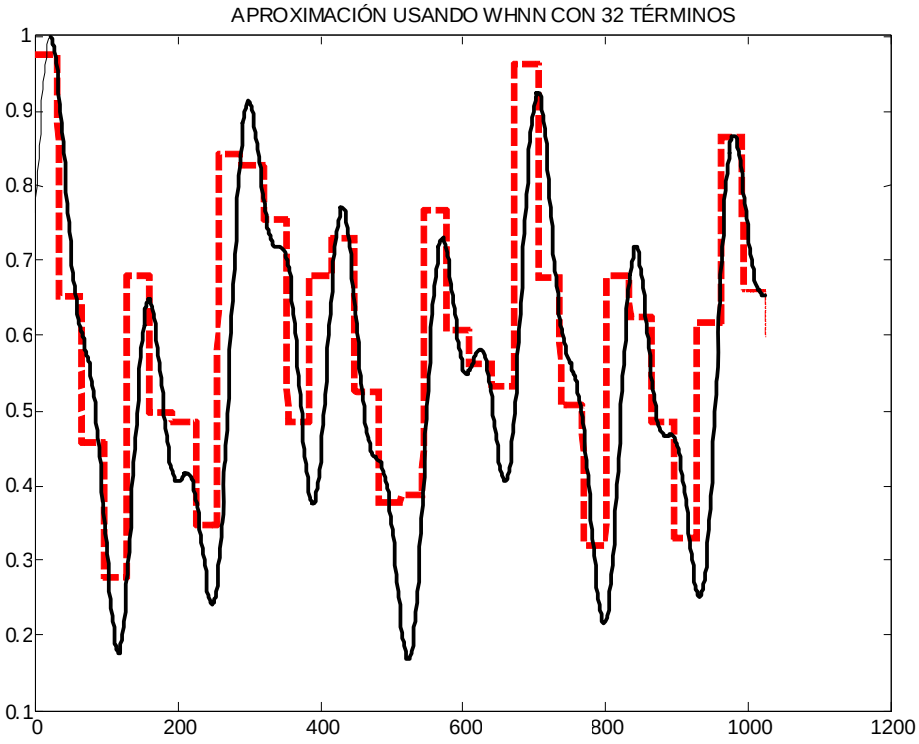


Figura 5.3: Aproximación usando WHNN con 32 términos.

$$T_{k+1} = s + (1-\delta)T_k - \beta T_k V_k + \rho(T_k, V_k) \tag{5.66}$$

$$T^*_{k+1} = \beta T_k V_k + (1-\mu)T^*_k \tag{5.67}$$

$$V_{k+1} = kT^*_k + (1-c)V_k \tag{5.68}$$

incluye la dinámica de las células no infectadas CD4+ T-cells, las células infectadas CD4+ T-cells, y los virus. $\rho(T,V) = r\left(\frac{V}{KV}\right)$ es la proliferación de CD4+ T-cells. En el modelo 3-D, $T(CD4/mm^3)$ representa la cantidad de células no infectadas CD4+ T-cells, $T*(CD4/mm^3)$ representa la cantidad de células infectadas CD4+T-cells, y V (RNA *copies/ ml*) representa

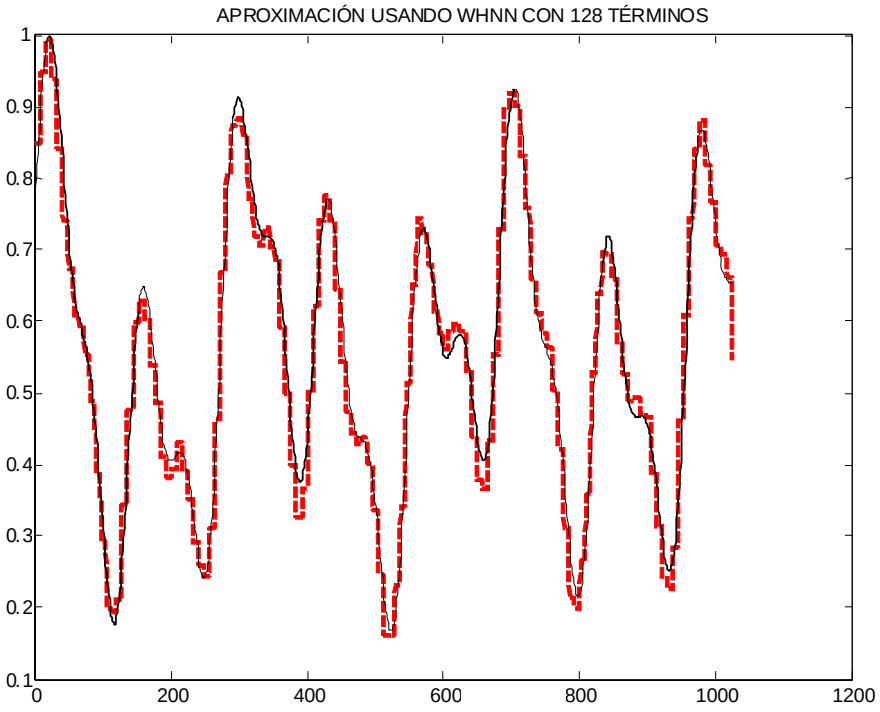


Figura 5.4: Aproximación usando WHNN con 128 términos.

los virus libres. Las partículas de virus libres infectan a las células sanas a una razón de proporcional tanto a T como a V (βTV). Ellas son removidas del sistema a una razón de c. En (5.66), se asume que las células sanas CD4+ T-cells, son producidas a un ritmo constante s. Esta es un modelo simple de la producción de las células CD4+T-cells. μ representa la taza con la que las células infectadas son removidas del sistema. Los términos $\rho(T,V) = r\left(\frac{V}{KV}\right)$ modelan la proliferación de células infectadas, donde r es índice máximo de proliferación. K es la constante media de proliferación.

Los parámetros usados en la simulación fueron tomados de [79]: $s = 10, \delta = 0{,}01, \mu = 0{,}09, k = 1000, c = 0{,}031, r = 0$. Y las condiciones iniciales fueron elegidas como:

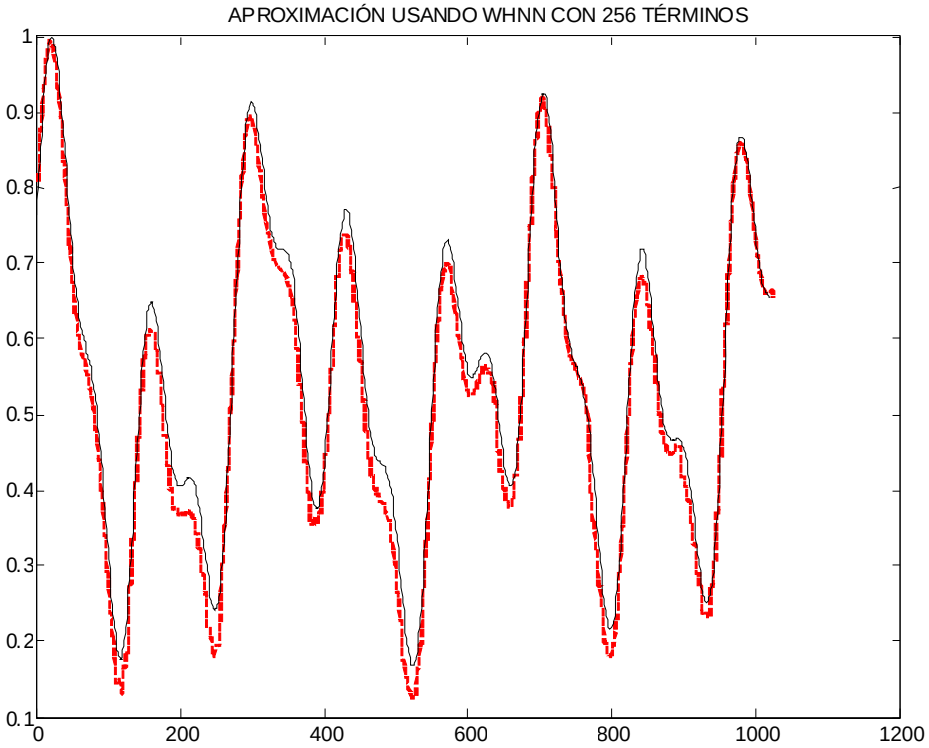


Figura 5.5: Aproximación usando WHNN con 256 términos.

$$T_0 = 1000CD_4/\text{mm}^3$$
$$T_0 = 50CD_4/\text{mm}^3$$
$$V_0 = 100 \text{ copies/ml}$$
$$V_0^* = 100 \text{ copies/ml}$$

De [80], tomamos la matriz A como:

$$A = \begin{bmatrix} -1x10^{-35} & 0 & 0 \\ 0 & -0{,}22 & 0 \\ 0 & 0 & -0{,}255 \end{bmatrix}$$

