Wavelets, Bancos de Filtros y Análisis de Señales Biomédicas

Por

María Inés Pisarello

TESIS PARA OPTAR AL GRADO ACADÉMICO DE DOCTOR DE LA UNIVERSIDAD NACIONAL DEL NORDESTE

ESPECIALIDAD: MATEMÁTICA

Facultad de Ciencias Exactas y Naturales y Agrimensura

2013

Agradecimientos

En este momento tan esperado y tan especial en mi vida, quiero mencionar a todas las personas que de algún u otro modo colaboraron durante el desarrollo de esta tesis.

Quiero agradecer en primer lugar a la Facultad de Ciencias Exactas y Naturales y Agrimensura, donde se desarrolló mi carrera de grado y ahora de posgrado. A la secretaría de Desarrollo y Posgrado de la FACENA, por sus gestiones y su acompañamiento. Al Departamento de Ingeniería de la FACENA donde desarrollo mi trabajo profesional.

A mi director de tesis, el Dr. Carlos D'Attellis que colaboró conmigo siempre con buena predisposición, con buenos consejos y brindándome toda su experiencia y sapiencia sin mezquindad alguna. Por sus correcciones y anotaciones al margen que fueron de gran ayuda.

Muy especialmente quiero agradecer al Dr. Jorge Monzón, mi maestro, que me guía, me sugiere, me acompaña y se involucra en cada etapa de mi formación académica, aportando lo mejor de sí para que alcance mis metas.

A mis compañeros siempre atentos a mis consultas, dispuestos a extender una mano amiga.

Quiero agradecer además a mis padres, que me enseñaron con su propio testimonio el deber ante el compromiso asumido, la fortaleza para enfrentar situaciones adversas y la confianza en Dios, para depositar en Sus manos mis momentos de flaqueza.

Por último, dedico este trabajo a mi esposo Sebastián y a mis hijos, Gerónimo, Juan Pablo y María Julieta, por acompañarme siempre, por sus abrazos, risas y besos después de cada jornada de trabajo. Por hacer mi vida completa.

Índice

Parte I

Parte I.

Marco Teórico

1. Introducción

1.1. Descripción de la Tesis

En esta tesis de doctorado se substancian desarrollos científicos en el amplio campo del procesamiento de señales biomédicas, en particular el análisis de la señal de electrocardiograma (ECG). La mayoría de los procesos que involucran señales biológicas, acarrean una gran dificultad para manejar los datos, debido al ruido interferente que se produce durante la captura de la señal. Otro aspecto complicado en el manejo de datos biológicos es la extracción de puntos característicos de la señal, debido, principalmente a las diferencias entre una señal y otra inclusive cuando analizamos datos de un mismo paciente. Los algoritmos para la detección de parámetros de estas señales deben incorporar aspectos generales y conllevan mucho proceso para lograr la generalidad buscada. La clasificación de latidos es también una tarea que puede resultar complicada, si la detección de los parámetros a clasificar no es adecuada. El ruido contaminante en las señales también interfiere en las tareas de detección y clasificación.

Por lo expuesto, planteamos estos tres desarrollos: (i) eliminación de ruido interferentes, de distintas fuentes y niveles de ruido, (ii) detección de parámetros característicos de la señal, como el complejo QRS y sus características y la onda P y por último, con estos procesos integrados, (iii) clasificación de latidos ventriculares prematuros. La importancia de estos procesos radica en la posibilidad cierta de aplicación en el ambiente clínico y hospitalario. Los procesos son realizados con señales reales, de pacientes reales e involucran algoritmos matemáticos y computacionales para la obtención de resultados. La buena performance de los procesos aquí desarrollados nos predispone a continuar con la investigación en esta rama de la Ingeniería Biomédica y posiciona a estos procesos como soporte para la toma de decisiones del especialista médico.

Los tópicos desarrollados en la tesis, están relacionados entre sí, de manera que unos sirven de pre-proceso para otros, o bien post-proceso, como es el caso de los Algoritmos Genéticos para un ajuste de pesos de las Redes Neuronales. Tanto las

técnicas de Análisis de Componentes Principales como El Análisis de Componentes Independientes permiten obtener datos de la señal de ECG que luego serán útiles en las Redes Neuronales, que son las encargadas de clasificar los latidos.

1.2. Aspectos Diferenciadores de la Tesis

A lo largo de esta tesis, presentamos distintas herramientas para el proceso de señales biomédicas, en particular, la señal de electrocardiogramas. Las técnicas de procesamiento van desde la eliminación de ruido presente en casi todos los procesos de captura de la señal, hasta la clasificación de arritmias ventriculares, capaces de ocasionar daño en la salud de los pacientes que la padecen. El interés particular de la tesis radica en que todos los procesos realizados, incorporan alguna novedad que no ha sido implementada o diseñada con anterioridad. Así, por ejemplo, en la implementación de Bancos de Filtros, hemos realizado tres procesos diferentes en paralelo, con la incorporación de algoritmos ya conocidos, como el filtro promedio para la eliminación de ruido gaussiano, pero con la innovación de su aplicación a una sub-banda de frecuencia. Lo mismo sucede con el algoritmo de detección del QRS y la posterior clasificación de arritmias. Los resultados de estos procesos fueron oportunamente publicados en:

– **Pisarello MI**, Monzón JE, 2002. "Utilización de Bancos de Filtros en el Análisis Digital del ECG", *Revista Argentina de Bioingeniería*, (ISBN 0329-5257). Vol 8, N°2.

> En este trabajo diseñamos un Banco de Filtros ortogonal, de fase lineal y con la característica de la reconstrucción perfecta. En las cuatro sub-bandas de frecuencia resultantes realizamos tres procesos en paralelo. Los resultados demuestran que los Bancos de Filtros así diseñados representan una alternativa viable para la eliminación de ruido, detección de parámetros característicos y clasificación de latidos ectópicos prematuros en la señal de ECG.

Una herramienta que despierta especial interés en los diseñadores de procesamiento de señales es la utilización del Análisis de Componentes Principales para la caracterización de patrones de una señal o bien para la reducción de las dimensiones de una matriz de datos de entrada. Esta última característica fue en la que nos basamos para la reducción de la matriz de datos del complejo QRS, que fuera luego una de las entradas a nuestra red neuronal. Con este método logramos reducir a un tercio del tamaño la matriz original de entrada conservando más del 95% de la información total. Como resultado anexo, la red neuronal diseñada logró un entrenamiento más rápido y efectivo. Nuestros resultados en esta técnica fueron ya publicados en:

- **Pisarello MI**, Álvarez Picaza C, Veglia JI, 2005. "Compresión de Bio-Datos – Utilización del MATLAB aplicado al Análisis de Componentes Principales". *Primer Congreso Latinoamericano de Ingeniería Prospectiva – Exposición Internacional de Ingeniería.*

- **Pisarello MI**, Álvarez Picaza C, Monzón JE, 2006. "Análisis de Componentes Principales para la Compresión del ECG" *5ta Conferencia Iberoamericana en Sistemas, Cibernética e Informática: CISCI*. *CISCI*. ISBN: 980-6560-89-2. Vol I – Pp 48.

> Los trabajos mencionados demuestran que es posible reducir una matriz de datos de 30 variables a una matriz de 5 variables en las que se concentra casi el 100% de la información. Con esta herramienta se eliminó la redundancia de datos para agilizar los tiempos computacionales, lo que constituye un objetivo primordial en el procesamiento de información. Estos nuevos datos, que están además estandarizados, serán las nuevas entradas a nuestras redes neuronales.

El análisis de componentes independientes (ICA) brinda una herramienta novedosa para la separación ciega de fuentes (BSS), lo que nos permite obtener señales libres de ruido, sin la necesidad de conocer las características de la señal ruidosa, con solo tener la seguridad de que la señal de interés y la interferente son estadísticamente independientes. Propusimos en esta tesis un novedoso algoritmo basado en la ecuación

matricial de Lyapunov, con la cual pudimos encontrar la matriz de mezcla, desconocida hasta entonces, y así obtener dos señales independientes entre sí. Los algoritmos diseñados para realizar este proceso involucran tareas recursivas y filtrado de la señal. Proponemos aquí algo diferente, que utiliza las herramientas de Control Moderno, verdadera matemática aplicada, y la extracción de Componentes Principales para lograr el objetivo. Nuestros resultados fueron publicados en:

- Monzón JE, **Pisarello MI**, Álvarez Picaza C, 2007. "Blind source separation of electrocardiographic signals using system stability criteria", *Proceedings of the 29th Annual International Conference IEEE-EMBS.* ISBN: 1-4244-0788-5. Pp 3.493.

- **Pisarello MI**, Álvarez Picaza C, Monzón JE, 2007. "Eliminación de ruido en bioseñales utilizando la ecuación algebraica de Lyapunov", *IV Congreso Latinoamericano de Ingeniería Biomédica, CLAIB.* ISBN: 979-980-7062-12-1. Vol 18 – Pp 78.

- **Pisarello MI**, Álvarez Picaza C, Monzón JE, 2008. "Separación de Frecuencias No Deseadas en la señal cardíaca utilizando ICA", *XII Jornadas Internacionales de Ingeniería Clínica y Tecnología Médica.* ISBN: 978-950-698-212-6.

> Los resultados alcanzados en estos trabajos permiten identificar las señales no deseadas y sus componentes de frecuencias y recuperar aquellas de la señal de interés.
>
> Los datos obtenidos resultan alentadores pues revelan la posibilidad de utilizar la ecuación algebraica de Lyapunov como alternativa para la separación ciega de fuentes, no sólo en el espacio del tiempo, sino también en el espacio de las frecuencias, ampliando las perspectivas de eliminación de los ruidos que perturban a la señal de interés.

Para clasificar latidos ventriculares prematuros, además de los Bancos de Filtros Digitales ya mencionados, utilizamos Redes Neuronales Artificiales. Optamos por la topología Perceptrón Multicapa (MLP), con el entrenamiento de retroalimentación y aprendizaje supervisado. El estudio y la aplicación de las redes neuronales estuvieron siempre regidos por la heurística. En nuestra presentación, después de varias

experiencias, definimos la estructura de la red y las entradas que involucran características temporales, morfológicas y frecuenciales de los latidos cardíacos. La novedosa incorporación de los componentes principales a la matriz de datos, y que luego formará parte estructural de la red, nos permitió obtener resultados más óptimos en la clasificación, con la ventaja de entrenamientos más rápidos, debido a la estandarización de los datos y de la conveniente reducción de la matriz de entrada. Diseñamos también una red neuronal con reglas de aprendizaje difusas, una técnica que tiene una amplia aplicación no solo en desarrollos científicos sino también en la industria. Estos híbridos de computación inteligente, permiten el manejo de decisiones que se hallan en una delgada zona de incertidumbre, donde la lingüística cobra interés y los límites no tienen bordes definidos. Demostraremos en esta tesis la habilidad de estas redes para tomar decisiones y manejar datos biológicos. Los resultados obtenidos fueron ya publicados en:

- Monzón JE, **Pisarello MI**, 2005. "Cardiac Beat Classification using a Fuzzy Inference System" Proceedings of the 27th Annual International Conference IEEE-EMBS. (ISBN 0-7803-8741-4), Pp. 5582-5584.

  El sistema allí diseñado identifica contracciones ventriculares prematuras con un acierto razonable, lo que se compara favorablemente a otros métodos reportados en la literatura, que abordan el problema desde un enfoque de sistema expertos híbridos. La tasa de clasificación basada únicamente en características temporales de la señal de electrocardiogramas es moderada.

- Monzón JE, **Pisarello MI**, 2005. "Evaluación del ECG con sistemas neuronales nítidos y difusos". XV Congreso de Bioingeniería. IV Jornadas de Ingeniería Clínica. SABI. (ISBN 950-698-156-6). 2005.

  En este trabajo, logramos clasificar latidos ectópicos prematuros de sujetos cuyos registros de ECG fueron tomados in vivo en nuestro laboratorio. La población de voluntarios estuvo conformada por 12 sujetos del sexo masculino, en el rango etario de 27 a 57 años. Se realizó una comparación entre tres arquitecturas neuronales diferentes, MLP, LVQ y ANFIS (sistema de inferencia difuso).

Otro híbrido inteligente utilizado en esta tesis, son los Algoritmos Genéticos diseñados para ajustar los pesos de las Redes Neuronales una vez que han sido entrenadas. Esta variante, permite obtener resultados más cercanos al óptimo, sin tener que modificar la estructura de la red o bien los datos de entrada. Otra vez, después de experimentaciones, encontramos la estructura con la que mejores resultados obtuvimos. Resultados preliminares fueron presentados en:

– Benitez EG, **Pisarello MI**, Álvarez Picaza C, Monzón JE. 2009. "Un sistema inteligente para la clasificación de latidos en la señal electrocardiográfica." XVII Congreso Argentino de Bioingeniería. VI Jornadas de Ingeniería Clínica. SABI.

> Aquí presentamos un sistema experto híbrido que combina las bondades de las redes neuronales con lo algoritmos genéticos. Los resultados demuestran que la incorporación de los AG mejora la calidad de clasificación de la red, sin un costo computacional elevado.

Para resumir, a lo largo de esta tesis obtuvimos los siguientes resultados diferenciadores:

– La aplicación de un Banco de Filtros Digitales, con Reconstrucción Perfecta y filtros de Fase Lineal, de coeficientes reales, capaces de dividir a la señal de interés en sub-bandas de frecuencia, en las que realizamos tres diferentes tareas en paralelo: eliminación de ruido gaussiano, detección del complejo QRS y clasificación de latidos ventriculares prematuros a una tasa de procesamiento menor a la original.

– Diseño de un algoritmo basado en conceptos de estabilidad, capaz de separar señales mezcladas de manera desconocida y obtener así dos señales independientes, una de interés, limpia, y otra que arrastra la interferencia no deseada. Para ello utilizamos la ecuación matricial de Liapunov y la extracción de componentes principales.

- El diseño de un Red Neuronal Artificial que logra diferenciar latidos ventriculares prematuros de los normales. La Red además incorpora el Análisis de Componentes Principales para lograr estandarizar los datos, reducir la matriz de entrada y optimizar los procesos de entrenamiento, lo que acelera la convergencia del aprendizaje.

- Incorporación de Algoritmos Evolutivos a las Redes Neuronales Artificiales, que optimizan sus pesos y logran mejorar los índices ya obtenidos.

1.3. Organización de la Tesis

La tesis está organizada de la siguiente manera: Cuenta con dos partes: Parte I y Parte II, cuyos capítulos son los siguientes:

Parte I: es el marco teórico, incorporamos los conceptos necesarios para el desarrollo de la tesis. Está compuesta por:

El **Capítulo 1**, donde hacemos una breve introducción a la tesis, sus aspectos diferenciadores y justificación conceptual.

En el **Capítulo 2** describimos las señales biomédicas con las que trabajamos y las señales ruidosas que serán parte de nuestro proceso.

El **Capítulo 3**, dedicado a los Bancos de Filtros Digitales, una breve mención de los fundamentos matemáticos necesarios para el desarrollo de la teoría y la descripción detallada de los aspectos relevantes de esta herramienta.

El **Capítulo 4 y 5** están dedicados al Análisis de Componentes Principales e Independientes respectivamente. La teoría que los sustenta y la descripción de la nueva metodología desarrollada para la Separación Ciega de Fuentes.

En el **Capítulo 6** abordamos las Redes Neuronales Artificiales, sus estructuras, aprendizaje, entrenamiento y otros aspectos importantes como las condiciones para la convergencia de los procesos. También incluimos la descripción de las redes neuronales con reglas difusas.

Por último el **Capítulo 7** comprende una presentación de los aspectos más relevantes de los Algoritmos Genéticos, útiles para el ajuste de pesos de las redes neuronales.

Parte II: es la descripción de las aplicaciones desarrolladas, cada una con sus correspondientes resultados y discusión. Está compuesta por los siguientes capítulos:

El **Capítulo 8**, dedicado a la descripción de los índices de performance utilizados para la evaluación de los algoritmos desarrollados a lo largo de toda la tesis.

El **Capítulo 9**, donde hacemos una descripción detallada de la estructura de Banco de Filtros que diseñamos, los procesos de sub-muestreo y sobre-muestreo y los resultados obtenidos luego del diseño de los algoritmos necesarios para desarrollar las tres tareas ya mencionadas.

En el **Capítulo 10**, de aplicaciones del Análisis de Componentes Principales, describimos el método de reducción de la matriz de datos del complejo QRS, diseñada para caracterizar los latidos cardíacos.

El **Capítulo 11**, de resultados del Análisis de Componentes Independientes, presenta la descripción del método diseñado por nosotros para lograr la separación de señales, mezcladas de manera desconocida. Presentamos también una comparación de nuestros resultados con los obtenidos con algoritmos ampliamente probados.

En el **Capítulo 12**, describimos las redes neuronales implementadas para la clasificación de latidos cardíacos, así como una descripción detallada de los datos de entrada a la red.

El **Capítulo 13** detalla el algoritmo utilizado para la optimización de los pesos de la red neuronal diseñada en el capítulo 12.

En el **Capítulo 14** presentamos las conclusiones del trabajo presentado y las perspectivas futuras de cara a nuevos desarrollos continuando con esta línea de investigación.-

2. Las Bioseñales Cardíacas

2.1. El Corazón como una Bomba

El corazón es el órgano principal del sistema circulatorio. Está ubicado en la cavidad torácica, apuntando levemente hacia la izquierda. Un poco más grande que el puño de su portador, el corazón está dividido en cuatro cavidades: dos superiores, llamadas **aurículas** y dos inferiores, **ventrículos,** separadas por una pared muscular denominada tabique, Figura 2.1. Está conformado por fibras musculares especializadas que suministran la fuerza motriz para impulsar la sangre a través del organismo, como una bomba. Esta función de bombeo está principalmente realizada por los ventrículos, mientras que las aurículas, son antecámaras que almacenan la sangre. La fase de llenado del ciclo cardíaco se llama **diástole**, la fase contráctil se denomina **sístole**. El flujo sanguíneo se controla por medio de cuatro válvulas: Tricúspide, Mitral, Pulmonar y Aórtica, las cuales, se encargan de permitir el paso de la sangre entre las cavidades al interior y al exterior del corazón. Para que la sangre llegue a todos los rincones del cuerpo, las partes del corazón laten en sucesión ordenada: la contracción auricular va seguida de la contracción ventricular.

Diferentes tipos de tejido conforman el corazón, por lo que la estructura anatómica de sus células difiere ampliamente. Son eléctricamente excitables y cada tipo de célula posee su propio potencial de acción, Figura 2.1. Todas las células cardíacas tienen la propiedad de generar dipolos eléctricos cuando reciben la señal de activación (despolarización) y también cuando, tras su contracción, regresan al estado de reposo (repolarización). Estos dipolos eléctricos son los que se registran, utilizando un amplificador de biopotenciales, el electrocardiógrafo, y permiten obtener el electrocardiograma (ECG).

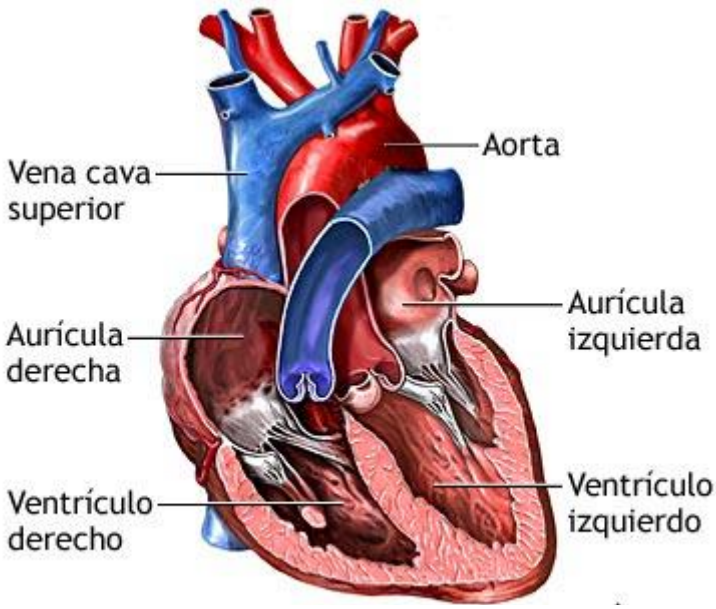

Figura 2.1. Anatomía del corazón humano

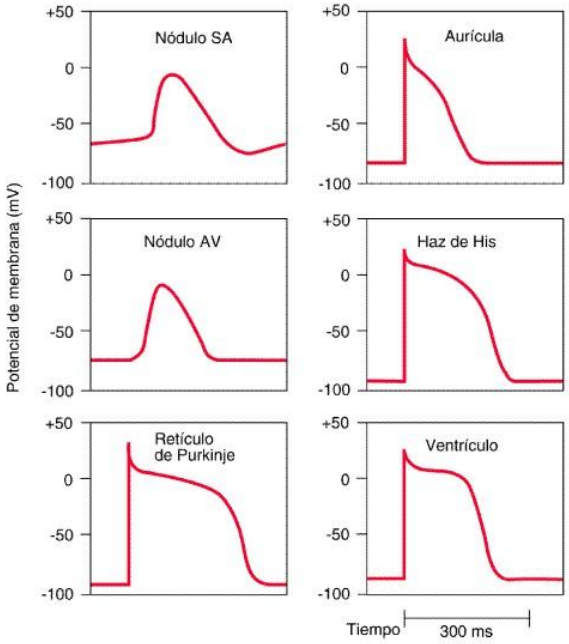

Figura 2.2. Potenciales de acción de varias regiones del corazón.
Fuente:http://articulosdemedicina.com/wp-content/uploads/2008/08/paste231.jpg .

2.2. Electrocardiografía

La propagación del impulso eléctrico a través de los tejidos cardíacos mencionada en la sección anterior, produce corrientes de acción de lazo cerrado que fluyen en el volumen torácico. La medición de estos potenciales en la superficie del cuerpo es lo que comúnmente llamamos **electrocardiograma**.

La electrocardiografía, por su parte, es el estudio de la actividad eléctrica del corazón que representa los cambios en el potencial de acción ocurridos durante el ciclo cardiaco. Estos cambios son descriptos como un vector cardíaco que, según la dirección a la que apunte, indica la dirección de despolarización celular en cada una de las cavidades cardíacas, Figura 2.3. El análisis de la secuencia de propagación del impulso hace factible la deducción del comportamiento electrofisiológico de las estructuras del corazón y de posibles anormalidades. A través de electrodos de superficie ubicados sobre el cuerpo según normas establecidas, se captura esa señal eléctrica que luego el electrocardiógrafo es el encargado de traducirla en una traza de dos dimensiones. Las distintas combinaciones en que se ubican los electrodos sobre la superficie corporal, se denominan derivaciones y son registradas por diferencia de potencial. La señal electrocardiográfica puede interpretarse de acuerdo a la derivación con la que es captada, ya que cada una de estas, representa la lectura de la magnitud y dirección del vector cardíaco desde distintos lugares de referencia.

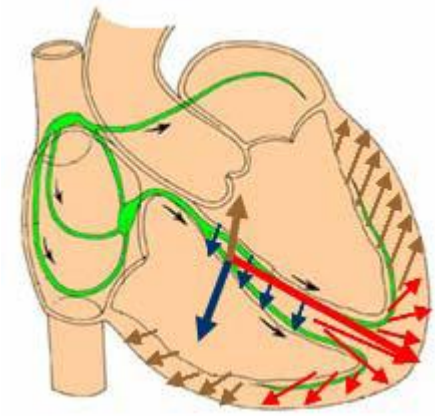

Figura 2.3. Direcciones del Vector cardíaco
Fuente: http://uuhsc.utah.edu/healthinfo/spanish/cardiac/electric.htm.

2.2.1. Derivaciones electrocardiográficas

La disposición de los electrodos sobre el cuerpo del paciente durante el registro del ECG es denominada derivación. Un electrocardiograma completo esta compuesto por 12 derivaciones que pueden ser clasificadas dependiendo de su posición en el cuerpo y del tipo de polaridad utilizada. Por convención, los grados de dirección del vector son interpretados como positivos en el hemisferio inferior del círculo unitario.

2.2.2. Derivaciones del plano frontal

Están a su vez subdividas en bipolares y unipolares. En la Figura 2.4 se muestran todas las derivaciones frontales.

Derivaciones bipolares: Fueron desarrolladas por Willem Einthoven en 1906 y descriptas en el artículo "Le telecardiogramme". *Arch Int Physiol*.

Son tres, cada una de ellas corresponde a dos electrodos aproximadamente equidistantes al corazón.

- *Derivación I (DI):* diferencia de potencial entre el brazo derecho (polo negativo) y el izquierdo (polo positivo). El eje de la derivación es 0º.
- *Derivación II (DII):* diferencia de potencial entre el brazo derecho (polo negativo) y la pierna izquierda (polo positivo). El eje de la derivación es +60º.
- *Derivación III (DIII):* diferencia de potencial entre el brazo izquierdo (polo negativo) y la pierna izquierda (polo positivo). El eje de la derivación es +120º o -60º.

Derivaciones aumentadas: están basadas en señales obtenidas de más de un par de electrodos. Se las denomina *unipolares* pues registran potenciales que aparecen en un electrodo tomado con respecto a un electrodo de referencia equivalente. Este electrodo de referencia es el promedio de las señales "vistas" por dos o más electrodos, se lo conoce como *terminal central de Wilson*, en honor a su descubridor.

- *aVR:* polo positivo en el brazo derecho. El eje de la derivación es 210º (–150º).
- *aVL:* polo positivo en el brazo izquierdo. El eje de la derivación es -30º.
- *aVF:* polo positivo en la pierna izquierda. El eje de la derivación es +90º.

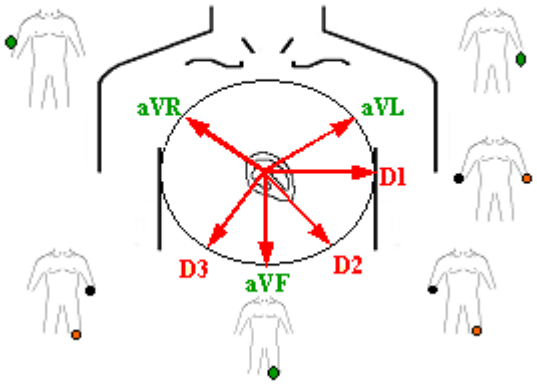

Figura 2.4. Derivaciones del plano frontal.
Fuente: http://uvirtual.sld.cu/wssc/slides/slides.php3?c=71&d=5&sz=small.

2.2.3. Derivaciones del plano horizontal

Derivaciones precordiales: son unipolares y se registran en el tórax desde la posición 1 a la 6, Figura 2.5. Los electrodos móviles registran el potencial eléctrico que hay bajo ellos mismos respecto a la conexión terminal central, que se hace conectando los cables del brazo derecho, el brazo izquierdo, y la pierna izquierda.

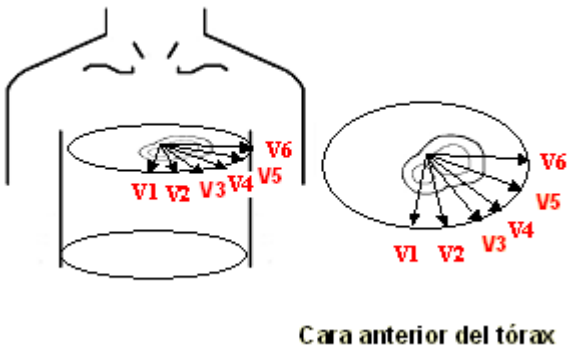

Figura 2.5. Derivaciones Precordiales
Fuente: http://uvirtual.sld.cu/wssc/slides/slides.php3?c=71&d=6&sz=small.

La posición de V1 está en el IV espacio intercostal a la derecha del esternón; V2 está en el IV espacio intercostal a la izquierda del esternón; V4 está a la izquierda de la línea medioclavicular en el V espacio intercostal; V3 está a medio camino entre V2 y V4; V5 está en el V espacio intercostal en la línea axilar anterior, y V6 está en el V espacio intercostal en la línea medioaxilar izquierda.

2.3. El Electrocardiograma Normal

El electrocardiograma es la herramienta de diagnóstico principal de la electrofisiología cardíaca y tiene una función relevante en el diagnóstico de las enfermedades cardiovasculares, alteraciones metabólicas y la predisposición a una muerte súbita cardiaca. No es invasivo, es económico y sus resultados se obtienen en forma inmediata. Si bien, a través de él no es posible conocer el estado completo en el que se encuentra el corazón, es el primer procedimiento al que acude el especialista cuando necesita realizar un control o bien un diagnóstico cardiológico.

La forma de onda de la señal de ECG, la duración de sus intervalos, la amplitud de sus picos característicos, son los datos que el médico “observa” en el ECG para determinar si el paciente está padeciendo alguna cardiopatía. De allí la importancia de conocer los parámetros en el ECG que determinan que éste sea normal.

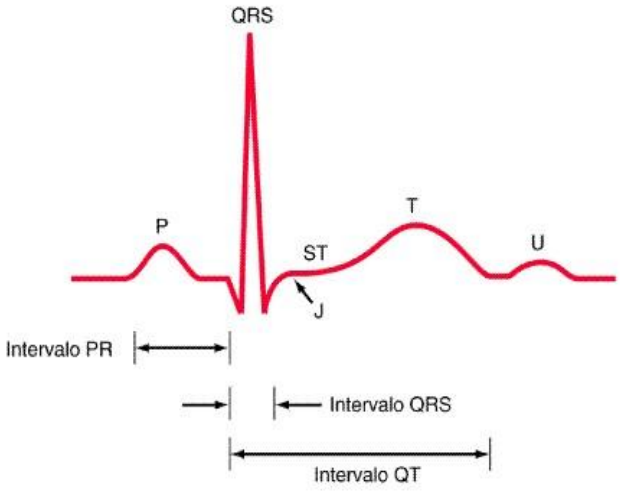

Figura 2.6. ECG normal
Fuente: www.beliefnet.com/.../images/EM00027_ma.jpg

La Figura 2.6 muestra un ciclo cardíaco típico normal. La gráfica de este latido representa a un ECG obtenido de la Derivación II. La onda P en la figura representa la despolarización auricular, mencionada previamente. El intervalo P-R representa el retardo de 0.1 segundo en el nodo AV. El complejo QRS formado por tres ondas, representa la despolarización del ventrículo provocando su contracción. La re-polarización de la aurícula sucede también en este momento, pero queda enmascarada por este complejo. La repolarización del ventrículo provoca la onda T. Seguida a esta onda, aparece la onda U, que es una deflexión de bajo voltaje usualmente positiva y muestra la misma dirección de la onda T. Cuando el nodo SA dispara todo este proceso, decimos que existe ritmo sinusal (Tompkins y Webster, 1998). Un periodo normal dura aproximadamente un segundo, en los adultos, mientras que los bebes alcanzan 90 latidos por segundo.

2.4. Latidos Ventriculares Prematuros

En una sección previa, hicimos referencia a las propiedades rítmicas del tejido cardíaco, que hacen posible la propagación ordenada del impulso eléctrico a través de todo el corazón. Cuando se altera el ritmo cardíaco, estamos en presencia de una

arritmia. Su detección temprana es de vital importancia ya que si bien en muchos casos son benignas y no tienen causa aparente, otras veces, sin embargo, están asociadas con anomalías electrolíticas en la sangre que deben corregirse. Igualmente, pueden estar relacionados con isquemia o reducción local del suministro de sangre al corazón. Además, los latidos ectópicos pueden ser causados o agravados por un excesivo consumo de tabaco, consumo de alcohol, cafeína, ciertos medicamentos y algunas drogas ilícitas.

Una arritmia es un cambio de ritmo, éste puede estar por debajo de lo normal (bradicardia) o bien por encima de lo normal (taquicardia). Los latidos denominados aberrantes, las *extrasístoles auriculares* (EA) y *extrasístoles ventriculares* (EV), pueden producirse aún si el sistema de conducción eléctrico está intacto. Los más frecuentes y potencialmente más peligrosos son los que tienen origen en los ventrículos. La taquicardia ventricular es una alteración seria del ritmo cardiaco. Puede causar la caída brusca de presión arterial y provocar la pérdida de conocimiento. Esta clase de taquicardia hace referencia a una serie de latidos ventriculares prematuros que se suceden rápidamente. Estos ritmos pueden ser cortos o durar más de 30 segundos. El nombre más habitual para denominar las EV es *contracción ventricular prematura* (PVC – Premature Ventricular Contraction).

Una PVC es un latido extra causado por la actividad anormal eléctrica. Se origina en los ventrículos. Las PVC interfieren con el ritmo normal del corazón y causan una pausa antes del latido siguiente. Entre los adultos mayores, es una patología que se encuentra frecuentemente. Puede ocurrir en personas sanas, en cuyo caso no son dañinas. Sin embargo, cuando ocurren luego de un ataque cardiaco o una cardiocirugía pueden producir arritmias peligrosas. En algunos casos, tales arritmias pueden causar la muerte.

En el registro del ECG, las PVC se ven como lo muestra la Figura 2.1. La onda P desaparece y por ello no se puede identificar el intervalo P-R. El complejo QRS dura más de 120 milisegundos y tiene una forma “exagerada”. La ocurrencia de una PVC se manifiesta antes del tiempo que correspondería a un latido normal, seguido de un pulso compensatorio, por esto es que los pacientes lo sienten como un latido omitido.

Se verifica que el intervalo R-R (RR_0) previo a la PVC es menor que el intervalo R-R (RR_{-1}) normal. El R-R (RR_1) posterior a la PVC es notablemente mayor a un intervalo R-R normal. La pausa compensatoria se debe a que es poco probable que la EV se propague retrógradamente a la aurícula y reajuste el nodo sinusal. La suma de los latidos previo y posterior a la PVC es igual al doble de un intervalo R-R normal.