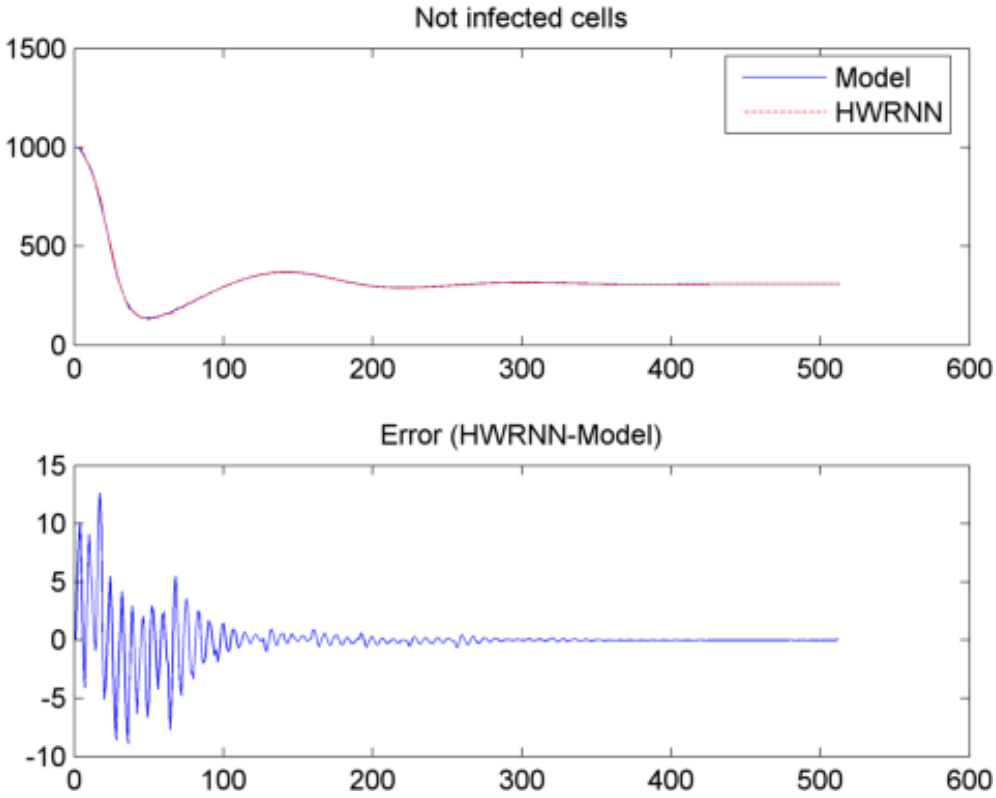


Figura 5.6: Celulas no Infectadas

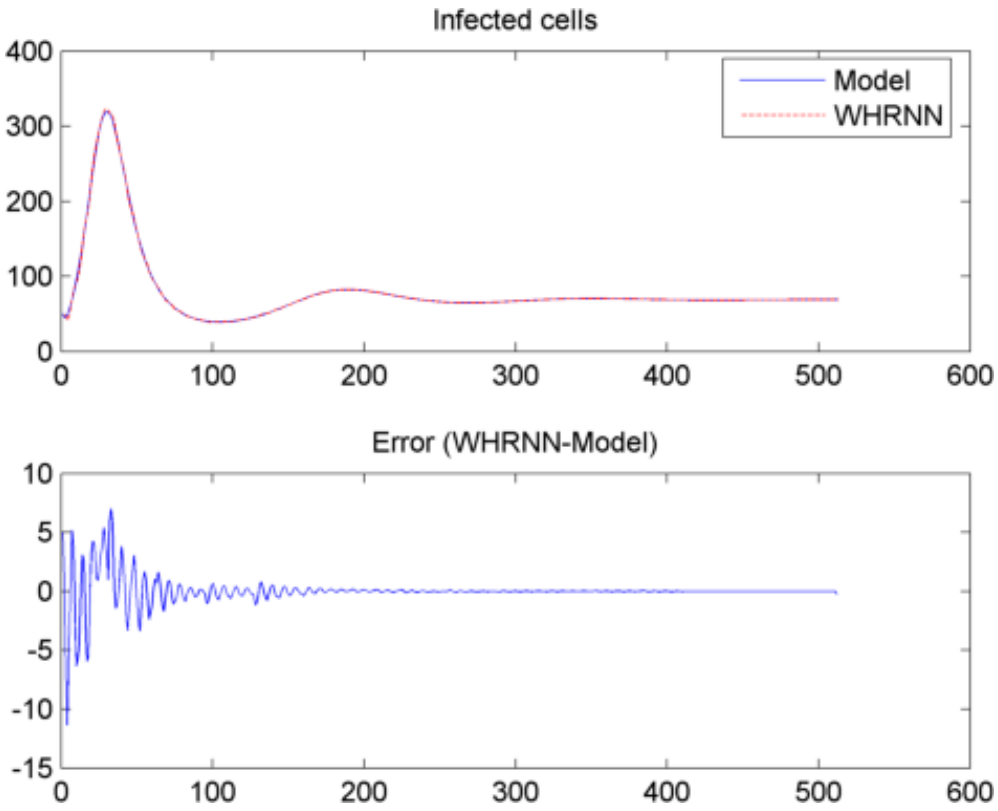


Figura 5.7: Celulas Infectadas

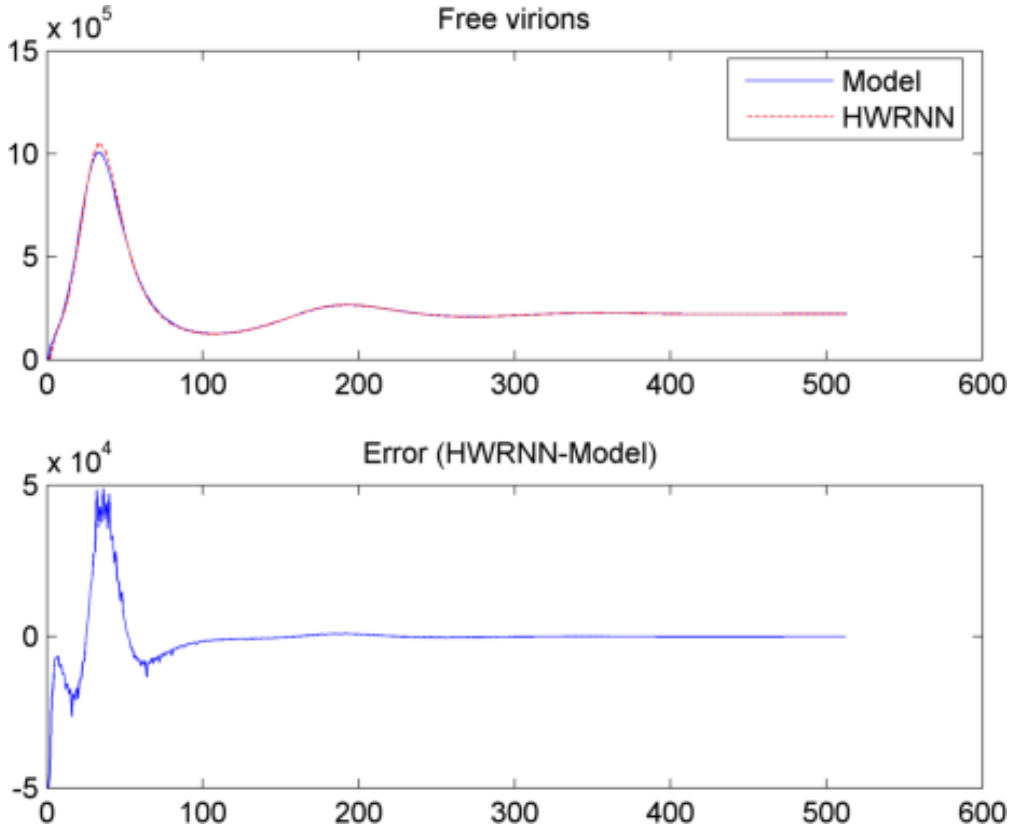


Figura 5.8: Celulas Libres de Virus

Capítulo 6

Modelado de Series de Tiempo vía Redes Neuro Difusas Wavelet.

6.1. Introducción

El concepto de la lógica difusa está basado en lo relativo, y se encuentra en los fenómenos que se producen comúnmente en el medio natural mundo y como el cerebro humano para percibir, reconocer y categorizar utiliza una serie de reglas que establecen una diferenciación en las observaciones. La lógica difusa se encuentra donde los límites en los conceptos y las observaciones no son claras y son vagas., La lógica difusa permite tratar información imprecisa, en términos de conjuntos difusos se tiene que estos conjuntos se combinan en reglas para definir acciones, la teoría de lógica difusa parte de la teoría clásica de conjuntos, añadiendo una función de pertenencia al conjunto, definida ésta como un número real entre 0 y 1, así se introduce el concepto de lógica difusa determinado a un valor lingüístico. Para cada conjunto o subconjunto difuso se define una función de pertenencia $\mu_A(t)$, que indica el grado en el cual la variable t está incluida en el concepto que esta representado por la etiqueta A. En se tiene la definición de un conjunto difuso.

Definición 6.1 *Un conjunto difuso A en un universo dado U es aquel que para cualquier*

$u \in U$, hay un correspondiente número real en $\mu_A(u) \in [0,1]$ para u, donde $\mu_A(u)$ es llamada el grado de membresía de u perteneciente a el conjunto A.

Esto significa que hay un mapeo

$$\mu_A : U \to [0,1], \quad u \to \mu_A(u)$$

dicho mapeo es llamado la función de membresía del conjunto A.

Un sistema difuso es un sistema basado en el conocimiento y una colección de reglas, del tipo **SI-ENTONCES**, tres tipos de reglas son utilizadas usualmente en la literatura [64]:

1. Sistemas difusos puros.
2. Takagi-Sugeno-Kang (TSK).
3. Sistemas difusos con fusificador y dedifusificador.

La configuración básica del sistema difuso puro se muestra en la Figura 6.1 .En el sistema básico tenemos en la entrada un conjunto de reglas difusos, que son mapeadas por un motor de inferencia difuso a otro conjunto de reglas difusas, el principal problema con los sistemas difusos puros es que las entradas y las salidas con conjuntos difusos, los sistemas TSK hacen un mapeo de variables de valor real a una salida de valores reales, su estructura se observa en la Figura 6.2. El sistema difuso con fusificador-defusificador, transforma una entrada de valores reales a un conjunto de variables difusas para tratarlas usando un motor de inferencia difusa que mapea con un dedifusificador a variables reales.

Al combinar los sistemas difusos con las redes neuronales se adquiere la capacidad de aprendizaje de reglas lingüísticas y optimizar las ya existentes. Lo que significa crear una base de reglas o funciones de membresía basadas en el entrenamiento de datos presentados como un problema de aprendizaje. Los sistemas neuro difusos combinan la habilidad de aprender de las redes neuronales, con la propiedad de inferencia de los sistemas difusos [65], [63], [67] Para crear la base de reglas se debe tener una función de membresía inicial, se encuentran tres formas de crear está base de reglas :

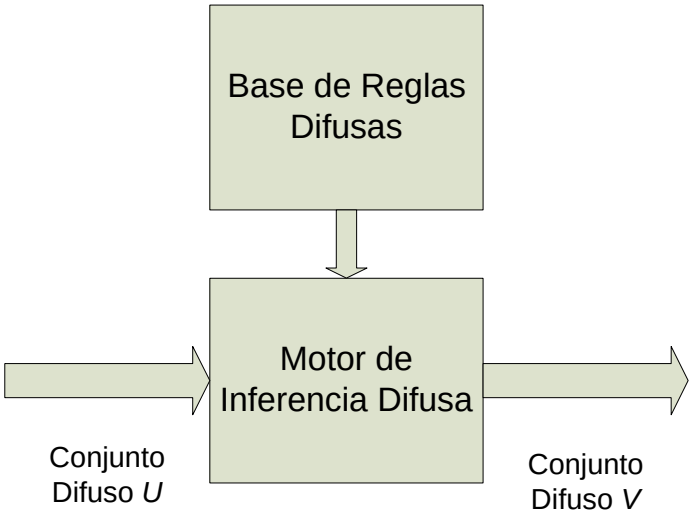


Figura 6.1: Configuración Básica de un Sistema Difuso Puro

1. El sistema difuso comienza sin reglas, y crea nuevas reglas cuando el problema de aprendizaje se resuelve. La creación de una nueva regla se obtiene por un patrón de entrenamiento.

2. El sistema comienza con todas las reglas que pueden ser creadas debido a la partición de las variables y borra las reglas insuficientes de la base de reglas

3. El sistema comienza con una base de reglas (posiblemente escogida al azar) con un cierto número de reglas que son reemplazadas durante el aprendizaje mientras se revisa la consistencia de la base de reglas a cada paso.

Los modelos neuro difusos mas usados son Mamdani y TSK, en los sistemas neuro difusos del tipo Mamdani, la implicación difusa mínima es usada en el razonamiento difuso [68], por otro lado, el modelo TSK para redes neuro difusas, la consecuencia de cada regla es la entrada a una función.[69]. Los sistemas neuro difusos han sido probados ampliamente y han demostrado su efectividad en problemas de aproximación de funciones, identificación de sistemas no lineales y control.

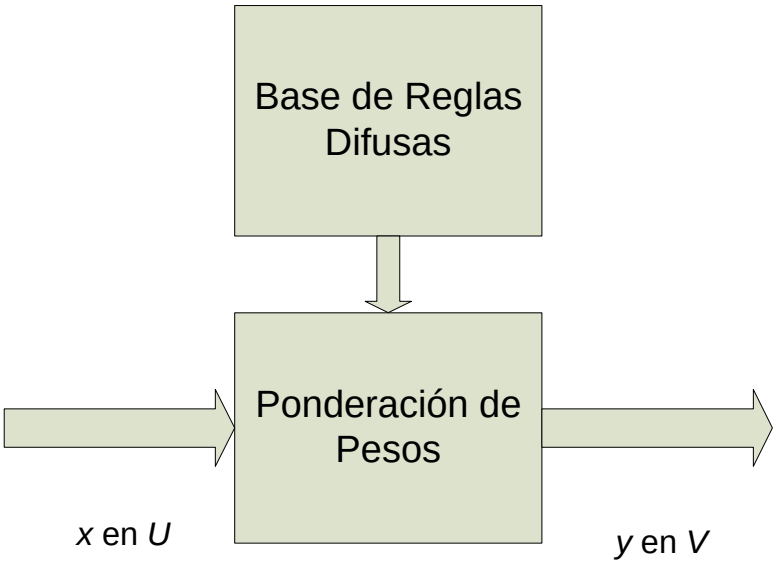


Figura 6.2: Configuración Básica de un Sistema Difuso del típo Takagi-Sugeno-Kang.