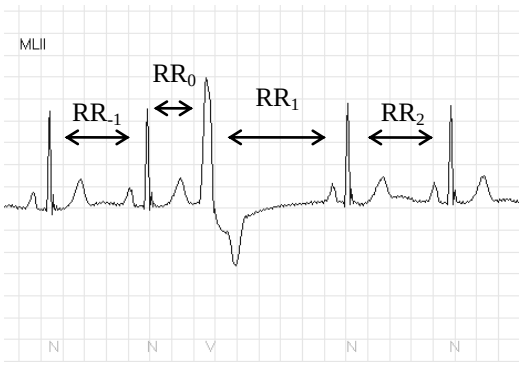

Figura 2.7 Forma típica de una PVC.
N: Latidos normales. V: PVC

Las PVC pueden presentarse en forma aislada, como se muestra en la Figura 2.7. La combinación de un latido normal y una PVC en forma reiterada se denomina bigeminismo, Figura 2.8(a) y Figura 2.8. Cuando aparecen dos latidos normales seguidos de una PVC también en forma repetitiva se denomina trigeminismo, Figura 2.8(b). También pueden aparecer PVC sucesivas, conocidas como pares, Figura 2.8(c). Todos estos casos son de difícil análisis mediante un algoritmo computacional, es por ello que su detección resulta de interés.

(a)

(b)

(c)

Figura 2.8. Extrasístoles Ventriculares. (a) Bigeminismo. (b) Trigeminismo. (c) Pares. Registros originales de la Base de Datos de MIT-BIH

2.5. La Base de Datos de MIT-BIH

Para lograr algoritmos computacionales capaces de procesar señales biomédica y que los resultados obtenidos tengan validez, es necesario contar con una base de datos completa, representativa y que además sea sencillo acceder a ella. Todas estas virtudes le son propias a la Base de Datos de MIT-BIH disponible en (http://www.physionet.org/), de acceso libre y gratuito.

Desde 1999 y gracias a investigadores del Centro Médico de Boston Beth Israel Deaconess (Boston's Beth Israel Deaconess Medical Center), la Universidad de Boston, la Universidad McGill y el Instituto de Tecnología de Massachusset (MIT), la comunidad científica dispone de una completa colección de señales biomédicas que además está en continuo crecimiento, ya que las nuevas contribuciones son tenidas en cuenta y compartidas con los usuarios. Tres bloques independientes conforman el sistema completo (Moody et al., 2001):

- *Physionet* es un foro comunitario de intercambio de señales biomédicas y programas de libre distribución para el análisis y evaluación de estas señales.

- *PhysioBank* es una gran colección de archivos, bien caracterizados. Actualmente incluye bases multiparamétricas de sujetos sanos y pacientes con una variedad de condiciones con implicancias en la salud pública.

- *PhysioToolkit* es una librería de programas para el procesamiento y análisis de señales fisiológicas, detección de eventos característicos, creación de nuevas bases de datos, simulación de señales fisiológicas, evaluación cuantitativa y comparación de métodos de análisis. Todo el software de PhysioToolkit está disponible, en código fuente, bajo la Licencia Pública General (GNU - General Public Licence – http://www.fsf.copyleft/gpl.html)

Las bases de datos que conforman el PhysioBank así como los programas del PhysioToolkit están clasificadas en tres clases:

- **Clase 1**: Las bases de datos de esta clase han sido cuidadosamente revisadas y sus anotaciones fueron hechas por al menos dos expertos, en forma independiente y revisadas por un tercero, si el caso lo ameritaba. Los programas de este grupo, han sido rigurosamente testeados.

- **Clase 2**: Las señales y los programas son copias de resultados que fueron publicados contribuidos por autores o por las mismas revistas.

- **Clase 3**: Las bases de datos incluyen colecciones de datos que han sido estudiados con menos rigor pero que igualmente resultan de interés para la comunidad científica. Los programas de esta clase incluyen copias que pueden necesitar más testeo o desarrollo. Los usuarios de *PhysioNet* pueden contribuir con correcciones o ampliaciones de estos códigos.

2.5.1. La base de Datos de Arritmias de MIT-BIH

Esta base de datos es la más antigua y la que mayor interés despierta en los investigadores. La fuente de las señales es un conjunto de más de 4000 registros ambulatorios de larga duración -24 horas. Pertenece a la Clase 1 y consiste en 48 registros de ECG anotados de 30 minutos de duración, tomadas de 47 sujetos estudiados por el Laboratorio de Arritmias de BIH; 25 hombres de 32 a 89 años de edad y 22 mujeres de 23 a 89 años de edad. Los registros 201 y 202 pertenecen al mismo sujeto (Moody y Mark, 1994).

Las señales están registradas con dos derivaciones simultáneas que varían según el registro. Se seleccionaron 23 registros al azar y 25 fueron cuidadosamente escogidos para incluir arritmias menos comunes pero de interés clínico. Del primer grupo, 10 registros se hallan disponibles en su totalidad (registros 100-1007, 118-119) y 15 del segundo grupo (registros 201-219). De los restantes, se dispone de los primeros 10 minutos de registro con sus anotaciones.

La serie 100 (100-124, con algunos números faltantes) es una muestra representativa de las diversas formas de onda que un detector de arritmia podría

encontrar en uso clínico rutinario. La serie 200 (200-234, con algunos números faltantes) incluyen arritmias ventriculares y supraventriculares y trastornos en la conducción. La variación de la morfología de QRS o la calidad de las señales de esta serie resultan hace que los algoritmos detectores de arritmias disminuyan su performance.

2.5.2. La Base de Datos de Ruidos de MIT-BIH

Esta base de datos contiene señales de interferencia de magnitud calibrada. Consiste en 12 registros de 30 minutos cada uno. Son 3 señales de ruidos que frecuentemente aparecen en la captura de ECG ambulatorio. Los registros de ruido se capturaron de sujetos voluntarios, se utilizaron electrocardiógrafo, derivaciones y electrodos. Éstos últimos ubicados en miembros de tal manera que no sea visible el ECG del sujeto. Los tres ruidos registrados son ruido de base (bw –baseline wander), ruido de contracciones musculares (ma –muscle artifact) y ruido por movimiento de electrodos (em –electrode motion). Este último, muy difícil de eliminar debido a su apariencia similar a la de los latidos ectópicos.

Los registros de esta base de datos fueron creados con la rutina *nstdbgen*, disponible en PhysioToolkit, que permite agregar cantidades calibradas de cualquiera de los ruidos descriptos anteriormente a una señal de la Base de Datos de Arritmias (118 y 119 en este caso). Los niveles de magnitud de la relación señal-ruido (SNR, *signal to noise ratio*) son 6 y van desde 24db a -6db en escala de 6, de menos ruidoso a más ruidoso.

Otro ruido que frecuentemente aparece durante la toma del ECG clínico es el ruido electromagnético de 50Hz. Es un ruido aleatorio, llamado también blanco que tiene frecuencias similares a las del ECG. Por ello es vital su eliminación, ya que de otro modo podría enmascarar la señal de interés y perder así características relevantes para su análisis. A los fines prácticos, desarrollamos una rutina que agrega ruido blanco de amplitud variable a los registros biomédicos, con el objetivo de evaluar nuestros algoritmos en entornos ruidosos.

3. Bancos de Filtros

Antes de abordar este capítulo en profundidad, entendimos que es necesario hacer una introducción acerca de los tres temas más relevantes que tienen que ver con la teoría de Bancos de Filtros. Empezamos nuestra introducción con el análisis de Fourier, que luego en una sección posterior será tratado en forma más exhaustiva, continuamos con la teoría de onditas, muy relacionadas con los Bancos de Filtros, de los que también haremos una breve mención sobre sus comienzos. Por último, mencionaremos las Diracs, útiles en hacer la transición de funciones de una variable real a sucesiones discretas, aunque más adelante también nos referiremos a ellas nuevamente.

3.1 Introducción (análisis de Fourier, Onditas, Bancos de Filtros)

3.1.1 El análisis de Fourier

El análisis de Fourier representa cualquier función de energía finita $f(t)$ como la suma de ondas sinusoides $e^{i\omega t}$ del modo siguiente

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(\omega) e^{i\omega t} d\omega . \qquad (0.1)$$

Su transformada está presente en la física y la matemática debido a que diagonaliza operadores de convolución invariantes en el tiempo y gobierna el procesamiento de señales lineales invariantes en el tiempo.

En la ecuación (0.1), la amplitud $\hat{f}(\omega)$ de cada onda sinusoide $e^{i\omega t}$ es igual a su correlación con f, llamada también transformada de Fourier (FT –Fourier Transform):

$$\hat{f}(\omega) = \int_{-\infty}^{+\infty} f(t) e^{-i\omega t} dt . \qquad (0.2)$$

Para $f(t)$ definidas en un intervalo [0,1], por ejemplo, la transformada de Fourier se convierte en una descomposición en bases ortonormales de Fourier $\{e^{i2\pi mt}\}_{m\in\mathbb{Z}}$ de

$\mathbf{L}^2[0,1]$. Sobre señales discretas, la FT es una descomposición en una base ortogonal discreta de Fourier $\left\{e^{i2k\pi n/N}\right\}_{0\le k<N}$ de $\mathbb{C}^N$, que tiene las propiedades similares a la FT sobre funciones. Su estructura conduce a los algoritmos de la transformada rápida de Fourier (FFT), que calculan sus coeficientes con menos operaciones. Este algoritmo FFT es una pieza fundamental en el proceso de señales discretas. Mientras se utilicen operadores lineales invariantes en el tiempo, la FT provee respuestas simples a muchas preguntas. Su potencia la hace apropiada para un amplio rango de aplicaciones tales como transmisión de señales o procesamiento de señales estacionarias, (Mallat, 2009). Sin embargo, para representar un fenómeno transitorio, la FT requiere muchos coeficientes para representar un evento localizado. En efecto, el soporte de $e^{i\omega t}$ cubre toda la línea real, entonces $\hat{f}(\omega)$ depende de los valores de $f(t), \forall t \in \mathbb{R}$. Esto hace que sea difícil de representar cualquier propiedad local de $f(t)$ a partir de $\hat{f}(\omega)$. Como ejemplo de esto, podemos citar la delta de Dirac $\delta(t)$ que será descripta en una sección posterior.

3.1.2 Bases de Onditas

Al igual que las bases de Fourier, las bases de onditas, revelan la regularidad de la señal a través de la amplitud de los coeficientes y su estructura nos conduce a un algoritmo rápido. Sin embargo, las onditas están bien localizadas y además se necesitan pocos coeficientes para representar estructuras transitorias. En oposición a una base de Fourier, una base de onditas define una representación incluye la localización de fenómenos transitorios y singularidades.

Un poco de Historia:

En 1910 A. Haar presentó en su trabajo original, "Zur theorie der orthogonalen funktionensysteme" (Haar, 1910) la construcción de una función constante por tramos

$$\psi(t)=\begin{cases}1 & \text{si } 0\le t<1/2\\ -1 & \text{si } 1/2\le t<1\\ 0 & \text{en otro caso}\end{cases},$$

de su dilatación y traslación se genera una base ortonormal

$$\left\{\psi_{j,n}(t)=\frac{1}{\sqrt{2^j}}\psi\left(\frac{t-2^j n}{2^j}\right)\right\}_{(j,n)\in\mathbb{Z}^2}$$

del espacio $\mathbf{L}^2(\mathbb{R})$ de señales con energía finita

$$\|f\|^2=\int_{-\infty}^{+\infty}|f(t)|^2\,dt<+\infty.$$

Si escribimos $\langle f,g\rangle=\int_{-\infty}^{+\infty}f(t)g^*(t)dt$ como el producto interior en $\mathbf{L}^2(\mathbb{R})$,cualquier señal de energía finita f puede ser representada por los coeficientes del producto interior de sus onditas

$$\langle f,\psi_{j,n}\rangle=\int_{-\infty}^{+\infty}f(t)\psi_{j,n}(t)\,dt,$$

y recuperados a partir de su suma en esta base ortonormal de onditas:

$$f=\sum_{j=-\infty}^{+\infty}\sum_{n=-\infty}^{+\infty}\langle f,\psi_{j,n}\rangle\psi_{j,n}\,. \qquad (0.3)$$

Nuevos avances se produjeron en 1980, cuando Strömberg encontró una función lineal por tramos ψ que también genera una base ortonormal y da además mejores aproximaciones de funciones suaves (funciones cuyas derivadas sucesivas existen infinitamente). A partir de este hallazgo, Mayer en 1986 construyó toda una familia de bases ortonormales de onditas, con funciones ψ que son infinitamente y continuamente diferenciables. Este fue es impulso fundamental que condujo a la búsqueda de nuevas bases ortonormales de onditas, que culminó en las onditas tan conocidas de Daubechies de soporte compacto.

Un estudio más profundo de las propiedades de las onditas ortonormales y aproximaciones de multiresolución, traen a la luz una conexión con Bancos de Filtros construidos con filtros en espejo.

3.1.3 Bancos de Filtros

La historia comienza en 1976 cuando Croisier, Esteban y Galand presentaron un banco de filtros inversible, que descomponía una señal discreta $f[n]$ en dos señales cuyas longitudes eran la mitad de la señal original usando procedimientos de filtrado y submuestreo. Ellos demostraron que $f[n]$ puede ser recuperada a partir de estas señales submuestreadas cancelando los términos de aliasing con una clase particular de filtros llamados *filtros conjugados en espejo*. La construcción de una teoría completa de bancos de filtros llevó 10 años de esfuerzo. Las condiciones necesarias y suficientes para la descomposición de una señal en componentes submuestreados con un esquema de filtros, y la recuperación de la misma señal con una transformada inversa, fueron establecidas por (Smith y Barnwell, 1984), (Vetterli, 1986) y (Vaidyanathan, 1987).

La teoría de la multiresolución (Mallat, 2009) prueba que cualquier filtro conjugado en espejo caracteriza una ondita ψ que genera una base ortonormal de $\mathbf{L}^2(\mathbb{R})$ y que la transformada rápida de ondita discreta se implementa iterando en cascada esos filtros conjugados en espejo. La equivalencia entre esta teoría de onditas en tiempo continuo y los bancos de filtro discretos conducen a una provechosa interfase entre procesamiento digital de señales y análisis armónico.

3.1.4 Delta de Dirac

Las Diracs son útiles en hacer la transición de funciones de una variable real a sucesiones discretas. Los cálculos simbólicos con Diracs simplifican la computación, sin tener que preocuparse por temas como la convergencia. Esto está bien justificado por la teoría de las distribuciones (D'Attellis, 1985).

Una Dirac δ tiene un soporte reducido a $t=0$ y asocia cualquier función continua ϕ su valor en $t=0$,

$$\int_{-\infty}^{+\infty}\delta(t)\phi(t)dt=\phi(0). \tag{0.4}$$

Convergencia de las Diracs

Una Dirac puede ser obtenida a partir de una función integral *g* de modo que $\int_{-\infty}^{+\infty}g(t)dt=1$ Sea $g_s(t)=s^{-1}g\left(s^{-1}t\right)$. Para cualquier función continua ϕ se cumple que

$$\lim_{S\to 0}\int_{-\infty}^{+\infty}g_s(t)\phi(t)dt=\phi(0)=\int_{-\infty}^{+\infty}\delta(t)\phi(t)dt. \tag{0.5}$$

Luego, una Dirac puede ser formalmente definida como el límite $\delta=\lim_{s\to 0}g_s$, en el sentido de (0.5). A esto se lo denomina convergencia débil. Una Dirac no es una función, pues vale cero en $t=0$ aunque su integral sea igual a 1. La integral a la derecha en (0.5) simboliza el hecho de que una Dirac aplicada a una función continua ϕ asocia su valor a $t=0$.

Las distribuciones generales están definidas sobre el espacio $\mathbb{C}_0^\infty$ de *funciones de testeo* que son continuas e infinitamente diferenciable con soporte compacto. Una distribución *d* es una forma lineal que asocia cualquier $\phi\in\mathbb{C}_0^\infty$ con un valor que se escribe como $\int_{-\infty}^{+\infty}d(t)\phi(t)dt$. Debe además satisfacer propiedades de convergencia débil que son satisfechas por una Dirac, (D'Attellis, 1985). Dadas dos distribuciones d_1 y d_2, se verifica la igualdad entre ellas si

$$\forall\phi\in\mathbb{C}_0^\infty,\quad \int_{-\infty}^{+\infty}d_1(t)\phi(t)=\int_{-\infty}^{+\infty}d_2(t)\phi(t)dt. \tag{0.6}$$

3.2 Análisis de Fourier: una herramienta esencial

En 1807 Fourier presentó un memorándum al Instituto de Francia, donde afirmaba que cualquier función periódica puede ser representada como una serie de sinusoides relacionadas armónicamente. Tal idea tuvo un alto impacto en el análisis matemático, la

física y la ingeniería, pero tomó un siglo y medio comprender la convergencia de las series de Fourier y completar la teoría de las integrales de Fourier. Su trabajo estuvo motivado por el estudio de la difusión del calor, gobernada por una ecuación diferencial parcial lineal. Sin embargo, la transformada de Fourier diagonaliza todos los operadores lineales invariantes en el tiempo -los bloques de construcción del procesamiento de señales. Éste es el punto inicial de nuestra exposición. Abordaremos primeramente el análisis de Fourier con funciones en espacios de dimensión infinita $\mathbf{L}^1$ y $\mathbf{L}^2$, mencionaremos teoremas que estudian sus propiedades. Luego haremos lo propio con las series de Fourier y la transformada de Fourier para señales discretas, sin dejar de mencionar un teorema de muestreo.

3.2.1 Filtros lineales invariantes en el tiempo

Los procesos clásicos que involucran señales tales como la transmisión de datos, eliminación de ruido estacionario, etc., son implementados con operadores lineales invariantes en el tiempo. Un filtro es un operador lineal.

La invariancia en el tiempo de un operador *L* significa que si la entrada $f(t)$ es retrasada por *τ*, $f_\tau(t) = f(t-\tau)$, luego la salida también es retrasada por *τ:*

$$g(t) = Lf(t) \Rightarrow g(t-\tau) = Lf_\tau(t). \tag{0.7}$$

El operador *L* debe tener continuidad, lo que significa que a poca variación de *f, Lf* se modifica también en una pequeña cantidad, esto proporciona una estabilidad numérica. Esta continuidad está formalizada por la teoría de las distribuciones, lo que nos asegura que estamos en un terreno seguro sin tener que preocuparnos, (D'Attellis, 1985).

Respuesta al impulso

Los sistemas lineales invariantes en el tiempo se caracterizan por su respuesta al impulso de una Dirac. Si *f* es continua, su valor en *t* se obtiene por una integración contra una Dirac localizada en *t*. Sea $\delta_u(t) = \delta(t-u)$

$$f(t) = \int_{-\infty}^{+\infty} f(u)\delta_u(t)du\,.$$

La continuidad y linealidad de L implican que

$$Lf(t) = \int_{-\infty}^{+\infty} f(u)L\delta_u(t)du\,.$$

Sea *h* la respuesta al impulso de *L*, es decir

$$h(t) = L\delta(t).$$

La invariancia en el tiempo prueba que $L\delta_u(t) = h(t-u)$, luego

$$Lf(t) = \int_{-\infty}^{+\infty} f(u)h(t-u)du = \int_{-\infty}^{+\infty} h(u)\,f(t-u)du = h * f(t). \qquad (0.8)$$

Es decir que un filtro lineal e invariante en el tiempo es equivalente a una convolución con la respuesta al impulso *h*. La fórmula en (0.8) sigue siendo válida para cualquier señal *f* para la cual la integral de convolución converge.

Algunas propiedades del producto de convoluciones

Conmutatividad $f * h(t) = h * f(t).$ (0.9)

Diferenciación $\frac{d}{dt}(f * h)(t) = \frac{df}{dt} * h(t) = f * \frac{dh}{dt}(t).$ (0.10)

Convolución de la Dirac $f * \delta_\tau(t) = f(t-\tau).$ (0.11)

Estabilidad y Causalidad

Se dice que un filtro es *causal* si $Lf(t)$ no depende de los valores de $f(u)$ para $u > t$. Como

$$Lf(t) = \int_{-\infty}^{+\infty} h(u)\,f(t-u)du,$$

esto significa que $h(u)=0$ para $u<0$. Se dice que esta respuesta al impulso es *causal*.

La característica de *estable* garantiza que $Lf(t)$ está acotado si $f(t)$ también lo está.

$$|Lf(t)| \le \int_{-\infty}^{+\infty} |h(u)|\,|f(t-u)|\,du \le \sup_{u\in\mathbb{R}} |f(u)| \int_{-\infty}^{+\infty} |h(u)|\,du,$$

es suficiente que $\int_{-\infty}^{+\infty} |h(u)|\,du < +\infty$. Esta condición también es necesaria siempre que *h* sea una función. Luego podemos afirmar que *h* es *estable* si es integrable.

Funciones Transferencia.

Los exponenciales complejos $e^{i\omega t}$ son autovectores de los operadores de convolución. Más aún

$$Le^{i\omega t} = \int_{-\infty}^{+\infty} h(u) e^{i\omega(t-u)}\,du,$$

lo que produce

$$Le^{i\omega t} = e^{i\omega t} \int_{-\infty}^{+\infty} h(u) e^{-i\omega u}\,du = \hat{h}(\omega) e^{i\omega t}.$$

El autovalor

$$\hat{h}(\omega) = \int_{-\infty}^{+\infty} h(u) e^{-i\omega u}\,du$$

es la transformada de Fourier de *h* en la frecuencia ω. Los autovectores del sistema lineal e invariante en el tiempo son las ondas sinusoidales complejas $e^{i\omega t}$, por ende, podríamos pensar en descomponer cualquier función f como la suma de esos autovectores. Es posible, entonces expresar *Lf* directamente a partir de sus autovalores $\hat{h}(\omega)$.

3.2.2 Integrales de Fourier

Definiremos primeramente las integrales de Fourier sobre el espacio $\mathbf{L}^1(\mathbb{R})$ de funciones integrables. Luego extenderemos el análisis sobre el espacio $\mathbf{L}^2(\mathbb{R})$ de funciones de energía finita.

Transformada de Fourier en $\mathbf{L}^1(\mathbb{R})$

La integral de Fourier mide cuántas oscilaciones en la frecuencia ω hay en f. Su fórmula está dada por

$$\hat{f}(\omega) = \int_{-\infty}^{+\infty} f(t) e^{-i\omega t} dt \,. \tag{0.12}$$

Si $f \in \mathbf{L}^1(\mathbb{R})$, entonces la integral converge y además está acotada, es decir:

$$\left|\hat{f}(\omega)\right| \leq \int_{-\infty}^{+\infty} \left|f(t)\right| dt < +\infty \,.$$

Si además $\hat{f}$ también es integrable, entonces también es posible hallar su transformada inversa. Debido a que no es específicamente de nuestro interés, sólo enunciaremos el teorema.

La Transformada Inversa de Fourier está dada por el siguiente Teorema:

Teorema 3.1: Sea $f \in \mathbf{L}^1(\mathbb{R})$ y $\hat{f} \in \mathbf{L}^1(\mathbb{R})$ luego se cumple que:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(\omega) e^{i\omega t} d\omega \tag{0.13}$$

A continuación enunciaremos un teorema que constituye la propiedad más importante en las aplicaciones de procesamiento de señales, es el teorema de la convolución. Es además otra manera de expresar que las ondas sinusoides $e^{it\omega}$ son autovalores de los operadores de convolución (Mallat, 2009).

Teorema 3.2: *Convolución*. Sea $f \in \mathbf{L}^1(\mathbb{R})$ y $h \in \mathbf{L}^1(\mathbb{R})$. La función $f * h \in \mathbf{L}^1(\mathbb{R})$ y además

$$\hat{g}(\omega) = \hat{h}(\omega)\,\hat{f}(\omega) \tag{0.14}$$

La demostración se encuentra en (Mallat, 2009)

La respuesta $Lf = g = f * h$ de un sistema lineal e invariante en el tiempo puede calcularse a partir de su transformada de Fourier $\hat{g}(\omega) = \hat{f}(\omega)\hat{h}(\omega)$ con la fórmula de la inversa de Fourier:

$$g(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \hat{g}(\omega) e^{i\omega t} d\omega, \tag{0.15}$$

lo que produce

$$Lf(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \hat{h}(\omega)\,\hat{f}(\omega) e^{i\omega t} d\omega\,.$$

Cada componente de frecuencia $e^{i\omega t}$ de amplitud $\hat{f}(\omega)$ es amplificada o atenuada por $\hat{h}(\omega)$. Esta convolución recibe el nombre de *filtrado de frecuencia* y $\hat{h}(\omega)$ es la *función transferencia* del filtro.

Transformada de Fourier en $\mathbf{L}^2(\mathbb{R})$

La transformada de Fourier de la función $f = 1_{[-1,1]}$ es

$$\hat{f}(\omega) = \int_{-1}^{1} e^{-i\omega t} dt = \frac{2\operatorname{sen}\omega}{\omega}\,.$$

Esta función $\in \mathbf{L}^1(\mathbb{R})$ y su cuadrado es integrable. Esto motivó la extensión de la transformada de Fourier al espacio $\mathbf{L}^2(\mathbb{R})$ de funciones f con energía finita $\int_{-\infty}^{+\infty} |f(t)|^2 \, dt < +\infty$. Al trabajar en espacios de Hilbert (espacios con producto interior completos) $\mathbf{L}^2(\mathbb{R})$**,** tenemos disponibles todas las propiedades de un producto interior. El producto interior de $f \in \mathbf{L}^2(\mathbb{R})$ y $g \in \mathbf{L}^2(\mathbb{R})$ es

$$\langle f, g \rangle = \int f(t) g^*(t) dt,$$

donde $g^*(t)$ denota el complejo conjugado y verifica además que su FT es $\hat{g}^*(-\omega)$; su norma en $\mathbf{L}^2(\mathbb{R})$ es

$$\|f\|^2 = \langle f, f \rangle = \int_{-\infty}^{+\infty} |f(t)|^2 \, dt .$$

El teorema que se menciona a continuación prueba que el producto interior y la norma en $\mathbf{L}^2(\mathbb{R})$ se mantienen por la transformada de Fourier hasta el factor 2π.

Teorema 3.3. Si $f, g \in \mathbf{L}^1(\mathbb{R}) \cap \mathbf{L}^2(\mathbb{R})$**,** entonces

$$\int_{-\infty}^{+\infty} f(t) h^*(t) dt = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(\omega) \hat{h}^*(\omega) d\omega, \tag{0.16}$$

llamada fórmula de *Parseval*.

Para $h = f$ se verifica que

$$\int_{-\infty}^{+\infty} |f(t)|^2 \, dt = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \left| \hat{f}(\omega) \right|^2 d\omega, \tag{0.17}$$

que es la fórmula de *Plancherel*

La demostración puede encontrarse en Mallat, 2009

Diracs

Las Diracs siempre son utilizadas en los cálculos; ya hemos mencionado algo de ellas y su convergencia en la introducción de este capítulo.

Una Dirac δ asocia su valor a una función en $t=0$. Como $e^{i\omega t}=1$ en $t=0$, podríamos definir su transformada de Fourier como

$$\hat{\delta}(\omega)=\int_{-\infty}^{+\infty}\delta(t)e^{-i\omega t}dt=1.$$

3.3 El Mundo Discreto

El universo digital ha colmado el procesamiento de señales. Usados desde los años 50 en el servicio de procesamiento de señales para simular transformadas analógicas, los algoritmos digitales han invadido nuestras comunicaciones, nuestras cámaras fotográficas, televisores, música. El procesamiento analógico, si bien es más rápido que el procesamiento digital, resulta menos preciso y menos flexible, lo que provoca que en muchos casos los circuitos analógicos sean reemplazados por sus equivalentes digitales.

La mayoría de las señales digitales, incluidas las imágenes, sonidos y las señales biológicas se obtienen a partir de señales analógicas que son muestreadas. La conversión analógica-digital es una aproximación lineal que introduce un error que depende de la tasa de muestreo. La transformada de Fourier otra vez juega un papel fundamental; pero ahora está discretizada para señales finitas e implementada con el algoritmo de la transformada rápida de Fourier (FFT – Fast Fourier Transform).

3.3.1 Muestreo de Señales Analógicas

Para obtener una señal discreta a partir de una señal analógica *f*, se toman valores de muestra $\{f(ns)\}_{n\in\mathbb{Z}}$ en intervalos *s*. La reconstrucción de $f(t)$ puede obtenerse mediante la interpolación de sus muestras, en realidad hablamos de una aproximación de $f(t)$. El teorema de muestreo de Shannon-Whittaker brinda una condición

suficiente sobre el soporte de la transformada de Fourier $\hat{f}$ para recuperar en forma exacta $f(t)$. Cuando esta condición no se satisface, se estudian los errores de aproximación y aliasing (concepto que será explicado más adelante). Los dispositivos de adquisición de datos no siempre satisfacen la condición restrictiva del teorema de muestreo mencionado, sin embargo aproximan coeficientes discretos con una precisión dada para almacenarlos con un número limitado de bits.

El teorema citado estudia las bases para reconstruir $f(t)$ a partir de sus muestras $\{f(ns)\}_{n\in\mathbb{Z}}$. Una señal discreta puede ser representada como la suma de Diracs. Dada una muestra cualquiera $f(ns)$ podemos asociar una Dirac $f(ns)\delta(t-ns)$ ubicada en $t=ns$. Un muestreo uniforme de f corresponde entonces a la suma ponderada de las Diracs

$$f_d(t)=\sum_{n=-\infty}^{+\infty} f(ns)\delta(t-ns). \tag{0.18}$$

La transformada de Fourier de $\delta(t-ns)$ es e^{insw}, luego la transformada de Fourier de f_d es una serie de Fourier

$$\hat{f}_d(\omega)=\sum_{n=-\infty}^{+\infty} f(ns)e^{-insw}. \tag{0.19}$$

Para comprender cómo calcular $f(t)$ a partir de sus valores de muestras $f(ns)$ y por ende f a partir de f_d, relacionamos sus transformadas de Fourier $\hat{f}$ y $\hat{f}_d$.

Teorema 3.4 La transformada de Fourier de la señal discreta obtenida por el muestreo de f en intervalos *s* es

$$\hat{f}_d(\omega)=\frac{1}{s}\sum_{k=-\infty}^{+\infty}\hat{f}\left(\omega-\frac{2k\pi}{s}\right). \tag{0.20}$$

La demostración puede encontrarse en Mallat, 2009

Este teorema prueba que muestreando f en intervalos *s* es equivalente a hacer su transformada de Fourier con período $2\pi/s$ sumando todas sus traslaciones $\hat{f}\left(\omega - 2k\pi/s\right)$.

Aliasing

Las restricciones de almacenamiento o computacionales son las encargadas de imponer el intervalo de muestreo *s* y el soporte de $\hat{f}$ generalmente no está incluido en $\left[-\pi/s, \pi/s\right]$. En este caso la fórmula de interpolación

$$f(t) = \sum_{n=-\infty}^{+\infty} f(ns)\phi_s(t - ns),$$

no recupera f.

El teorema 3.4 prueba que

$$\hat{f}_d(\omega) = \frac{1}{s}\sum_{k=-\infty}^{+\infty} \hat{f}\left(\omega - \frac{2k\pi}{s}\right). \tag{0.21}$$

Supongamos que el soporte de $\hat{f}$ va más allá de $\left[-\pi/s, \pi/s\right]$. En general, el soporte de $\hat{f}\left(\omega - \frac{2k\pi}{s}\right)$ interseca $\left[-\pi/s, \pi/s\right]$ para algunos $k \neq 0$. Este solapamiento de componentes de alta frecuencia sobre un intervalo de baja frecuencia se denomina *aliasing.*

Cuando este efecto se produce, la señal interpolada

$$\phi_s * f_d(t) = \sum_{n=-\infty}^{+\infty} f(ns)\phi_s(t - ns)$$

tiene una transformada de Fourier

$$\hat{f}_d(\omega)\hat{\phi}_s(\omega) = s\hat{f}_d(\omega)1_{[-\pi/s,\pi/s]}(\omega)$$
$$= 1_{[-\pi/s,\pi/s]}(\omega)\sum_{k=-\infty}^{+\infty}\hat{f}\left(\omega - \frac{2k\pi}{s}\right), \tag{0.22}$$

la que puede ser completamente diferente de $\hat{f}(\omega)$ sobre $\left[-\pi/s, \pi/s\right]$. La señal $\phi_d * f_d$ puede que no sea una buena aproximación de *f*.

3.3.2 Filtros Discretos Invariantes en el Tiempo

Respuesta al Impulso y Función Transferencia

La mayoría de los algoritmos de procesamiento de señales, incluidos los diseñados por nosotros, están basados en operadores lineales invariantes en el tiempo. Para simplificar la notación, el intervalo de muestreo está normalizado a $s = 1$, y $f[n]$ es la señal muestreada o bien son los valores de muestra.

Un operador lineal discreto *L* es invariante en el tiempo si una entrada, retrasada por $p \in \mathbb{Z}$, $f_p[n] = f[n-p]$, produce una salida también retrasada por *p*:

$$Lf_p[n] = Lf[n-p].$$

Respuesta al Impulso

La Dirac discreta es $\delta[n]$

$$\delta[n] = \begin{cases} 1 & \text{si } n = 0 \\ 0 & \text{si } n \neq 0 \end{cases}. \tag{0.23}$$

Cualquier señal $f[n]$ puede ser descompuesta en una suma de Diracs:

$$f[n] = \sum_{p=-\infty}^{+\infty} f[p]\delta[n-p].$$

Sea $L\delta[n]=h[n]$ la respuesta al impulso. La linealidad y la invariancia en el tiempo implica que

$$Lf[n]=\sum_{p=-\infty}^{+\infty} f[p]h[n-p]=f*h[n]. \tag{0.24}$$

Entonces el cálculo de un operador lineal e invariante en el tiempo se realiza a través de la convolución. Si $h[n]$ tiene soporte finito, es decir si $f\in\ell^1$ y $h\in\ell^1$ la convolución tiene sentido, así como en el caso de funciones continuas, cómo lo vimos en el teorema 3.5 de la convolución. Entonces la suma (0.24) se calcula con un número finito de operaciones. Estos son llamados filtros con respuesta finita al impulso (FIR - *finite impulse response*). Las convoluciones con filtros de respuesta infinita al impulso pueden ser calculadas también con un número finito de operaciones si pueden escribirse con una ecuación recursiva.

Causalidad y Estabilidad

Un filtro discreto es *causal* si $Lf[p]$ depende solamente de los valores de $f[n]$ para $n\le p$. La fórmula de convolución (0.24) implica que es $h[n]=0$ si $n<0$.

El filtro es *estable* si cualquier señal de entrada acotada $f[n]$ produce una señal de salida también acotada $Lf[n]$. Como

$$\left|Lf[n]\right|\le \sup_{n\in\mathbb{Z}}\left|f[n]\right|\sum_{k=-\infty}^{+\infty}\left|h[k]\right|,$$

es suficiente que $\sum_{n=-\infty}^{+\infty}\left|h[n]\right|<+\infty$, lo que significa que $h\in\ell^1(\mathbb{Z})$. Esta condición no solo es suficiente sino necesaria, (Mallat, 2009). Luego, el filtro es estable sí y sólo sí $h\in\ell^1(\mathbb{Z})$.

3.3.3 Función Transferencia

La transformada de Fourier, como ya lo mencionamos, juega un papel fundamental en el análisis de operadores discretos invariantes en el tiempo debido a que las ondas sinusoidales $e_\omega[n] = e^{i\omega n}$ son autovectores o autofunciones

$$Le_\omega[n] = \sum_{p=-\infty}^{+\infty} e^{i\omega(n-p)} h[p] = e^{i\omega n} \sum_{p=-\infty}^{+\infty} h[p] e^{-i\omega p} . \tag{0.25}$$

El autovalor es una serie de Fourier

$$h(\omega) = \sum_{p=-\infty}^{+\infty} h[p] e^{-i\omega p} . \tag{0.26}$$

Esta es la *función transferencia* del filtro.

3.3.4 Series de Fourier

Las series de Fourier son instancias particulares de las transformadas de Fourier de sumas de Diracs, por lo tanto, las propiedades son en esencia las mismas. Si $f(t) = \sum_{n=-\infty}^{+\infty} f[n]\delta(t-n)$, luego $\hat{f}(\omega) = \sum_{n=-\infty}^{+\infty} f[n] e^{-i\omega n}$. Para cualquier $n \in \mathbb{Z}$, $e^{-i\omega n}$ tiene período *2π*, luego las series de Fourier tienen período *2π.* Un punto importante a comprender es si todas las funciones con período 2π pueden escribirse como series de Fourier. Dichas funciones están caracterizadas por su restricción a $[-\pi, \pi]$. Se consideran entonces funciones del tipo $\hat{a} \in \mathbf{L}^2[-\pi, \pi]$ que son de cuadrado integrable sobre $[-\pi, \pi]$. El espacio $\mathbf{L}^2[-\pi, \pi]$ es un espacio de Hilbert con producto interior

$$\left\langle \hat{a}, \hat{b} \right\rangle = \frac{1}{2\pi} \int_{-\pi}^{\pi} \hat{a}(\omega) \hat{b}^*(\omega) d(\omega) \tag{0.27}$$

y la norma resultante

$$\|\hat{a}\|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |\hat{a}(\omega)|^2 \, d\omega .$$

El teorema que será mencionado a continuación prueba que cualquier función en $\mathbf{L}^2[-\pi,\pi]$ puede escribirse como una serie de Fourier.

Convergencia Puntual

La igualdad de la ecuación **¡Error! No se encuentra el origen de la referencia.** está definida en este sentido de la convergencia

$$\lim_{N\to+\infty}\left\|\hat{f}(\omega)-\sum_{k=-N}^{N} f[k]e^{-i\omega k}\right\|=0,$$

lo que no implica una convergencia puntual en todo $\omega\in\mathbb{R}$.

Debido a que la demostración no es de interés específico para esta tesis, sólo nos limitaremos a comentar que en 1873, Dubois-Reymond construyó una función periódica $\hat{f}(\omega)$ que es continua y tiene una serie de Fourier que diverge en algunos puntos. Ahora bien , si $\hat{f}(\omega)$ es continuamente diferenciable, la demostración del teorema 3.5 pone en claro que su serie de Fourier converge uniformemente a $\hat{f}(\omega)$ en $[-\pi,\pi]$. No fue sino hasta 1966 que Carleson probó que si $\hat{f}\in\mathbf{L}^2[-\pi,\pi]$ entonces su serie de Fourier converge en casi todo punto, (Mallat, 2009).

3.3.5 Convoluciones

Como $\{e^{-i\omega k}\}_{k\in\mathbb{Z}}$ son autovectores de operadores discretos de convolución, vamos a enunciar un teorema de convolución discreta.

Teorema 3.6. Si $f\in\ell^1(\mathbb{Z})$ y $h\in\ell^1(\mathbb{Z})$, luego $g=f*h\in\ell^1(\mathbb{Z})$ y

$$\hat{g}(\omega)=\hat{f}(\omega)\hat{h}(\omega). \tag{0.28}$$

La demostración puede encontrarse en Mallat, 2009

3.3.6 La transformada Discreta de Fourier

El espacio de señales de período *N* es un espacio Euclideano de dimensión *N* y el producto interior de dos señales dadas *f* y *g* es

$$\langle f, g \rangle = \sum_{n=0}^{N-1} f[n] g^*[n]. \tag{0.29}$$

El teorema que sigue prueba que cualquier señal con periodo *N* puede descomponerse como una suma de ondas discretas sinusoidales.

Teorema 3.7. La familia

$$\left\{ e_k[n] = \exp\left(\frac{i2\pi kn}{N} \right) \right\}_{0 \le k < N}$$

es una base ortogonal del espacio de señales de período *N*.

Como el espacio es de dimensión *N*, cualquier familia ortogonal de *N* vectores es una base ortogonal. Con verificar que $\{e_k\}_{0 \le k < N}$ es ortogonal con respecto al producto interior (0.29) probamos este teorema. Cualquier señal *f* de período *N* puede descomponerse en esta base:

$$f = \sum_{k=0}^{N-1} \frac{\langle f, e_k \rangle}{\|e_k\|^2} e_k. \tag{0.30}$$

Por definición, la *transformada discreta de Fourier* (DFT –Discrete Fourier Transform) de *f* es

$$\hat{f}[k] = \langle f, e_k \rangle = \sum_{n=0}^{N-1} f[n] \exp\left(\frac{i2\pi kn}{N} \right). \tag{0.31}$$

Como $\|e_k\|^2 = N$, (0.30) proporciona una fórmula de la inversa discreta de Fourier:

$$f[n] = \frac{1}{N} \sum_{k=0}^{N-1} \hat{f}[k] \exp\left(\frac{i2\pi kn}{N} \right). \tag{0.32}$$

La ortogonalidad de la base también implica una fórmula de Plancherel:

$$\|f\|^2 = \sum_{n=0}^{N-1} \left| f[n] \right|^2 = \frac{1}{N} \sum_{k=0}^{N-1} \left| \hat{f}[k] \right|^2.$$

La DFT de una señal *f* de período *N* se calcula a partir de sus valores para $0 \le n < N$. A pesar de esto, es más importante considerar una señal periódica con período *N* que una señal finita con *N* muestras. La razón radica en la interpretación de los coeficientes de Fourier. En la suma discreta (0.31), donde se define una señal de período *N*, si $f[0]$ y $f[N-1]$ son muy diferentes entre sí, esto produce una transición brutal en la señal periódica, creando coeficientes de Fourier de relativamente alta amplitud a altas frecuencias, (Mallat, 2009).

3.3.7 Transformada Rápida de Fourier (FFT – Fast Fourier Transform)

Para una señal *f* de *N* puntos, un cálculo directo de *N* sumas discretas de Fourier

$$\hat{f}[k] = \sum_{n=0}^{N-1} f[n] \exp\left(\frac{-i2\pi kn}{N} \right), \quad 0 \le k \le N, \tag{0.33}$$

require N^2 multiplicaciones y sumas complejas. Reorganizando los cálculos, la FFT reduce el número de complejidad a $O(N \log_2 N)$. Cuando el índice de frecuencia es par, agrupamos los términos *n* y $n + N/2$:

$$\hat{f}[2k] = \sum_{n=0}^{N/2-1} \left(f[n] + f\left[n + N/2 \right] \right) \exp\left(\frac{-i2\pi kn}{N/2} \right). \tag{0.34}$$

Cuando el índice de frecuencia es impar, la misma agrupación se transforma en

$$\hat{f}[2k+1] = \sum_{n=0}^{N/2-1} \exp\left(\frac{-i2\pi n}{N} \right) \left(f[n] - f[n + N/2] \right) \exp\left(\frac{-i2\pi kn}{N/2} \right). \tag{0.35}$$

La ecuación (0.33) prueba que las frecuencias pares se obtienen calculando la transformada discreta de Fourier de la señal periódica con período $N/2$

$$f_e[n] = f[n] + f[n+N/2].$$

Las frecuencias impares se deducen de (0.35) calculando la transformada de Fourier de la señal periódica con período $N/2$

$$f_0[n] = \exp\left(\frac{-i2\pi n}{N}\right)\left(f[n] - f[n+N/2]\right).$$

Una DFT de tamaño N puede ser calculada entonces con dos transformadas discretas de Fourier de tamaño $N/2$ más $O(N)$ operaciones.

La FFT inversa de $\hat{f}$ se deduce de la FFT de sus complejos conjugados $\hat{f}^*$ teniendo en cuenta que

$$f^*[n] = \frac{1}{N}\sum_{k=0}^{N-1} \hat{f}^*[k]\exp\left(\frac{-i2\pi kn}{N}\right). \qquad (0.36)$$

3.4 Introducción a la Teoría Multitasa

Antes de describir un Banco de Filtros con decimación, vamos a definir algunos conceptos que serán necesarios para la comprensión del tema a tratar.

3.4.1 Algunos conceptos previos

Desarrollo en serie de Laurent

Debido a que trabajamos con señales discretas, para poder analizarlas con técnicas digitales, es necesario previamente realizar un muestreo de los valores que toma la función. Su tratamiento matemático requiere el uso de la denominada transformada

Zeta. Previa a su definición, demostraremos el desarrollo en serie de Laurent, que da la justificación matemática necesaria para comprender la transformada Zeta.

Todas las sucesiones $(X_n)_n$ en esta definición se suponen causales, es decir que si están definidas para algún subíndice *n* negativo los elementos correspondientes x_n son nulos.

Teorema 3.8: (del desarrollo en serie de Laurent) Si *f* en una función holomorfa, en el interior D de una corona circular limitada por dos circunferencias centradas en $a \in \mathbb{C}$, existen $c_n \in \mathbb{C}, n \in \mathbb{Z}$ tales que

$$f(z) = \sum_{n=-\infty}^{\infty} c_n (z-a)^n, z \in D. \tag{0.37}$$

La demostración puede encontrarse en (Sánchez Ruiz y Legua Fernández, 2006)

Transformada z

Definición 3.1: Se denomina Transformada Z de una sucesión de números complejos $(x_n)_{n\geq 0}$ y su notación es $Z\left[(x_n)_{n\geq 0}\right]$ o simplemente $Z[x_n]$, a la función de variable compleja $X(z)$ definida por la serie de Laurent

$$X(z) = Z[x_n] = \sum_{n=0}^{\infty} x_n z^{-n}.$$

Submuestreo o decimación

En la Figura 3.1 podemos observar el operador de submuestreo o decimación, con un factor *M* en este caso. Dada una señal discreta $x(n)$, el bloque de submuestreo es un bloque de procesamiento que consiste en tomar cada *M*-ésima muestra de $x(n)$, es decir que la nueva señal, $x_S(n)$, tiene una taza de muestreo igual a 1/*M*. La relación entonces entre ambas señales está dada por,

$$x_S(n) = x(Mn). \tag{0.38}$$

La transformada Z de la operación de submuestreo está dada por

$$X_S(z) = \frac{1}{M}\sum_{k=0}^{M-1} X(z^{\frac{1}{M}} W^k), \tag{0.39}$$

donde

$$W = e^{\frac{-j2\pi}{M}}. \tag{0.40}$$

Los componentes de la salida $X_S(z)$ son versiones cambiadas y estiradas de los componentes de la señal de entrada $X(z)$.

La operación de submuestrear una señal no es inversible y es posible que $x(n)$ no pueda ser recuperada a partir de $x_S(n)$. Supongamos un Banco de Filtros con $M = 2$. La operación de submuestreo estaría representada por (↓2). En la práctica significa que los componentes impares de la señal $x(n)$ se pierden. Por ejemplo:

Sea x(n) = [1, 2, 3, 0, 2, 1, 0, 3, 1, 2, 1, 0]; resulta que (↓2)x(n) = x_S(n); con x_S(n) = [2, 0, 1, 3, 2, 0].

La Figura 3.1 esquematiza el proceso de submuestreo del ejemplo mostrado.

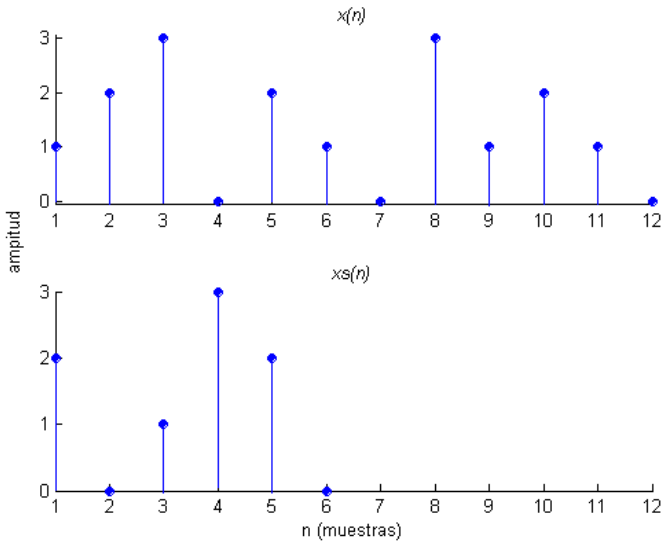

Figura 3.1. Operador de submuestreo. Toma una de cada *M* muestras de *x(n)* para obtener $x_D(n)$.

En el ejemplo, $M = 2$.

La recuperación de *x* a partir de $(\downarrow 2)x$ es posible si la transformada $X(\omega)$ es cero sobre una media banda de frecuencias. Esta señal tiene banda limitada y podría estar limitada en la parte superior o inferior, es decir,

$$X(\omega) = 0; \quad \text{con } 0 \le |\omega| \le \frac{\pi}{2}, \qquad \text{o, } X(\omega) = 0; \text{ con } \frac{\pi}{2} \le |\omega| \le \pi$$

Y esto surge del conocido Teorema del Muestreo de Shannon. Cuando la señal de entrada, *X(z)* en el dominio $\mathbb{Z}$, está limitada por la frecuencia π/M, no hay solapamiento en los componentes de los términos de (0.39). Es decir, no se produce aliasing en la señal submuestreada $X_S(z)$. Luego, para evitar este efecto no deseado sobre la señal submuestreada, en necesario limitar el ancho de banda de la señal de entrada *X(z)* por un factor *M*, si es que ésta no lo estuviera.

Sobremuestreo

El operador de sobremuestreo es el encargado de insertar *M*-1 ceros luego de cada muestra en la señal de entrada. Si partiéramos de una señal de entrada cualquiera y luego se aplica el operador de sobremuestreo de factor *M*, entonces no hay pérdida de

información en la señal original, pues simplemente aplicado luego un operador de submuestreo, de factor *M*, eliminamos los ceros introducidos en el sobremuestreo. En general, y también ocurre en los Bancos de Filtros, esta operación se da en el orden inverso, es decir primero se aplica el submuestreo y después el sobremuestreo.

La relación entre las señales de entrada $x(n)$ y salida del bloque $x_U(n)$ es:

$$x_U(n) = \begin{cases} x\left(\frac{n}{M}\right); & \text{si } n \text{ es un entero múltiplo de } M \\ 0; & \text{en otro caso} \end{cases} . \qquad (0.41)$$

La Figura 3.2 muestra el funcionamiento del operador para $M = 2$.

La transformada Z de la operación de sobremuestreo de la ecuación (0.41) está dada por

$$X_U(z) = X(z^M) \qquad (0.42)$$

La salida sobremuestreada $X_U(z)$ es un versión comprimida de la entrada $X(z)$.

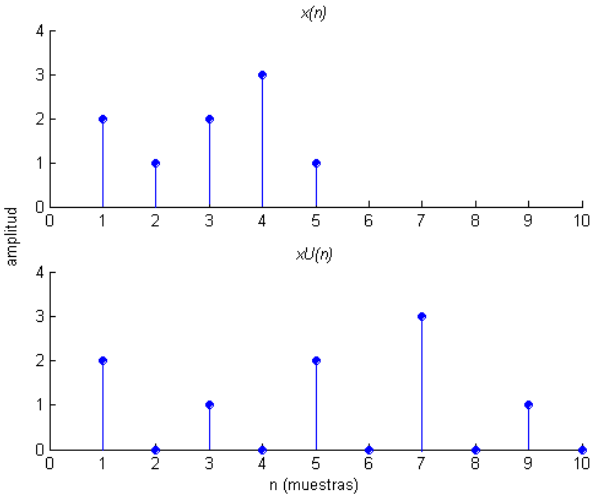

Figura 3.2 . Operador de sobremuestreo. Inserta *M*-1 ceros después de cada muestra de *x(n)* para obtener $x_U(n)$. En el ejemplo $M = 2$.

El operador de submuestreo, *diezmador*, genera aliasing. El operador de sobremuestreo, *expansor*, crea imágenes. Los filtros evitan que se produzca el alias y

remueven las imágenes. Un filtro previo al diezmador, limita el ancho de banda de la señal. Los términos de aliasing se hacen cero. Un filtro posterior al expansor, elimina las imágenes.

3.5 Análisis en Tiempo-Frecuencia. Conceptos Básicos de Onditas

Como ya lo mencionamos en el principio de este capítulo, la FT es la herramienta matemática más comúnmente utilizada en el análisis de señales. Los valores de la transformada representan la contribución de las funciones seno y coseno para cada frecuencia. Aplicando la TF a una función, puede analizarse su contenido espectral a partir de los coeficientes obtenidos. Al hacerlo, la función se integra sobre todo el tiempo, obteniendo la amplitud total de cada frecuencia sobre toda la señal. Dado que $e^{i\omega t}$ cubre toda la línea real, $F(\omega)$ depende de los valores de *f* para todo tiempo $t \in \mathbb{R}$

El contenido frecuencial de una señal discreta se obtiene por medio de la aplicación de la Transformada Discreta de Fourier (DFT). Esta operación se lleva a cabo por medio de algoritmos computacionales, utilizando la Transformada Rápida de Fourier (FFT).

La FT implica que la noción de tiempo desaparece luego de aplicada la transformación y el interés reside sobre el contenido en frecuencia de la señal, desconociendo el momento en que estos componentes aparecen. Su principal limitación es que no permite localizar en el tiempo eventos que forman parte de la señal, por lo que se hace difícil analizar propiedades locales de la señal a partir de su transformada. Si bien muchas aplicaciones pueden resolverse aplicando este tipo de operadores lineales, invariantes en el tiempo, existe una gran diversidad de señales biológicas de gran importancia que no son estacionarias. Por lo tanto, cuando se tiene interés en fenómenos transitorios, la Transformada de Fourier pierde validez.

Una alternativa propuesta por Dennis Gabor para solucionar el problema de la localización es el tipo de representaciones tiempo-frecuencia que se basa en el empleo de *ventanas temporales* caracterizadas por un centro y un ancho. Éstas toman una porción de la señal y permiten aplicar localmente la Transformada de Fourier. De este modo, se releva la información en frecuencia localizada temporalmente en el dominio efectivo de la ventana. Desplazando temporalmente la ventana se cubre el dominio de la

señal obteniéndose una completa información tiempo-frecuencia de la misma. La Transformada de Gabor puede entenderse como un tratamiento localizado de la señal mediante filtros pasabanda deslizantes, de ancho de banda constante. Así Gabor demostró la importancia del procesamiento localizado en tiempo – frecuencia. La ventana elegida por Gabor es una función gaussiana, debido a que su transformada también lo es. El inconveniente reside en que el ancho de la ventana se mantiene constante, limitando la precisión de la localización temporal. Así, los resultados obtenidos no son buenos cuando la señal está formada por componentes frecuenciales diferentes entre sí.

Cuando las señales tienen un amplio rango de componentes frecuenciales, una alternativa es utilizar ventanas de dimensión variable, ajustadas a la frecuencia de oscilación. Esta es la base de la Transformada Wavelet, que introduce una familia de funciones elementales mediante las dilataciones o contracciones y traslaciones en el tiempo de sólo una función básica $\psi(t)$. Esta función tiene la característica de ser finita en el tiempo, o de soporte acotado o bien que tiende a cero rápidamente, a diferencia del seno y coseno utilizados como bases de Fourier.

A partir de la ondita básica $\psi(t) \in \mathbf{L}^2(\mathbb{R})$, pueden generarse las demás funciones:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{(t-b)}{a}\right). \tag{0.43}$$

Una elección típica de ψ es $\psi(t) = (1-t^2)\exp(-t^2/2)$, la segunda derivada de la gaussiana, "llamada también sombrero mejicano". Esta función está bien localizada en tiempo y frecuencia.

Dado que estas funciones no cubren toda la recta, debe introducirse un parámetro *(b)* para trasladarlas por $\mathbb{R}$. Además, el parámetro de escala *a* ($a \neq 0$) permite las dilataciones y contracciones. A medida que *a* cambia, $\psi^{a,0}(s) = |a|^{-1/2} \omega(s/a)$ cubre diferentes rangos de frecuencia. Cambiando el parámetro *b* podemos movernos en la localización del tiempo: cada $\psi^{a,b}(s)$ está localizada alrededor de $s = b$. Las dilataciones y traslaciones generan una base ortonormal de $\mathbf{L}^2(\mathbb{R})$. Se preserva la energía de las funciones mediante un factor de normalización. Puede observarse que la

longitud de la ventana ya no es fija, sino que depende de *a*, (Daubechies, 1992), (D'Attellis y Fernandez Berdaguer, 1995).

Entonces, dada una señal $f(t)$, de energía finita, la *Transformada Wavelet Continua* se define como:

$$W_\psi f(a,b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{+\infty} f(t) \overline{\psi\left(\frac{(t-a)}{a}\right)} dt \tag{0.44}$$

y su antitransformada:

$$f(t) = \frac{1}{C_\psi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} W_\psi f(a,b) \frac{1}{\sqrt{|a|}} \psi\left(\frac{(t-a)}{a}\right) \frac{dadb}{a^2} \tag{0.45}$$

,

donde C_ψ es una constante positiva, definida por Calderon (Calderon, 1962), y que tiene la forma,

$$C_\psi = -\int_{-\infty}^{+\infty} \frac{|\hat{\psi}(\omega)|}{|\omega|} d\omega < \infty .$$

A través de la antitransformada, $f(t)$ queda expresada como una superposición de los vectores $\psi_{a,b}$.

En la aplicación práctica es común discretizar lo anterior. La discretización elegida es la diádica, restringiéndose a las escalas $a = 2^{-j}$ y a las traslaciones $b = k2^{-j}$, con $j,k \in \mathbb{Z}$. Reescribiendo las funciones ondita y la señal $f(t)$, se obtienen las siguientes fórmulas:

$$\psi_{j,k}(t) = 2^{\frac{j}{2}} \psi\left(2^j t - k\right) \tag{0.46}$$

$$f(t) = \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} d_j(k) \psi_{j,k}(t) \tag{0.47}$$

siendo $d_j(k)$ los *coeficientes de detalle* de la función.

Una base de Wavelets ortogonal descompone el eje de frecuencias en intervalos diádicos, cuyos tamaños tienen un crecimiento exponencial. Cada intervalo de

frecuencia es cubierto por los rectángulos tiempo – frecuencia de funciones Wavelet, que se trasladan uniformemente en tiempo con el objeto de cubrir todo el plano. La descomposición de una señal discreta utilizando la Transformada Wavelet Discreta (TWD) se logra por medio de la implementación de un banco de filtros que surgen a través del análisis de multirresolución. El análisis de multirresolución permite la aproximación de señales en varios niveles de resolución con proyecciones ortogonales en diferentes subespacios $\left(V_j\right)_{j\in\mathbb{Z}}$. La aproximación de una función en la resolución 2^{-j} está definida como una proyección ortogonal en un subespacio $V\in\mathbf{L}^2\left(\mathbb{R}\right)$. El subespacio V_j contiene todas las aproximaciones posibles en la resolución 2^{-j}. La proyección ortogonal de f es la función $f_j\in V_j$ que minimiza $\left\|f-f_j\right\|$. Estos subespacios tienen las siguientes propiedades:

i) $\cdots\subset V_2\subset V_1\subset V_0\subset V_{-1}\subset V_{-2}\subset\cdots$

ii) $\bigcap_{j\in\mathbb{Z}}V_j=\{0\};\quad\overline{\bigcup_{j\in\mathbb{Z}}V_j}=\mathbf{L}^2\left(\mathbb{R}\right);$

iii) $f\in V_j\Leftrightarrow f\left(2^j\cdot\right)\in V_0$

iv) $f\in V_0\Rightarrow f\left(\cdot-n\right)\in V_0;\quad\forall n\in\mathbb{Z}.$

v) $\exists\phi\in V_0$ tal que $\phi_{o,n}\left(x\right)=\phi\left(x-n\right)$ constituye una base ortonormal para V_0.

Los espacios Wavelets son complementos ortogonales a V_j en V_{j-1} y contienen la información necesaria para ir desde 2^{-j} hasta $2^{-(j-1)}$.

La Ondita madre $\psi\left(t\right)$ está construida a partir de las funciones de escala de forma tal que el conjunto $\psi\left(t-k\right)$ es una base ortonormal de w_0.

$$\varphi\left(t\right)=\sum_{k=-\infty}^{+\infty}p_k\varphi\left(2t-k\right) \tag{0.48)}$$

$$\psi\left(t\right)=\sum_{k=-\infty}^{+\infty}q_k\varphi\left(2t-k\right) \tag{0.49}$$

donde p_k y q_k gobiernan la estructura del subespacio. La ecuación (0.48) representa la relación de dos escalas de $\phi(t)$.

Para cualquier valor de escala *l* podemos escribir

$$\varphi(2t-l)=\sum_{k=-\infty}^{+\infty}\left(a_{l-2k}\varphi(t-k)+b_{l-2k}\psi(t-k)\right);\quad l\in\mathbb{Z}, \tag{0.50}$$

claramente, a_k y b_k gobiernan la descomposición.

La función $f(t)$ puede expandirse en términos de la escala $j-1$ a la luz de la re relación general entre las escalas, sumado a que $V_j = V_{j-1}\oplus W_{j-1}$. Luego,

$$f_j(t)= f_{j-1}+g_{j-1}, \tag{0.51}$$

$$f=\sum_{k=-\infty}^{+\infty}c_{k,j}\varphi_{k,j}=\sum_{k=-\infty}^{+\infty}c_{k,j-1}\varphi_{k,j-1}+\sum_{k=-\infty}^{+\infty}d_{k,j-1}\psi_{k,j-1}, \tag{0.52}$$

con c_k, d_k componentes $f_{j-1}(t), g_{j-1}(t)$ de $f_j(t)$ y se utilizan como las secuencias de datos de entrada o salida en los algoritmos de procesamiento (Juli, Portu, 2008).

$$c_{k,j-1}=\sum a_{l-2k}c_{i,j} \tag{0.53}$$

$$d_{k,j-1}=\sum_{l=-\infty}^{+\infty}b_{l-2k}c_{i,j} \tag{0.54}$$

Los coeficientes de aproximación en (0.53) y detalle en (0.54) de un nivel dado se calculan como la convolución de los coeficientes de aproximación del nivel inmediato superior, con los coeficientes a_k y b_k característicos de la función base wavelet respectivamente y luego tomando los coeficientes pares del resultado ($l\rightarrow 2l$). Esta operación es la denominada submuestreo, Figura 3.1 y se simboliza como $\downarrow 2$ que ya fuera mencionada.

Aplicando recursivamente *N* veces estos pasos se obtiene una descomposición de *N* niveles de *f*:

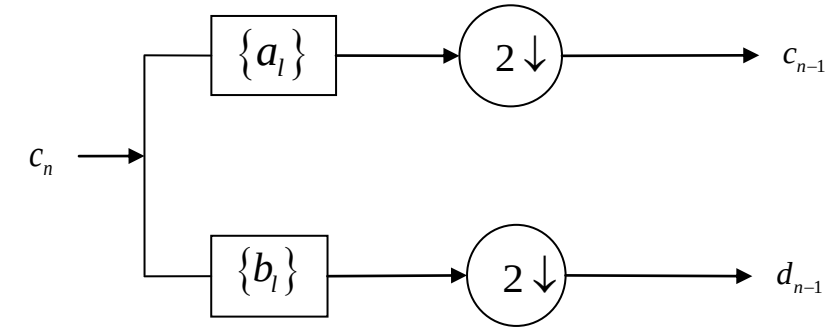

Figura 3.3 Descomposición de la ondita. Diagrama esquemático

Para realizar el camino inverso, teniendo c_{j-1} y d_{j-1} podemos reconstruir el coeficiente *c j* :

$$c_{k,j} = \sum_{l=-\infty}^{+\infty} \left(c_{l,j-1} p_{k-2l} + d_{l,j-1} q_{k-2l} \right) \tag{0.55}$$

que se puede interpretar como un sobremuestreo ($2l \rightarrow l$) , mencionado también con anterioridad. Este proceso está seguido de una convolución con los coeficientes de reconstrucción. Esta operación (Figura 3.4), se simboliza como $\uparrow 2$.

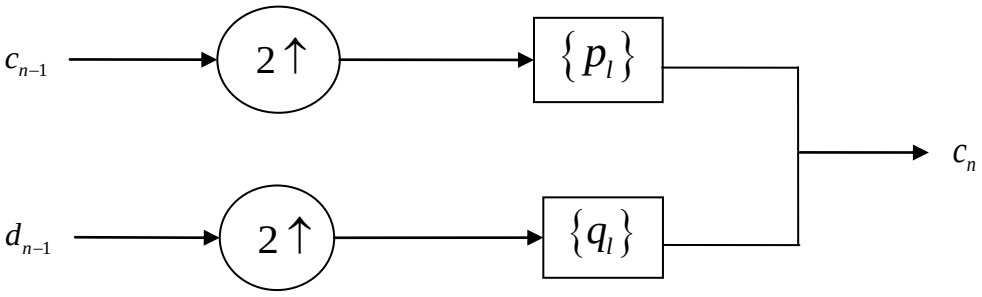

Figura 3.4 Composición de la ondita. Diagrama esquemático

La Figura 3.5 es una combinación de la Figura 3 y Figura 3.4. Obtenemos de este modo una estructura de banco de filtros (Chui, 1997).

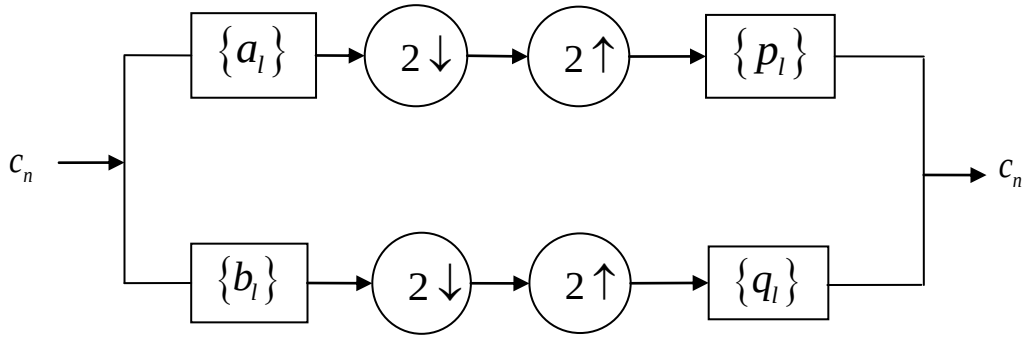

Figura 3.5 Descomposición – Reconstrucción. La señal c_n es recuperada en su totalidad.

3.6 Bancos de Filtros Digitales

3.6.1 ¿Qué es un Banco de Filtros Digitales?

Por definición, un *Banco de Filtros* (FB) es un conjunto de filtros, pasa bajos y pasa altos, cada uno de los cuales cubre un ancho de banda en el espectro de las frecuencias. Otros elementos que forman parte del FB son los operadores de muestreo (sub-muestreo y sobre-muestreo) y operadores de retardo. En la Figura 3.6, se muestra un FB de M canales con sub-muestreo o decimación. El FB cuenta con dos etapas bien definidas, la primera, llamada *análisis*, donde la señal de entrada $x(n)$ pasa a través del banco de M filtros de análisis $H_i(z)$.

Los *M* filtros de análisis son filtros pasa bajos y pasa altos. Juntos son capaces de separar la señal en bandas de frecuencias uniformes, de ancho *π/M*. La dificultad que se produce, es que la longitud de la señal se ve aumentada en un factor *M* y los datos resultan redundantes. La solución que se encontró fue sub-muestrear la señal. Es decir que luego de la etapa de *análisis*, deviene la etapa de *sub-muestreo*. La reducción de la tasa de muestreo se llama *decimación*. La decimación está compuesta por dos etapas: filtrado y submuestreo. Las señales de sub-bandas resultantes pueden ser luego procesadas en el bloque *procesamiento*, juntas o en forma independiente. Tal bloque podría no considerarse como parte del FB propiamente dicho.