El modelo Mandani se representa por las reglas difusas en la forma siguiente:

$$R^i: \text{ si } x_1 \text{ es } A_{1i} \text{ y } x_2 \text{ es } A_{2i} \text{ y,...,} x_n \text{ es } A_{ni} \text{ entonces } y_1 \text{ es } B_{1i} \text{ y,...,} y_m \text{ es } B_{mi} \tag{6.1}$$

en donde tanto la parte antecedente como la parte consecuente esta formado por etiquetas lingüisticas (A, B) de conjuntos difusos, este tipo de técnica ofrece una expresión cualitativa del sistema, es decir hay un mapeo de la entrada lingüística $X = [x_1, ..., x_n] \in \mathbb{R}^n$ a una variable de salida lingüística. $A_{1i}, ..., B_{mi}$ son conjuntos difusos, para cada variable lingüística x_i, U_i es el universo de colección de posibles patrones; $x_i \in U_i$, $i = 1, 2, .., n$. Sea U un producto cartesiano de universos $U = U_1 \times U_2 \times ... \times U_n$. Y sea V el universo de y. Se tienen m funciones de membresía para cada x_i, $i = 1, 2, ..., n$. Produciendo medidas de membresía para cada variable con respecto a los conjuntos difusos A_{ni} y B_{mi}, y $\mu_{A_{ni}}(x_n) : U_i \to [0, 1]$ y $\mu_{B_m}(y) : V \to [0, 1]$.

Cada una de las reglas lógicas difusas puede ser representado por una relación difusa $R^j = (A_{1i} \text{ y } A_{2i} \text{ y } ... \text{ y } A_{ni}) \to B^m$ definido en $U_1 \times U_2 \times ... \times U_n \times V$. Donde el operador

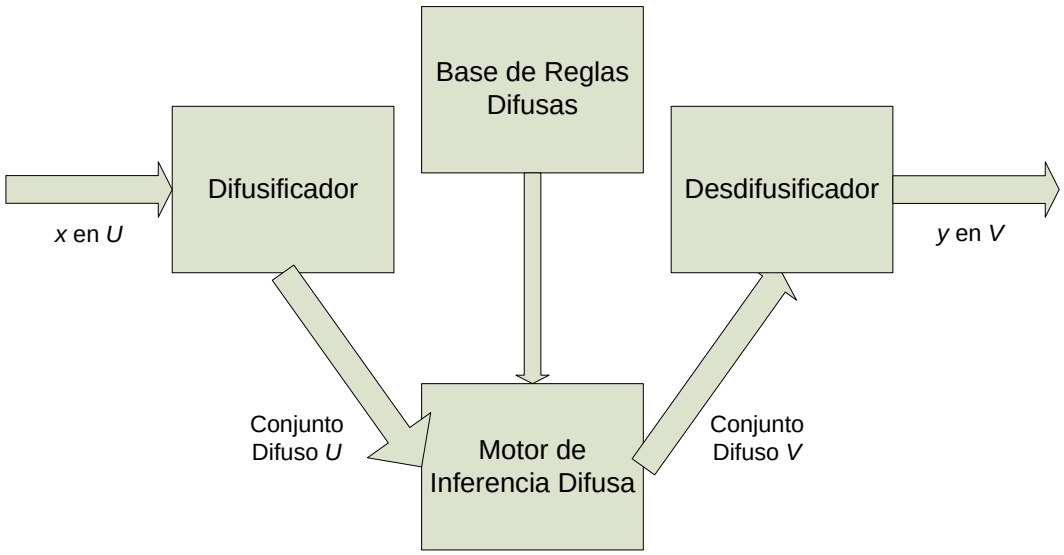


Figura 6.3: Configuración Básica de un Sistema Difuso con Fusificador y Desfusificador.

producto para la relación difusa se escribe como:

$$\mu_{R^i}(x_1, ..., x_n, y) = \mu_{A_{1i}}(x_1) \cdot ... \cdot \mu_{A_{ni}}(x_n) \cdot \mu_{B_{1i}}(y) \tag{6.2}$$

Las relaciones difusas R^i forman la relación completa R ENTONCES como una unión difusa de la siguiente manera:

$$\mu_R(x_1, ..., x_n, y) = \bigvee_{i=1}^{m} \mu_{R^i}(x_1, ..., x_n, y) \tag{6.3}$$

donde $\bigvee$ denota el operador máx.

Las típicas reglas de RWF se describen a continuación:

$$R^i : \text{Si } x_1 \text{ es } A_1^i \text{ y } x_1 \text{ es } A_1^i \text{ y ... y } x_q \text{ es } A_q^i,$$
$$entonces\ \hat{y}_i = \sum_{k=1}^{Ti} w_{M_i t^k} \Psi^k_{M_i t^k}$$
$$\text{y } w_{m_i \varepsilon R}$$

Usando el bien conocido mecanismo de inferencia TSK se tiene

$$\hat{\mu}_i(x) = \frac{\mu_i(x)}{\sum_{i=1}^{c} \mu_i(x)} \tag{6.4}$$

La salida de la Red Neuro Difusa Wavelet esta dada por:

$$\hat{y} = \frac{\sum_{i=1}^{c} \mu_i(x)\hat{y}_i}{\sum_{i=1}^{c} \mu_i(x)} = \sum_{i=1}^{c} \hat{\mu}_i(x)\hat{y}_i \tag{6.5}$$

donde $\mu_i(x) = \prod_{j=1}^{q} A_j^i(x_j)$, se puede observar que segun este planteamiento cada $i-esima$ regla corresponde a un solo nivel de escalamiento,

Como se ha visto en capítulos anteriores, hay dos tipos de estructuras de redes neuronales wavelets, una donde los parámetros de traslación y ensanchamiento son fijos y otra estructura donde esos parámetros son ajustados por algún sistema de aprendizaje, se propone un algoritmo para encontrar los mejores valores para m y n, de una red wavelet, sin embargo se parte de pre establecer cuantas neuronas se utilizaran, y para conocer los valores óptimos que tomara el coeficiente de ensanchamiento es necesario

$$f(x) \approx \sum_{m,n}^{N} w_{mn} \Psi_{mn}(x)$$

donde $w_{mn} = \langle f, S^{-1}\Psi_{mn} \rangle$.

Los sistemas neuro difusos en combinación con una estructura Wavelet es una buena idea, donde se pueden aprovechar tanto las ventajas de un sistema neuro difuso del tipo Mamdani como un TSK, como sabemos un sistema de tipo Mamdani, se puede ver como

aproximaciones lineales locales, en donde cada regla junto con su respuesta ENTONCES representa una partición de el espacio de salida.

Este tipo de estructura ha sido utilizada en [58], [54], sin embargo se ha desaprovechado el valor de el conocimiento previo y de las variables lingusticas, del algoritmo order-FWT se observa que partiendo de arreglo, se inicia el algoritmos haciendo diferencias entre dos valores continuos, y se determina el valor de el coeficiente wavelet de la iteración l, y la siguiente iteración hará la diferencia de las diferencias, lo que hace evidente que si no hay cambio entre dos valores continuos, la diferencia sera cero, y el coeficiente wavelet producto de esa diferencia sera cero, esta diferencia entre dos elementos continuos del arreglo es la determinación de la pendiente de la función a el paso k, lo que nos resulta que si establecemos una serie de reglas determinadas por la pendiente a el paso k, podemos tener una estructura neuronal donde se eliminen los coeficientes wavelets que tendrán valor igual a cero.

6.2. Algoritmo de Aprendizaje

Las RNW tienen grandes ventajas como aproximador universal sobre las redes neuronales multicapa, sin embargo determinar los valores de la traslación y ensanchamiento representa un problema que se necesita solucionar, usualmente dos diferentes métodos se han propuesto, el primero determina un número determinado de valores para el ensanchamiento y traslación y solo la capa de salida es ajustada con un peso, el otro tipo es una base variable donde los parámetros de traslación y ensanchamiento son ajustados utilizando capas ocultas de la red neuronal, como se vio en los capítulos anteriores, es posible establecer una estructura creciente de una red neuronal que modifica su estructura dependiendo un índice de desempeño. Una alternativa al uso de capas ocultas para ajustar los parámetros de traslación y ensanchamiento ha sido utilizar sistemas difusos, en [54], se aprovecha que el parámetro de resolución tiene interpretaciones físicas claras para ajustar la aproximación y se establece un conjunto de reglas segun el modelo fuzzy Takagi-Sugeno-Kang (TSK) [63].

Un modelo Red Neuro Difuso Wavelet (RNDW) permite hacer mejores aproximaciones sin la necesidad de aumentar el número de neuronas, el modelo TSK consiste en una serie

de reglas que particiona el espacio en modelos locales, cada regla fuzzy corresponde a una wavelet con un valor especifico de resolución. Una típica RNDW puede describirse por las siguientes reglas:

Un modelo difuso se presenta como una colección de reglas difusas de la siguiente forma (Mamdani fuzzy model [62])

$$\mathrm{R}^i\text{: IF } x_1 \text{ is } A_{1i} \text{ and } x_2 \text{ is } A_{2i} \text{ and } \cdots \; x_n \text{ is } A_{ni} \text{ THEN } \widehat{y}_1 \text{ is } B_{1i} \text{ and} \cdots \widehat{y}_m \text{ is } B_{mi} \quad (6.6)$$

Se tienes $l(i = 1, 2 \cdots l)$ IF-THEN reglas para mapear de un vector de entrada lingüístico $X = [x_1 \cdots x_n] \in \Re^n$ a una salida linguistica $\widehat{Y}(k) = [\widehat{y}_1 \cdots \widehat{y}_m]^T \in R^{m \times 1}$. $A_{1i}, \cdots A_{ni}$ y $B_{1i}, \cdots B_{mi}$ son conjuntos difusos. Cada variable de entrada x_i tiene l_i conjuntos difusos. En el caso de una conexión completa, $l = l_1 \times l_2 \times \cdots l_n$. De [68] sabemos que la salida p, de un sistema difuso puede expresarse como

$$\widehat{y}_p = \left(\sum_{i=1}^{l} w_{pi} \left[\prod_{j=1}^{n} \mu_{A_{ji}} \right] \right) / \left(\sum_{i=1}^{l} \left[\prod_{j=1}^{n} \mu_{A_{ji}} \right] \right) = \sum_{i=1}^{l} w_{pi} \phi_i \quad (6.7)$$

donde $\mu_{A_{ji}}$ es la función de membresía del conjunto difuso A_{ji}, w_{pi} es el punto donde $\mu_{B_{pi}} = 1$.

$$R^i \quad : \quad \text{Si } x_1 \text{ es } A_1^i \text{ y } x_2 \text{ es } A_2^i \text{ y ... y } x_q \text{ es } A_q^i, \quad (6.8)$$

$$\text{entonces } \hat{y}_i = \sum_{k=1}^{T_i} w_{M_i,t^k} \Psi_{M_i,t^k}^{(k)}(\mathbf{x}), \qquad M_i \in Z, t^k \in R^q \quad (6.9)$$

$$\text{y } w_{M_i,t^k} \in R, \; x \in R^q$$

Donde R^i es la $i - esima$ regla $(1 \leq i \leq c)$; $x_j \, (1 \leq j \leq q)$ es la variable j de $\mathbf{x}$;T_i es el número total de wavelets para la regla $i - esima$, Se tiene también que $\hat{y}_i$ es la salida del modelo local para la regla R^i, que es igual a la combinación lineal de el conjunto finito de wavelets $\Psi_{M_i,t^k}^{(k)}(\mathbf{x})$ con el mismo valor de ensanchamiento $M_i \in Z.t_i^k$ es el valor de ensanchamiento correspondiente a la wavelet k.Finalmente, A_j^i es el conjunto fuzzy caracterizado por la siguiente función Gausiana de membresía, y $A_j^i(x_j)$ es el grado de membresía de x_j en A_j^i denotado por:

$$
\begin{aligned}
A_j^i(x_j) &= e^{-\left|\left(x_j - p_{j1}^i\right)/p_{j2}^i\right|^{p_{j3}^i}} \\
&= e^{-\left(\left(\left(x_j - p_{j1}^i\right)/p_{j2}^{ip_{j3}^i}\right)^2\right)^{p_{j3}^i/2}} \\
p_{j1}^i, p_{j2}^i &\in R \text{ y } 0 < p_{j3}^i \leq 5
\end{aligned}
$$

donde p_{j1}^i representa el centro de la función de membresía, p_{j2}^i y p_{j3}^i determinan la anchura y la forma de la función de membresía. La Red Neuro Difusa Wavelet tiene una estructura de siete capas que se muestra en la Figura 6.4.