En la etapa de *síntesis*, las sub-bandas son nuevamente combinadas por un bloque de *sobre-muestreo* o interpolación y nuevamente *M* filtros $F_i(z)$ pasa bajos y pasa altos reconstruyen la señal $\hat{x}(n)$. Si consideramos que los filtros son ideales, $\hat{x}(n) = x(n)$, es decir que no se produce solapamiento y podemos decir que se logró la reconstrucción perfecta (PR –Perfect Reconstruction), característica buscada en todos los FB, y que será luego descripta en detalle. El rasgo de PR de los Bancos de Filtros se refiere al hecho de producirse el retardo como única diferencia entre la entrada y la salida. En la práctica esto no ocurre, es por ello que la elección de los coeficientes de los filtros resulta de vital importancia para que la señal $\hat{x}(n)$ sea lo más cercana a $x(n)$ posible.

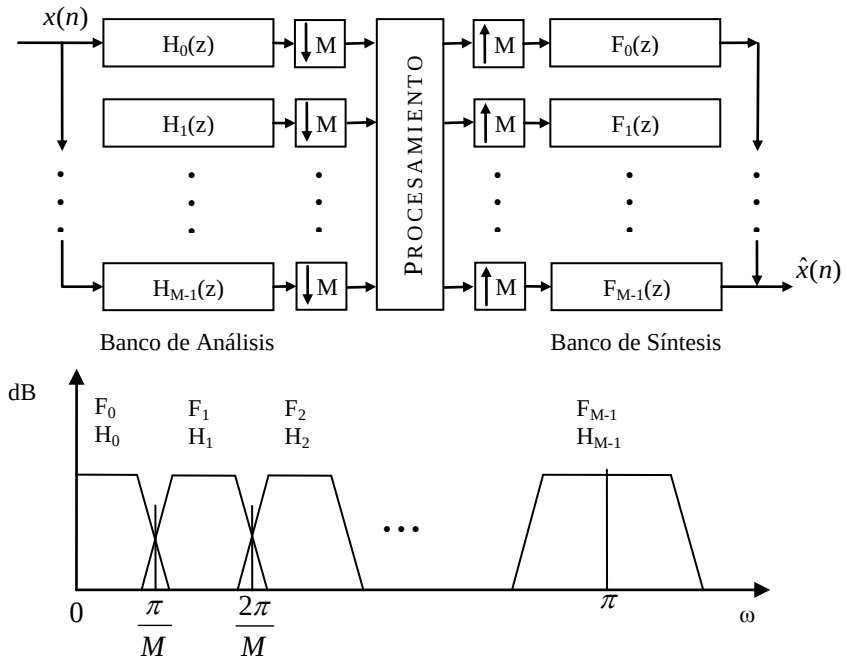

Figura 3.6 Banco de Filtros de M canales con sub-muestreo y sobre-muestreo. Respuesta de magnitud ideal de los filtros

3.6.2 El Banco de Filtros Identidad

Para comprender el funcionamiento de un Banco de Filtros Digitales, describiremos el funcionamiento de un Banco de Filtros de 2 canales, y su extensión a M canales será directa.

Los comienzos de la teoría de Bancos de Filtros fueron a partir del banco de filtros de la Figura 3.7 a los principios de los ’80. Con sólo dos canales, este FB con reconstrucción perfecta, ha sido estudiado y caracterizado con diferentes métodos de diseño basados en factorización espectral, estructura de red, optimización en el dominio del tiempo, etc, (Tran, 1999).

El diseño de Bancos de Filtros con 2 canales qué más alcance ha tenido es el diseño de *factorización espectral*, que constituye un diseño simple conceptualmente hablando. Su explicación también es sencilla y es la siguiente:

Si examinamos la función transferencia del Banco de Filtros podemos deducir la relación entre la salida $\hat{X}(z)$ y la entrada $X(z)$ dada por:

$$\hat{X}(z)=\frac{1}{2}\left[F_0(z)H_0(z)+F_1(z)H_1(z)\right]X(z)$$
$$+\frac{1}{2}\left[F_0(z)H_0(-z)+F_1(z)H_1(-z)\right]X(-z).$$

La reconstrucción perfecta con ℓ retardos puede obtenerse con filtros no ideales si $\hat{X}(z)=z^{-\ell}X(z)$, sujeto a las condiciones siguientes, que resultan intuitivas:

$$F_0(z)=H_0(-z)+F_1(z)H_1(-z)=0\text{. Cancelación del solapamiento} \quad (0.56)$$
$$F_0(z)H_0(z)+F_1(z)H_1(z)=2z^{-\ell}\text{. Eliminación de la distorsión.} \quad (0.57)$$

La elección de $F_0(z)=H_1(-z)$ y $F_1(z)=-H_0(-z)$ cancela totalmente el solapamiento y reduce la ecuación (0.57) a un filtro pasa bajos de media banda $P_0(z)\triangleq F_0(z)H_0(z)$ pues la eliminación de la distorsión luego se simplifica a

$$P_0(z)-P_0(-z)=2z^{-\ell}. \quad (0.58)$$

Luego, el diseño de un banco de filtros de 2 canales puede ser realizado en dos pasos:

1.- Diseño de un filtro pasa bajos que satisface la ecuación (0.58).

2.- Factorizar $P_0(z)$ para obtener $H_0(z)$ y $F_0(z)$. Luego, $H_1(z)$ y $F_1(z)$ se eligen de manera que cancelen el solapamiento descripto más arriba.

Podemos obtener los filtros de fase lineal si el filtro pasa bajos de media banda $P_0(z)$ se elige de tal manera que tenga fase lineal y entonces la factorización espectral se produce apropiadamente.

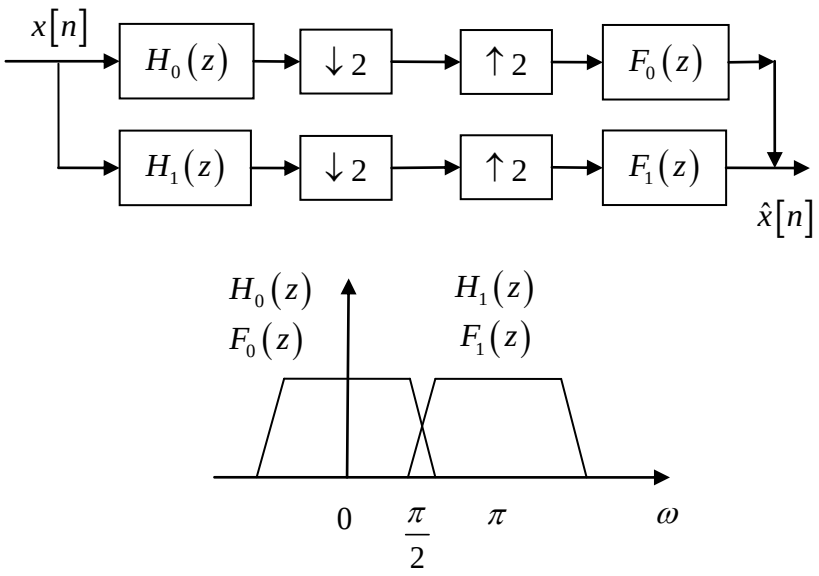

Figura 3.7 Banco de Filtros de 2 canales

3.6.3 Representación Polifásica

Antes de introducirnos en esta característica importante de los Bancos de Filtros con decimación, es necesario definir las identidades nobles. Estas igualdades (Figura 3.8), permiten obtener alguna simplificación adicional en términos de las operaciones realizadas (Tran, 1999).

En la Figura 3.8 puede observarse que los bloques de submuestreo y sobremuestreo aparecen luego del filtrado en los términos de la derecha del signo de equivalencias. Esto produce que el banco de filtros de la Figura 3.9(a) se transforme en el banco de filtros de la Figura 3.9(b), la cual es una estructura más útil, tanto desde el punto de vista teórico como práctico.

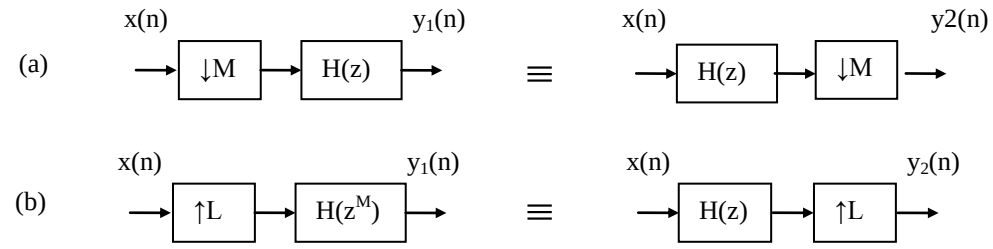

Figura 3.8 Identidades Nobles. Permiten la aplicación de la representación polifásica

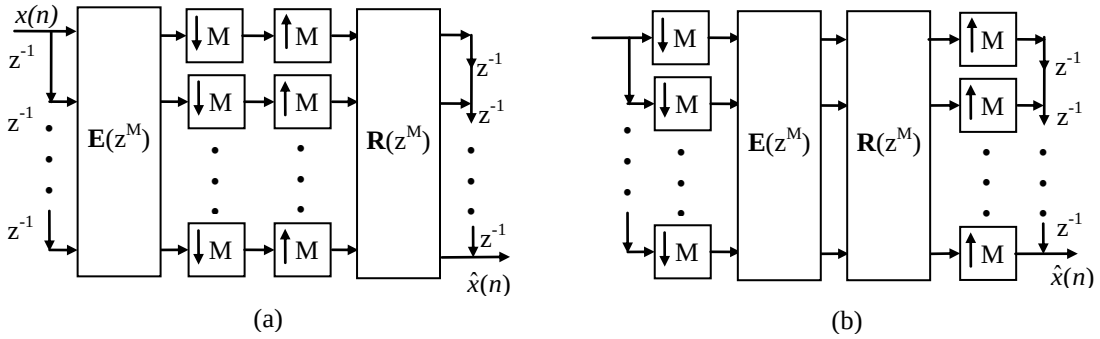

Figura 3.9 Representación Polifásica equivalente de un Banco de Filtros de *M* canales. (a) Puede ser reemplazado por el Tipo I y II de representación polifásica y obtener (b) para una aplicación más eficiente

La representación Polifásica de los filtros digitales ha contribuido significativamente al diseño, implementación y análisis teórico de sistemas de múltiples velocidades (Strang and Nguyen, 1996) y es especialmente útil en la implementación computacional de estructuras operacionales como la decimación e interpolación. Si consideramos la expresión general de la transformada Z de la respuesta al impulso de un filtro digital:

$$H(z) = \sum_{n=-\infty}^{\infty} h(n) z^{-n} \, . \tag{0.59}$$

Podríamos separar los términos pares e impares para producir la siguiente expresión:

$$H(z) = \sum_{n=-\infty}^{\infty} h(2n) z^{-2n} + \sum_{n=-\infty}^{\infty} h(2n+1) z^{-2n} \, . \tag{0.60}$$

Ambos términos de la suma pueden ser representados por las siguientes definiciones:

$$E_0(z) = \sum_{n=-\infty}^{\infty} h(2n) z^{-n} \ , \tag{0.61}$$

$$E_1(z) = \sum_{n=-\infty}^{\infty} h(2n+1) z^{-n} \ . \tag{0.62}$$

Se definen así dos componentes polifásicos de *H(z)*. Ahora *H(z)* puede ser escrito como la suma de ambas componentes

$$H(z) = E_0(z^2) + z^{-1} E_1(z^2). \tag{0.63}$$

En forma más general, *H(z)* puede ser caracterizado equivalentemente por sus *M* componentes polifásicos, para cualquier *M* entero positivo:

$$H(z) = \sum_{l=0}^{M-1} z^{-l} E_l(z^M), \tag{0.64}$$

donde

$$E_l(z) = \sum_{n=-\infty}^{\infty} h(nM + l) z^{-n} \ ; \qquad 0 \le l \le M-1 \tag{0.65}$$

La expresión de la ecuación (0.63) es conocida como la *representación polifásica de Tipo I*. Otra representación polifásica de *H(z)* está dada por:

$$H(z) = \sum_{l=0}^{M-1} z^{-(M-1-l)} R_l(z^M) \tag{0.66}$$

con $R_l(z) = E_{M-1-l}(z)$.

La representación polifásica permite el procesamiento de señales a tasas más bajas y simplifica la teoría de banco de filtros. A partir de estos conceptos podemos definir algunos aspectos matemáticos acerca de los FB más precisos y que contribuirán a la aplicación de los mismos.

3.6.4 Reconstrucción Perfecta

Definición 3.2: Se dice que un banco de filtros tiene reconstrucción perfecta si sus matrices polifásicas R(z) y E(z) satisfacen la siguiente ecuación (Tran, 99):

$$\mathbf{R}(z)\mathbf{E}(z) = z^{-l}\mathbf{I}\,; \quad l \in \mathcal{Z} \tag{0.67}$$

Si observamos la Figura 3.9 y consideramos que $\mathbf{R}(z)\cdot\mathbf{E}(z) = \mathbf{I}$, entonces la salida $\hat{x}(n)$ es simplemente una versión retrasada de $x(n)$. Luego la Reconstrucción Perfecta estaría garantizada si la matriz polifásica **E**(z) es invertible. Si bien la definición 3.2 nos da una idea de cuál es la principal restricción para diseñar un FB con PR, es todavía un concepto muy general y a los fines prácticos es necesario imponer aun más condiciones. Para obtener bancos de filtros con filtros FIR en el banco de análisis y en el banco de síntesis, el determinante de cada una de las matrices polifásicas debe ser un retraso (Strang and Nguyen, 1997).

$$|\mathbf{E}(z)| = z^{-n} \quad \text{y} \quad |\mathbf{R}(z)| = z^{-m}; \quad m,n \in \mathcal{Z} \tag{0.68}$$

Es decir, los determinantes de las matrices debes ser monomios, (Tran, 1999).

3.6.5 Propiedad Paraunitaria

Una matriz **E**(z) es paraunitaria si satisface

$$\tilde{\mathbf{E}}(z)\mathbf{E}(z) = \mathbf{I} \tag{0.69}$$

donde $\tilde{\mathbf{E}}(z) = \mathbf{E}_*^T(z^{-1})$, $\mathbf{E}^T(z)$ es la transpuesta de la matriz $\mathbf{E}(z)$, y $\mathbf{E}_*(z)$ la conjugación de sus coeficientes.

Las ventajas que se pueden mencionar con respecto al diseño de $\mathbf{E}(z)$ paraunitaria es que luego el BF tendrá la propiedad de RP, si bien ésta no es una condición necesaria. Otra ventaja muy importante es el hecho de que los filtros de síntesis tienen la misma longitud que los filtros de análisis y éstos pueden ser obtenidos simplemente por la inversión del tiempo y la conjugación de los coeficientes del filtro de análisis. La

matriz paraunitaria **E**(z) se puede implementar de manera que tenga un costo bajo de complejidad computacional de los filtros de análisis y que aseguren además una rápida convergencia.

3.6.6 Bancos de filtros de *M* canales

Nuestro foco de estudio es un banco de filtros de M canales con filtros de fase lineal y con la característica de reconstrucción perfecta. Sus ventajas son muchas y su potencial es enorme. Ejemplos de estos bancos de filtros son la transformada discreta de coseno (DCT –Discrete Cocine Transform) y la transformada ortogonal superpuesta (LOT –Lapped Orthogonal Transform). Un banco de filtros de *M* canales provee una partición en el espectro de las frecuencias sin cometer faltas en el retardo o en la performance de los filtros. El aumento en los grados de libertad permiten a los bancos de filtros de *M* canales tener mayor flexibilidad para adaptarse a una clase particular de señales o una clase particular de aplicación, (Tran, 1999). Los bancos de filtros pueden satisfacer simultáneamente las características de fase lineal y paraunitario, mientras que los sistemas de 2-canales no. Para finalizar, un banco de filtros de *M* canales con fase lineal y reconstrucción perfecta da lugar a las onditas ortogonales o biortogonales de *M* canales.

3.6.7 Wavelets y su relación con Bancos de Filtros

Las wavelets son ondas localizadas. La transformada Wavelet está relacionada con los Bancos de Filtros debido a que puede ser obtenida por iteración de un banco de filtros de 2 canales con reconstrucción perfecta a la salida del filtro pasa bajos, (Figura 3.10).

Las funciones de escala y wavelets heredan la ortogonalidad, o biortogonalidad del banco de filtros. Las wavelets y las funciones de escala son el resultado de aplicar repetidamente un re-escalonamiento, por ello las wavelets descomponen una señal en detalles en todas las escalas. Las funciones biortogonales $\tilde{\phi}(t)$ y $\tilde{\psi}(t)$ vienen de iterar el banco de síntesis (Transformada Inversa).

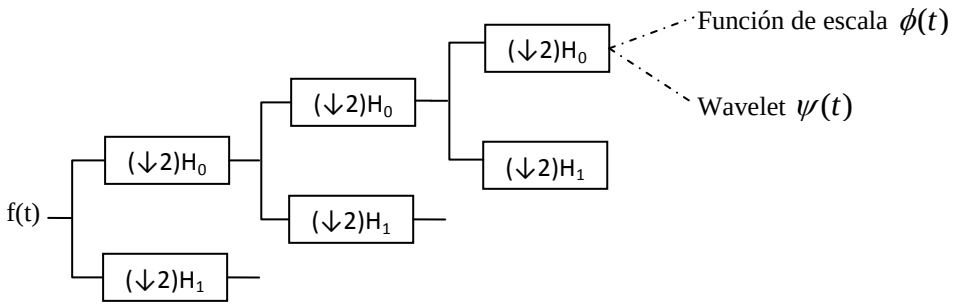

Figura 3.10 Función de escala y wavelets a partir de la iteración de filtros pasa bajos

La transformada wavelet puede ser pensada como una descomposición multiresolución de la señal en sus componentes más gruesos y de detalle. El campo del diseño de los bancos de filtros introdujo nuevas consideraciones a partir de la idea de la iteración que se incorporó con las wavelets. Un filtro que no fue diseñado con la idea de la iteración de las wavelets en mente hace que la función de escala y función wavelet rápidamente diverjan en pocas iteraciones (Strang, Nguyen, 1996). Daubechies demostró que imponiendo *N momentos nulos* (Daubechies, 1992)

$$\int t^k \psi(t) dt = 0; \text{ con } 0 \le k \le N-1 \tag{0.70}$$

Esta condición es equivalente a imponer un *N*-ésimo orden cero $(N \le L-1)$ en $(\pi = \omega)$ en el filtro pasa bajos

$$\sum_n (-1)^n n^k h_0(n) = 0; \tag{0.71}$$

con $k \in \mathbb{Z}; \quad 0 \le k \le N-1$

En el dominio de las frecuencias, la transformada wavelet emula una partición no uniforme en *M*-bandas del espectro de frecuencias (Figura 3.10).

A partir de las aplicaciones realizadas a señales de ECG, podemos decir que los Bancos de Filtros Digitales resultaron ser más convenientes y fáciles de implementar que las Wavelets, por ello optamos por lo primero.

3.7 Transformadas Superpuestas

La idea de las transformadas superpuestas (LT –Lapped Transforms) que mantengan la ortogonalidad y la no expansión de las muestras fue desarrollada a comienzos de los ’80 en el MIT por un grupo de investigadores disconformes con los artefactos de bloques tan comunes en las transformadas de bloque tradicionales en la codificación de imágenes. El objetivo principal fue extender la función de las bases más allá de los límites del bloque, creando una superposición, para eliminar justamente el efecto del bloque mencionado anteriormente. Esto no fue lo novedoso, sino que lo que hicieron con la misma cantidad de coeficientes de la transformada que en el caso en que no se producía la superposición y la transformada mantendría la ortogonalidad. Cassereu introdujo la Transformada Ortogonal Superpuesta (LOT -) y Malvar diseñó la estrategia y el algoritmo rápido de resolución. Más tarde, Malvar demostró la equivalencia entre una LOT y un banco de filtros multitasa.

Basadas en los bancos de filtros de coseno modulado, se diseñaron las transformadas moduladas superpuestas, las que fueron generalizadas por una superposición arbitraria, creando la clase de transformadas superpuestas extendidas (ELT -). Posteriormente, se desarrolló una nueva clase de LT con base simétrica produciendo las LOT Generalizadas o GenLOT.

Como dijimos, los bancos de filtros y las LTs son lo mismo, aunque fueron estudiados en el pasado en forma separada, (Queiroz, 1996).

Las LTs son bancos de filtros paraunitarios, uniformes y con filtros de fase lineal y respuesta al impulso finita.

Describiremos en este apartado la transformada DCT que dio origen a la LOT, la que fue utilizada para en diseño de nuestro Banco de Filtros.

3.7.1 La Transformada Discreta de Coseno (DCT)

Los coeficientes X de la DCT y los valores de muestra de la señal de entrada x están relacionados por la siguiente ecuación:

$$X[m] = \alpha_m \sum_{n=0}^{M-1} x[n] \cos\left[\frac{\pi m}{2M}(2n+1)\right], \quad 0 \le m \le M-1 \tag{0.72}$$

con

$$\alpha_m = \begin{cases} \sqrt{1/M}, & m = 0 \\ \sqrt{2/M}, & 1 \le m \le M-1. \end{cases}$$

Desde el punto de vista de un Banco de Filtros, la DCT es el Banco de Filtros de *M* canales con fase lineal y reconstrucción perfecta más básico. Su matriz de polifase tiene orden 0 (es independiente de z) y puede escribirse de la siguiente forma:

$$\mathbf{E}_0 = \frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{U}_0 & \mathbf{U}_0\mathbf{J} \\ \mathbf{V}_0\mathbf{J} & -\mathbf{V}_0 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{U}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_0 \end{bmatrix}\begin{bmatrix} \mathbf{I} & \mathbf{J} \\ \mathbf{J} & -\mathbf{I} \end{bmatrix}$$

$\mathbf{E}_0$ es ortogonal sólo si $\mathbf{U}_0$ y $\mathbf{V}_0$ lo son. Para representar la DCT mediante $\mathbf{E}_0$ necesitamos dos matrices ortogonales. Todas las funciones de la base de la DCT tienen fase lineal; la mitad de ellas son simétricas, aquellas asociadas a $\mathbf{U}_0$ y aquellas asociadas a $\mathbf{V}_0$, la otra mitad, son antisimétricas. Además de esto, al tener coeficientes reales y un algoritmo de implementación rápido, pone especial énfasis en el espectro de frecuencias pasa bajos. Provee mayor compactación de energía sobre todas las transformadas clásicas tales como la FFT. Sus características más importantes podrían resumirse como:

- Es independiente de la señal.
- Tiene fase lineal.
- Coeficientes reales.
- Un algoritmo rápido.

3.7.2 La Transformada Ortogonal Superpuesta (LOT-Lapped Orthogonal Transform)

La transformada ortogonal superpuesta, popularizada por Malvar (Malvar 1989 y 1990), está especialmente diseñada para solucionar los problemas de las transformadas

de bloques tradicionales. Por ejemplo en la DFT (Transformada Discreta de Fourier), las señales son particionadas en bloques que no se superponen y luego procesados con los filtros en la DFT. Cuando se reconstruye la señal mediante la antitransformada, aparecen discontinuidades en la señal reconstruida. Las LOTs son expansiones en bases ortonormales y son computacionalmente eficientes debido a que pueden aplicarse con bancos de filtros. Tienen bases de vectores de soporte solapado para eliminar los artefactos de la separación en bloque. (Sandryhaila, 2010).

Como mencionamos previamente, un banco de filtros con decimación, reduce el número de muestras y logra así el procesamiento de la señal a una tasa más baja. Esta transformación debería ser invertible para que la señal original pueda ser reconstruida a partir de los coeficientes de la transformada –propiedad conocida como reconstrucción perfecta. Si hablamos de condiciones ideales, esto se logra mediante una transformación unitaria para que la energía de la señal permanezca sin cambios y para que la diferencia de unidad de ruido blanco gaussiano se transforme en diferencia de unidad de ruido blanco gausiano (Kunz, 2008).

(Sandryhaila, 2010) (Nesbit, 2009) (Kunz, 2008) muestran algunos ejemplos de aplicación de transformadas ortogonales superpuestas. La LOT tiene la ventaja de que los filtros de análisis y síntesis del FB pueden ser calculados eficientemente usando algoritmos de rápida implementación y cálculo (Afonso, 2000). En particular, la que utilizamos para computar los coeficientes del banco de filtros que diseñamos tiene las siguientes características.

Sea **P**, una matriz de la LOT en la cual cada columna es una de las funciones de síntesis en el FB. La matriz LOT representa *M* funciones base cada una de longitud *2M*.

La ortogonalidad de las funciones base de ***P*** es importante, no sólo porque deben ser ortogonales a ellas mismas, sino también a las funciones en dos solapamientos adyacentes del filtro cuando opera en la señal. Ambos requerimientos pueden ser representados como

$$\mathbf{P}\mathbf{P}' = \mathbf{I} \tag{0.73}$$

y

$$\mathbf{P}'\mathbf{W}\mathbf{P}=\mathbf{0} \tag{0.74}$$

donde

$$\mathbf{W}=\begin{bmatrix}\mathbf{0} & \mathbf{I}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}$$

Condiciones que son satisfechas si

$$\mathbf{P}=\frac{1}{2}\begin{bmatrix}\mathbf{D}_e-\mathbf{D}_o & \mathbf{D}_e-\mathbf{D}_o\\ \mathbf{J}(\mathbf{D}_e-\mathbf{D}_o) & -\mathbf{J}(\mathbf{D}_e-\mathbf{D}_o)\end{bmatrix}\mathbf{Z} \tag{0.75}$$

donde ***J*** es la matriz contra-identidad. $\boldsymbol{D_e}$ y $\boldsymbol{D_o}$ son las transformadas discreta de coseno (DCT –discrete cosine transform) pares e impares respectivamente. La *n*-ésima fila y la *k*-ésima columna de estas matrices *M* x *M/2* están dadas por

$$[\mathbf{D}_e]_{nk}=c(k)\sqrt{\frac{2}{M}}\cos\left(\frac{\pi}{M}2k\left(n+\frac{1}{2}\right)\right), \tag{0.76}$$

$$[\mathbf{D}_o]_{nk}=\sqrt{\frac{2}{M}}\cos\left(\frac{\pi}{M}(2k+1)\left(n+\frac{1}{2}\right)\right), \tag{0.77}$$

con $0\le n\le M-1$ y $0\le k\le\frac{M}{2}-1$, donde

$$c(k)=\begin{cases}1/\sqrt{2} & k=0\\ 1 & \text{en otro caso}\end{cases} \tag{0.78}$$

Z está dada por

$$\mathbf{Z}=\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{C}^{II}_{M/2}\mathbf{S}^{IV}_{M/2}\end{bmatrix}\mathbf{R}, \tag{0.79}$$

donde

$$\left[\mathbf{C}^{II}_{K}\right]_{kr}=c(k)\sqrt{\frac{2}{K}}\cos\left(\frac{\pi}{K}k\left(r+\frac{1}{2}\right)\right),$$

$$\left[\mathbf{S}_K^{IV}\right]_{kr} = \sqrt{\frac{2}{K}}\sin\left(\frac{\pi}{K}\left(k+\frac{1}{2}\right)\left(r+\frac{1}{2}\right)\right),$$

con *c(k)* según ecuación (0.78). ***R*** en (0.79) es una matriz permutación.

Los filtros análisis y síntesis diseñados en este trabajo fueron computados con esta transformada superpuesta recién descripta. La implementación computacional de la LOT fue realizada en MATLAB®, de manera eficiente y con un mínimo de multiplicaciones y sumas en el proceso de filtrado.

Desde el punto de vista de un FB, la transformada ortogonal superpuesta es justamente un banco de filtros paraunitario de canales pares con fase lineal y longitud de filtros $L = 2M$. En la Figura 3.11 puede observarse la respuesta en frecuencia y al impulso de la LOT para $M = 8$.

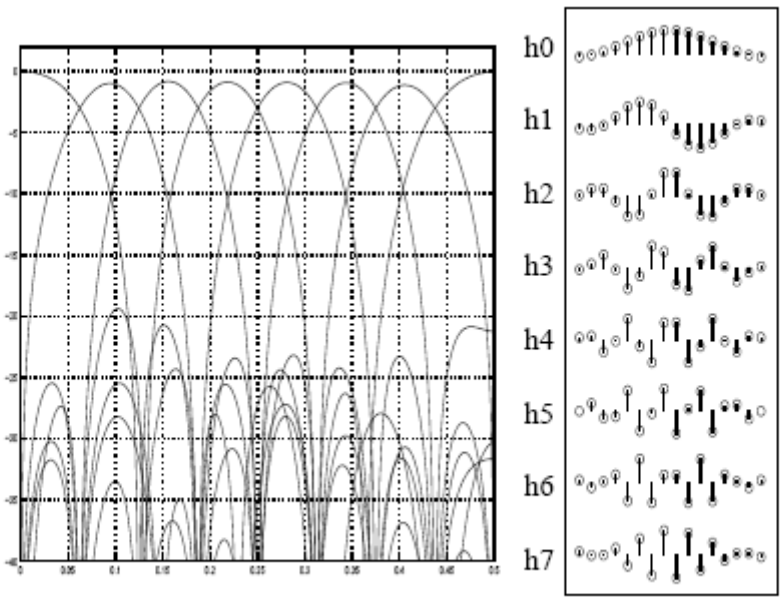

Figura 3.11 Respuesta en frecuencia y al impulso de la LOT para M = 8

4. Análisis de Componentes Principales

4.1 Introducción

En la constante búsqueda del método automático que mejor clasifique los latidos cardíacos, ya sea un algoritmo que contemple los aspectos temporales o morfológicos de la señal, es inevitable la acumulación de gran cantidad de datos. Como solución a este problema surgen técnicas estadísticas capaces de reducir el número de variables necesarias. Una consecuencia directa es la obtención de datos normalizados y procesos más eficientes que agilizan los tiempos computacionales y además economizan en espacio de almacenamiento, (Pisarello et al, 2006).

El análisis y la clasificación de los latidos cardíacos pueden realizarse a partir de estudios temporales y morfológicos de la señal electrocardiográfica. En general, para lograr la caracterización de la forma de un complejo QRS por medio de sus puntos distinguibles, es inevitable la acumulación de gran cantidad de datos que no siempre resultan imprescindibles para obtener resultados certeros (Moody and Mark, 1989).

El *Análisis de Componentes Principales* o *PCA (Principal Component Analysis)*, es una técnica estadística de síntesis de la información o reducción de la dimensión (número de variables). En bancos de datos de muchas variables, la técnica de PCA permite reducir el número de tales variables, sin perder información substancial. Los nuevos *factores* o *componentes* serán una combinación lineal de las variables originales, e independientes entre sí.

El Análisis de Componentes Principales tiene sus antecedentes en la Psicología mediante las técnicas de regresión lineal. Fue Pearson en 1901 quien presentó la primera propuesta formal del *método* de componentes principales. A pesar de esto, el nombre de *Componentes Principales* se lo debemos a Hotelling, que en 1933 publicó el artículo "Analysis of a complex of statistical variables into principal components" en *Journal of Educational Psychology*, donde desarrolló la teoría que lo sustenta y diseñó un método para la extracción de factores.

El PCA se empleó inicialmente en la psicología, las ciencias sociales y naturales. Sin embargo, desde hace ya algunos años se ha extendido su aplicación a las ciencias físicas, la ingeniería, la economía, el reconocimiento de patrones, la compresión de datos, etc.

Nuestra exposición estará centrada en espacios vectoriales reales de dimensión finita, que es el caso particular en el análisis de señales de ECG.

4.2 Algunos Conceptos Previos

Antes de iniciar una descripción más detallada, vamos a introducir algunos conceptos estadísticos que se usan en PCA. Ellos son, la *desviación estándar*, la *covarianza*, los *autovectores* y los *autovalores.*

Para la definición de estos conceptos, vamos a asumir el hecho de que trabajamos con *muestras* de una población.

Desviación Estándar

Sobre un conjunto de datos, *n*, es posible realizar infinidad de mediciones, entre ellas la media aritmética cuya fórmula es:

$$\overline{X} = \frac{\sum_{i=n}^{n} X_i}{n} \tag{0.1}$$

La media nos da una idea de cuál es el punto medio, pero no nos dice nada acerca de la distribución de los puntos.

La Desviación Estándar (SD –Standard Deviation) de un conjunto de datos es la medida de cómo están dispersos los puntos. La definición formal es: *La distancia promedio desde la media del conjunto de datos a un punto.* Su fórmula es:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2}{n}}. \tag{0.2}$$

Varianza

Es otra medida de la dispersión de datos en el conjunto de datos. Es el cuadrado de la desviación estándar. Su fórmula es:

$$\text{var}\left(X\right) = \sigma^2 = \frac{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2}{n} \tag{0.3}$$

Covarianza

Las últimas dos mediciones son puramente unidimensional. Sin embargo, muchos conjuntos de datos poseen más de una dimensión y el objetivo de un análisis estadístico de estos datos es ver qué relaciones existen entre las dimensiones, si las hubiera. La covarianza siempre se mide entre dos dimensiones. Si se calcula la covarianza entre una dimensión y ella misma, se obtiene la varianza. La formula de covarianza es muy similar a la fórmula de varianza de la ecuación (4.3) si expandimos el término cuadrático y está dada por:

$$\text{cov}\left(X,Y\right) = \frac{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{n}. \tag{0.4}$$

Matriz de Covarianza

La covarianza es una medición entre dos dimensiones. Cuando contamos con un conjunto de datos con más de dos dimensiones, hay más de una medición de covarianza a ser calculada. Para ser precisos, para un conjunto de datos de n dimensiones, se debe calcular $\frac{n!}{2(n-2)!}$ diferentes valores de covarianzas. En general se organizan los

cálculos en una matriz. La definición para la matriz de covarianza para un conjunto de datos con n dimensiones es:

$$\mathbf{C}^{n\times n} = \left(c_{ij}; c_{ij} = \text{cov}\left(\text{Dim}_i, \text{Dim}_j\right)\right), \tag{0.5}$$

donde $\mathbf{C}$ es una matriz con n filas y n columnas, y Dim_k es la k-ésima dimensión. La matriz es obviamente cuadrada y cada elemento es el cálculo de la covarianza entre dos dimensiones. Supongamos las dimensiones (x, y, z), tendremos una matriz 3x3 de la forma:

$$\mathbf{C} = \begin{bmatrix} \text{cov}(x,x) & \text{cov}(x,y) & \text{cov}(x,z) \\ \text{cov}(y,x) & \text{cov}(y,y) & \text{cov}(y,z) \\ \text{cov}(z,x) & \text{cov}(z,y) & \text{cov}(z,z) \end{bmatrix}.$$

En la diagonal principal, el valor de covarianza está calculado para cada dimensión con ella misma. Son las varianzas para esa dimensión.

Debido a que $\text{cov}(a,b) = \text{cov}(b,a)$ la matriz es simétrica alrededor de la diagonal principal.

Autovectores y Autovalores de una matriz

El escalar λ es un autovalor, llamado también valor propio o eigenvalor, de una matriz $\mathbf{A} \in \mathbb{K}^{n\times n}$;con $\mathbb{K}$ cuerpo si y solo si existe un vector no nulo $\mathbf{x} \in \mathbb{K}^{n\times 1} / \mathbf{Ax} = \lambda\mathbf{x}$. Un vector que satisfaga la relación $\mathbf{Ax} = \lambda\mathbf{x}$ se llama vector propio o autovector o eigenvector de $\mathbf{A}$ asociado al autovalor λ.

Se puede decir entonces que los autovalores y autovectores siempre vienen en pares. Una condición necesaria pero no suficiente para la existencia de los autovectores es que la matriz asociada sea cuadrada. Si una matriz $n\times n$ posee autovectores, entonces posee n autovectores.

4.3Análisis de Componentes Principales

El Análisis de Componentes Principales es un método que reduce la dimensión de los datos realizando un análisis de covarianza entre factores. De este modo, resulta útil para conjuntos de datos en múltiples dimensiones.

La selección de patrones o la extracción de patrones resulta un problema crucial para la estadística de reconocimiento de patrones. Esta selección hace referencia a un proceso donde se transforma un *espacio de datos* en un *espacio de patrones* que en teoría tiene las mismas dimensiones que el espacio de datos original. Sin embargo, la transformación está diseñada en tal sentido que el grupo de datos pueda ser representado por un número reducido de patrones efectivos y conservar aún el mayor contenido de la información intrínseca de los datos. Podríamos decir que los datos sufren una reducción en su dimensión (Haykin, 1996).

La reducción es útil y muchas veces posible, pues con frecuencia mucha de la variabilidad de los datos puede explicarse por un número pequeño de m componentes principales, con $m < p$, siendo p el número original de variables X_i. La reducción también puede ser útil como una primera etapa de un análisis más completo, (Hernandez Rodriguez, 1998). El objetivo sería encontrar una transformación $\mathbf{T}$ tal que la relación $\mathbf{TX}$ es óptima en el sentido de producir el mínimo error cuadrático medio. La transformación $\mathbf{T}$ debería tener la propiedad de que algunos de sus componentes tengan menor varianza.

Esta técnica tiene tres **efectos** a destacar:

- Ortogonaliza los componentes de los vectores de entrada de tal modo que estén no-correlacionados entre ellos.
- Reordena los componentes ortogonales (componentes principales) de modo que aquellos con mayor variación estén en los primeros lugares.
- Elimina aquellos componentes que contribuyen menos a la variación del conjunto de datos.

4.3.1 Definición de los Componentes Principales

Consideremos un conjunto de datos de interés representado por un vector aleatorio de dimensión p al que llamaremos $\mathbf{x}$. Asumimos además que dicho vector tiene media aritmética cero, es decir:

$$E[\mathbf{x}] = 0, \tag{0.6}$$

donde $E[\cdot]$es el operador de esperanza estadística (también llamada esperanza, valor esperado, media poblacional o media) de una variable aleatoria y formaliza la idea de *valor medio* de un fenómeno aleatorio. Si $\mathbf{x}$ tiene media distinta de cero, se centran las variables, lo que significa hacer como en (0.6). Esto simplifica el análisis, sin embargo no altera la forma del objeto, ni las relaciones que en él se dan. Si las variables de los datos de partida no están centradas, se aplica el cálculo que sigue

$$\mathbf{x}^c = \mathbf{x} - \mathrm{E}[\mathbf{x}]. \tag{0.7}$$

Sea $\mathbf{u}$ un vector unidad, también de dimensión p sobre el cual el vector $\mathbf{x}$ será proyectado. Esta proyección está definida por el producto interior de los vectores $\mathbf{x}$ y $\mathbf{u}$ de la siguiente manera

$$a = \mathbf{x}^T\mathbf{u} = \mathbf{u}^T\mathbf{x}. \tag{0.8}$$

Sujeto a la siguiente restricción

$$\|\mathbf{u}\| = \left(\mathbf{u}^T\mathbf{u}\right)^{1/2} = 1. \tag{0.9}$$

La proyección a es una variable aleatoria con la media aritmética y varianza relacionadas con la estadística del vector de datos $\mathbf{x}$. Como supusimos que $\mathbf{x}$ tiene media cero, entonces se cumple también para la proyección a como pude ser fácilmente demostrado:

$$E[a] = \mathbf{u}^T E[\mathbf{x}] = 0. \tag{0.10}$$

La varianza de la proyección a está dada por:

$$\begin{aligned}\sigma^2 &= E\left[a^2\right] \\ &= E\left[\left(\mathbf{u}^T\mathbf{x}\right)\left(\mathbf{x}^T\mathbf{u}\right)\right] \\ &= \mathbf{u}^T E\left[\mathbf{x}\mathbf{x}^T\right]\mathbf{u} \\ &= \mathbf{u}^T\mathbf{R}\mathbf{u}\end{aligned} \tag{0.11}$$

La matriz $\mathbf{R}$ de dimensión $p \times p$ es la matriz de correlación del vector de datos. La podemos definir también más formalmente como la esperanza del producto exterior del vector $\mathbf{x}$ con él mismo, de esta manera:

$$\mathbf{R} = E\left[\mathbf{x}\mathbf{x}^T\right]. \tag{0.12}$$

Puede observarse que la matriz $\mathbf{R}$ es simétrica, lo que significa que

$$\mathbf{R} = \mathbf{R}^T. \tag{0.13}$$

Como consecuencia de esta propiedad, se cumple si $\mathbf{a}$ y $\mathbf{b}$ son vectores de dimensión $p \times 1$ luego

$$\mathbf{a}^T\mathbf{R}\mathbf{b} = \mathbf{b}^T\mathbf{R}\mathbf{a}. \tag{0.14}$$

De las ecuaciones en (0.11) podemos ver que la varianza σ^2 de la proyección a es función del vector unitario $\mathbf{u}$; por lo que podemos escribir:

$$\psi(\mathbf{u}) = \sigma^2 = \mathbf{u}^T\mathbf{R}\mathbf{u}. \tag{0.15}$$

Entonces, debemos pensarla a $\psi(\mathbf{u})$ como un *varianza de sondeo.*

4.3.2 Estructura del Análisis de Componentes Principales

La idea del PCA es encontrar un vector unitario $\mathbf{u}$ de tal modo que $\psi(\mathbf{u})$ tenga valores extremos o estacionarios (máximo local o mínimo local), respetando las

restricciones de la norma Euclideana de $\mathbf{u}$, $\|\mathbf{u}\| = 1$. La solución a este problema radica en la autoestructura (estructura de los autovalores y autovectores) de la matriz de correlación $\mathbf{R}$. Si $\mathbf{u}$ es un vector unitario tal que la varianza de sondeo $\psi(\mathbf{u})$ tiene un valor extremo, entonces para cualquier pequeña perturbación $\delta\mathbf{u}$ del vector unitario $\mathbf{u}$, encontramos que,

$$\psi(\mathbf{u}+\delta\mathbf{u}) = \psi(\mathbf{u}). \tag{0.16}$$

Ahora, de la definición de varianza de sondeo, dada en (0.15), tenemos que:

$$\begin{aligned}\psi(\mathbf{u}+\delta\mathbf{u}) &= (\mathbf{u}+\delta\mathbf{u})^T \mathbf{R}(\mathbf{u}+\delta\mathbf{u}) \\ &= \mathbf{u}^T\mathbf{R}\mathbf{u} + 2(\delta\mathbf{u})^T \mathbf{R}\mathbf{u} + (\delta\mathbf{u})^T \mathbf{R}\delta\mathbf{u}\end{aligned} \tag{0.17}$$

donde hicimos uso de (0.14) en la segunda línea.

Si ignoramos el término de segundo orden $(\delta\mathbf{u})^T \mathbf{R}\delta\mathbf{u}$ y llamamos a la definición establecida en (0.15), podemos escribir:

$$\begin{aligned}\psi(\mathbf{u}+\delta\mathbf{u}) &= \mathbf{u}^T\mathbf{R}\mathbf{u} + 2(\delta\mathbf{u})^2 \mathbf{R}\mathbf{u} \\ &= \psi(\mathbf{u}) + 2(\delta\mathbf{u})^T \mathbf{R}\mathbf{u}\end{aligned} \tag{0.18}$$

Reemplazando ahora lo hallado en la ecuación (0.16) en (0.18) resulta que

$$(\delta\mathbf{u})^T \mathbf{R}\mathbf{u} = 0. \tag{0.19}$$

No cualquier perturbación $\delta\mathbf{u}$ del vector $\mathbf{u}$ resulta aceptable, más aún, serán admitidas sólo aquellas perturbaciones para las cuales la norma Euclidea del vector perturbado $\mathbf{u}+\delta\mathbf{u}$ permanezca igual a la unidad, esto es:

$$\|\mathbf{u}+\delta\mathbf{u}\| = 1, \tag{0.20}$$

que equivale a decir:

$$(\mathbf{u}+\delta\mathbf{u})^T (\mathbf{u}+\delta\mathbf{u}) = 1. \tag{0.21}$$

Luego y a la luz de la ecuación (0.9) requerimos que para $\delta\mathbf{u}$,

$$\left(\delta\mathbf{u}\right)^T \mathbf{u} = 0, \tag{0.22}$$

esto significa que las perturbaciones $\delta\mathbf{u}$ deben ser ortogonales a $\mathbf{u}$, y de este modo solamente se permite un cambio en la dirección de $\mathbf{u}$.

Por convención, los elementos del vector unidad $\mathbf{u}$ son adimensionales en un sentido físico. Si, de este modo, vamos a combinar las ecuaciones (0.19) y (0.22), debemos introducir un factor de escala λ en una ecuación posterior con las mismas dimensiones que las entradas en la matriz de correlación $\mathbf{R}$. Haciendo todo esto, podemos entonces escribir

$$\left(\delta\mathbf{u}\right)^T \mathbf{R}\mathbf{u} - \lambda\left(\delta\mathbf{u}\right)^T \mathbf{u} = 0, \tag{0.23}$$

o de manera equivalente,

$$\left(\delta\mathbf{u}\right)^T \left(\mathbf{R}\mathbf{u} - \lambda\mathbf{u}\right) = 0. \tag{0.24}$$

Para que la condición de la ecuación (0.24) se mantenga, es necesario y suficiente que se cumpla

$$\mathbf{R}\mathbf{u} = \lambda\mathbf{u}. \tag{0.25}$$

Esta es la ecuación que gobierna los vectores unitarios $\mathbf{u}$ para los cuales la varianza de sondeo $\psi\left(\mathbf{u}\right)$ tiene valores extremos.

La ecuación (0.25) se reconoce como un problema de autovalores, comúnmente encontrado en el álgebra lineal. Este problema tiene soluciones no triviales sólo para valores especiales de λ llamados autovalores de la matriz de correlación $\mathbf{R}$. Los valores asociados de $\mathbf{u}$ son los autovectores. Una matriz de correlación se caracteriza por tener autovalores reales no-negativos. Los autovectores asociados son únicos, asumiendo que los autovalores son distintos. Sean los autovalores de la matriz $\mathbf{R}$ de dimensión $p \times p$, a los que llamaremos $\lambda_0, \lambda_1, ..., \lambda_{p-1}$, y los autovectores asociados se denominaran $\mathbf{u}_0, \mathbf{u}_1, ..., \mathbf{u}_{p-1}$ respectivamente. Podemos, por lo tanto escribir:

$$\mathbf{R}\mathbf{u}_j = \lambda_j \mathbf{u}_j; \qquad j = 0,1,..., p-1. \tag{0.26}$$

Si ordenamos los autovalores de manera decreciente:

$$\lambda_0 > \lambda_1 > ... > \lambda_j > ... > \lambda_{p-1}; \tag{0.27}$$

de modo que $\lambda_o = \lambda_{\max}$ y construimos una matriz de dimensión $p \times p$ con los autovectores asociados, tenemos que:

$$\mathbf{U} = [\mathbf{u}_0, \mathbf{u}_1, ..., \mathbf{u}_j, ..., \mathbf{u}_{p-1}]. \tag{0.28}$$

Podemos combinar las p ecuaciones de (0.26) en una única ecuación:

$$\mathbf{R}\mathbf{U} = \mathbf{U}\mathbf{\Lambda}. \tag{0.29}$$

Donde $\mathbf{\Lambda}$ es una matriz diagonal definida por los autovalores de la matriz $\mathbf{R}$:

$$\mathbf{\Lambda} = \text{diag}\left[\lambda_{0,}\lambda_1, ..., \lambda_j, ..., \lambda_{p-1}\right]. \tag{0.30}$$

La matriz $\mathbf{U}$ es una matriz ortogonal en el sentido que sus vectores columnas (los autovectores de $\mathbf{R}$) satisfacen las condiciones de ortonormalidad:

$$\mathbf{u}_i^T \mathbf{u}_j = \begin{cases} 1, & j = 1 \\ 0, & j \neq 1 \end{cases} \tag{0.31}$$

La ecuación (0.31) requiere autovalores distintos. Equivalentemente, podemos escribir:

$$\mathbf{U}^T\mathbf{U} = \mathbf{I}. \tag{0.32}$$

De lo que deducimos que la inversa de la matriz $\mathbf{U}$ es igual a su transpuesta,

$$\mathbf{U}^T = \mathbf{U}^{-1}. \tag{0.33}$$

Esta igualdad nos da la posibilidad de escribir la ecuación (0.29) en una forma conocida como la *transformación de similitud ortogonal*, según lo justifican los siguientes teoremas (4.1 y 4.2)

$$\mathbf{U}^T\mathbf{R}\mathbf{U} = \mathbf{\Lambda}. \tag{0.34}$$

Teorema 4.1: (Teorema de Schur). Sea $\mathbf{A}$ una matriz arbitraria. Existe una matriz no singular $\mathbf{U}$ con la propiedad que

$$\mathbf{T} = \mathbf{U}^{-1}\mathbf{A}\mathbf{U},$$

donde $\mathbf{T}$ es una matriz triangular superior cuyos elementos diagonales constan de los valores característicos de $\mathbf{A}$.

Teorema 4.2: Si $\mathbf{A}$ es simétrica y $\mathbf{D}$ es una matriz ortogonal cuyos elementos diagonales son los valores característicos de $\mathbf{A}$, entonces existe una matriz $\mathbf{Q}$ tal que

$$\mathbf{D} = \mathbf{Q}^{-1}\mathbf{A}\mathbf{Q} = \mathbf{Q}^T\mathbf{A}\mathbf{Q}.$$

Lo que equivale a escribir

$$\mathbf{u}_j^T\mathbf{R}\mathbf{u}_k = \begin{cases} \lambda_j, & k = j \\ 0, & k \neq j \end{cases} \tag{0.35}$$

De las ecuaciones (0.15) y (0.35) se deduce que las varianzas de sondeo y autovalores son iguales

$$\psi\left(\mathbf{u}_j\right) = \lambda_j; \quad j = 0,1,..., p-1. \tag{0.36}$$

Resumiendo los dos puntos importantes hallados a partir de la autoestructura de los análisis de componentes principales:

- Los autovectores de la matriz de correlación $\mathbf{R}$ relacionados al vector $\mathbf{x}$ cuya media es igual a cero define los vectores unitarios $\mathbf{u}_j$ que representan las direcciones principales a lo largo de las cuales la varianza de sondeo $\psi\left(\mathbf{u}_j\right)$ tiene sus valores extremos.

- Los autovalores asociados definen los valores extremos de las varianzas de sondeo $\psi(\mathbf{u}_j)$.

4.3.3. Representación de datos

Existen p posibles proyecciones del vector de datos $\mathbf{x}$ a ser consideradas para p posibles soluciones del vector unidad $\mathbf{u}$. Específicamente, de la ecuación (0.8) podemos notar que

$$a_j = \mathbf{u}_j^T \mathbf{x} = \mathbf{x}^T \mathbf{u}_j; \qquad j = 0,1,..., p-1. \tag{0.37}$$

Donde las a_j son las proyecciones de $\mathbf{x}$ en la dirección principal representada por los vectores unitarios $\mathbf{u}_j$. Las a_j se denominan componentes principales; tienen la misma dimensión física que el vector de datos $\mathbf{x}$.

En general, el *j*-ésimo componente principal de $\mathbf{x}$ es $a_j = \mathbf{u}_j^T \mathbf{x}$, y $\text{var}(\mathbf{u}_j^T \mathbf{x}) = \lambda_j$ donde λ_j es el j-ésimo mayor autovalor de $\mathbf{R}$ y $\mathbf{u}_j$ el autovector correspondiente.

Vamos a demostrarlo con $j = 2$:

El segundo componente principal a_1 maximiza $\mathbf{u}_1^T \mathbf{R} \mathbf{u}_1$ que está no-correlacionado con $a_0 = \mathbf{u}_0^T \mathbf{x}$, que equivale a decir que $\text{cov}[\mathbf{u}_0^T \mathbf{x}, \mathbf{u}_1^T \mathbf{x}] = \text{cov}[a_0, a_1] = 0$. Pero

$$\text{cov}[a_0, a_1] = \mathbf{u}_0^T \mathbf{R} \mathbf{u}_1 = \mathbf{u}_1^T \mathbf{R} \mathbf{u}_0 = \mathbf{u}_1^T \lambda_0 \mathbf{u}_0^T = \lambda_0 \mathbf{u}_1^T \mathbf{u}_0 = \lambda_0 \mathbf{u}_0^T \mathbf{u}_1 .$$

Luego cualquiera de las ecuaciones

$$\mathbf{u}_0^T \mathbf{R} \mathbf{u}_1 = 0, \qquad \mathbf{u}_0^T \mathbf{u}_1 = 0,$$

$$\mathbf{u}_1^T \mathbf{R} \mathbf{u}_0 = 0, \qquad \mathbf{u}_1^T \mathbf{u}_0 = 0,$$

podría usarse para especificar la correlación entre $\mathbf{u}_0^T\mathbf{x}$ y $\mathbf{u}_1^T\mathbf{x}$. Elegimos la última arbitrariamente; teniendo presente la restricción de la ecuación (4.10):

$$\mathbf{u}_1^T\mathbf{R}\mathbf{u}_0 = \lambda\left(\mathbf{u}_1^T\mathbf{u}_1 - 1\right) - \phi\mathbf{u}_1^T\mathbf{u}_0;$$

donde λ y ϕ son multiplicadores de Lagrange. Derivando con respecto a $\mathbf{u}_1$ resulta

$$\mathbf{R}\mathbf{u}_1 - \lambda\mathbf{u}_1 - \phi\mathbf{u}_0 = 0.$$

Premultiplicando por $\mathbf{u}_0^T$ nos da

$$\mathbf{u}_0^T\mathbf{R}\mathbf{u}_1 - \lambda\mathbf{u}_0^T\mathbf{u}_1 - \phi\mathbf{u}_0^T\mathbf{u}_0 = 0,$$

que como los primeros dos términos son cero y $\mathbf{u}_0^T\mathbf{u}_0 = 1$, entonces resulta que $\phi = 0$.

Luego λ es una vez más un autovalor de $\mathbf{R}$ y $\mathbf{u}_1$ su autovector correspondiente. Nuevamente $\lambda = \mathbf{u}_1^T\mathbf{R}\mathbf{u}_1$, entonces λ debe ser tan grande como sea posible. Asumiendo que $\mathbf{R}$ no tiene autovalores repetidos, si así fuera $\mathbf{u}_1 = \mathbf{u}_0$ lo que viola la restricción $\mathbf{u}_0^T\mathbf{u}_1 = 0$, por lo tanto λ es el segundo mayor autovalor de $\mathbf{R}$, y $\mathbf{u}_1$ su correspondiente autovector.

Como se demostró con *j*=2, puede demostrarse que para los 3ros, 4tos, …,p-ésimos componentes principales, los vectores de coeficientes $\mathbf{u}_2, \mathbf{u}_{3,} ..., \mathbf{u}_{p-1}$ con los autovectores de $\mathbf{R}$ correspondientes a $\lambda_2, \lambda_3, ..., \lambda_{p-1}$ el 3ro y 4to más grande y así hasta el autovalor más chico respectivamente, más aún

$$\text{var}\left[\mathbf{u}_j^T\mathbf{x}\right] = \lambda_j; \quad j = 0,1,..., p-1.$$

La fórmula de la ecuación (0.37) puede ser vista como una fórmula de *análisis*. La etapa de *síntesis* está dada por la reconstrucción del vector $\mathbf{x}$ a partir de sus componentes principales a_j. La transformación entonces que sufren los datos originales mediante el análisis de componentes principales, puede revertirse para reconstruir el vector original a partir de sus proyecciones. Para realizar este proceso resulta necesario

seguir los pasos que a continuación se detallan. Primero, se combinan un grupo de proyecciones $\{a_j \mid j = 0,1,..., p-1\}$ en un único vector, como se muestra en:

$$\begin{aligned}\mathbf{a} &= \left[a_0, a_1, ..., a_{p-1} \right]^T \\ &= \left[\mathbf{x}^T\mathbf{u}_0, \mathbf{x}^T\mathbf{u}_1, ..., \mathbf{x}^T\mathbf{u}_{p-1} \right]^T \\ &= \mathbf{U}^T\mathbf{x}.\end{aligned} \tag{0.38}$$

Si premultiplicamos ambos miembros de la igualdad por la matriz $\mathbf{U}$ y utilizamos luego la relación que establece que $\mathbf{U}^T = \mathbf{U}^{-1}$, el vector de datos originales $\mathbf{x}$ puede ser reconstruido como sigue:

$$\mathbf{x} = \mathbf{U}\mathbf{a} = \sum_{j=0}^{p-1} a_j \mathbf{u}_j \ . \tag{0.39}$$

Ésta puede ser vista como una fórmula de *síntesis*, como mencionamos previamente. En este caso, los vectores unitarios $\mathbf{u}_j$ representan una base del espacio de datos. En realidad la ecuación (0.39) no es más que una transformación coordinada, de acuerdo a la cual un punto $\mathbf{x}$ en el espacio de datos es transformado en un punto correspondiente $\mathbf{a}$ en el espacio de patrones.

4.3.4 Reducción de la Dimensión

Como lo mencionamos con anterioridad, el valor práctico del análisis de componentes principales es que proporciona una técnica efectiva para la reducción de la dimensión de los datos. En particular, podemos reducir el número de patrones necesarios para la representación efectiva de los datos descartando aquellas combinaciones lineales de la ecuación (0.39) que tienen varianzas pequeñas y reteniendo aquellas cuyos términos tienen varianzas más grandes.

Sean $\lambda_0, \lambda_1, ..., \lambda_{m-1}$ de modo que signifiquen los m mayores autovalores de la matriz de correlación $\mathbf{R}$. Entonces podemos aproximar el vector de datos $\mathbf{x}$ truncando la expansión de la ecuación (0.39) luego de m términos como sigue:

$$\mathbf{x}' = \sum_{j=0}^{m-1} a_j \mathbf{u}_j; \quad m < p. \tag{0.40}$$

Debe notarse que los valores mayores de $\lambda_0, \lambda_1, ..., \lambda_{m-1}$ no participan del cálculo del vector $\mathbf{x}'$; simplemente determinan el número de los términos retenidos en la expansión actual utilizada para calcular $\mathbf{x}'$.

El vector de error de aproximación de $\mathbf{e}$ iguala la diferencia entre el vector de datos original $\mathbf{x}$ y el vector de datos de aproximaciones $\mathbf{x}'$, como se muestra:

$$\mathbf{e} = \mathbf{x} - \mathbf{x}'. \tag{0.41}$$

Si sustituimos las ecuaciones (0.39) y (0.40) en (0.41) se produce

$$\mathbf{e} = \sum_{j=m}^{p-1} a_j \mathbf{u}_j \ . \tag{0.42}$$

El vector de error $\mathbf{e}$ es ortogonal al vector de datos de aproximaciones $\mathbf{x}^T$ como se puede observar en forma gráfica en la Figura 4.1). En otras palabras, el producto interior del vector $\mathbf{e}$ y $\mathbf{x}'$ es cero, lo que podemos demostrar usando las ecuaciones (0.40) y (0.42) de la siguiente manera:

$$\begin{aligned} \mathbf{e}^T \mathbf{x}' &= \sum_{i=m}^{p-1} a_j \mathbf{u}_i^T \sum_{j=0}^{m-1} a_j \mathbf{u}_j \\ &= \sum_{i=m}^{p-1} \sum_{j=0}^{m-1} a_i a_j \mathbf{u}_i^T \mathbf{u}_j \\ &= 0, \end{aligned} \tag{0.43}$$

donde hicimos uso de la segunda condición en la ecuación (0.31). La ecuación (0.43) se conoce como el *principio de ortogonalidad.*

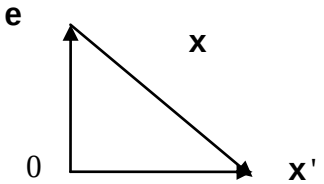

Figura 4.1 Relación entre el vector $\mathbf{x}$, su versión reconstruida $\mathbf{x}'$ y el vector error $\mathbf{e}$

La varianza total de los p componentes del vector aleatorio $\mathbf{x}$ es, vía ecuación (0.15), y la primera línea de la ecuación (0.35),

$$\sum_{j=0}^{p-1}\sigma_j^2 = \sum_{j=0}^{p-1}\lambda_j\,, \tag{0.44}$$

donde σ_j^2 es la varianza del j-ésimo componente principal a_j. La varianza total de los m elementos del vector de aproximaciones $\mathbf{x}'$es

$$\sum_{j=0}^{m-1}\sigma_j^2 = \sum_{j=0}^{m-1}\lambda_j\,. \tag{0.45}$$

La varianza total de los elementos $(m-p)$en el vector de error de aproximación $\mathbf{x}-\mathbf{x}'$ es por lo tanto,

$$\sum_{j=m}^{p-1}\sigma_j^2 = \sum_{j=m}^{p-1}\lambda_j\,. \tag{0.46}$$

Los autovalores $\lambda_m,...,\lambda_{p-1}$son los *menores* autovalores $(p-m)$de la matriz de correlación $\mathbf{R}$. Este subconjunto de autovalores corresponde a los términos descartados de la expansión de la ecuación (0.40) utilizada para construir el vector de aproximación $\mathbf{x}'$. Cuanto más se acerquen a cero estos autovalores, más efectiva será la reducción de

la dimensión, que es el resultado de aplicar el análisis de componentes principales al vector de datos $\mathbf{x}$.

Luego para reducir la dimensionalidad sobre algunos datos de entrada, se deben calcular los autovalores y autovectores de la matriz de correlación del vector de datos de entrada y luego proyectar los datos en forma ortogonal sobre el subespacio generado por los autovectores que pertenecen a los autovalores más grandes. Este método de representación de datos se conoce con el nombre de *descomposición del subespacio*, como lo describiera Oja en su trabajo original, "Subspace methods of Pattern Recognition" en 1983.

4.4 Propiedades Matemáticas y Estadísticas de los Componentes Principales

Los componentes principales pueden ser hallados usando puramente argumentos matemáticos: están dados a partir de una transformación lineal ortogonal de un grupo de variables que optimizan un criterio algebraico. Los componentes principales optimizan varios criterios algebraicos diferentes y esas propiedades de optimización conjuntamente con sus implicancias estadísticas, serán descriptas a continuación.

Consideremos el vector $\mathbf{z}$ cuyo *k*-ésimo elemento es $\mathbf{z}_k$ el *k*-ésimo componente principal (CP), $k = 1,2,\ldots, p$ (el *k*-ésimo PC será considerado con la *k*-ésima máxima varianza con las interpretaciones correspondientes del *k*-ésimo autovalor y *k*-ésimo autovector). Luego

$$\mathbf{z} = \mathbf{A}^T\mathbf{x}, \tag{4.49}$$

donde $\mathbf{A}$ es una matriz ortogonal cuya *k*-ésima columna $\mathbf{u}_k$ es el *k*-ésimo autovector de $\mathbf{R}$ (matriz de covarianza). Entonces los componentes principales están definidos por una transformación lineal ortogonal de $\mathbf{x}$. Más aún, de la sección anterior tenemos que

$$\mathbf{RA} = \mathbf{A\Lambda}, \tag{4.50}$$

donde $\mathbf{\Lambda}$ es una matriz diagonal cuyo *k*-ésimo elemento de la diagonal es $\lambda_{k,}$ el *k*-ésimo autovalor de $\mathbf{R}$, y $\lambda_k = \text{var}\left(\mathbf{u}_k^T\mathbf{x}\right) = \text{var}\left(\mathbf{z}_k\right)$, como $\mathbf{A}$ es ortogonal, podemos expresar (4.50) de estas dos maneras equivalentes,

$$\mathbf{A}^T\mathbf{R}\mathbf{\Lambda} \tag{4.51}$$

y

$$\mathbf{R} = \mathbf{A}\mathbf{\Lambda}\mathbf{A}^T. \tag{4.52}$$

A continuación enunciaremos las propiedades óptimas de a transformación lineal expresada en (4.49) que define a $\mathbf{z}$.

Propiedad 4.1: para cualquier entero $q; 1 \le q \le p$, consideremos la siguiente transformación lineal

$$\mathbf{y} = \mathbf{B}^T\mathbf{x}, \tag{4.53}$$

donde $\mathbf{y}$ es un vector de q elementos y $\mathbf{B}^T$ es una matriz ($q \times p$) y sea $\mathbf{R}_y = \mathbf{B}^T\mathbf{R}\mathbf{B}$ la matriz de covarianza de $\mathbf{y}$. Luego la traza de $\mathbf{R}_y$, $\text{tr}\left(\mathbf{R}_y\right)$ se maximiza si se toma $\mathbf{B} = \mathbf{A}_q$ donde $\mathbf{A}_q$ es la matriz con las primeras q columnas de $\mathbf{A}$.

Demostración (Jolliffe, 2002)

Sea β_k la *k*-ésima columna de $\mathbf{B}$; como las columnas de $\mathbf{A}$ forman una base para un espacio de dimensión *p*, tenemos que,

$$\beta_k = \sum_{j=1}^{p} c_{jk}\mathbf{u}_j; \quad \text{k=1,...,q}; \tag{4.54}$$

donde c_{jk} ; $j=1,\ldots,p$ y $k=1,\ldots,q$, son constantes. Entonces $\mathbf{B} = \mathbf{AC}$ donde $\mathbf{C}$ es una matriz $\left(p \times q\right)$ con el elemento (j,k)-ésimo c_{jk} y

$$\begin{aligned}\mathbf{B}^T\mathbf{R}\mathbf{B} &= \mathbf{C}^T\mathbf{A}^T\mathbf{R}\mathbf{A}\mathbf{C} = \mathbf{C}^T\mathbf{A}\mathbf{C} \\ &= \sum_{j=1}^{p}\lambda_j c_j c_j^T\end{aligned} \tag{4.55}$$

donde c^T_{j} es la j-ésima fila de $\mathbf{C}$. Por ende

$$\begin{aligned}\mathrm{tr}\left(\mathbf{B}^T\mathbf{R}\mathbf{B}\right) &= \sum_{j=1}^{p}\lambda_j \mathrm{tr}\left(c_j c_j^T\right) \\ &= \sum_{j=1}^{p}\lambda_j \mathrm{tr}\left(c_j^T c_j\right) \\ &= \sum_{j=1}^{p}\lambda_j c_j^T c_j \\ &= \sum_{j=1}^{p}\sum_{k=1}^{q}\lambda_j c_{jk}^2 .\end{aligned} \tag{4.56}$$

Ahora

$$\mathbf{C} = \mathbf{A}^T\mathbf{B}, \text{ entonces} \tag{4.57}$$

$$\mathbf{C}^T\mathbf{C} = \mathbf{B}^T\mathbf{A}\mathbf{A}^T\mathbf{B} = \mathbf{B}^T\mathbf{B} = \mathbf{I}_q ,$$

pues $\mathbf{A}$ es ortogonal y las columnas de $\mathbf{B}$ son ortonormales. Por lo tanto,

$$\sum_{j=1}^{p}\sum_{k=1}^{q} c_{jk}^2 = q , \tag{4.58}$$

y las columnas de $\mathbf{C}$ son también ortonormales. La matriz $\mathbf{C}$ puede pensarse como las primeras q columnas de una matriz ortogonal $\left(p \times q\right)$. Llamemos a esa matriz $\mathbf{D}$, las columnas de $\mathbf{D}$ son ortonormales y satisfacen por ende, $\mathbf{d}'_j\,\mathbf{d}_j = 1;\ j = 1,..., p$. Como las columnas de $\mathbf{C}$ consiste en los primeros q elementos de las filas de $\mathbf{D}$, luego $\mathbf{c}_j^T\mathbf{c}_j \le 1;\ j = 1,..., p$, esto es

$$\sum_{k=1}^{q} c_{jk}^2 \le 1. \tag{4.60}$$

Ahora $\sum_{k=1}^{q} c_{jk}^2$ es el coeficiente de λ_j en (4.56), la suma de estos coeficientes es q de (4.57), y ninguno de los coeficientes pude ser mayor a 1, de (4.58). Debido a que $\lambda_1 > \lambda_2 > ... > \lambda_p$, es claro que $\sum_{j=1}^{p}\left(\sum_{k=1}^{q} c_{jk}^2\right)\lambda_j$ será maximizado si podemos encontrar un conjunto de c_{jk} para el cual

$$\sum_{k=1}^{q} c_{jk}^2 = \begin{cases} 1, & j = 1,...,q \\ 0, & j = q+1,...,p \end{cases} \tag{4.61}$$

Pero $\mathbf{B}^T = \mathbf{A}_q^T$, luego

$$c_{jk} = \begin{cases} 1, & 1 \le j = k \le q \\ 0, & \text{de otro modo} \end{cases} \tag{4.62}$$

Lo que satisface (4.61). Luego $\text{tr}\left(\mathbf{R}_y\right)$ alcanza su máximo valor cuando $\mathbf{B}^T = \mathbf{A}_q^T$

Propiedad 4.2: Consideraremos nuevamente la transformación ortonormal

$$\mathbf{y} = \mathbf{B}^T\mathbf{x}, \tag{4.63}$$

con $\mathbf{x, B, A, R}_y$ definidas como en la propiedad anterior, luego la $\text{tr}\left(\mathbf{R}_y\right)$ es minimizada si tomamos $\mathbf{B} = \mathbf{A}_q^*$ donde $\mathbf{A}_q^*$ consiste en las últimas q columnas de **A.**

Demostración:

El cálculo de los componentes principales descripto en la sección anterior, puede perfectamente efectuarse pero en un sentido contrario, es decir buscar sucesivamente funciones lineales de $\mathbf{x}$ cuyas varianzas son tan pequeñas como sean posibles y además estén no correlacionadas con las funciones lineales previas. Como en el caso visto, la solución se obtiene encontrando los autovectores de $\mathbf{R}$, pero en este caso el orden está invertido, es decir, los autovalores estarán ordenados de menor a mayor. La implicancia estadística de esta propiedad es que los componentes principales con varianzas mínimas

no son simplemente descartados, sino que son útiles en el sentido que pueden ayudar a detectar relaciones lineales entre elementos de $\mathbf{x}$, también pueden ser útiles en regresión, en la selección de subconjuntos de variables de $\mathbf{x}$.