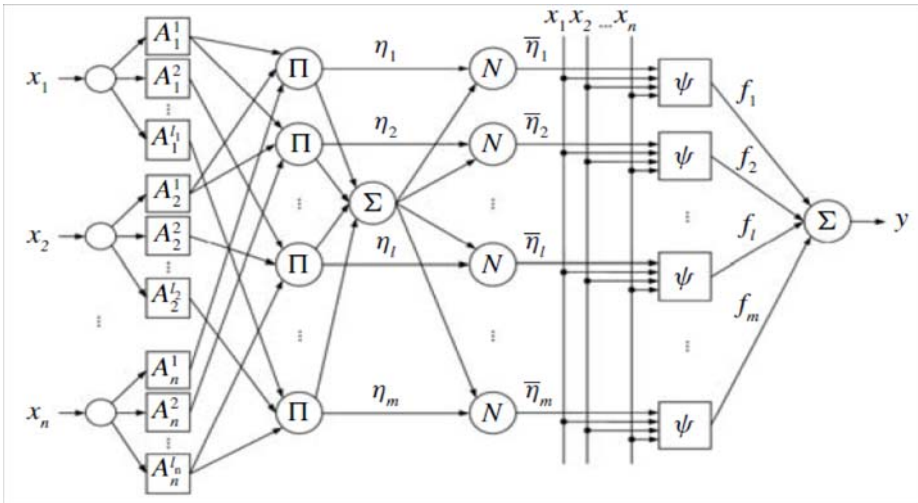


Figura 6.4: Estructura de la Red Neuro Difúsa Wavelet

Donde la Capa 1 es la capa de entrada. Las neuronas pasan la señal de entrada $x_1, x_2, ..., x_n$ a la segunda capa. La segunda capa es la capa de fusificación, cada neurona en esta capa es una función de membresía en la parte SI de la regla, la salida de esta capa es el valor de la función de membresía, hay en total l_1 Funciones de membresía para la primera entrada, l_2 para la segunda entrada, y así consecutivamente, la siguiente ecuación muestra la función de membresía i_j del tipo Gausiana para la entrada j.

$$\begin{aligned} A_j^{i_j} &= e^{\left(-\frac{1}{2}\left(\frac{x_j-\mu_{ij}}{\sigma_{ij}}\right)^2\right)}, \\ j &= 1,2,...,n \\ i_j &= 1,2,...,l_j \end{aligned}$$

La capa tercera es la capa de la regla difusa, cada neurona en está capa presenta una regla difusa. En está capa se usa la multiplicación como el operador AND. La salida del nodo l es:

$$\eta_l = \prod_{j=1}^{n} A_j^{i_j}(x_j) \tag{6.10}$$

cada posible combinación de entradas en la función de membresía representa una regla difusa.

La cuarta capa es la capa de normalización, en donde cada neurona en esta capa calcula el factor de normalización de la regla l por:

$$\bar{\eta}_l = \frac{\eta_l}{\sum_{i=1}^{m} \eta_i}, (l = 1,...,m) \tag{6.11}$$

la salida de está capa es la contribución a el resultado final.

La capa número cinco computa el valor del peso de salida de la siguiente forma:

$$f_l = \bar{\eta}_l \Psi_l, \ \ (l = 1,...,m)\,. \tag{6.12}$$

donde:

$$\Psi_l = \sum_{i=l}^{n} w_{il} H_{b,c}(x) \tag{6.13}$$

recordando que $H_{b,c}(x)$ corresponde a la función wavelet Haar

$$H_{b,c}(x) = \begin{cases} 1 \text{ si } b2^c \leq x < (b+\frac{1}{2})2^c, \\ -1 \text{ si } (b+\frac{1}{2})2^c \leq x < (b+1)2^c \\ 0 \text{ otro caso.} \end{cases}$$

La sexta capa calcula la salida como la sumatoria de la salidas de la capa previa, y esta definida como:

$$y = \sum_{l=1}^{m} f_l \tag{6.14}$$

Está red neuro difusa wavelet es capaz de seguir e identificar cualquier sistema no lineal Considerando la planta no lineal siguiente:

$$x(k+1) = f(x(k), u(k)) \tag{6.15}$$

$$y(k) = Cx(k) \tag{6.16}$$

En esta red Neuro Difusa Wavelet usamos un algoritmo de gradiente descendente con un índice de aprendizaje adaptable para asegurar la convergencia y la velocidad de aprendizaje. Los parámetros que se aprenderán serán los de la función de membresía y los parámetros de traslación y ensanchamiento de la función wavelet, (b_{il}, c_{il}). Se presenta dividido en tres distintos algoritmos, el primero damos por fijo el valor de los parámetros de ensanchamiento y traslación de la función Wavelet Haar y de igual forma se da por conocida la función de membresía A_j^i y aprendemos solo pesos de red neuronal wavelet. El segundo algoritmo considera como fijos el ensanchamiento y traslación de la ficción Haar y entrena el valor de la función de membresía., en el tercer caso, se entrenan los parámetros de traslación y ensanchamiento, así como el coeficiente de la serie wavelet.

Iniciamos por definir el error a la salida de la RNDH como la diferencia con respecto a la función a aprender,

$$e(k) = \widehat{y}(k) - y(k). \tag{6.17}$$

y error cuadratico medio ε, se escribe como sigue:

$$\varepsilon = \frac{1}{2}\sum_{t=0}^{n} e(k)^2 \tag{6.18}$$

los cuales son comunes en los tres algoritmos.

Caso 1: Parámetros de función Haar Fijos, Función de Membresía Conocida.

Se propone la ley que ajusta los parámetros de la red que muestra a continuación: ,

$$w_{il}\left(k+1\right) = w_{il}\left(k\right) - \eta\frac{\partial\varepsilon\left(k\right)}{\partial w_{il}\left(k\right)} \tag{6.19}$$

donde η es la razón de aprendizaje y $1 > \eta > 0$. El termino $\frac{\partial\varepsilon(k)}{\partial w_j(k)}$ se calcula como

$$\frac{\partial\varepsilon\left(k\right)}{\partial w_{il}\left(k\right)} = \frac{\partial\varepsilon\left(k\right)}{\partial e\left(k\right)}\frac{\partial e\left(k\right)}{\partial y\left(k\right)}\frac{\partial y\left(k\right)}{\partial w_{il}\left(k\right)}. \tag{6.20}$$

Las derivadas parciales de la ecuación, se calculan como sigue,

$$\begin{aligned}
\frac{\partial\varepsilon}{\partial e} &= e\left(k\right), \\
\frac{\partial e\left(k\right)}{\partial y\left(k\right)} &= \frac{\mu_l}{\sum_{i=1}^{n}\mu_l} \\
\frac{\partial y\left(k\right)}{\partial w_{il}\left(k\right)} &= \frac{\eta_l}{\sum_{i=1}^{m}\eta_i} H_{b,c}(x) = \psi_l(k) \\
\eta_l &= \prod_{j=1}^{n} A_j^{ij}\left(x_j\right), \qquad A_j^{ij} = e^{\left(-\frac{1}{2}\left(\frac{x_j-\mu_{ij}}{\sigma_{ij}}\right)^2\right)},
\end{aligned}$$

finalmente escribimos la ecuación (6.19) de la siguiente forma:

$$w_{il}\left(k+1\right) = w_{il}\left(k\right) - \eta\psi_l(k)e\left(k\right)\frac{\mu_l}{\sum_{i=1}^{n}\mu_l}. \tag{6.21}$$

Caso 2: Parámetros de Haar fijos, Función de Membresía y Coeficiente de Serie Haar Desconocidos.

Para ajustar la función de membresía se tiene la ley:

$$\frac{\partial \varepsilon(k)}{\partial \mu_{ij}(k)} = \frac{\partial \varepsilon(k)}{\partial e(k)} \frac{\partial e(k)}{\partial y(k)} \frac{\partial y(k)}{\partial \mu_{ij}(k)}$$

$$\frac{\partial y(k)}{\partial \mu_{ij}(k)} = \frac{\partial y(k)}{\partial A_j^{ij}(k)} \frac{\partial A_j^{ij}(k)}{\partial \mu_{ij}(k)} = \sum e \frac{y_j - y}{\sum \mu_j} \mu_j \frac{2(x_i - \mu)}{\sigma_{ij}^2}$$

$$\mu_{ij}(k+1) = \mu_{ij}(k) - \eta e(k) \sum \frac{y_j - y}{\sum \mu_j} \mu_j \frac{2(x_i - \mu)}{\sigma_{ij}^2}. \tag{6.22}$$

Caso 3: Parámetros de Función Haar, Función de Membresía y Coeficiente de Serie Haar Desconocidos.

Para los parámetros de traslación y ensanchamiento de la función wavelet Haar se tienen las siguientes leyes:

$$\begin{aligned} \frac{\partial \varepsilon(k)}{\partial b_{ij}(k)} &= \frac{\partial \varepsilon(k)}{\partial y(k)} \frac{\partial y(k)}{\partial y_l(k)} \frac{\partial y_l(k)}{\partial \psi_l(k)} \frac{\partial \psi_l(k)}{\partial b_{ij}(k)} \\ \frac{\partial \varepsilon(k)}{\partial c_{ij}(k)} &= \frac{\partial \varepsilon(k)}{\partial y(k)} \frac{\partial y(k)}{\partial y_l(k)} \frac{\partial y_l(k)}{\partial \psi_l(k)} \frac{\partial \psi_l(k)}{\partial c_{ij}(k)} \end{aligned}$$

$$\frac{\partial e\left(k\right)}{\partial y\left(k\right)}=\frac{\mu_l}{\sum_{i=1}^{n}\mu_l},\qquad \frac{\partial y\left(k\right)}{\partial y_l\left(k\right)}=1$$

$$\frac{\partial y_l\left(k\right)}{\partial \psi_l\left(k\right)}=w_{il},\qquad \psi_l(k)=\frac{\eta_l}{\sum_{i=1}^{m}\eta_i}H_{b,c}(x),\qquad \Psi_l=\sum_{i=l}^{n}w_{il}H_{b,c}(x)$$

$$\frac{\partial \psi_l\left(k\right)}{\partial b_{ij}\left(k\right)}=\frac{\eta_l}{\sum_{i=1}^{m}\eta_i}\frac{\partial H_{b,c}(x)}{\partial b_{ij}\left(k\right)}=\frac{\eta_l}{\sum_{i=1}^{m}\eta_i}$$

$$\frac{\partial \psi_l\left(k\right)}{\partial c_{ij}\left(k\right)}=\frac{\eta_l}{\sum_{i=1}^{m}\eta_i}\frac{\partial H_{b,c}(x)}{\partial c_{ij}\left(k\right)}=\frac{\eta_l}{\sum_{i=1}^{m}\eta_i}$$

$$H_{n,m}\left(x\right)=\begin{cases}1 \text{ si } m2^n\leq x<(m+\frac{1}{2})2^n,\\ -1 \text{ si } (m+\frac{1}{2})2^n\leq x<(m+1)2^n\\ 0 \text{ otro caso.}\end{cases}$$

$$m_{ij}\left(k+1\right)=m_{ij}\left(k\right)-\eta\frac{\partial\varepsilon\left(k\right)}{\partial b_{ij}\left(k\right)}=m_{ij}\left(k\right)-\eta w_{il}e\left(k\right)\frac{\mu_l}{\sum_{i=1}^{n}\mu_l} \tag{6.23}$$

$$n_{ij}\left(k+1\right)=n_{ij}\left(k\right)-\eta\frac{\partial\varepsilon\left(k\right)}{\partial c_{ij}\left(k\right)}=n_{ij}\left(k\right)-\eta w_{il}e\left(k\right)\frac{\mu_l}{\sum_{i=1}^{n}\mu_l} \tag{6.24}$$

Comentario 6.1 *El sistema neuro difuso wavelet Haar proporciona un método para encontrar los valores de ensanchamiento y traslación, sin embargo aun el problema de decidir el número de reglas a utilizar se encuentra abierto, aun así la estructura neuro difusa wavelet proporciona un arreglo neuronal de dimensión menor al que se obtiene de una red neuronal wavelet.*

6.3. Modelado con Sistemas Neuro Difusos Wavelet con Conocimiento de Función de Membresía y Fijando Parámetros de Función Wavelet.