Propiedad 4.3: La descomposición espectral de $\mathbf{R}$ es

$$\mathbf{R} = \lambda_1 \mathbf{u}_1 \mathbf{u}_1^T + \lambda_2 \mathbf{u}_2 \mathbf{u}_2^T + ... + \lambda_p \mathbf{u}_p \mathbf{u}_p^T . \tag{4.64}$$

Demostración

De la ecuación (4.52) sabemos que $\mathbf{R} = \mathbf{A}\Lambda\mathbf{A}^T$. Si expandimos el producto de matrices a la derecha de la igualdad, tenemos que

$$\mathbf{R} = \sum_{k=1}^{p} \lambda_k \mathbf{u}_k \mathbf{u}_k^T$$

como queríamos probar.

Este resultado tiene importantes implicaciones estadísticas. Si observamos los elementos de la diagonal, podemos ver que

$$\text{var}\left(x_j\right) = \sum_{k=1}^{p} \lambda_k \mathbf{u}_{kj}^2 .$$

Es decir que no solo podemos descomponer las varianzas combinadas de todos los elementos de $\mathbf{x}$ en sus contribuciones decrecientes debido a cada CP, sino que también podemos descomponer la matriz de covarianza total en sus contribuciones $\lambda_k \mathbf{u}_k \mathbf{u}_k^T$ a partir de cada CP. Aunque no de manera estrictamente decreciente, los elementos de $\lambda_k \mathbf{u}_k \mathbf{u}_k^T$ se irán haciendo cada vez más pequeños a medida que *k* aumenta, al igual que

λ_k disminuye cuando aumenta *k*, mientras los elementos de $\mathbf{u}_k$ tienden a quedarse del mismo tamaño, debido a la restricción de la norma: $\mathbf{u}_k\mathbf{u}_k^T = 1; \quad k = 1,..., p$.

La propiedad 4.1 hace énfasis en que los CP pueden explicar, en forma sucesiva, tanto como sea posible acerca de la $\text{tr}(\mathbf{R})$, pero esta propiedad, la 4.3, muestra, de un modo intuitivo, que también son útiles para explicar los elementos fuera de la diagonal de $\mathbf{R}$.

De la ecuación (4.64) podemos ver que la matriz de covarianza puede ser construida, dados los coeficientes y varianzas de los primeros r componentes principales, siendo r el rango de la matriz de covarianza.

Propiedad 4.4. Como en la propiedad 4.1 y 4.2, consideremos la transformación $\mathbf{y} = \mathbf{B}^T\mathbf{x}$. Si el determinante de la matriz de covarianza $\mathbf{y}$ que lo llamamos $\det(\mathbf{R}_y)$, luego $\det(\mathbf{R}_y)$ es maximizado cuando $\mathbf{B} = \mathbf{A}_q$; con $1 \le q \le p$.

Demostración:

Consideremos un entero *k* cualquiera de modo que $1 \le k \le q$ y sea S_k el subespacio de vectores de dimensión *p* ortogonales a $\mathbf{u}_1$, $\mathbf{u}_2$,...$\mathbf{u}_{k-1}$. Luego, la dimensión del subespacio mencionado está dada por: $\dim(S_k) = p - k + 1$. El *k*-ésimo autovalor λ_k de $\mathbf{R}$ satisface:

$$\lambda_k = \sup_{\substack{\mathbf{u}\in S_k \\ \mathbf{u}\neq 0}} \left\{ \frac{\mathbf{u}^T\mathbf{R}\mathbf{u}}{\mathbf{u}^T\mathbf{u}} \right\}.$$

Supongamos que $\mu_1 > \mu_2 > ... > \mu_q$ son los autovalores de $\mathbf{B}^T\mathbf{R}\mathbf{B}$ y que $\gamma_1, \gamma_2, ..., \gamma_q$ los correspondientes autovectores. Sea T_k el subespacio de vectores de

dimensión q ortogonales a $\gamma_{k+1}, \gamma_{k+2}, ..., \gamma_q$ con $\dim(T_k) = k$. Luego para cualquier vector distinto de cero $\gamma \in T_k$, se verifica que

$$\frac{\gamma^T \mathbf{B}^T \mathbf{R} \mathbf{B} \gamma}{\gamma^T \gamma} \geq \mu_k .$$

Consideremos el subespacio $\hat{S}_k$ de vectores de dimensión p de la forma $\mathbf{B}\gamma$ para $\gamma \in T_k$.

$$\dim\left(\hat{S}_k\right) = \dim(T_k) = k ;$$

pues **B** es 1x1, en realidad **B** mantiene la longitud de los vectores.

De un resultado general que involucra la dimensión de dos espacios de vectores, tenemos que

$$\dim\left(\hat{S}_k \cap S_k\right) + \dim\left(\hat{S}_k + S_k\right) = \dim\left(\hat{S}_k\right) + \dim(S_k)$$

pero

$$\dim\left(\hat{S}_k + S_k\right) \leq p ; \qquad \dim(S_k) = p - k + 1 \qquad \text{y} \qquad \dim\left(\hat{S}_k\right) = k$$

por lo tanto,

$$\dim\left(\hat{S}_k \cap S_k\right) \geq 1 .$$

Luego hay un vector **u** distinto de cero, tal que $\mathbf{u} \in S_k$ y que tiene la forma $\mathbf{u} = \mathbf{B}\gamma$ para un $\gamma \in T_k$, y resulta que

$$\mu_k \leq \frac{\gamma^T \mathbf{B}^T \mathbf{R} \mathbf{B} \gamma}{\gamma^T \gamma} = \frac{\gamma^T \mathbf{B}^T \mathbf{R} \mathbf{B} \gamma}{\gamma^T \mathbf{B}^T \mathbf{B} \gamma} = \frac{\mathbf{u}^T \mathbf{R} \mathbf{u}}{\mathbf{u}^T \mathbf{u}} \leq \lambda_k .$$

Es decir, el k-ésimo autovalor de $\mathbf{B}^T\mathbf{R}\mathbf{B} \leq k-$ésimo autovalor de **R** para $k = 1, 2, \ldots, q$. Esto significa que,

$$\det\left(R_y\right) = \prod_{k=1}^{q}\left(k-\text{ésimo autovalor de } \mathbf{B}^T\mathbf{R}\mathbf{B}\right) \leq \prod_{k=1}^{q}\lambda_k \ .$$

Pero si $\mathbf{B} = \mathbf{A}_q$, luego los autovalores de $\mathbf{B}^T\mathbf{R}\mathbf{B}$ son:

$$\lambda_1, \lambda_2, ..., \lambda_q \ ,$$

de modo que

$$\det\left(R_y\right) = \prod_{k=1}^{q}\lambda_k$$

en este caso, y de este modo $\det\left(R_y\right)$ es maximizado cuando $\mathbf{B} = \mathbf{A}_q$.

■

El resultado puede ser extendido al caso en que las columnas de $\mathbf{B}$ no son necesariamente ortonormales, pero los elementos de la diagonal de $\mathbf{B}^T\mathbf{B}$ son iguales a uno (Jolliffe, 2002). O´Hagan (1984) afirma que esta propiedad propone una derivación alternativa de los componentes principales.

La importancia estadística de este resultado se halla en que el determinante de la matriz de covarianza, llamado también varianza generalizada, puede usarse como una medida simple de dispersión de una variable aleatoria multivariada. La raíz cuadrada de la varianza normalizada de una distribución normal multivariada, es proporcional al volumen en el espacio de dimensión *p* que encierra una porción fija de la distribución de probabilidad de $\mathbf{x}$. Para $\mathbf{x}$ normales multivariadas, los primeros *q* componentes principales son como una consecuencia de la propiedad 4.4, *q* funciones lineales de $\mathbf{x}$ cuya distribución de probabilidad conjunta tiene contornes de probabilidad fija que encierran el máximo volumen.

5. Análisis de Componentes Independientes

5.1 Introducción

En el ambiente clínico frecuentemente encontramos señales electrocardiográficas contaminadas por el ruido. Un conocimiento integral de la actividad eléctrica del corazón para propósitos diagnósticos, requieren de señales limpias de ruidos. Cientos de trabajos científicos están dedicados a la eliminación, o al menos la minimización de estas señales no deseadas (Monzón et al, 2007).

En este apartado, describiremos un método llamado Análisis de Componentes Independientes (ICA –Independent Component Analysis), que es un método de transformaciones lineales recientemente desarrollado (en 1986, Herault and Jutten fueron los primeros en describir el problema de ICA y así también lo llamaron debido a su similitud con el análisis de componentes principales PCA; Comon, 1994) , en el cual la representación deseada es aquella que minimiza la dependencia estadística de los componentes de la representación. Tal representación captura la estructura esencial de los datos en muchas aplicaciones (Hyvärinen, 1999). El concepto de ICA puede ser visto también como una extensión del PCA que sólo puede imponer independencia hasta segundo orden y consecuentemente define direcciones que son ortogonales, (Comon, 1994).

Una aplicación particular de ICA es la *separación ciega de fuentes* (BSS –Blind Source Separation), (Comon, 1994) que representa una herramienta poderosa para recuperar señales de una mezcla de componentes estadísticamente independientes. Podríamos plantear el problema de la siguiente forma: un determinado proceso combina, superpone o, genéricamente, *mezcla* señales a las que se conoce como *fuentes* Las señales mezcladas reciben el nombre de *observaciones*. El adjetivo *ciega* hace referencia al hecho de que, por hipótesis, tanto las fuentes como el proceso de mezcla son, en cada situación particular, desconocidos. En estas condiciones, se desea diseñar un sistema capaz de invertir todo el proceso de la mezcla y recuperar las fuentes. La

separación se debe basar tanto en el conocimiento de las observaciones como en algunas hipótesis genéricas sobre la naturaleza de las fuentes y de la mezcla.

Como se mencionara previamente, es necesario plantear las hipótesis bajo las cuales tiene sentido aplicar la BSS. Planteamos la hipótesis de que el corazón es un sistema estable. Nos basamos para ello en la homeostasis, que alude a la tendencia a mantener el equilibrio fisiológico por compensación química, y que permite, por lo tanto, considerar al corazón como un sistema biológico estable y retroalimentado. Esto convalidaría la aplicación de los criterios generales de estabilidad de un sistema físico, y la utilización de procedimientos computacionales propios para la separación de fuentes de señal.

La teoría general de estabilidad considera que un sistema físico causal está en equilibrio si, en ausencia de cualquier perturbación a la entrada, la salida permanece inalterada. También se admite que un sistema lineal e invariante en el tiempo es estable si la respuesta a una perturbación tiende al punto de equilibrio transcurrido un cierto tiempo. Contrariamente, será inestable, cuando la respuesta sea divergente alejando al sistema del equilibrio que presentaba al momento de la perturbación.

En este capítulo se presentarán las bases teóricas del ICA, BSS como una aplicación de ICA y los criterios de estabilidad de sistemas necesarios para aplicar los métodos de separación de fuentes para la señales.

5.2 Análisis de Componentes Independientes

5.2.1 Definición

Según lo mencionado en la sección anterior, podríamos decir que el problema en el ICA se reduce a encontrar una representación lineal en la cual los componentes son estadísticamente independientes. En el mundo real, encontrar una representación con esta característica no siempre es posible, pero al menos se podrían encontrar los componentes lo más independientes posible.

Sea el siguiente conjunto observaciones de variables aleatorias $\left(x_1(t), x_2(t), ..., x_N(t)\right)$ donde t es la variable tempo o bien muestra, asumimos que están generadas como una mezcla lineal de componentes independientes de modo que:

$$\begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_N(t) \end{bmatrix} = \mathbf{A} \begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_N(t) \end{bmatrix} \tag{5.1}$$

Donde $\mathbf{A}$ es una matriz desconocida. Entonces ICA trata de estimar $\mathbf{A}$ y $s_i(t); i = 1, ..., N$, teniendo como único dato a $x_i(t); i = 1, ..., N$. Por simplicidad asumimos que el número de señales observadas y el número de señales fuentes son iguales, sin embargo, no es completamente necesario.

Debido a que ICA es un tópico relativamente nuevo, existen al menos tres diferentes definiciones mencionadas en la literatura, (Comon, 1994; Jutten and Herault, 1991). Nos focalizaremos en la siguiente definición que resulta la más apropiada para nuestro propósito:

Definición 5.1 : El análisis de Componentes Principales de un vector $\mathbf{x}$ consiste en estimar el siguiente modelo para los datos

$$\mathbf{x} = \mathbf{A} \times \mathbf{s} \tag{5.2}$$

donde $\mathbf{A}$ es una matriz constante llamada de mezcla de dimensión $N \times M$ y el vector $\mathbf{s}(t) = \left[s_1(t), s_2(t) ..., s_M(t) \right]^T$ que se asume es independiente.

Hemos decidido basar nuestro trabajo en esta definición simple debido a que la gran mayoría de las investigaciones en ICA basan sus modelos en este concepto.

5.2.2 Condiciones de identificabilidad del Modelo ICA

En estadística, la *identificabilidad* de un modelo es la propiedad que debe cumplir el modelo para que la inferencia sea posible. Se puede decir que el modelo es *identificable* si es teóricamente posible aprender los valores verdaderos de sus parámetros más relevantes luego de obtener de él un infinito número de observaciones. Usualmente el modelo es identificable solo bajo restricciones técnicas, que reciben el nombre de condiciones de identificación o identificabilidad.

Además de la independencia estadística mencionada previamente, se deben considerar las siguientes restricciones para que la identificabilidad del modelo esté asegurada.

a) Todos los componentes independientes s_i, con la posible excepción de uno de ellos, debe ser *no-Gaussiano*.

b) El número de mezclas lineales observadas N debe ser al menos tan grande como el número de componentes independientes M, es decir $N \geq M$

c) La matriz $\mathbf{A}$ debe ser una matriz cuyas columnas sean todas linealmente independientes (full column rank)

También se asume, con regularidad, que *x* y *s* están centradas, lo que no constituye una restricción en la práctica, pues siempre se la puede obtener sustrayendo su media aritmética.

Asumiremos que se cumplen las tres condiciones anunciadas en a), b) y c). Asumiremos también que la dimensión de los datos observados es igual a la dimensión de los componentes independientes, es decir $N = M$. Esta simplificación está justificada en el hecho que si $N > M$, la dimensión del vector de observaciones siempre puede ser reducida para que $N = M$. Tal reducción de la dimensión puede lograrse con métodos como PCA.

5.2.3 Definiciones y propiedades de la independencia

El concepto de independencia estadística hace referencia a la información que puede ser obtenida de una variable, a partir de los valores de otra. Sean dos variables cualesquiera, aleatorias con valores escalares, y_1 e y_2, básicamente se dice que son independientes si la información sobre el valor de y_1 no proporciona información sobre el valor de y_2 y viceversa, (Hyvärinen, Oja, 2000). Este es el caso de las variables s_i mencionadas previamente, sin embargo no lo es de las variables de mezcla x_i. Técnicamente, la independencia puede ser definida por las densidades de probabilidad. Si llamamos $p(y_1, y_2)$ la función de densidad de probabilidad conjunta (fdpc) de y_1 e y_2 y llamamos $p_1(y_1)$ la función de densidad de probabilidad (fdp) de y_1 cuando la consideramos sola, resulta:

$$p_1(y_1) = \int p(y_1, y_2)\, dy_2 \tag{5.3}$$

y de modo similar para y_2.

Luego definimos que y_1 e y_2 son independientes, sí y solo sí la fdpc es *factorizable* en el modo que sigue:

$$p(y_1, y_2) = p_1(y_1)\, p_2(y_2) \tag{5.4}$$

Esta definición se extiende naturalmente para cualquier número n de variables aleatorias, en cuyo caso la densidad conjunta será el producto de n términos.

De esta definición surge una propiedad importante de las variables aleatorias. Dadas dos funciones h_1 y h_2, se cumple que

$$E\{h_1(y_1) h_2(y_2)\} = E\{h_1(y_1)\} E\{h_2(y_2)\}. \tag{5.5}$$

La demostración es la siguiente:

$$\begin{aligned} E\{h_1(y_1)h_2(y_2)\} &= \iint h_1(y_1)h_2(y_2)\,p(y_1,y_2)\,dy_1dy_2 \\ &= \iint h_1(y_1)\,p_1(y_1)h_2(y_2)\,p_2(y_2)\,dy_1dy_2 \\ &= \int h_1(y_1)p_1(y_1)\,dy_1 \int h_2(y_2)\,p_2(y_2)dy_2 \\ &= E\{h_1(y_1)\}\,E\{h_2(y_2)\}. \end{aligned} \tag{5.6}$$

La no correlación es una independencia parcial

La no correlación es un modo débil de independencia. Dos variables aleatorias y_1, y_2 son no correlacionas si su covarianza es cero:

$$E\{y_1y_2\} - E\{y_1\}E\{y_2\} = 0. \tag{5.7}$$

La independencia implica la no correlación, lo que se deduce directamente de la ecuación (5.5), tomando $h_1(y_1) = y_1$ y $h_2(y_2) = y_2$. Sin embargo la no correlación no implica independencia. El ejemplo que sigue lo demuestra:

Sean (y_1, y_2) valores discretos que siguen una distribución tal que el par tiene una probabilidad de $1/4$ igual a cualquiera de los valores siguientes (0,1), $(0,-1)$, $(1,0)$, $(-1,0)$. Luego y_1 e y_2 están no correlacionadas. Sin embargo,

$$E\{y_1^2y_2^2\} = 0 \neq \frac{1}{4} = E\{y_1^2\}E\{y_2^2\}. \tag{5.8}$$

Por lo tanto no se cumple la condición de la ecuación (5.5), es decir que las variables no son independientes.

Condición de no-Gaussianidad

La restricción fundamental en ICA es que los componentes deben ser no-Gaussianos. Lo ilustraremos con un ejemplo, asumiremos que la matriz de mezcla es ortogonal y que las s_i son gaussianas. Luego x_1 y x_2 son gaussianas, no correlacionadas y con varianza unitaria. La función de densidad de probabilidad conjunta está dada por

$$p(x_1, x_2) = \frac{1}{2\pi} \exp\left(-\frac{x_1^2 + x_2^2}{2} \right), \tag{5.9}$$

cuya distribución se muestra en la Figura 5.1). Como podemos observar de la Figura 5.1), la densidad es completamente simétrica, es decir que no contiene información acerca de las direcciones de las columnas de la matriz de mezcla $\mathbf{A}$, es por ello que no podemos estimarla. En el caso de variables gaussianas, solo podemos estimar el modelo de ICA hasta una transformación ortogonal.

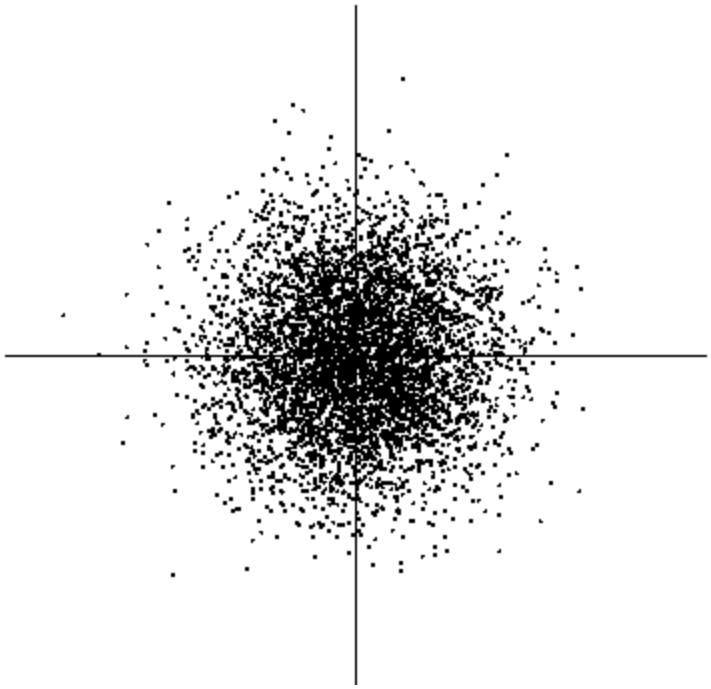

Figura 5.1. Distribución de densidad de variables gaussianas

5.3 Principios para la estimación de ICA

5.3.1 No gaussianidad es independencia

En forma intuitiva, podemos decir que evaluando la no gaussianidad de las variables, estamos en condiciones de saber si es posible estimar ICA. Esta es quizás la razón del resurgimiento tardío de la investigación del ICA: en mucha de la teoría clásica de la estadística, se asume que las variables aleatorias poseen una distribución gaussiana, lo que desalienta la aplicación del método ICA. El teorema central del límite dice que la distribución de una suma de variables aleatorias independientes tiende a una distribución gaussiana, bajo ciertas condiciones. Por lo tanto la suma de dos variables aleatorias independientes usualmente tiene una distribución que se aproxima más a la gaussiana que cualquiera de las dos variables aleatorias originales.

Consideremos el vector de datos $\mathbf{x}$, con una distribución de acuerdo al modelo de datos de ICA de la ecuación (5.2), es una mezcla de componentes independientes. Asumamos, por simplicidad, que todos los componentes independientes tienen idéntica distribución. Para poder estimar uno de los componentes independientes, consideramos una combinación lineal de los x_i de la ecuación $\mathbf{s} = \mathbf{W}\mathbf{x}$, que surge luego de estimar la matriz de mezcla $\mathbf{A}$, la matriz $\mathbf{W}$ es la inversa de $\mathbf{A}$, $\mathbf{s}$ son las fuentes. Llamamos $y = \mathbf{w}^T\mathbf{x} = \sum_i w_i x_i$ donde $\mathbf{w}$ es un vector que será determinado y que si fuese uno de las filas de $\mathbf{W}$, la combinación lineal sería igual a uno de los componentes independientes. En la práctica, no se puede determinar exactamente $\mathbf{w}$ de ese modo, debido a que desconocemos $\mathbf{A}$, pero podríamos obtener una buena aproximación. Haremos ahora aun cambio de variables, definiendo $\mathbf{z} = \mathbf{A}^T\mathbf{w}$. Luego tenemos $y = \mathbf{w}^T\mathbf{x} = \mathbf{w}^T\mathbf{A}\mathbf{s} = \mathbf{z}^T\mathbf{s}$; y es una combinación lineal de s_i con pesos dados por z_i. Como ya lo mencionamos previamente, la suma de dos variables aleatorias independientes es más gaussiana que las variables originales; $\mathbf{z}^T\mathbf{s}$ es más gaussiana que cualquiera de las s_i y se convierte en menos gaussiana cuando es igual a una de las s_i. En este caso, solo uno de los elementos z_i de $\mathbf{z}$ es distinto de cero. Podemos tomar entonces a $\mathbf{w}$ como a un vector que maximiza la no-gaussianidad de $\mathbf{w}^T\mathbf{x}$. Este vector

se corresponderá con un vector $\mathbf{z}$ que sólo tiene un componente distinto de cero. Esto significa que $\mathbf{w}^T\mathbf{x} = \mathbf{z}^T\mathbf{s}$ iguala a uno de los componentes independientes.

Maximizar la no-gaussianidad de $\mathbf{w}^T\mathbf{x}$ nos da uno de los componentes independientes. La optimización para la no-gaussianidad en el espacio *n*-dimensional de los vectores $\mathbf{w}$ tiene 2*n* máximos locales, dos para cada componente independiente, que corresponde a s_i y $-s_i$. Para encontrar varios componentes independientes, necesitamos encontrar todos esos máximos locales. Lo que no es difícil pues los diferentes componentes independientes están no-correlacionados: es decir que siempre podemos restringir la búsqueda en el espacio que da estimaciones no-correlacionadas con las previas. Esto corresponde a una ortogonalización en un espacio transformado adecuado.

5.3.2 La kurtosis como una medida de la no-gaussianidad

Para medir la no-gaussianidad de las variables a ser analizadas por el ICA, es necesario contar con un índice numérico que nos dé tal información. Para seguir con la nomenclatura utilizada, supongamos a y nuestra variable de salida y supongamos que está centrada, $E[y] = 0$ y $\sigma^2(y) = 1$, por simplicidad.

La medida clásica de la no-gaussianidad es la kurtosis, que está definida como

$$\text{kurt}(y) = E\{y^4\} - 3\left(E\{y^2\}\right)^2$$

Como supusimos $\sigma^2(y) = 1$, el lado derecho se simplifica a $E\{y^4\} - 3$, lo que muestra que la kurtosis es simplemente una versión normalizada del cuarto momento $E\{y^4\}$. Para una variable gaussiana y, el cuarto momento es igual a $3\left(E\{y^2\}\right)^2$, es decir que la kurtosis es cero para variables aleatorias gaussianas. Para la gran mayoría de variables aleatorias no-gaussianas, la kurtosis es distinta de cero.

La kurtosis puede tomar valores negativos o positivos. Cuando la kurtosis es menor que cero, las variables aleatorias se denominan sub-gaussianas o platikúrticas , y cuando la kurtosis toma valores positivos se denominan super-gaussianas o leptokúrticas. En el

caso de las super-gaussianas típicamente tienen una forma puntiaguda de función de probabilidad de densidad con colas pesadas. Una distribución típica es la de Laplace cuya fórmula de función de probabilidad de densidad se muestra a continuación y su gráfica en la Figura 5.2):

$$p(y) = \frac{1}{\sqrt{2}} \exp\left(\sqrt{2}|y|\right).$$

Las variables sub-gaussianas se caracterizan por tener una función de probabilidad de densidad "plana", cuya distribución típica es la uniforme,

$$p(y) = \begin{cases} \dfrac{1}{2\sqrt{3}} & \text{si } |y| \le \sqrt{3} \\ 0 & \text{en otro caso} \end{cases}.$$

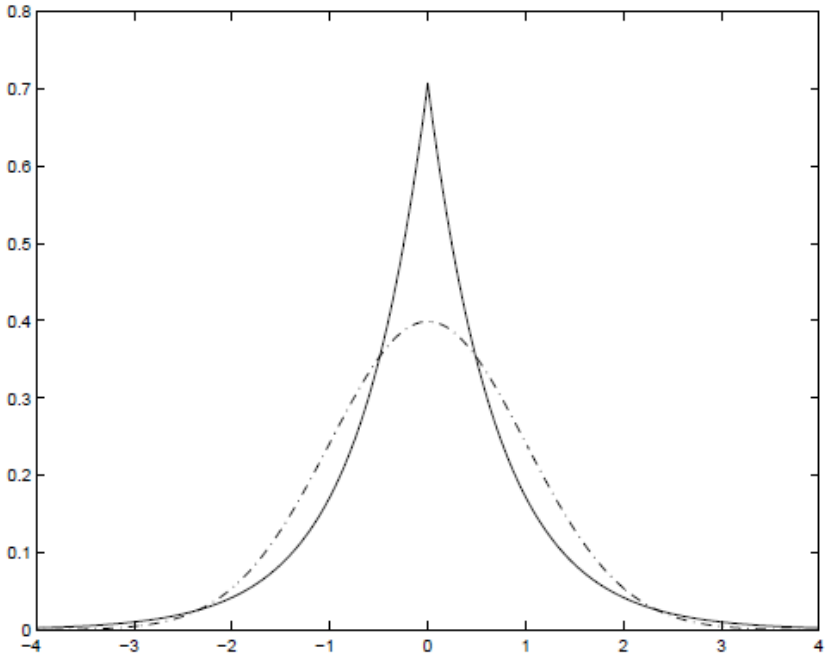

Figura 5.2 Distribución de densidad de una distribución de Laplace, típica supergaussiana. La línea punteada representa una distribución gaussiana

La no-gaussianidad se mide típicamente por el valor absoluto de la kurtosis. El cuadrado de la kurtosis también es utilizado. Estos valores son cero para variables gaussianas y mayores que cero para la mayoría de las variables no-gaussianas. Las variables aleatorias no-gaussianas con kurtosis cero son un caro excepcional.

La kurtosis como medida de la no-gaussianidad ha sido ampliamente usado en ICA, esto es debido a su simplicidad, tanto teórica como computacional.

La simplicidad del análisis teórico se debe a las siguiente propiedades lineales: sean x_1 y x_2 son dos variables independientes aleatorias, se cumple que:

$$\text{kurt}\left(x_1 + x_2\right) = \text{kurt}\left(x_1\right) + \text{kurt}\left(x_2\right)$$

$$\text{kurt}\left(\alpha x_1\right) = \alpha^4 \text{kurt}\left(x_1\right)$$

con α escalar

La desventaja que tiene la kurtosis como medida de la no-gaussianidad de variables aleatorias es que puede ser sensible a valores atípicos. Los valores de una muestra pueden depender de pocas observaciones en las colas de distribución, lo que puede originar observaciones irrelevantes o erróneas. La kurtosis no representa una medida robusta de la no-gaussianidad. Por lo tanto, otras medidas de la no-gaussianidad pueden resultar mejores que la kurtosis en ciertas ocasiones.

A continuación describiremos la negentropía como una medida alternativa de la no-gaussianidad.

5.3.3 Negentropía

Otra medida de la no-gaussianidad es la negentropía, basada en información de la cantidad teórica de la entropía. Entropía es el concepto básico de la información teórica. La entropía de una variable aleatoria es el grado de información que da la observación de la variable. Cuanto más aleatoria es la variable, mayor es la entropía. La entropía está íntimamente relacionada con la longitud del código de la variable, en realidad bajo

algunas suposiciones simplicadoras, la entropía es el código de la longitud de la variable aleatoria.

La entropía H está definida para una variable aleatoria discreta Y como

$$H(Y) = -\sum_i P(Y = a_i) \log P(Y = a_i)$$

donde los a_i son los valores posibles de Y. Esta definición puede ser generalizada para variables aleatorias de valores continuos, en cuyo caso puede denominarse entropía diferencial.

La entropía diferencial H de una vector aleatorio con densidad $f(y)$ es definido como

$$H(y) = -\int f(y) \log f(y)\, dy$$

Una variable gaussiana tiene la mayor entropía entre todas las variables aleatorias de igual varianza. Este resultado importante significa que la entropía puede efectivamente usarse como medida de la no-gaussianidad. Ciertamente, esto muestra que la distribución gaussiana es lo más aleatorio o lo menos estructurado de todas las distribuciones. La entropía es pequeña para distribuciones que claramente están concentradas en ciertos valores, por ejemplo cuando la variable tiene una función de probabilidad de densidad muy "puntiaguda".

La negentropía siempre tiene valores no-negativos, y es cero si y solo si y tiene una distribución gaussiana. Además de la característica recién mencionada, tiene la propiedad de ser invariante para las transformaciones lineales inversibles (Comon, 1994; Hyvärinen, 1999).

La ventaja de usar la negentropía o la entropía diferencial como medida de la no-gaussianidad es que está bien justificada por la teoría estadística. En lo que a propiedades estadísticas concierne, la negentropía es en un sentido el estimador óptimo de no-gaussianidad.

Como desventaja podemos decir que su cálculo computacional es dificultoso.

5.4 Aplicaciones de ICA

5.4.1 Separación Ciega de Fuentes (BSS)

Es la aplicación clásica de ICA y la utilizaremos en nuestro caso. En la separación ciega de fuentes, los valores observados de x corresponden a la realización de una señal en tiempo discreto de dimensión N del tipo $x(t);t=1,2,....$ Los componentes independientes, se los llamará fuentes $s_i(t)$, que son por lo general señales originales incorruptas o señales de ruido. Un ejemplo clásico y simplificado para que resulte ilustrativo, es el problema de la fiesta cóctel en el que se supone que varias personas conversan frente a micrófonos; a menos que cada uno hable por turnos, los micrófonos recogerán la *superposición* de todas las voces. Cómo hacer para, a partir de esas grabaciones, separar una voz de la otra. En principio, esto se corresponde con el modelo de datos de ICA, donde $x_i(t)$ es el registro del i-ésimo micrófono y $s_i(t)$ son las formas de onda de las voces. Una aplicación más práctica es la reducción de ruido, que es justamente la que hemos desarrollado. Si una de las fuentes es la original, es decir la señal limpia, las otras fuentes son las señales ruidosas. Estimar la fuente incorrupta es en realidad una operación de eliminación de ruido.