Cuando tenemos al información a priori de la planta, podemos construir reglas difusas del tipo (6.1) o (6.6). En está sección asumimos conocida la función de membresía $A_{1i} \cdots A_{ni}$, *i.e.* están dado, $\phi_i = \prod_{j=1}^{n} \mu_{A_{ji}} / \sum_{i=1}^{l} \prod_{j=1}^{n} \mu_{A_{ji}}$. Es conocido que los modelos (ver [74], [68]). Mamdani y TSK tienen la misma forma si $\Phi[X(k)]$ es conocida, la única diferencia es la función $W(k)$.

El objetivo del sistema neuro difuso es encontrar en valor central de $B_{1i} \cdots B_{mi}$. Que son los pesos entre la capa III y la capa IV y que se muestra en la Figura.6.4, dado que la salida $\widehat{Y}(k)$ de la red neuro difusa wavelet (6.13) puede seguir la salida $Y(k)$ de una planta no lineal (6.15). Vamos a definir el vector de error de identificación $e(k) \in R^{m \times 1}$ como

$$e(k) = \widehat{Y}(k) - Y(k) \tag{6.25}$$

Vamos a utilizar el error $e(k)$ para entrenar en línea la red neuronal difusa wavelet (6.13) dado que $\widehat{Y}(k)$ puede aproximarse $Y(k)$. De acuerdo a la teoría de aproximación de la lógica difusa [68] y de las redes neuronales [20], el proceso de identificación no lineal (6.15) puede representarse como

$$Y(k) = W^* \Phi[X(k)] - \mu(k) \tag{6.26}$$

donde W^* es un peso desconocido que puede minimizar la dinámica desconocida $\mu(k)$. El error de identificación puede ser representado por (6.25) y (6.26)

$$e(k) = \widetilde{W}(k) \Phi[X(k)] + \mu(k) \tag{6.27}$$

donde $\widetilde{W}(k) = W(k) - W^*$. En este trabajo solo tomamos la identificación del sistemas en lazo abierto identificación, asumimos que la planta (6.15) es acotada a la entrada y acotada a la salida (BIBO) estable, *i.e.*, $y(k)$ y $u(k)$ en (6.15) son acotadas. Por definición la función de membresía Φ, $\mu(k)$ es acotada (6.26). El siguiente teorema formula un algoritmo estable para el modelado difuso.

Teorema 6.1 *Si usamos una red neurona difusa wavelet (6.13) para identificar la planta no lineal (6.15), el siguiente algoritmo gradiente descendente con un índice de aprendizaje que varia en el tiempo, se tiene el error de identificación estable* $e(k)$.

$$W(k+1) = W(k) - \eta_k e(k) \Phi^T [X(k)] \tag{6.28}$$

donde el escalar $\eta_k = \dfrac{\eta}{1 + \|\Phi [X(k)]\|^2}$, $0 < \eta \leq 1$. *La normalización del error de identificación.*

$$e_N(k) = \frac{e(k)}{1 + \max\limits_k \left(\|\Phi [X(k)]\|^2 \right)}$$

satisface el siguiente promedio de desempeño.

$$\limsup_{T \to \infty} \frac{1}{T} \sum_{k=1}^{T} \|e_N(k)\|^2 \leq \overline{\mu} \tag{6.29}$$

donde $\overline{\mu} = \max\limits_k \left[\|\mu(k)\|^2 \right]$.

Demostración. Seleccionamos un escalar positivo L_k definido como:

$$L_k = \left\| \widetilde{W}(k) \right\|^2. \tag{6.30}$$

Por la ley de actualización (6.28), tenemos

$$\widetilde{W}(k+1) = \widetilde{W}(k) - \eta_k e(k) \Phi^T [X(k)]$$

Usando la desigualdades

$$\|a - b\| \geq \|a\| - \|b\|, 2\|ab\| \leq a^2 + b^2$$

para cualquier a y b. Usando (6.27) y $0 \leq \eta_k \leq \eta \leq 1$, se tiene

$$\begin{aligned}
\Delta L_k &= L_{k+1} - L_k = \left\| \widetilde{W}(k) - \eta_k e(k) \Phi^T(X) \right\|^2 - \left\| \widetilde{W}(k) \right\|^2 \\
&= \left\| \widetilde{W}(k) \right\|^2 - 2\eta_k \left\| e(k) \Phi^T(X) \widetilde{W}(k) \right\| + \eta_k^2 \|e(k) \Phi [X(k)]\|^2 - \left\| \widetilde{W}(k) \right\|^2 \\
&= \eta_k^2 \|e(k)\|^2 \|\Phi [X(k)]\|^2 - 2\eta_k \|e(k) [e(k) - \mu(k)]\| \\
&\leq \eta_k^2 \|e(k)\|^2 \|\Phi [X(k)]\|^2 - 2\eta_k \|e(k)\|^2 + 2\eta_k \|e(k) \mu(k)\| \\
&\leq \eta_k^2 \|e(k)\|^2 \|\Phi [X(k)]\|^2 - 2\eta_k \|e(k)\|^2 + \eta_k \|e(k)\|^2 + \eta_k \|\mu(k)\|^2 \\
&= -\eta_k \|e(k)\|^2 \left(1 - \eta_k \left\| \Phi^T(X) \right\|^2 \right) + \eta_k \|\mu(k)\|^2
\end{aligned} \tag{6.31}$$

Teniendo $\eta_k = \dfrac{\eta}{1+\left\|\Phi\left[X\left(k\right)\right]\right\|^2}$,

$$\begin{aligned}
&\eta_k\left(1-\eta_k\left\|\Phi\left[X\left(k\right)\right]\right\|^2\right) = \eta_k\left(1-\frac{\eta}{1+\left\|\Phi\left[X\left(k\right)\right]\right\|^2}\left\|\Phi\left[X\left(k\right)\right]\right\|^2\right)\\
&\geq \eta_k\left(1-\eta\frac{\max\limits_k\left(\left\|\Phi[X(k)]\right\|^2\right)}{1+\max\limits_k\left(\left\|\Phi[X(k)]\right\|^2\right)}\right) \geq \eta_k\left(1-\frac{\max\limits_k\left(\left\|\Phi[X(k)]\right\|^2\right)}{1+\max\limits_k\left(\left\|\Phi[X(k)]\right\|^2\right)}\right)\\
&= \frac{\eta_k}{1+\max\limits_k\left(\left\|\Phi[X(k)]\right\|^2\right)} \geq \frac{\eta}{\left[1+\max\limits_k\left(\left\|\Phi[X(k)]\right\|^2\right)\right]^2}
\end{aligned}$$

Entonces

$$\Delta L_k \leq -\pi\left\|e\left(k\right)\right\|^2 + \eta\left\|\mu\left(k\right)\right\|^2 \tag{6.32}$$

donde π es definido como

$$\pi = \frac{\eta}{\left[1+\max\limits_k\left(\left\|\Phi\left[X\left(k\right)\right]\right\|^2\right)\right]^2}$$

Por

$$n\min\left(\widetilde{w}_i^2\right) \leq L_k \leq n\max\left(\widetilde{w}_i^2\right)$$

donde $n\min\left(\widetilde{w}_i^2\right)$ y $n\max\left(\widetilde{w}_i^2\right)$ son funciones $\mathcal{K}_\infty$,y $\pi\left\|e\left(k\right)\right\|^2$ es un función $\mathcal{K}_\infty$, $\eta\left\|\mu\left(k\right)\right\|^2$ es una función $\mathcal{K}$. Entonces L_k admite una función ISS-Lyapunov como en la Definición *2.* Por el *Teorema 1*, la dinámica del error de identificación es estable de la entrada a la salida. De (6.27) y (6.30) sabemos que L_k es la función de $e\left(k\right)$ y $\mu\left(k\right)$. La "ENTRADA" corresponde a el segundo término de (6.32), *i.e.*, el modelado del error $\mu\left(k\right)$. El "ESTADO" corresponde a en primer término de (6.31), *i.e.*., el error de identificación $e\left(k\right)$. Porque la "ENTRADA" $\mu\left(k\right)$ es acotada y la dinámica es ISS, el "ESTADO" $e\left(k\right)$ es acotado.

(6.31) puede re escribirse como

$$\Delta L_k \leq -\eta\frac{\left\|e\left(k\right)\right\|^2}{\left[1+\max\limits_k\left(\left\|\Phi\left[X\left(k\right)\right]\right\|^2\right)\right]^2} + \eta\left\|\mu\left(k\right)\right\|^2 \leq -\eta\frac{\left\|e\left(k\right)\right\|^2}{\left[1+\max\limits_k\left(\left\|\Phi\left[X\left(k\right)\right]\right\|^2\right)\right]^2} + \eta\overline{\mu} \tag{6.33}$$

Sumando (6.33) de 1 a T, y usando $L_T > 0$ y L_1 es constante, nosotros obtenemos

$$\begin{aligned}
L_T - L_1 &\leq -\eta\sum_{k=1}^{T}\left\|e_N\left(k\right)\right\|^2 + T\eta\overline{\mu}\\
\eta\sum_{k=1}^{T}\left\|e_N\left(k\right)\right\|^2 &\leq L_1 - L_T + T\eta\overline{\mu} \leq L_1 + T\eta\overline{\mu}
\end{aligned}$$

(6.29) es establecida. ■

Comentario 6.2 *En general el modelo neuronal difuso wavelet no hará una aproximación exacta de la planta. Esto debido a que los parámetros de la función wavelet han sido truncado a conveniencia, además de que los parámetros de un sistema difuso con convergen completamente. La idea de una identificación en línea propuesta es forzar a la salida del sistema neuronal difuso wavelet a seguir la salida de la planta. Aun cuando sus parámetros que se encuentren no sean los óptimos, (6.29) viendo que el error normalizado converge al radio* $\overline{\mu}$*. Si el sistema neuronal difuso wavelet (6.13) puede igualar a la planta no lineal (6.15) exactamente* ($\mu(k) = 0$)*, i.e., se puede encontrar la mejor función de membresía* $\mu_{A_{ji}}$ *y* W^* *tal que el sistema no lineal pues escribirse como* $Y(k) = W^*\Phi\left[\mu_{A_{ji}}\right]$*. Teniendo* $\|e(k)\|^2 > 0$*, la misma ley de aprendizaje (6.28) haciendo el error de identificación* $\|e(k)\|$ *asintóticamente estable,*

$$\lim_{k\to\infty} \|e(k)\| = 0 \tag{6.34}$$

Comentario 6.3 *Normalizando el índice de aprendizaje* η_k *en (6.28) es variable en el tiempo con el fin de asegurar la estabilidad del error de identificación. Ese índice de aprendizaje es fácil de establecer como en [68] (por ejemplo seleccionamos* $\eta = 1$*), sin requerimiento y sin ninguna información a priori. El índice de aprendizaje variable en el tiempo puede encontrarse por medio de algún sistema adaptativo [75]. Pero ello necesita sin embargo modificaciones robustas para garantizar estabilidad de identificación. (6.28) es similar al resultado de [76], sin embargo la aproximación es diferente.*

6.4. Modelado con Sistemas Neuro Difusos Wavelet sin Conocimiento de Función de Membresía y Fijando Parámetros de Función Wavelet.

Cuando la planta se modela como una caja negra,y tanto como la premisa son desconocidas. Ahora el objetivo del modelado neuro difuso wavelet es encontrar el centro de $B_{1i} \cdots B_{mi}$,

así como las funciones de membresía $A_{1i} \cdots A_{ni}$, tal que la red neuro difusa wavelet (6.13) pueda seguir una planta no lineal (6.15).

Se utiliza la función de membresía Gausiana para identificar las reglas difusas, la cual está definida como

$$\mu_{A_{ji}} = \exp\left(-\frac{(x_j - c_{ji})^2}{\sigma_{ji}^2}\right) \tag{6.35}$$

q-esisma salida de la función difusa esta dada por

$$\widehat{y}_q = \sum_{i=1}^{l} w_{qi} \prod_{j=1}^{n} \exp\left(-\frac{(x_j - c_{ji})^2}{\sigma_{ji}^2}\right) / \left[\sum_{i=1}^{l} \prod_{j=1}^{n} \exp\left(-\frac{(x_j - c_{ji})^2}{\sigma_{ji}^2}\right)\right] \tag{6.36}$$

Definimos

$$z_i = \prod_{j=1}^{n} \exp\left(-\frac{(x_j - c_{ji})^2}{\sigma_{ji}^2}\right), \qquad a_q = \sum_{i=1}^{l} w_{qi} z_i, \qquad b = \sum_{i=1}^{l} z_i$$

Entonces

$$\widehat{y}_q = \frac{a_q}{b}$$

De forma similar a (6.26), la planta no lineal (6.15) puede representarse como:

$$y_q = \sum_{i=1}^{l} w_{qi}^* \prod_{j=1}^{n} \exp\left(-\frac{\left(x_j - c_{ji}^*\right)^2}{\sigma_{ji}^{*2}}\right) / \left[\sum_{i=1}^{l} \prod_{j=1}^{n} \exp\left(-\frac{\left(x_j - c_{ji}^*\right)^2}{\sigma_{ji}^{*2}}\right)\right] - \mu_q \tag{6.37}$$

donde w_{qi}^*, c_{ji}^* y σ_{ji}^{*2} son parámetros desconocidos que pueden minimizar la dinámica μ_q.

En el caso de tres variables independientes, una función suave f tiene una expansión de Taylor:

$$f(x_1, x_2, x_3) = \sum_{k=0}^{l-1} \frac{1}{k!} \left[\left(x_1 - x_1^0\right) \frac{\partial}{\partial x_1} + \left(x_2 - x_2^0\right) \frac{\partial}{\partial x_2} + \left(x_3 - x_3^0\right) \frac{\partial}{\partial x_3}\right]_0^k f + R_l$$

donde R_l es el residuo de la formula de Taylor. Si x_1, x_2, x_3 corresponde a w_{pi}^*, c_{ji}^* y σ_{ji}^{*2}, x_1^0, x_2^0, x_3^0 corresponde w_{pi}, c_{ji} y σ_{ji}^2,

$$\begin{aligned} y_q + \mu_q = \widehat{y}_q + \sum_{i=1}^{l} \left(w_{qi}^* - w_{qi}\right) z_i / b + \sum_{i=1}^{l} \sum_{j=1}^{n} \tfrac{\partial}{\partial c_{ji}} \left(\tfrac{a_q}{b}\right) \left(c_{ji}^* - c_{ji}\right) \\ + + R_{1q} \end{aligned} \tag{6.38}$$

donde R_{1q} es la aproximación de segundo orden de la serie de Taylor, $q = 1 \cdots m$. Usando la regla de la cadena, tenemos

$$\mu_{ij}(k+1) = \mu_{ij}(k) - \eta e(k) \sum \frac{y_j - y}{\sum \mu_j} \mu_j \frac{2(x_i - \mu)}{\sigma_{ij}^2}$$

En la matriz formada por,

$$y_q + \mu_q = \widehat{y}_q - Z(k) \widetilde{W}_q - D_{Zq} \overline{C}_k E - D_{Zq} \overline{B}_k E + R_{1q} \tag{6.39}$$

donde

$$Z(k) = [z_1/b \cdots z_l/b]^T, \qquad W_q = [w_{q1} \cdots w_{ql}], \qquad \widetilde{W}_q = W_q - W_q^*$$
$$D_{Zq} = \left[2z_1 \tfrac{w_{q1} - \widehat{y}_q}{b}, \cdots, 2z_l \tfrac{w_{ql} - \widehat{y}_q}{b}\right], \qquad E = [1, \cdots 1]^T$$
$$\overline{C}_k = \begin{bmatrix} \frac{x_1 - c_{11}}{\sigma_{11}^2}(c_{11} - c_{11}^*) & & \frac{x_n - c_{n1}}{\sigma_{n1}^2}(c_{n1} - c_{n1}^*) \\ & \ddots & \\ \frac{x_1 - c_{1l}}{\sigma_{1l}^2}(c_{1l} - c_{1l}^*) & & \frac{x_n - c_{nl}}{\sigma_{nl}^2}(c_{nl} - c_{nl}^*) \end{bmatrix}$$

En forma vectorial

$$e(k) = \widetilde{W}_k Z(k) + D_z(k) \overline{C}_k E + \zeta(k) \tag{6.40}$$

donde $e(k) = [e_1 \cdots e_m]^T$,

$$\widetilde{W}_k = \begin{bmatrix} w_{11} - w_{11}^* & & w_{m1} - w_{m1}^* \\ & \ddots & \\ w_{1l} - w_{1l}^* & & w_{ml} - w_{ml}^* \end{bmatrix}, D_z(k) = \begin{bmatrix} 2z_1 \frac{w_{11} - \widehat{y}_1}{b} & & 2z_l \frac{w_{1l} - \widehat{y}_1}{b} \\ & \ddots & \\ 2z_1 \frac{w_{m1} - \widehat{y}_m}{b} & & 2z_l \frac{w_{ml} - \widehat{y}_m}{b} \end{bmatrix}$$

$\zeta(k) = \mu - R_1$, $\mu = [\mu_1 \cdots \mu_m]^T$, $R_1 = [R_{11} \cdots R_{1m}]^T$.

Dado que la función Gausiana ϕ es acotada y la planta es BIBO, μ y R_1 en (6.37) y (6.38) son acotadas. Entonces $\zeta(k)$ en (6.40) es acotado. El siguiente teorema establece un algoritmo estable para redes difusas wavelet, basadas en el tipo difuso Mamdani.

Teorema 6.2 *Se se una red neuronal difusa wavelet del tipo Mamdani (6.36) para identificar una planta no lineal (6.15), el siguiente algoritmo hace que el error de identificación $e(k)$ sea acotado*

$$\begin{aligned} W_{k+1} &= W_k - \eta_k e(k) Z(k)^T \\ \mu_{ij}(k+1) &= \mu_{ij}(k) - \eta e(k) \sum \tfrac{y_j - y}{\sum \mu_j} \mu_j \tfrac{2(x_i - \mu)}{\sigma_{ij}^2} \end{aligned} \tag{6.41}$$

donde $\eta_k = \dfrac{\eta}{1+\|Z\|^2+2\|D_z\|^2}$, $0<\eta\leq 1$. *El promedio de el error de identificación satisface*

$$J=\lim_{T\to\infty}\sup\frac{1}{T}\sum_{k=1}^{T}\|e(k)\|^2\leq\frac{\eta}{\pi}\overline{\zeta} \tag{6.42}$$

donde $\pi=\dfrac{\eta}{(1+\kappa)^2}>0$, $\kappa=\max_k\left(\|Z\|^2+2\|D_z\|^2\right)$, $\overline{\zeta}=\max_k\left[\|\zeta(k)\|^2\right]$

Demostración. Se define $\widetilde{c}_{ji}(k)=c_{ji}(k)-c^*_{ji}$, $\widetilde{b}_{ji}(k)=\sigma_{ji}(k)-\sigma^*(k)$, el elemento de $\widetilde{C}_k$ se expresa como $\widetilde{c}_{ji}(k)=\left[\widetilde{C}_k\right]$. Entonces

$$\left[\widetilde{C}_{k+1}\right]=\left[\widetilde{C}_k\right]-2\eta_k z_i\frac{w_{qi}-\widehat{y}_q}{b}\frac{x_j-c_{ji}}{\sigma^2_{ji}}(\widehat{y}_q-y_q)$$

Seleccionamos el escapar positivo definido L_k como

$$L_k=\left\|\widetilde{W}_k\right\|^2+\left\|\widetilde{C}_k\right\|^2+\left\|\widetilde{B}_k\right\|^2 \tag{6.43}$$

De la ley (6.41)

$$\widetilde{W}_{k+1}=\widetilde{W}_k-\eta_k e(k)\,Z(k)^T$$

Utilizando (6.40) se tiene

$$\begin{aligned}
\Delta L_k &= \left\|\widetilde{W}_k-\eta_k e(k)\,Z(k)^T\right\|^2+\left\|\widetilde{C}_k-\left[2\eta_k z_i\tfrac{w_{qi}-\widehat{y}_q}{b}\tfrac{x_j-c_{ji}}{\sigma^2_{ji}}(\widehat{y}_q-y_q)\right]\right\|^2\\
&+\left\|\widetilde{B}_k-\left[2\eta_k z_i\tfrac{w_{qi}-\widehat{y}_q}{b}\tfrac{(x_j-c_{ji})^2}{\sigma^3_{ji}}(\widehat{y}_q-y_q)\right]\right\|^2-\left\|\widetilde{W}_k\right\|^2-\left\|\widetilde{C}_k\right\|^2\\
&=\eta_k^2\|e(k)\|^2\left(\left\|Z(k)^T\right\|^2+2\left\|D_z^T\right\|^2\right)-2\eta_k\|e(k)\|\left\|\widetilde{W}_k Z(k)^T+D_z^T\overline{C}_k E+E\right\|\\
&=\eta_k^2\|e(k)\|^2\left(\|Z\|^2+2\|D_z\|^2\right)-2\eta_k\|e(k)\,[e(k)-\zeta(k)]\|\\
&\leq-\eta_k\|e(k)\|^2\left[1-\eta_k\left(\|Z\|^2+2\|D_z\|^2\right)\right]+\eta\|\zeta(k)\|^2\\
&\leq-\pi\|e(k)\|^2+\eta\|\zeta(k)\|^2
\end{aligned} \tag{6.44}$$

donde π es definida como

$$\pi=\frac{\eta}{\left[1+\max_k\left(\|Z\|^2+2\|D_z\|^2\right)\right]^2}$$

Dado que

$$n\left[\text{mín}\left(\widetilde{w}_i^2\right)+\text{mín}\left(\widetilde{c}_{ji}^2\right)+\text{mín}\left(\widetilde{b}_{ji}^2\right)\right]\leq L_k\leq n\left[\text{máx}\left(\widetilde{w}_i^2\right)+\text{máx}\left(\widetilde{c}_{ji}^2\right)+\text{máx}\left(\widetilde{b}_{ji}^2\right)\right]$$

donde $n\left[\text{mín}\left(\widetilde{w}_i^2\right)+\text{mín}\left(\widetilde{c}_{ji}^2\right)+\text{mín}\left(\widetilde{b}_{ji}^2\right)\right]$ y $n\left[\text{máx}\left(\widetilde{w}_i^2\right)+\text{máx}\left(\widetilde{c}_{ji}^2\right)+\text{máx}\left(\widetilde{b}_{ji}^2\right)\right]$ son $\mathcal{K}_\infty$-función, y $\pi\left\|e\left(k\right)\right\|^2$ es una $\mathcal{K}_\infty$-función, $\eta\left\|\zeta\left(k\right)\right\|^2$ es una $\mathcal{K}$-función. Entonces L_k es ISS-Lyapunov función. Del T*eorema 6.2*, la dinámica del error de identificación es estable de la entrada a los estados. De (6.40) y (6.43) sabemos que V_k es función de $e\left(k\right)$ y $\zeta\left(k\right)$. Debido a que la "ENTRADA" $\zeta\left(k\right)$ es acotada y la dinámica es ISS, el "ESTADO" $e\left(k\right)$ es acotado.