Otro ejemplo importante es la aplicación a los registros potenciales de superficie de electroencefalografía (EEG) en humanos. Las señales eléctricas originadas del cerebro son bastante débiles para los sensores, en el rango de microvoltage y además existen grandes componentes de artefacto que vienen del movimiento ocular y muscular. Ha sido un desafío muy grande eliminar esos artefactos sin alterar las señales provenientes del cerebro. ICA está idealmente diseñado para esta tarea debido a que las señales cerebrales son independientes de aquellas que generan artefactos (Vigário, 1997; Vigário et al., 1998).

Similarmente, ICA puede usarse para la descomposición de potenciales evocados medidos por EEG y MEG que son aplicaciones de considerable interés en la neurociencia. Aplicaciones en el campo de las imágenes, reportadas por (Vigário, 2000). Otras aplicaciones que involucran imagenología cerebral obtenida con

resonancia magnética funcional (fMRI) reportada por (Calhoun y Adali, 2006; Nakamura et al., 2006).

Debido a que muchos investigadores en ICA han desarrollado sus aplicaciones para la separación ciega de fuentes, muchos autores tratan el problema de ICA pero lo llaman BSS. Sin embargo es necesario aclarar nuevamente la diferencia entre un problema teórico o modelado de datos con diferentes aplicaciones y la separación ciega de fuentes, que es una aplicación que puede ser resuelta usando varias construcciones teóricas, incluyendo, pero no limitando al ICA. Para ser precisos, el problema de la separación ciega de fuentes puede resolverse usando diferentes muy distintos a ICA. En particular métodos que utilizan información de frecuencia o propiedades espectrales. Estos métodos pueden ser usados para señales correlacionadas en el tiempo, que es por supuesto un caso usual de la BSS, pero no en muchas otras aplicaciones de ICA (Hyvärinen, 1999).

5.4.2 Extracción de Patrones

En la ecuación (5.2), las columnas de $\mathbf{A}$ representas patrones y s_i es el coeficiente del i-ésimo patrón en el vector de datos observados $\mathbf{x}$. El uso de ICA para la extracción de patrones está motivado por la teoría de la reducción de la redundancia. Fueron analizados algunos trabajos destacados muy recientes que lo demuestran (Zarzoso y Comon, 2010; Lan et al., 2010; Kharabian et al. 2009).

5.4.3 Otras aplicaciones

Debido a la estrecha relación entre ICA y la búsqueda de proyección por un lado y entre ICA y el análisis de factores por el otro, sería posible usar ICA en muchas de las aplicaciones donde se utilicen estos métodos. Esto incluye análisis de datos en áreas como la economía, psicología y otras áreas sociales, tanto como la estimación de densidad y la regresión, (Hyvärinen, 1999).

5.5 Separación Ciega de Fuentes

El problema de la Separación Ciega de Fuentes ha recibido la atención de la comunidad científica, que en algunos casos también lo nombraron como procesamiento ciego de arreglos, análisis de componentes independientes, estimación de preservación de forma de onda, convolución ciega, etc. En todos los casos, el modelo está planteado como: Dadas M señales estadísticamente independientes, de las cuales es posible observar N combinaciones lineales ruidosas; el problema consiste en recuperar las señales originales a partir de su mezcla. La calificación de ciega se refiere a que no hay información disponible a priori acerca de los coeficientes de la mezcla, (Cardozo y Laheld, 1994). A continuación se amplía el modelo que representa este problema. y se mencionan las aplicaciones más relevantes.

5.5.1 Aplicaciones de la Separación Ciega de Fuentes

Estudio de señales de naturaleza biomédica. En particular, los algoritmos de Separación de Fuentes se han empleado para el análisis del electroencefalograma, de las resonancias magnéticas (Vigário et al, 2000) el registro del electrocardiograma fetal (Kharabian et al, 2009).

Tratamiento de señales de audio. Evidentemente, la separación de voces (problema del c*ocktail party* en el que se supone que varias personas conversan frente a micrófonos; a menos que cada uno hable por turnos, los micrófonos recogerán la *superposición* de todas las voces. Cómo hacer para, a partir de esas grabaciones, separar una voz de la otra) parece ser una aplicación natural de los algoritmos de Separación de Fuentes. Puede ser útil en videoconferencias, conciertos y, en general, en cualquier situación en la que se desee destacar una señal sobre el nivel del ruido o de las interferencias (Nakamura et al, 2006).

Extracción de patrones de una señal. Mediante las técnicas de Separación de Fuentes, se trata de descomponer la señal de interés en otras, independientes entre sí, que pongan mejor de manifiesto la información. Esta aplicación se considera una extensión del

clásico Análisis de Componentes Principales (Haykin, 1994), otras aplicaciones más recientes se pueden encontrar en (Langley, 2006)

5.5.2 Algunos Conceptos Importantes: Fuentes, Observaciones y Señales de Salida

Supondremos que las **fuentes** son el producto de procesos estocásticos, de media cero y *estadísticamente independientes entre sí*. Las llamaremos con la letra s (source – fuente).

La hipótesis de que la media sea cero no es en absoluto restrictivo, sin embargo sí lo es la característica de estadísticamente independientes entre sí, que hace referencia al hecho de que los valores de ninguno de los componentes da información sobre los valores de los demás componentes (Hyvärinen, 2001).

Dadas M fuentes distintas, $s_1(t), s_2(t), ..., s_M(t)$, se agruparán en el vector

$$\mathbf{s}(t) = \left[s_1(t), s_2(t) ..., s_M(t) \right]^T$$

Las **observaciones** serán representadas por la letra x. Dadas N observaciones que representan N sensores, se denotarán como $x_1(t), x_2(t), ..., x_N(t)$ y serán agrupadas en el vector

$$\mathbf{x}(t) = \left[x_1(t), x_2(t), ... x_N(t) \right]^T$$

El sistema de separación nos devolverá M **señales de salida** $y_1(t), y_2(t), ..., y_M(t)$ que deberán reproducir, en una situación ideal, fielmente a las *fuentes.* El vector de salida quedará definido así:

$$\mathbf{y}(t) = \left[y_1(t), y_2(t), ..., y_M(t) \right]^T$$

5.5.3 Mezcla lineal, e invariante en el tiempo

La **mezcla** es el proceso por el cual las fuentes se transforman en observaciones. Como ya lo enunciamos previamente, en esta tesis trabajaremos bajo la hipótesis de que la mezcla es *lineal e invariante* en el tiempo, que, a su vez, se dividen en *instantáneas* y *convolutivas.* Nos concentraremos en las primeras que, si bien constituyen un problema más simple, proporcionan la base para estudiar los casos más generales.

Las Mezclas Lineales e Instantáneas.

Sea el vector $\mathbf{s}(t)=\left[s_1(t), s_2(t)..., s_M(t)\right]^T$ que contiene a las M señales *fuente*, el vector $\mathbf{x}(t)=\left[x_1(t), x_2(t),...x_N(t)\right]^T$, con las N *observaciones* e $\mathbf{y}(t)=\left[y_1(t), y_2(t),..., y_M(t)\right]^T$ el vector que contiene las M salidas.

Se dice que la *mezcla* es *lineal* e *instantánea* si

$$\mathbf{x}(t)=\mathbf{A}\cdot\mathbf{s}(t) \tag{5.10}$$

donde $\mathbf{A}$ es una matriz $N\times M$ que recibe el nombre de *matriz de mezcla*. Nótese que la transformación es *lineal e invariante en el tiempo,* donde N es el número de sensores y M es el número de fuentes.

Por hipótesis, tanto $\mathbf{A}$ como $\mathbf{s}(t)$ *son desconocidos* y sólo se cuenta con las o*bservaciones y la hipótesis de independencia estadística* entre las fuentes.

Las salidas se determinan como

$$\mathbf{y}(t)=\mathbf{B}\cdot\mathbf{x}(t) \tag{5.11}$$

siendo $\mathbf{B}$ una matriz $M\times N$ que recibe el nombre de *matriz de separación*. La relación que liga *fuentes* y *salidas* es, por lo tanto,

$$\mathbf{y}(t) = \mathbf{B} \cdot \mathbf{x}(t) = \mathbf{B} \cdot \mathbf{A} \cdot \mathbf{s}(t) = \mathbf{G} \cdot \mathbf{s}(t) \tag{5.12}$$

donde $\mathbf{G}$ es una matriz $M \times M$ a la que se llama *matriz global de transferencia*. La Figura 5.3) muestra la relación entre las distintas señales.

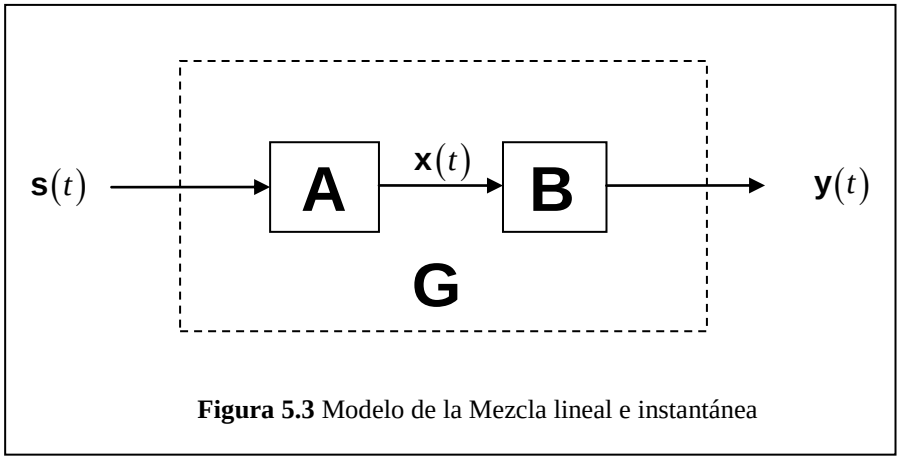

Figura 5.3 Modelo de la Mezcla lineal e instantánea

Este modelo puede generalizarse suponiendo que las fuentes están filtradas por el medio antes de llegar a los sensores. Esto significa que la i-ésima observación se construye como sigue:

$$x_i(t) = \sum_{j=1}^{N} h_{ij}(t) * s_j(t) \tag{5.13}$$

donde '$*$' denota la operación de convolución y $h_{ij}(t)$ es la *respuesta al impulso* del medio por el que se propaga la *j*-ésima fuente hasta llegar al *i*-ésimo sensor.

Las Mezclas Lineales y Convolutivas.

El modelo de *mezcla convolutiva* recoge básicamente dos fenómenos:

- Que las *fuentes* no alcancen *simultáneamente* todos los *sensores*. En este caso, el medio de propagación tan sólo retrasa las señales.

- Que las fuentes sean además distorsionadas por el canal.

La Transformada de Fourier permite convertir (5.13) en una relación algebraica:

$$X_i(f) = \sum_{i=1}^{N} H_{ij}(f) \cdot S_i(f) \tag{5.14}$$

por lo que, *para cada frecuencia,* (5.13) nos define una *mezcla lineal* e *instantánea*.

Separación de Fuentes Basada en la Independencia

El problema de la BSS se limita encontrar los coeficientes de la matriz de separación $\mathbf{B}$ en (5.11). Para ello sería necesario encontrar un método general que pueda ser aplicado en muchas circunstancias y que provea una solución general además a la representación de datos multivariantes. Contamos únicamente con las observaciones, en el vector $\mathbf{x}(t)$ y queremos hallar una matriz $\mathbf{B}$ tal que la representación está dada por las fuentes de las señales, en el vector $\mathbf{s}(t)$. Si consideramos la independencia estadística de las señales, podríamos resolver el problema. En realidad, si las señales son *no Gaussianas*, esto es suficiente para determinar los coeficientes de $\mathbf{B}$ a los que llamaremos b_{ij}. De este modo las señales:

$$\begin{aligned} y_1(t) &= b_{11}x_1(t) + ... + b_{1N}x_N(t) \\ &. \\ &. \\ y_M(t) &= b_{M1}x_1 + ... + b_{MN}x_N(t) \end{aligned} \tag{5.15}$$

son estadísticamente independientes. Si las señales $y_i(t); i = 1,...,M$ son independientes, luego son iguales a las señales $s_i(t); i = 1,...,M$. Basándonos en esta propiedad de la independencia estadística, se podrían calcular los coeficientes de la matriz $\mathbf{B}$. Entonces en el problema de la separación ciega de fuentes, las señales originales son "componentes independientes" del conjunto de datos, (Hyvärinen et al, 2001).

Una vez planteadas la hipótesis y establecidas las restricciones del modelo, estamos en condiciones de plantear una solución al sistema (5.10), repetido aquí por conveniencia

$$\mathbf{x}(t) = \mathbf{A} \cdot \mathbf{s}(t) \tag{5.16}$$

La solución estará dada por la matriz $\mathbf{B}$, donde $\mathbf{B}$ es la inversa de $\mathbf{A}$. Proponemos para ello hallar la matriz $\mathbf{B}$ mediante la ecuación de estabilidad de sistemas lineales de Liapunov, que permite hallar una matriz incógnita bajo ciertas restricciones. Para ello, es necesario hacer previamente una introducción teórica acerca de la estabilidad de los sistemas.

5.6 Estabilidad de sistemas

A menudo utilizamos los conceptos de estabilidad e inestabilidad en la vida cotidiana. Dar la característica de estable a un objeto o situación como la salud de una persona o la moneda de un país, por ejemplo, nos resulta familiar y de uso común. Incluso, en muchas áreas del conocimiento, se manejan estos conceptos de manera intuitiva. En ingeniería hablamos de que una estructura es estable, en química se relaciona la estabilidad con las características de las reacciones, un economista puede utilizar el concepto para describir el precio de un producto determinado, en física lo corresponderíamos, por ejemplo, con el movimiento de una partícula, etc. Pero un concepto que aparece frecuentemente en todas las ciencias, merece ser definido en términos precisos (Álvarez Picaza et al, 2006).

No fue sino hasta 1892 cuando Aleksandr Liapunov (1857–1918), matemático e ingeniero ruso estableció las bases de la teoría que hoy lleva su nombre. Formuló de manera precisa el concepto de estabilidad y ese ha sido el punto de partida para establecer otras variantes de tan importante concepto.

Definiremos los conceptos de estabilidad sobre sistemas de la forma

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}\left[t, \mathbf{x}(t)\right], \\ \mathbf{x}(t_0) &= \mathbf{x}_0 \end{aligned} \tag{5.17}$$

donde $\mathbf{f}\left[t, \mathbf{x}(t)\right]: \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$ es una función continua. La solución para este sistema está dado por $\mathbf{x}(t)$.

La estabilidad de un sistema se determina en un punto específico, punto de equilibrio. En este punto de equilibrio todas las derivadas del sistema valen cero, esto nos dice que cuando un sistema está en su punto de equilibrio, allí permanece. La definición de punto de equilibrio es la siguiente:

Definición 5.2: Para un sistema como (5.17) se denomina punto de equilibrio $\mathbf{x}_e$ al punto que verifica

$$\mathbf{f}\left[t, \mathbf{x}_e\right] = 0_n$$

Para un sistema lineal e invariante en el tiempo del tipo $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)$, se cumple que $\mathbf{f}\left[\mathbf{x}(t)\right] = \mathbf{A}\mathbf{x}(t)$ y tiene solamente un estado de equilibrio si $\mathbf{A}$ es no singular; en efecto,

$$\mathbf{f}\left[\mathbf{x}_e\right] = \mathbf{A}\mathbf{x}_e = 0_n \Rightarrow \mathbf{x}_e = \mathbf{A}^{-1} 0_n = 0_n .$$

Si $\mathbf{A}$ es singular, entonces existe $\mathbf{x}_a \neq 0_n$ tal que $\mathbf{A}\mathbf{x}_a = 0_n$, por lo que

$$\mathbf{f}\left[\alpha \mathbf{x}_a\right] = \mathbf{A}\alpha\mathbf{x}_a = \alpha\mathbf{A}\mathbf{x}_a = 0_n \Rightarrow \mathbf{x}_e = \alpha\mathbf{x}_a ; \ \forall \alpha ,$$

entonces tiene infinitos puntos de equilibrio. Existen también sistemas sin puntos de equilibrio, por ejemplo $\dot{x}(t) = 1$.

5.6.1 Definiciones de estabilidad

Consideremos $\mathbf{x}(t)$ con $t \in [t_0, \infty)$, que es solución del sistema (5.17) con $\mathbf{x}_e = 0_n$ el punto de equilibrio, mencionamos las siguientes definiciones:

i. Estable (E.) en $t \in [t_0, \infty)$, si dado un $\varepsilon > 0$ existe un $\delta = \delta(\varepsilon, t_0) > 0$ tal que $|\mathbf{x}(t_0)| < \delta$, implica que la solución existe y cumple $|\mathbf{x}(t)| < \varepsilon$ para $t \geq t_0$. Esta definición también se la conoce como estabilidad en el sentido de Liapunov.

ii. Asintóticamente Estable (A.E.) en $t \in [t_0, \infty)$, si es estable y además $\lim_{t \to \infty} \mathbf{x}(t) = 0_n$.

iii. Inestable (I.) en $t \in [t_0, \infty)$, si no es estable.

iv. Uniformemente Estable (U.E) en $t \in [t_0, \infty)$, es la misma definición de (E.) pero con $\delta = \delta(\varepsilon) > 0$, es decir, independiente de t_0.

v. Uniforme y Asintóticamente Estable (U.A.E.) si es (U.E.) y existe $\delta > 0$ tal que para cualquier $\varepsilon > 0$, existe $T = T(\varepsilon)$ tal que si $|\mathbf{x}(t_1)| < \delta$ con $t_1 \geq t_0$, entonces $|\mathbf{x}(t)| < \varepsilon$ para $t_1 + T(\varepsilon) \leq t < \infty$. En otras palabras, $\mathbf{x}(t) \to 0_n$, para $(t - t_1) \to \infty$ uniformemente en $|\mathbf{x}(t_1)| < \delta$.

A partir de estas definiciones surge el siguiente lema

Lema 5.1: Las siguientes implicaciones son válidas. La demostración de este lema puede ser consultado en (Rautenberg y D'Attellis, 2004).

$$\begin{array}{ccc} (U.A.E) & \Rightarrow & (A.E.) \\ \Downarrow & & \Downarrow \\ (U.E) & \Rightarrow & (E.) \end{array}$$

Un concepto importante en la teoría de la estabilidad de sistemas es el de *matriz de transición* que será mencionado a continuación.

5.6.2 Matriz de Transición

Sea la siguiente ecuación matricial diferencial

$$\begin{aligned}\dot{\mathbf{X}}(t) &= \mathbf{F}\left[\mathbf{X}(t), t\right] = \mathbf{A}(t)\mathbf{X}(t)\\ \mathbf{X}(t_0) &= \mathbf{X}_0;\end{aligned} \tag{5.18}$$

la matriz solución del sistema con condición inicial (5.18) es $\mathbf{X}(t)$ y recibe el nombre de *matriz fundamental*. Si consideramos ahora una matriz $\mathbf{C} \in \mathbb{R}^{n\times n}$ de modo que el sistema (5.18) verifique

$$\begin{aligned}&\dot{\mathbf{X}}(t)\mathbf{C} = \mathbf{A}(t)\,\mathbf{X}(t)\mathbf{C};\\ &\frac{d\left[\mathbf{X}(t)\mathbf{C}\right]}{dt} = \mathbf{A}(t)\left[\mathbf{X}(t)\mathbf{C}\right];\end{aligned}$$

es decir que $\mathbf{X}(t)\mathbf{C}$ también es una matriz fundamental, pues las matrices fundamentales difieren entre sí en una matriz constante multiplicativa.

A partir de una condición inicial dada, es posible determinar de forma unívoca la matriz fundamental, o dicho de otro modo, podemos determinar todas las matrices fundamentales como

$$\mathbf{X}(t) = \Phi(t, t_0)\mathbf{C},$$

con $\Phi(t, t_0)$, matriz fundamental que verifica que $\Phi(t_0, t_0) = \mathbf{I}$ y la matriz $\mathbf{C}$ quedará determinada cuando se fija la condición inicial del sistema.

Si planteamos el sistema (5.18) de forma vectorial, tenemos

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}(t)\mathbf{x}(t)\\ \mathbf{x}(t_0) &= \mathbf{x}_0;\end{aligned} \tag{5.19}$$

con $\mathbf{x}(t) \in \mathbb{R}^n$; en este caso, la matriz $\Phi(t,t_0)$ sigue siendo solución del sistema, pero en forma vectorial resulta esta solución: $\mathbf{x}(t) = \Phi(t,t_0)\mathbf{c}$, donde $\mathbf{c} \in \mathbb{R}^n$ y cumple que

$$\mathbf{x}(t_0) = \Phi(t_0,t_0)\mathbf{c} = \mathbf{x}_0$$

de lo que deducimos que $\mathbf{c} = \mathbf{x}_0$; por lo tanto la ecuación fundamental del sistema (5.19) es única y resulta $\mathbf{x}(t) = \Phi(t,t_0)\mathbf{x}_0$. La matriz $\Phi(t,t_0)$ se denomina *matriz de transición* y brinda resultados muy importantes en el análisis de la estabilidad de sistemas. El significado físico de esta matriz es muy claro, gobierna las trayectorias de los estados de un intervalo de tiempo en el cual la entrada es igual a cero. Puede interpretarse también como una transformación lineal que mapea el vector de estado $\mathbf{x}(0)$ en t_0 en el vector de estado $\mathbf{x}(t)$ en t, (Rautenberg y D'Attellis, 2004).

5.7 Teoría de la estabilidad de Liapunov

Consideremos el sistema siguiente

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}\left[t,\mathbf{x}(t)\right], \\ \mathbf{x}(t_0) &= \mathbf{x}_0; \end{aligned} \tag{5.20}$$

donde $\mathbf{f}\left[t,\mathbf{x}(t)\right]: \mathbb{R}\times\mathbb{R}^n \to \mathbb{R}^n$ es una función continua y se cumple que para ciertos estados $\mathbf{x}_e$, $\mathbf{f}\left[t,\mathbf{x}_e(t)\right] = 0$. Definimos como *función de Liapunov* a aquella que determina la distancia del vector $\mathbf{x}(t)$ al origen de coordenadas (analizamos la estabilidad en el origen de coordenadas por simplicidad, pues mediante un cambio de coordenadas se produce el traslado del punto de equilibrio).

Este método se denomina **segundo método de Liapunov** y brinda información sobre la estabilidad de los puntos de equilibrio de sistemas lineales y no lineales, sin necesidad de conocer las soluciones del mismo.

Definición 5.3. Se denomina función de Liapunov del sistema (5.20) a una función $v\left[t,\mathbf{x}(t)\right]:\mathbb{R}\times\mathbb{R}^n\to\mathbb{R}$, que cumple para $t\geq t_0$ y para un entorno del origen:

i. $v\left[t,\mathbf{x}(t)\right]$ y sus primeras derivadas en $\mathbf{x}(t)$ y en t existen y son continuas.

ii. $v\left[t,0_n\right]=0$.

iii. $v\left[t,\mathbf{x}(t)\right]\geq\alpha\left(\left|\mathbf{x}(t)\right|\right)>0$ para $\mathbf{x}(t)\neq 0_n$ y $t\geq t_0$, donde $\alpha(0)=0$ y $\alpha(\xi)$ es una función continua y no decreciente de la variable ξ.

iv. $\dfrac{d}{dt}v\left[t,\mathbf{x}(t)\right]=\nabla_{\mathbf{x}}v\cdot\dot{\mathbf{x}}(t)+\dfrac{\partial v}{\partial t}<0$ para $\mathbf{x}(t)\neq 0_n$.

v. $v\left[t,\mathbf{x}(t)\right]\leq\beta\left(\left|\mathbf{x}(t)\right|\right)>0$ para $\mathbf{x}(t)\neq 0_n$ y $t\geq t_0$; donde $\beta(0)=0$ y $\beta(\xi)$ es una función continua y no decreciente de la variable ξ.

Este método, si bien es una herramienta muy poderosa, tiene el inconveniente que encontrar las funciones resulta un problema complejo en muchos casos.

Observación: Debido a que la siguiente definición será utilizada en la demostración de un teorema que se enunciará más adelante, creímos oportuno agregarla para una mejor comprensión de lo aquí de demostrará.

Previamente dijimos que la matriz de transición es una herramienta muy útil para definir la estabilidad de sistemas lineales, es más, la estabilidad de estos sistemas depende únicamente de su matriz de transición. Es posible y útil redefinir la estabilidad en términos de la matriz de transición, sin embargo, a los fines de este trabajo, sólo definiremos la estabilidad de tipo uniforme y asintóticamente estable, pues es la que nos interesa y es además análoga a la definición 5.3 v).

Proposición 5.1: Para un sistema lineal cuya matriz de transición es $\Phi(t,t_0)$, el origen de coordenadas es uniforme y asintóticamente estable sí y sólo sí existen las constantes $k>0,\alpha>0;$ tal que $\|\Phi(t,t_1)\| \le ke^{-\alpha(t-t_1)}$ $;t_0 \le t_1 \le t < \infty$.

Debido a que la demostración de esta proposición no es en particular de nuestro interés, la obviaremos.

En la sección siguiente describiremos la teoría de Liapunov para la estabilidad de sistemas lineales, que es el tópico que nos interesa y utilizaremos los resultados expuestos aquí.

5.8 Teoría de Liapunov para la estabilidad de sistemas lineales

Sea el sistema lineal uniforme y asintóticamente estable

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}(t)\mathbf{x}(t), \\ \mathbf{x}(t_0) &= \mathbf{x}_0; \end{aligned} \tag{5.21}$$

para el que se cumple que $\|\mathbf{A}(t)\|_2 \le r(t_0)$ y para el que existe

$$\int_t^\infty \|\Phi(\tau,t)\|^2 \, d\tau$$

para alguna norma. Este requerimiento se cumple para los sistemas uniformes y asintóticamente estables, pues por la proposición 5.1, satisfacen $\|\Phi(t,t_1)\| \le ke^{-\alpha(t-t_1)}$, luego

$$\int_t^\infty \|\Phi(\tau,t)\|^2 \, d\tau \le k^2 \int_t^\infty e^{-2\alpha(\tau-t)} d\tau = \frac{k^2}{2\alpha} < \infty$$

El teorema que sigue, establece la construcción de una función de Liapunov mediante una integral y será la base la definición de la ecuación de Liapunov, que es la que utilizamos en el algoritmo de BSS.

Teorema 5.1. Para el sistema de la forma (5.20), $v\left[t,\mathbf{x}(t)\right]=\mathbf{x}^T(t)\mathbf{P}(t)\mathbf{x}(t)$ es una función de Liapunov, con $\mathbf{P}(t)$ definida como

$$\mathbf{P}(t)=\int_t^{\infty}\Phi^T(\tau,t)\mathbf{Q}(\tau)\Phi(\tau,t)d\tau \tag{5.22}$$

donde $\mathbf{Q}(t)$ es una matriz continua cualquiera, simétrica definida positiva, que además cumple que $\|\mathbf{Q}(t)\|\le k(t_0)$ y $\mathbf{Q}(t)-\varepsilon\mathbf{I}$ es definida positiva para $t\ge t_0$ si $\varepsilon>0$ es lo suficientemente pequeño.

Demostración El lector puede consultarla en (Rautenberg y D'Attellis, 2004)

Esta construcción de funciones de Liapunov, involucra la integral

$$\mathbf{P}(t)=\int_t^{\infty}\Phi^T(\tau,t)\mathbf{Q}(\tau)\Phi(\tau,t)d\tau\,.$$

La matriz $\mathbf{P}(t)$ satisface la siguiente ecuación diferencial

$$\frac{d\mathbf{P}(t)}{dt}=-\mathbf{A}^T(t)\mathbf{P}(t)-\mathbf{P}(t)\mathbf{A}(t)-\mathbf{Q}(t) \tag{5.23}$$

Luego de la exposición de estos conceptos importantes, podemos mencionar el siguiente teorema que establece condiciones necesarias y suficientes para la estabilidad uniforme y asintótica.

Teorema 5.2: El sistema definido en (5.19) es uniforme y asintóticamente estable, sí y solo sí, la solución $\mathbf{P}(t)$ de la ecuación (5.23) es definida positiva.

Demostración

⇒)

Si el sistema es uniforme y asintóticamente estable, con las condiciones del teorema 5.1 es posible definir $\mathbf{P}(t)$ a partir de la ecuación (5.23).

⇐)

Si $\mathbf{P}(t) > 0$, entonces $v\left[t, \mathbf{x}(t)\right] = \mathbf{x}^T(t)\mathbf{P}(t)\mathbf{x}(t)$ es una función de Liapunov del sistema y $\mathbf{x}(t) \xrightarrow{t\to\infty} 0_n$ uniformemente.

■

Este teorema es de suma importancia para la estabilidad de sistemas lineales. La ecuación (5.23) nos da un método directo para obtener la matriz $\mathbf{P}(t)$. Como dijimos $\mathbf{P}(t)$ es simétrica, por lo tanto necesitamos resolver $n(n+1)/2$ ecuaciones diferenciales escalares para poder hallar una solución a (5.23)

En particular, para sistemas del tipo

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}(t)\mathbf{x}(t), \\ \mathbf{x}(t_0) &= \mathbf{x}_0; \end{aligned}$$

lineales e invariantes en el tiempo, que es el caso que nos interesa resolver, la ecuación (5.23) se convierte en

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} = -\mathbf{Q} \tag{5.24}$$

pues $\mathbf{P}(t) = \mathbf{P}$ y por lo tanto $\dot{\mathbf{P}}(t) = \dot{\mathbf{P}} = 0_{n\times n}$.

Esta ecuación es la conocida ecuación de Liapunov. A la luz de ella, la estabilidad se basa en encontrar **P** eligiendo previamente **Q**.

5.9 Uso de la ecuación de Liapunov para resolver la matriz de mezcla de BSS

Para lograr resolver la matriz de mezcla A de la ecuación (5.16) mediante la ecuación de Liapunov (5.24), planteamos la hipótesis de que el corazón es un sistema estable. Para ello, nos basamos en la homeostasis, que alude a la tendencia a mantener el equilibrio fisiológico por compensación química, y que permite, por lo tanto, considerar al corazón como un sistema biológico estable y retroalimentado. Esto convalidaría la aplicación de los criterios generales de estabilidad de un sistema físico, y la utilización de procedimientos computacionales propios para la separación de fuentes de señal.

La teoría general de estabilidad considera que un sistema físico causal está en equilibrio si, en ausencia de cualquier perturbación a la entrada, la salida permanece inalterada. También se admite que un sistema lineal e invariante en el tiempo es estable si la respuesta a una perturbación tiende al punto de equilibrio transcurrido un cierto tiempo. Contrariamente, será inestable, cuando la respuesta sea divergente, alejando al sistema del equilibrio que presentaba al momento de la perturbación.

Teniendo en cuenta estos conceptos, vamos a demostrar ahora cómo la resolución de la ecuación de Liapunov nos sirve para hallar una solución a la matriz de mezcla, para así eliminar el ruido interferente en señales eléctricas cardíacas.

La ecuación de Liapunov, (5.24) permite hallar **P** eligiendo previamente **Q**. **A** es una matriz de dimensión $(N \times N)$ y es constante. Recordemos que en el modelo de ICA, N es el número de observaciones y M es el número de fuentes. En nuestro caso particular, $N = M$, pues tenemos dos observaciones, donde aparecen mezcladas la señal de interés, el ECG y el ruido contaminante. Es decir que contamos con matrices cuadradas, lo que además simplifica el problema.

El sistema matricial de ecuaciones (5.24), tiene la forma general de un sistema del tipo

$$\mathbf{AX} + \mathbf{XB} = \mathbf{C}. \tag{5.25}$$

Si adecuamos ese sistema general al sistema particular de Liapunov, tenemos que

$$\begin{aligned} &\mathbf{AX} + \mathbf{XB} = \mathbf{C} \\ &\text{donde} \\ &\mathbf{B} = \mathbf{A}^T \\ &\mathbf{C} = -\mathbf{Q}. \end{aligned} \tag{5.26}$$

Si la matriz **Q** es definida positiva, entonces el sistema es asintóticamente estable. Por ejemplo, se puede escoger **Q** = **I**, la matriz identidad que es obviamente definida positiva, y resolver para **P**, y ver si ésta es también definida positiva.

5.9.1 Resolución del sistema de ecuaciones de Liapunov.

Para afrontar esta tarea, resolveremos como primer paso un sistema genérico, de la forma (5.26).

Si expandimos el sistema matricial de ecuaciones planteado en el sistema (5.26), obtenemos:

$$s_{ij} = \sum_{j=1}^{M} a_{ij} x_{ij} + \sum_{l=1}^{N} x_{il} b_{lk} \quad \text{para} \begin{cases} i = 1, 2, 3, \ldots, M \\ k = 1, 2, 3, \ldots, N. \end{cases} \tag{5.27}$$

Dada la naturaleza matricial de **X**, la solución de este sistema implica la resolución de un sistema de ecuaciones algebraicas lineales de $M \times N$ ecuaciones e igual número de incógnitas.