(6.44) se puede re escribir como

$$\Delta L_k\leq-\pi\left\|e\left(k\right)\right\|^2+\eta\left\|\zeta\left(k\right)\right\|^2\leq\pi\left\|e\left(k\right)\right\|^2+\eta\overline{\zeta}\tag{6.45}$$

Sumarizando (6.45) de 1 a T, y usando $L_T>0$ y L_1 es una constante, se obtiene

$$\begin{gathered}L_T-L_1\leq-\pi\sum_{k=1}^{T}\left\|e\left(k\right)\right\|^2+T\eta\overline{\zeta}\\ \pi\sum_{k=1}^{T}\left\|e\left(k\right)\right\|^2\leq L_1-L_T+T\eta\overline{\zeta}\leq L_1+T\eta\overline{\zeta}\end{gathered}$$

(6.42) es establecida. ■

6.5. Simulaciones

El sistema dinámico de la función logística esta dado por:

$$x(k+1)=\alpha*x1(k)*(1-x1(k));$$

donde dependiendo el valor del parámetro α puede tener distintos comportamientos, hacemos la identificación de ese sistema con distintos valores para el parámetro α, primero lo hacemos con $\alpha=0{,}9$ y se hace la identificación mostrando el impacto de establecer un correcto número de reglas.

En la Figura 6.5, se hace la aproximación a la serie de tiempo utilizando 4 reglas con 16 funciones de membresía en la última regla.

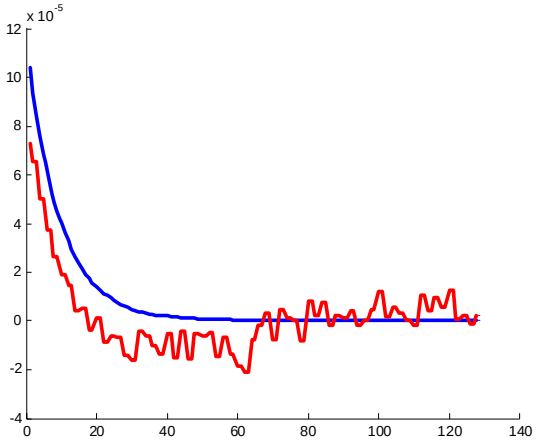


Figura 6.5: 4 reglas con 16 funciones de membresía en la última regla

En la Figura 6.6 se hace la identificación utilizando solo dos reglas, pero con un mayor número de funciones de membresía, con 64 funciones de membresía en la última regla.

El la Figura 6.7 se utilizan siete reglas, con 64 funciones de membresía en la regla número 7

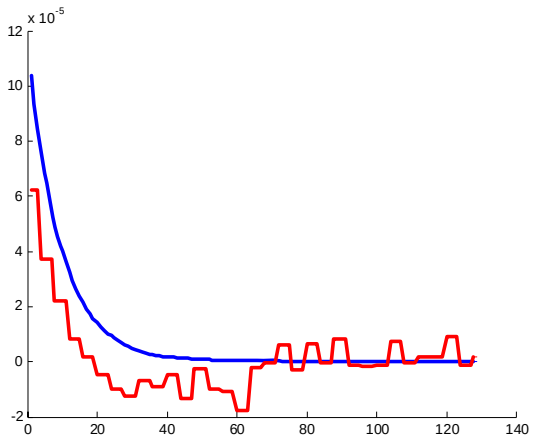


Figura 6.6: 64 funciones de membresía en la última regla

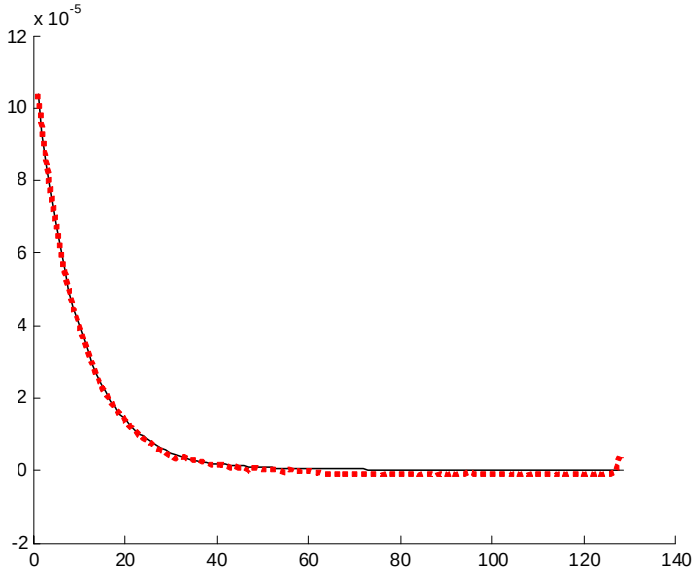


Figura 6.7: 64 funciones de membresía en la regla número 7

Capítulo 7

Conclusiones y Trabajo Futuro.

7.1. Conclusiones

Fueron expuestas las bondades de las funciones wavelets para ser usadas para aproximar funciones, además de que presentan mejores propiedades que el análisis de Fourier, su aplicación en redes neuronales fue adoptada hace varios años, sin embargo se ha desestimado el uso de la función wavelet Haar, que por no ser continua se puede creer que no puede aproximar funciones continuas, aun cuando en otras áreas técnicas han sido utilizadas, como lo es por ejemplo en el tratamiento de imágenes. El uso de funciones wavelets en redes neuronales ha encontrado el problema de determinar el número de neuronas necesarias para realizar una aproximación, el uso de funciones wavelets continuas como Morlet o Sombrero Mexicano, no presenta el problema de determinar el valor de resolución además de que el número de traslaciones a lo largo de la función a aproximar puede ser infinita por las mismas características que le identifican como una función continua.

En el algoritmo rápido para obtener la trasformada wavelet, está bien definido el número de traslaciones que debe hacer una función wavelet Haar para hacer una aproximación adecuada, bajo las consideraciones de la transformada rápida wavelet podemos conocer a priori el número de neuronas necesarias para hacer una aproximación a series de tiempo, una importante propiedad de este tipo de red, consiste en que el número de neuronas en la red de-

terminan la calidad de la aproximación, bajo las consideraciones de la transformada wavelet podemos conocer a priori el número de neuronas necesarias para hacer una aproximación exacta.

La función wavelet Haar se mostró como una buena opción para utilizarse en redes neuronales, siendo sencilla de implementar y de analizar, logrando aproximaciones donde el nivel de error puede ser ajustarlo a un valor deseado agregando mas neuronas a la red. El algoritmo de aprendizaje y de neuro identificación es estable, y no afecta en la estabilidad el hecho de usar una función no continua como lo es la función Haar.

El implementar redes wavelets con sistemas difusos nos permite aproximar a una serie de tiempo y ajustar el error de aproximación determinando el número de reglas que se utilicen, se propusieron tres algoritmos, en el tercer algoritmo no es necesario fijar una resolución de aproximación, lo que nos ofrece una herramienta sumamente útil.

7.2. Trabajo Futuro.

- Realizar Predicción en series de tiempo utilizando redes neuronales wavelets Haar.
- Implementar la red neuronal wavelet como un clasificador.

Bibliografía

[1] A. Haar, "Zur Theorie der Orthogonalen Funktionen-Systeme". Math. Ann. 69 (1910), 331-371.

[2] O. Christensen, K. L. Christensen, *Approximation Theory, From Taylor Polynomials to Wavelets*, Ed. Birkhäuser, 2004.

[3] T. Struts, *Bilddatenkompression, Grundlagen, Codierung, Wavelets MPG, JPG, H.264.* Vieweg 2005.

[4] J. M. Keiser, "Wavelet based approach to numerical solution of nonlinear partial differential equations and nonlinear waves in fully discrete dynamical systems", PhD Thesis, University of Colorado, 1995.

[5] I. Daubechies, *Ten Lectures on Wavelets,* volume 61 of CBMS-NFS Regional Conference series in Applied Mathematics, SIAM, Philadelphia 1992.

[6] S. Jaffard, Y. Meyer, R. D. Ryan, *Wavelets, Tools for Science & Technology*, SIAM, 2001.

[7] E. Hernandez, G. Weiss, *A First Course on Wavelets*, ser. Studies in Advanced Mathematics, CRC Press 1996.

[8] Y. Nievergelt, *Wavelets Made Easy*, Birkhäuser, 1999.

[9] H. Stark, *Wavelets and Signal Processing*, Springer, 2005.

[10] J. Benedetto, M. W. Frazier, *Wavelets Mathematics and Applications*, CRC Press, 1994.

[11] G. Beylkin, R. R. Coifman, and V. Rokhlin. *Wavelets and Their Applications*, Jones and Bartlett, 1992.

[12] N. Sureshbabu, J. A.Farrell, "Wavelet-Based System Identification for Nonlinear Control", *IEEE Trans. on Automatic Control*, Vol. 44, No 2, pp. 412-417,. Feb 1999.

[13] Yang Li, Hua-liang Wei, and S. A. Billings, "Identification of Time-Varying Systems Using Multi-Wavelet Basis Functions,"*IEEE Trans. on Control Systems Technoloy*, Vol. 19, No. 3, pp 656-663, May 2011,

[14] H. Ye, g. Wang, S. X. Ding, .A New Parity Space Approach for Feult Detection Based on Stationary Wavelet Transform", *IEEE Trans. on Automatic Control*, Vol 49 no. 2, pp. 281-287, Feb 2004.

[15] K. R. Davidson, A.P. Donsig, *Real Analysis and Applications , Theory in Practice*, ser. Undergraduate Text in Mathematics, Springer, 2000.

[16] Wen Yu, Xao Li. "Discrete-time neuro identification without robust modification", *IEEE Procedings Control Theory Applications*, vol 150, no. 3, May 2003.

[17] Wen Yu, "Nonlinear system identification using discrete-time recurrent neural networks with stable learning algorithms", *Information Sciencies,* vol 158, pp 131-147, 2004.

[18] G.Cybenko, .Approximation by Superposition of Sigmoidal Activation Function", *Math.Control, Sig Syst*, Vol.2, 303-314, 1989.

[19] Z.P.Jiang and Y.Wang, "Input-to-State Stability for Discrete-Time Nonlinear Systems", *Automatica*, Vol.37, No.2, 857-869, 2001.

[20] S. Haykin. *Neural Networks, A Comprehensive Foundation,*2a Edición, Prentice Hall, 1999.

[21] A. S. Poznyak, E.N. Sanchez, W.Yu,.*Differential Neural Networks for Robust Nonlinear Control, Identification, State Estimation and Trajectory Tracking*, World Scientific Publishing, 2001.

[22] A.S. Poznyak, W.Yu, E.N. Sanchez and Jose P. Perez, Stability Analysis of Dynamic Neural Control, *Expert System with Applications,* Vol. 14, No.1, 227-236, 1998.

[23] G.A.Rovithakis and M.A.Christodoulou, .Adaptive Control of Unknown Plants Using Dynamical Neural Networks", *IEEE Trans. on Syst., Man and Cybern.,* Vol. 24, 400-412, 1994.

[24] J. Park and I. W. Sandberg, "Universal approximation using radial-basisfunction networks", *Neural Computa.*, vol. 3, pp. 246–257, 1991.

[25] J. Zhang, G.G. Walter, Y. Miao anda N. Lee. "Wavelet Neural Networks for Function Learning", *IEEE Trans. on Signal Processing*, Vol 43. No. 6, pp. 1485-1497, June 1995.

[26] Ülo Lepik, Enn Tamme, .Application of the Haar Wavelets for Solution of Linear, Integral Equations", *Proc Dynamical Systems and Applications 2004*, pp. 494—507.

[27] Ülo Lepik, .Application of the Haar wavelet transform to solving integral and differential equations", *Proc. Estonian Acad. Sci. Phys. Math.*, 2007, 56, 1, 28–46.

[28] Qinghua Zhang, Albert Benveniste. "Wavelet Networks", *IEEE Transactions on Neural Networks*, Vol 3, No 6, November 1992.

[29] Qinghua Zhang. Üsing Wavelet Networks in Nonparametric Estimation", *IEEE Transactions on Neural Networks*, Vol 8, No 2, March 1997.

[30] Wei Sun, Yaonan Wang, Jianxu Mao. Üsing Wavelet Network for Identifying the Model of Robot Manipulator", *Procedings of the 4 Worl Congres on Intelligent Control and Automation*, June 10-14 2002, Shanghai China.

[31] (2010) Banco de Mexico, website, [online] Disponible: http://www.banxico.gob.mx/

[32] Schram Gerard, Tempel van der, Krijsman A.J. "Structure Determination of Radial Basis Function Network, With An Application to Flight Control", Control Laboratory, Deparment of Electrical Engineering, Delft University of Technology, Delft, The Netherlands, August (1996).

[33] Sherstinsky Alex and Rosalind Picard, "The Efficiency Of The Orthogonal Least Squares Training Method For Radial Function Networks", Departament of Electrical Engineering and Computer Science, MIT Media Laboratory, October (1994).

[34] Y. Miche, P. Bias, C. Jutten, O. Simula, A. Lendasse, "A methodology for Building Regression Models using Extreme Learning Machine", *Procc 16th European Symposium in Artificial Neural Networks*, pp. 247-252, Bruges, Belgium, 2008.

[35] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, "Least Angle Regression", *Annal of Statistics*, vol. 32, pp. 407-499, (2004).

[36] Stephen A. Billings and Hua-Liang Wei, "New Class of Wavelet Networks for Nonlinear System Identification", *IEEE Trans. on Neural Netorks,* Vol. 16, No.. 4, Jully 2005.

[37] Zhijun Zhang and ChaoZhao, .A Fast Learning Algorithm for Wavelet Network and its Application in Control", *on proc IEEE International Conference on Control and Automationc 2007,* pp 1403-1407.

[38] Jian-Xin Xu and Ying Tan, "Nonlinear Adaptive Wavelet Control Using Constructive Wavelet Networks", *IEEE Trans. on Neural Netorks*, VOL. 18, NO. 1,pp 115-127 JANUARY 2007.

[39] Ning Jin and Derong Liu, "Wavelet Basis Function Neural Networks for Sequential Learning", *IEEE Trans. on Neural Netorks*, Vol. 19, No 3, March 2008.

[40] K.S.Narendra and K.Parthasarathy, "Identification and Control of Dynamical Systems Using Neural Networks", *IEEE Trans. on Neural Networks,* Vol. 1, 4-27, 1990.

[41] Lennart Ljung, *System Identification, Theory for the User,* Prentice Hall, 1999.

[42] Lennart Ljunt, *MatLab User's Guide: Systems Identification Toolbox*, 1998.

[43] C.L Giles, G.M. Khun, and R.J. Williams, Eds., Special Issue on Dynamic Recurrent Neural Networks, IEEE Trans, on Neural Networks, vol. 5, March 1994

[44] Wlodzimierz Greblicki and Miroslaw Pawlak, *Non Parametric System Identification*, Ed.Cambridge University Press, 2008.

[45] Miroslaw Pawlak and Zygmunt Hasiewicz, "Nonlinear System Identification by the Haar Multiresolution Analysis,"*IEEE TRans. on Circuits and Systems I.* Vol 45, No 9, Sep 1998.

[46] Tomaso Poggio, Federico Girosi, "Networks for Approximation and Learning", *Proc of IEEE,* Vol. 78, No. 9, pp 1481-1497, 1990.

[47] Landau, I.D., *Adaptive Control: The Model Reference Approach,* Marcel Dekker, New York, 1979.

[48] Omidvar Omid, Elliot David L., *Neural Systems for Control,* Academic Press, 1997.

[49] M. Vetterli and C. Herley, "Wavelets and Filter Banks: Theory and Design,"*IEEE Trans. on Signal Processing,* Vol. 40, 1992, pp. 2207-2232.

[50] J. Bradley, C. Brislawn, and T. Hopper, "The FBI Wavelets/Scalar Quantization Standar for Gray-scale Fingerprint Image Compression,"Techh. Report LA-UR-93-1659, Los Alamos Nat'Lab, Los Alamos, M.N 1993.

[51] M. Bhatia,W. C. Karl and A. S. Willsky, Fellow, .A Wavelet-Based Method for Multiscale Tomographic Reconstruction,"*IEEE Trans. on Medical Imaging,* Vol. 15, No. 1, 1996, pp. 92-101.

[52] Douglas A. Newandee and Stanley S. Reisman, "Wavelet Representation Comparison for Heart Rate Variability Analysis"*on Proc. Bioengineering Conference 2003,* pp 112-113.

[53] Fatma H. Elfouly, Mohamed I. Mahmoud, Moawad I. M. Dessouky, and Salah Deyab, Çomparison between Haar and Daubechies Wavelet Transformations on FPGA Technology," *World Academy of Science, Engineering and Technology*, vol 37, 2008, pp 417-422.

[54] Daniel W. C. Ho, Ping-An Zhang and Jinhua Xu, "Fuzzy Wavelet Networks for Function Learning,"*IEEE Trans. on Fuzzy Systems,* Vol 9, No 1, pp. 200-211, Feb. 2001.

[55] Przemsyslaw Sliwinski, Jerzy Rozenblit, Michael W. Marcellin and Ryszard Klempous, "Wavelet Amendment of Polynomial Models in Hammerstein Systems Identification,"*IEEE Trans. on Automatic Control,* vol 54, No. 4, April 2009.

[56] Juan Jose Cordova and Wen Yu, Stable Fourier Neural Networks with Application to Modeling Lettuce Growth, 2009 International Joint Conference on Neural Networks, IJCNN'09, Atlanta, USA, 591-596, 2009

[57] Silvescu Adrian, (1999). Fourier neural networks. IJCNN 1999. International Joint Conference on Neural Networks, vol 1:488-491.

[58] Leonardo M. Reyneri, Ünification of Neural and Wavelet Networks and Fuzzy Systems", *IEEE Trans. on Neural Networks,* Vol 10, No 4, July 1999.

[59] Sevcan Yilmaz and Yusuf Oysal, "Fuzzy Wavelet Neural Network Model for Prediction and Identification of Dynamical Systems", *IEEE Trans on Neural Networks*, Vol 21, No 10 October 2010.

[60] Cheng-Jia Lin and Cheng-Chung Chin, "Prediction and Identification Using Wavelet-Based Recurrent Fuzzy Neural Networks", I*EEE Trans. on Systems, Man and Cybernetics*, vol 34, No. 5, october 2004.

[61] Rahib Hidayat Abiyev, "Fuzzy Wavelet Neural Networks for Identification and Control of Dynamics Plants- A Novel Structure and a Comparative Study", *IEEE Trans. on Industrial Electronics*, vol 55, No. 8, august 2008.

[62] Li-Xin Wang, *A course in Fuzzy Systems and Control*, Prentice Hall, 1997.

[63] J. S. R. Jang, C. T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*. Englewood Cliffs, NJ:Prentice-Hall, 1997.

[64] Hongxing Li, C.L. Philip Chen, Han-Pang Huang, *Fuzzy Neural Intelligent Systems, Mathematical Foundation and the Applications in Engineering*, CRC Press LLC, 2001

[65] S. Horikawa, T. Furuhashi, and Y. Uchikawa, .ºn Fuzzy modeling using fuzzy neural networks with the back propagation algorithm", *IEEE Trans. on Neural Networks.*, vol. 3, no. 5, pp. 801-806, sep 1992.

[66] J.-S. R. Jang, .ªNFIS: Adaptive-network based fuzzy inference systems,"*IEEE Trans. on Syst. Man. Cybern.*, vol 23, no. 3. pp. 665-685, May-Jun. 1993.

[67] J. Kim and N. Kasabov, "HyFIS: Adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems,"*Neural Networks.*, vol. 12, no. 9, pp 1301-1319, nov 1999.

[68] L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples,"IEEE Trans. Syst., Man, Cybern., vol. 22, pp. 1414-1427, Nov/Dec. 1992.

[69] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control,"IEEE Trans. Syst., Man, Cybern., vol. SMC-15, pp. 116-132, 1985.

[70] E.B. Kosmatopoulos, M.M. Polycarpou, M.A. Christodoulou, P.A. Ioannou, High-order neural network structures for identification of dynamical systems, IEEE Trans. Neural Networks 6 (2) (1995) 431–442.

[71] R.K. Boel, M.R.James and I.R.Petersen, Robustness and Risk-Sensitive Filtering, IEEE Trans. Automatic Control, Vol.47, No.3, 451-462, 2002.

[72] F.N.Chowdhury, A new approach to real-time training of dynamic neural networks, International Journal of Adaptive Control and Signal Processing, 509-521, 2003.

[73] G. V. Puskorius and L. A. Feldkamp, Neurocontrol of Nonlinear Dynamical Systems with Kalman Filter Trained Recurrent Networks,IEEE Transactions on Neural Networks, Vol.5, No.2, 279-297, March 1994.

[74] Guanrong Chen, Trung Tat Pham, *Introduction to fuzzy sets, fuzzy logic, and fuzzy control systems*, CRC Press, 2001.

[75] P. A. Ioannou, Jing Sun, *Robust Adaptive Contro,* PTR Prentice-Hall, 1996

[76] Danilo P Mandic, Jonathon.Chambers, .°n robust stability of time-variant discrete-time nonlinear systems with bounded parameter perturbations,"IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 47(2), pp. 185 - 188.

[77] George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, *Time Series Analysis, Forescasting and Control*, Prentice-Hall, 1994

[78] Editores Peter S.C. Heuberger, Paul M.J. Van den Hof and Bo Wahlberg, *Modelling and Identification with Rational Ortogonal Basis Function*, Springer, 2005

[79] Djomangan Adama Ouattara, Modélisation de l'infection par le VIH, identification et aide au diagnostic, *Doctoral Thesis, l'École Centrale de Nantes et l'Université de Nantes*, 22 September 2006.

[80] I. Salgado and I. Chairez, Discrete Time Recurrent Neural Network Observer, Proc. of International Joint Conference on Neural Networks 2009, 2764-2770.