Una técnica para abordar la solución de este problema algebraico es reacomodar el sistema dado por (5.26) en la forma tradicional de un sistema de ecuaciones del tipo $\mathbf{Wz} = \mathbf{y}$, donde **z** es un vector de $M \times N$ componentes y que tiene la forma:

$$\mathbf{z} = \begin{bmatrix} x_{11} \\ x_{12} \\ \cdots \\ x_{1M} \\ x_{21} \\ x_{22} \\ \cdots \\ x_{2M} \\ \cdots \\ x_{N1} \\ \cdots \\ x_{NM} \end{bmatrix}. \qquad (5.28)$$

Podemos observar en (5.28) que las componentes del vector **z** son las mismas de la matriz **X**, del sistema (5.26), dispuestas en forma de una única columna, conformada por las componentes de cada una de las columnas de la matriz **X** colocadas en forma vertical.

De igual manera, el vector **y** del sistema modificado se obtiene colocando en forma vertical, seguidas una tras otra, las columnas de la matriz **C** del sistema original. La matriz **S** quedará también definida en un paso posterior.

El sistema matricial de ecuaciones que resulta de la aplicación del Segundo Método de Estabilidad de Liapunov, tiene la característica particular de que todas las matrices son cuadradas, por lo que $M = N$, sumado a lo ya establecido en (5.26), relativo a que la matriz **B** es la transpuesta de la matriz **A**.

La matriz **A** es definida positiva y, naturalmente, no singular. La matriz **C** puede ser escogida arbitraria y convenientemente. En nuestro caso particular escogimos una matriz aleatoria $N \times N$, que sea definida positiva.

La expansión de la expresión (5.27) para el caso particular de $N = 3$, resulta en la formación de dos matrices auxiliares, **U** y **V**, las cuales se muestran a continuación:

$$\mathbf{U}=\begin{bmatrix} a_{11} & a_{21} & a_{31} & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{12} & a_{22} & a_{32} & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{13} & a_{23} & a_{33} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{11} & a_{21} & a_{31} & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{12} & a_{22} & a_{32} & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{13} & a_{23} & a_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{11} & a_{21} & a_{31} \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{12} & a_{22} & a_{32} \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{13} & a_{23} & a_{33} \end{bmatrix} \tag{5.29}$$

$$\mathbf{V}=\begin{bmatrix} b_{11} & 0 & 0 & b_{12} & 0 & 0 & b_{13} & 0 & 0 \\ 0 & b_{11} & 0 & 0 & b_{12} & 0 & 0 & b_{13} & 0 \\ 0 & 0 & b_{11} & 0 & 0 & b_{12} & 0 & 0 & b_{13} \\ b_{21} & 0 & 0 & b_{22} & 0 & 0 & b_{23} & 0 & 0 \\ 0 & b_{21} & 0 & 0 & b_{22} & 0 & 0 & b_{23} & 0 \\ 0 & 0 & b_{21} & 0 & 0 & b_{22} & 0 & 0 & b_{23} \\ b_{31} & 0 & 0 & b_{32} & 0 & 0 & b_{33} & 0 & 0 \\ 0 & b_{31} & 0 & 0 & b_{32} & 0 & 0 & b_{33} & 0 \\ 0 & 0 & b_{31} & 0 & 0 & b_{32} & 0 & 0 & b_{33} \end{bmatrix} \tag{5.30}$$

Podemos observar que los *N* bloques que aparecen sobre la diagonal principal de la matriz **U** son iguales entre sí y que, además, están conformados por la transpuesta de la matriz **A** del sistema matricial original, dado en (5.26).

Por otra parte, la matriz **V** está conformada por los elementos de la matriz **B**, dispuestos en bandas diagonales.

No podemos dejar de notar que las matrices así obtenidas presentan un gran número de ceros.

De la suma de **U** y **V** se obtiene la matriz **S**:

$$\mathbf{S}=\begin{bmatrix} a_{11}+b_{11} & a_{21} & a_{31} & b_{12} & 0 & 0 & b_{13} & 0 & 0 \\ a_{12} & a_{22}+b_{11} & a_{32} & 0 & b_{12} & 0 & 0 & b_{13} & 0 \\ a_{13} & a_{23} & a_{33}+b_{11} & 0 & 0 & b_{12} & 0 & 0 & b_{13} \\ b_{21} & 0 & 0 & a_{11}+b_{22} & a_{21} & a_{31} & b_{23} & 0 & 0 \\ 0 & b_{21} & 0 & a_{12} & a_{22}+b_{22} & a_{32} & 0 & b_{23} & 0 \\ 0 & 0 & b_{21} & a_{13} & a_{23} & a_{33}+b_{22} & 0 & 0 & b_{23} \\ b_{31} & 0 & 0 & b_{32} & 0 & 0 & a_{11}+b_{33} & a_{21} & a_{31} \\ 0 & b_{31} & 0 & 0 & b_{32} & 0 & a_{12} & a_{22}+b_{33} & a_{32} \\ 0 & 0 & b_{31} & 0 & 0 & b_{32} & a_{13} & a_{23} & a_{33}+b_{33} \end{bmatrix}. \tag{5.31}$$

Una vez preparadas las matrices **U**, **V** y **S**, y el vector **y**, el sistema matricial original,

$$\begin{aligned} &\mathbf{AX}+\mathbf{XB}=-\mathbf{C}; \\ &\text{con} \\ &\mathbf{B}=\mathbf{A}^{T}, \end{aligned} \tag{5.32}$$

queda expresado en forma de un sistema algebraico, lineal, de primer orden, de $N\times N$ ecuaciones e igual número de incógnitas, dado por

$$\mathbf{Wz}=\mathbf{y}, \tag{5.33}$$

cuya solución consistirá de $N\times N$ valores para los elementos el vector **z**, los cuales, de acuerdo a (5.28), corresponden con las componentes de la matriz **X**, y constituyen la solución del sistema matricial dado por (5.26).

Es por ello que, después de resolver el sistema modificado dado por (5.33), utilizando algún método de resolución de sistemas de ecuaciones algebraicas lineales, sólo restaría reacomodar los elementos del vector **z**, en la forma matricial de **X**, como se muestra a continuación.

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{2} & \cdots & x_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ x_{N1} & x_{N2} & \cdots & n_{NN} \end{bmatrix}. \tag{5.34}$$

La matriz dada por (5.34) constituye la solución del sistema matricial dado por (5.26) y que representa la matriz requerida por el Segundo Método de Estabilidad de Liapunov, y es para nuestro problema de estudio la matriz de mezcla del modelo planteado por ICA en (5.1).

La resolución de la ecuación de Liapunov (5.24), nos permite hacer una analogía directa entre (5.11) y (5.33), con lo cual podemos encontrar la matriz de **B** en (5.11) y a partir de ella, separar las fuentes, señal de ECG de las interferencias, ruido, que en las observaciones aparecen mezcladas.-

6. Redes Neuronales

6.1. Introducción

Las Redes Neuronales Artificiales (ANN, Artificial Neural Networks) han capturado la atención de la comunidad científica durante años debido a que la construcción de un modelo matemático-computacional inteligente que emule el comportamiento humano es un logro buscado por los investigadores. Las redes neuronales pueden ser vistas como soluciones de tipo arquitectónicas a problemas de ingeniería, como el reconocimiento de patrones, la clasificación de los mismos y la optimización de procesos, (Sanchez-Sinencio y Lau, 1992). Es decir que la elección de la arquitectura de la red podría definirse más por una cuestión de gustos o preferencias y estilos que por una cuestión operativa. En el campo del análisis de las señales biológicas, en particular las de electrocardiogramas (ECG), el desarrollo de sistemas automáticos de detección basados en redes neuronales artificiales ha producido una enorme cantidad de resultados que evidencian la utilidad de las mismas y el interés por continuar en el avance de este tema. A groso modo, podríamos distinguir dos grandes grupos de modelos de redes neuronales: aquellos que involucran el aprendizaje supervisado y aquellos que no. Dentro del primer grupo, destacamos, además los sistemas basados en lógica difusa. Para la elaboración de este trabajo, cada uno de los modelos recién mencionados, fueron diseñados y aplicados a señales de ECG para la clasificación de latidos cardíacos. Sus resultados serán mencionados posteriormente.

Las RNA (Redes Neuronales Artificiales) son modelos computacionales no convencionales, inspirados en el funcionamiento del sistema nervioso humano. Están compuestas por elementos simples que operan en paralelo. Realizan procesos distribuidos, no-lineales. Como en la naturaleza, las funciones de la red están determinadas mayormente por las conexiones entre los elementos. El objetivo de la red neuronal es lograr realizar una tarea particular luego de que haya sido entrenada, ajustando valores de las conexiones entre elementos.

Las redes neuronales aprenden a partir de datos de entrenamiento. Esta *capacidad para adaptarse* a nuevos datos, su *habilidad para generalizar* y manejar información imperfecta o incompleta y la *no-linealidad* que les permite capturar interacciones complejas entre las variables de entrada y salida, son propiedades que convierten a las redes neuronales, útiles para la implementación de sistemas inteligentes aplicables a un amplio abanico de procesos donde el objeto de estudio es la señal de ECG, (Monzón, 2004).

Algunas de las tareas que pueden realizar las redes neuronales están listadas en la Tabla 6.1 (Simpson, 1992).

Con respecto al campo de aplicación de las RNA, el diagnóstico clínico fue una de las primeras áreas objeto de estudio. Su habilidad para reconocer relaciones temporales, espaciales o de otro tipo y realizar tareas de predicción, clasificación o estimación, las ha convertido en una herramienta más que válida para el análisis de relaciones multidimensionales existentes en los datos biológicos y que con otros métodos de procesamiento, no resultaban evidentes (Hammerstrom, 1993).

Tabla 6.1 Algunas de las tareas más habituales de las redes neuronales

Operación	**Entrada**	**Salida**
Clasificación	Patrón de entrada.	Clase representativa.
Apareo de patrones	Patrón de entrada.	Patrón de salida correspondiente.
Complementación de patrones	Patrón incompleto.	Patrón de salida que completa las partes perdidas del patrón de entrada.
Eliminación de ruido	Patrón corrompido por ruido.	Versión limpia del patrón de entrada.
Optimización	Patrón de entrada que representa un valor inicial para un problema específico de optimización.	Grupo de variables que representa una solución al problema.
Control	Un patrón de entrada que representa el estado actual de un controlador y una respuesta deseada para el controlador.	Secuencia de comandos apropiada que creará la respuesta deseada.

6.2. El modelo biológico

En nuestro cerebro tenemos aproximadamente 12 billones de células nerviosas o neuronas. Cada una tiene de 5.600 a 60.000 conexiones dendríticas provenientes de otras neuronas (Figura 6.1). Estas conexiones, conectadas a la membrana de la neurona, son las encargadas de transportar los impulsos enviados desde otras neuronas. Cada neurona tiene una salida denominada axón. El contacto de cada axón con una dendrita se realiza a través de la sinapsis. Tanto el axón como las dendritas transmiten la señal en una única dirección.

La sinapsis consta de un extremo pre-sináptico de un axón conectado a un extremo post-sináptico de una dendrita, existiendo normalmente entre éstos un espacio denominado espacio sináptico.

Las neuronas son eléctricamente activas e interactúan entre ellas mediante un flujo de corrientes eléctricas locales. Estas corrientes se deben a diferencias de potencial entre las membranas celulares de las neuronas. Un impulso nervioso es un cambio de voltaje que ocurre en una zona localizada de la membrana celular. El impulso se transmite a través del axón hasta llegar a la sinapsis, produciendo la liberación de una sustancia química denominada neurotransmisor que se esparce por el fluido existente en el espacio sináptico. Cuando este fluido alcanza el otro extremo transmite la señal a la dendrita. Los impulsos recibidos desde la sinapsis se suman o restan a la magnitud de las variaciones del potencial de la membrana. Si las contribuciones totales alcanzan un valor determinado (alrededor de 10 milivoltios) se disparan uno o más impulsos que se propagarán a lo largo del axón.

Este impulso se inicia en la conexión entre el axón y la membrana. Su amplitud y velocidad dependen del diámetro del axón y su frecuencia del número de disparos que se efectúen.

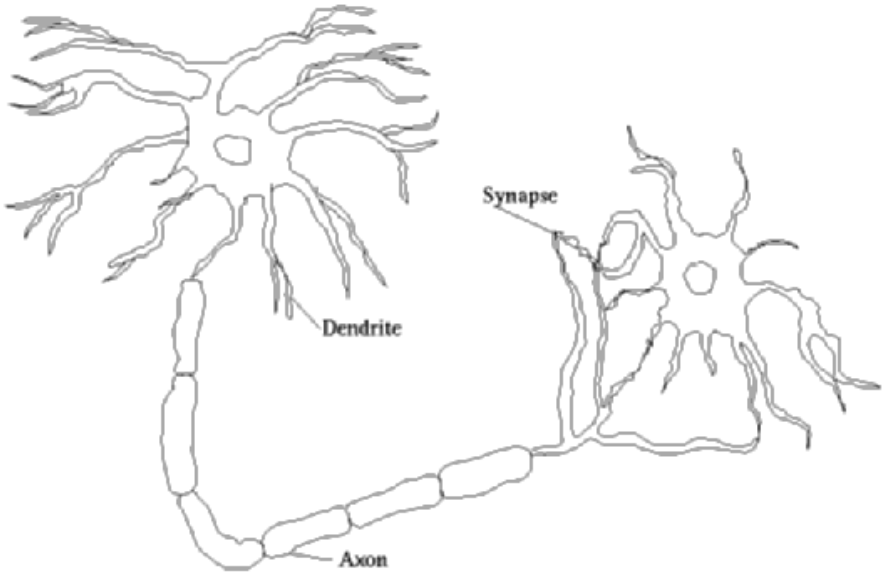

Figura 6.1. Estructura simplificada de una neurona y sus componentes

Las redes neuronales artificiales basan su funcionamiento en las redes neuronales reales, estando formadas por un conjunto de unidades de procesamiento conectadas entre sí. Por analogía con el cerebro humano se denomina «neurona» a cada una de estas unidades de procesamiento. Cada neurona recibe muchas señales de entrada y envía una única señal de salida (como ocurre en las neuronas reales). La Figura 6.2 grafica la analogía entre una neuronal real y una artificial. Las similitudes tiene que ver con que una neurona real tiene entradas (dentritas – x_i), utilizan pesos (sinapsis – w_{ji}) y generan una salida (axón – **J**).

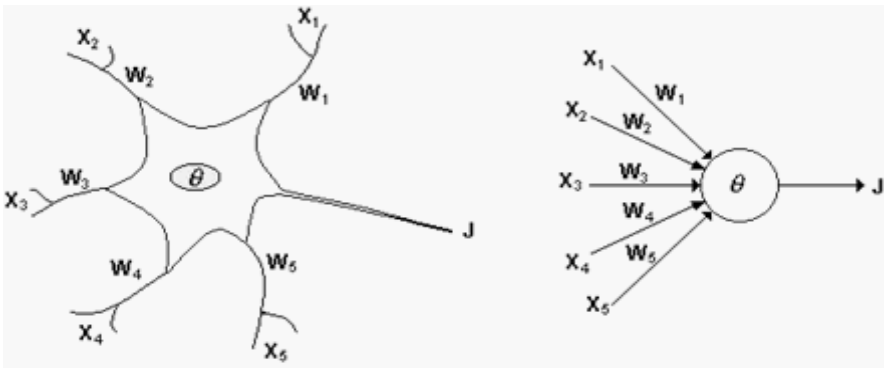

Figura 6.2 Analogía entre una neurona real y una neurona artificial

6.3. Red de una única neurona.

Una neurona es la unidad de proceso fundamental para el funcionamiento de una red neuronal. La primera aparición de un modelo que emulara la operatoria de una neurona fue gracias al aporte de Warren McColloch y Walter Pitts en 1943. Al primer modelo se le agregó luego la capacidad de aprendizaje. Este modelo recibe el nombre de *Perceptrón Simple*, Figura 6.3. El funcionamiento simple del modelo perceptrón permite comprender cómo trabaja la neurona y luego entonces, la generalización a una capa de múltiples neuronas será sencillo.

En la Figura 6.3 pueden distinguirse las entradas a la red, x_j que representan la *sinapsis* o *vínculo*, caracterizados por un *peso*, que indica la fuerza de la conexión. Una señal a la entrada de la sinapsis *j* conectada a la neurona *k* es multiplicada por el peso sináptico w_{kj}. El peso puede tener signo positivo o negativo, según sea la sinapsis asociada, exitatoria en el primer caso e inhibitoria en el segundo. Un nodo *sumador* para sumar las entradas, pesarlas (multiplicarlas) por las sinapsis respectivas, actúa como un operador lineal. Como último elemento de la red, aparece la *función de activación* φ que limita la amplitud de la salida de la neurona y emula del sistema biológico la sensibilidad no-lineal a los estímulos. El rango usual es [0,1] o [-1,1].

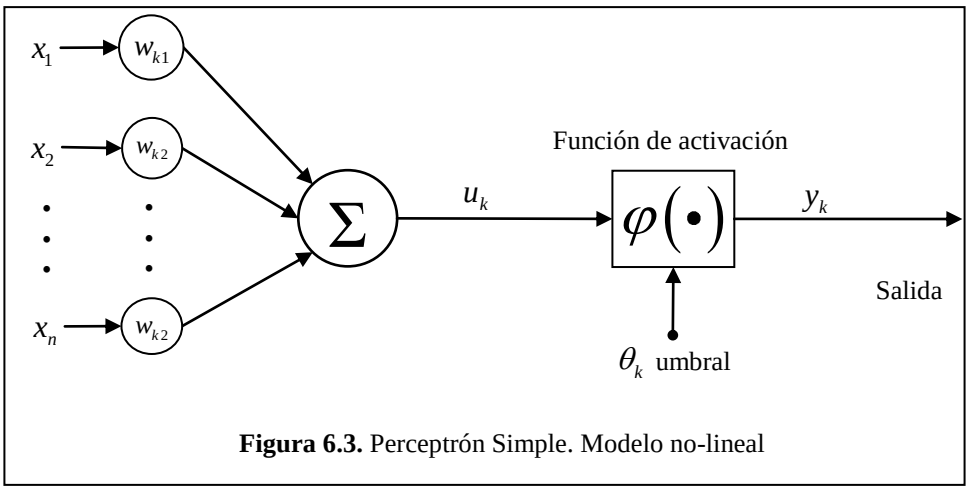

Figura 6.3. Perceptrón Simple. Modelo no-lineal

Las entradas individuales son *pesadas* cada una por los pesos correspondientes w_{kn}. Podemos describir la neurona *k* matemáticamente, mediante el par de ecuaciones siguientes:

$$u_k = \sum_{j=1}^{n} w_{kj} x_j \tag{6.1}$$

y

$$y_k = \varphi(u_k - \theta_k) \tag{6.2}$$

u_k es la salida del operador lineal; θ_k es el umbral; $\varphi(\cdot)$ es la función de activación y y_k es la señal de salida de la neurona *k*. θ_k tiene el efecto de aplicar una *transformación afín* a la salida u_k del operador lineal, Figura 6.3, (Haykin, 1996). Esta relación está descripta por

$$v_k = u_k - \theta_k \tag{6.3}$$

En la Figura 6.4 puede observarse cómo se modifica la relación entre el *potencial de activación* v_k y la salida del operador lineal u_k, dependiendo del signo del umbral, positivo o negativo será la gráfica, la que ya no pasará por el origen.

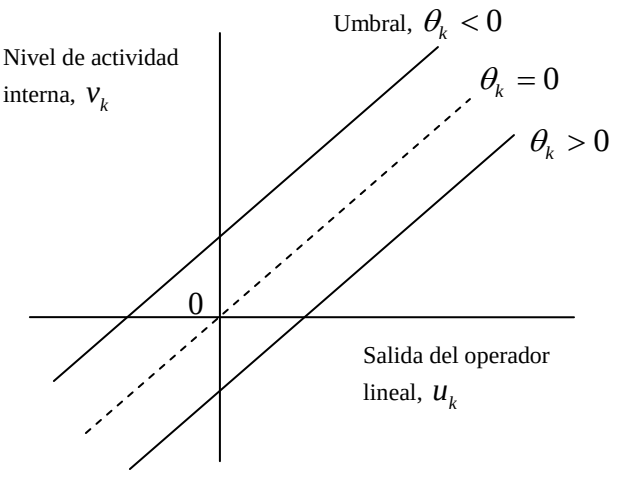

Figura 6.4. Transformación afín producida por la presencia del umbral

El umbral es un parámetro externo de la neurona artificial *k*. Podríamos reescribir las ecuaciones (1) y (2) de la siguiente manera

$$u_k = \sum_{j=0}^{n} w_{kj} x_j \tag{6.4}$$

y

$$y_k = \varphi(v_k) \tag{6.5}$$

Como puede observarse en la ecuación (6.4), ahora $0 \le j \le n$, es decir se agregó una nueva sinapsis, que tiene como entrada, $x_0 = -1$ y peso $w_{k0} = \theta_k$, lo que produce un modelo alternativo al descripto en la Figura 6.3.

6.4. Funciones de Activación

En la Figura 6.3 puede observarse el nodo de la función de activación $\varphi(\cdot)$. Define la salida de la neurona en términos del nivel de actividad de su entrada. Es además que es el responsable de introducir la no-linealidad a la red.

Una función de activación es una función transferencia de la neurona. Los tipos más usados son los que se describirán a continuación, aunque pueden utilizarse otras funciones de activación de soporte compacto. Existe muchos trabajos que así lo demuestran aplicados al análisis del ECG, (Yuehui, 2006), (Suranai, 2009), (Hongyu, 2009).

Los tres tipos básicos de funciones de activación son:

(a). Función umbral o escalón. Su ecuación está dada por (6.6) y su representación es la Figura 6.5(a)

$$\varphi(v)=\begin{cases}1, & v\geq 0\\ 0, & v<0\end{cases} \tag{6.6}$$

Esta función tiene un comportamiento binario, representa la activación o no de la neurona. Es la versión más simplificada de funciones de activación.

(b). Función rampa. Cuya ecuación es la que sigue y su gráfica está dada en la Figura 6.5(b).

$$\varphi(v)=\begin{cases}1, & v\geq \frac{1}{2}\\ v, & -\frac{1}{2}\leq v\leq \frac{1}{2}\\ 0, & v\leq -\frac{1}{2}\end{cases} \tag{6.7}$$

La función rampa o lineal por tramos es una aproximación a un amplificador no-lineal. Refleja el incremento del potencial de activación que se produce al incrementarse la entrada. Cuando se utiliza este tipo de funciones, surgen dos situaciones especiales. La primera es que funciona como un operador lineal en la región lineal, si no se llega a la saturación. Y la segunda, es que la función rampa se reduce a una función escalón si el factor de amplificación de la región lineal crece hasta infinito. Un inconveniente que se presenta, además de las limitaciones propias de una función lineal por partes es que, la convergencia de la redes neuronales que utilizan esta función de activación se vuelven inestables, pues las entradas a la neurona tienden a crecer sin límite.

(c).Función sigmoidea. Comúnmente es la más utilizada por ser una función estrictamente creciente, Figura 6.5(c). Posee además características asintóticas y por ser de clase $\mathscr{C}^{\infty}$ le proporciona más flexibilidad a la red neuronal artificial. Su funcionamiento es el siguiente: el modelo permanece en cero hasta que ingresa un dato, en este punto la frecuencia de activación se incrementa rápidamente, pero gradualmente llega a ser asíntota. Un ejemplo de función sigmoidea es la descripta en (6.8), donde a es el parámetro de la pendiente. Variando *a* se obtienen distintos valores de pendientes. En el límite, con $a\to\infty$, la función sigmoidea se aproxima a la función rampa, pero como asume valores continuos $\forall v\in[0,1]$, resulta también derivable en ese rango.

$$\varphi(v) = \frac{1}{1+\exp(-a \cdot v)} \tag{6.8}$$

Tanto la función escalón, como las funciones rampa y sigmoideas poseen un rango de variabilidad de [0,1]. En general, lo deseable es tener funciones de activación que varíen entre [-1,1], produciéndose una antisimetría con respecto al origen. La función umbral, que ahora se llamará *función signo*, adopta la forma

$$\varphi(v) = \begin{cases} 1, & v > 0 \\ 0 & v = 0 \\ -1 & v < 0 \end{cases} \tag{6.9}$$

La función sigmoidea que varía entre [-1,1] más comúnmente utilizada es la *función tangente hiperbólica*, (6.10).

$$\varphi(v) = \tanh\left(\frac{v}{2}\right) = \frac{1-\exp(-v)}{1+\exp(-v)} \tag{6.10}$$

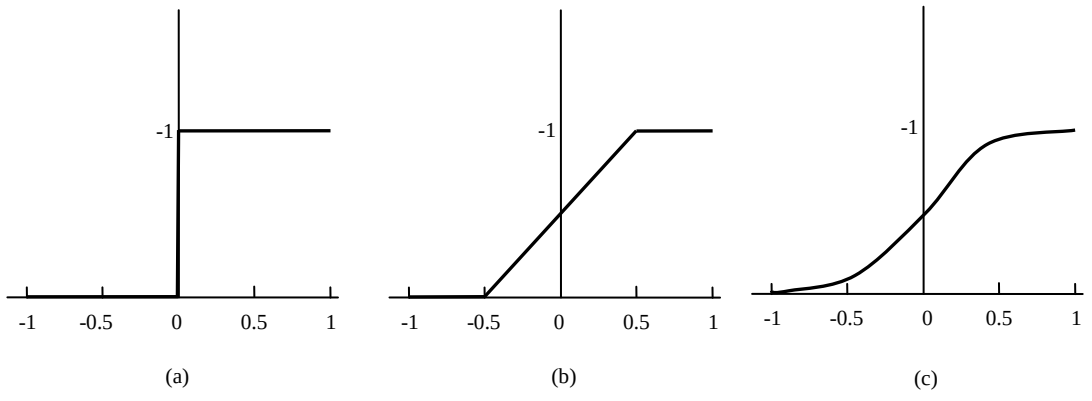

Figura 6.5. Funciones de Activación de una red neuronal artificial. (a) Función umbral. (b) función rampa o lineal por tramos y (c) función sigmoidea

6.5. Proceso de Aprendizaje

Las redes neuronales artificiales poseen como uno de sus principales atributos, la capacidad de aprender. Podemos definir aprendizaje de una red neuronal como:

La respuesta de la red neuronal a estímulos del entorno que recibe en un proceso continuo e iterativo y que resulta en cambios en los valores de los pesos de las conexiones entre neuronas.

El modo en que estos valores de pesos se modifican está determinado por el tipo de entrenamiento que se defina para la red. Como dos grandes grupos podíamos definir el tipo de aprendizaje *supervisado* y *no-supervisado* o competitivo. El modo más intuitivo es el mencionado en primer lugar, que consiste en que la red dispone de los patrones de entrada y los patrones de salida que deseamos para esa entrada y en función de ellos se modifican los pesos de las sinapsis para ajustar la entrada a esa salida. El aprendizaje no supervisado, consiste en no presentar patrones objetivos, si no solo patrones de entrada, y dejar a la red clasificar dichos patrones en función de las características comunes de los patrones.

El aprendizaje supervisado incluye aprendizaje de corrección del error, aprendizaje reforzado y aprendizaje estocástico. En este tipo de aprendizaje se define cuándo finalizar con el aprendizaje, de decide cuánto y cada cuánto presentar cada asociación para el entrenamiento y se provee la información de rendimiento (error). También podríamos hacer una nueva subdivisión (Figura 6.6) en dos categorías: aprendizaje estructural y aprendizaje temporal. El primero tiene que ver con la posibilidad de encontrar la mejor relación entrada/salida para cada par individual de patrones. El aprendizaje temporal tiene que ver con capturar una secuencia de patrones necesaria para lograr algún resultado final. A diferencia del aprendizaje estructural, la respuesta presente de la red depende de entradas y respuestas previas. Los ejemplos de estas sub-categorías están mencionados en la Figura 6.6.

El aprendizaje no-supervisado es un proceso que como no incorpora un supervisor externo, basa su aprendizaje en información interna durante todo el proceso de

aprendizaje. Su tarea es organizar los datos presentados y descubrir sus propiedades colectivas perceptibles. Los ejemplos más destacados están listados en la Figura 6.6

Otra clasificación válida para los tipos de aprendizaje de una red neuronal, seria *aprendizaje off-line*, mayormente utilizado y *aprendizaje on-lin*e. Se llama aprendizaje off-line a aquel que usa todos los parámetros para condicionar las conexiones previo al uso de la red. Un ejemplo es algoritmo de entrenamiento de *retro-propagación o back-propagation (BP)*, donde se ajustan las conexiones en la red neuronal multicapa a través de cientos de iteraciones que vinculan todos los pares de patrones hasta alcanzar el funcionamiento deseado. Llegado este punto, los valores de los pesos se *congelan* y a partir de ese momento, la red que resulta con estos valores se usa en un modo de re-llamada. Un requerimiento específico de este tipo de aprendizaje es que todos los patrones deben ser residentes para el entrenamiento, resulta imposible introducir algún nuevo patrón y que ésta incorpore automáticamente los cambios y actúe en consecuencia. Es necesario volver a entrenar la red y obtener nuevos valores de pesos.

Las RNA que utilizan aprendizaje on-line, deben pagar un alto costo si hablamos de performance. Este aprendizaje permite la incorporación automática de nuevos patrones y aprende in situ, sin embargo, las redes neuronales con aprendizaje off-line proveen soluciones muy superiores a cuestiones de clasificación no-lineal y a otros problemas que resultan de difícil resolución para redes con aprendizaje on-line.

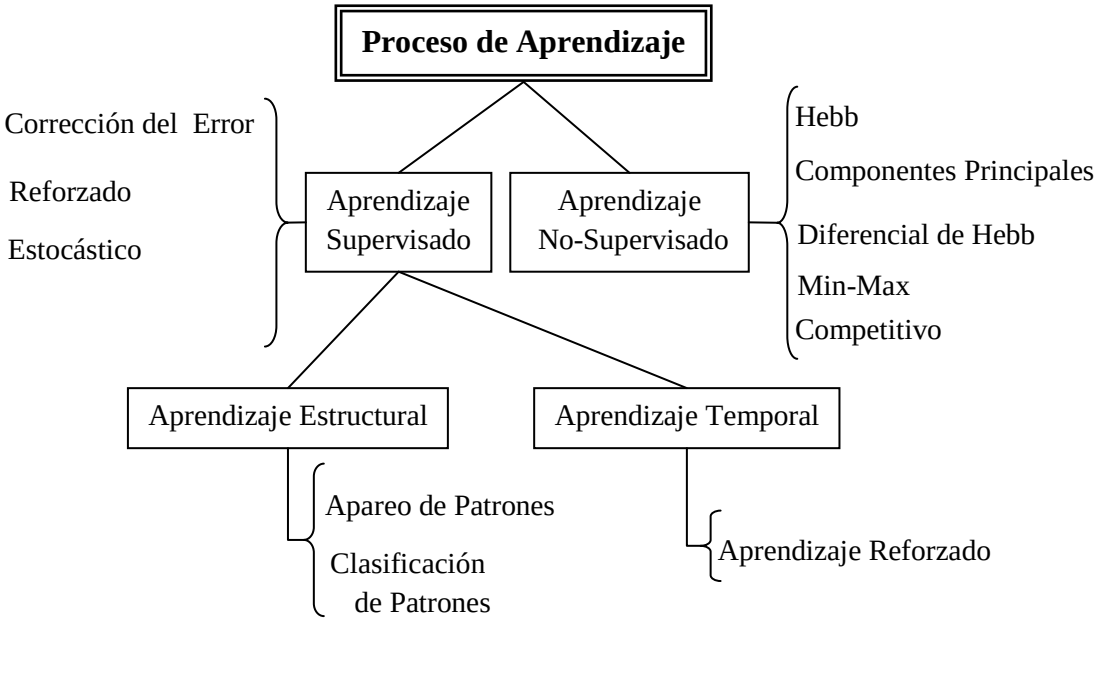

Figura 6.6. Clasificación de Tipos de Aprendizaje de una Red Neuronal Artificial

6.5.1. Aprendizaje Supervisado

Principalmente un sistema de aprendizaje supervisado funciona con un "maestro" externo que tiene conocimiento del entorno, entendiéndose por entorno a los pares de entrada-salida que actuarán en la fase de entrenamiento y aprendizaje de la red. Este entorno, además es completamente desconocido para la red. Cuando se introduce un parámetro de entrada a la red, el maestro es capaz de proveer a la red el valor deseado de la salida para esa entrada. Ésta representa la acción óptima a realizar por la red neuronal. Mientras se entrena a la red, es decir se ajustan lo parámetros, interviene la señal de error, definida como la diferencia entre la respuesta deseada y la respuesta presente de la red. El ajuste de parámetros se realiza de manera iterativa hasta alcanzar o emular el comportamiento del maestro, que se presume óptimo en un modo estadístico. Resumiendo, el maestro trasfiere a la red su conocimiento del entorno y cuando esto

ocurre, la red es capaz de funcionar en forma autónomo sin la supervisión del maestro, de un modo no-supervisado.

El error medio cuadrático, definido como una función, es una medida para la evaluación de la performance del algoritmo de aprendizaje. Los parámetros libres de la función, que es multidimensional, forman las coordenadas de una *superficie de error* (Haykin, 1994). Un punto en esta superficie representa una operación de la red. Cuando un punto se mueve en dirección a un punto mínimo local o global de la superficie de error, entonces podemos decir que la red está aprendiendo. Esto es posible gracias a la información del *gradiente* de la superficie que corresponde al comportamiento presente del sistema. Por gradiente de la superficie se entiende al vector que apunta en la dirección del *descenso más rápido* (steepest descent). Resumiendo, un sistema de aprendizaje supervisado está capacitado para realizar tareas de clasificación de patrones, siempre que cuente con un adecuado conjunto de ejemplos de entrada-salida, un algoritmo que minimice la función de interés y un tiempo suficiente para realizar el entrenamiento.

El algoritmo de retro-propagación (BP -Back Propagation) es el más utilizado de los sistemas de aprendizaje supervisado. En el algoritmo BP los términos erróneos del algoritmo son retro-propagados a través de la red, en una base capa por capa.

El modo off-line es el más utilizado en sistema de aprendizaje supervisado. Se entrena la red con un conjunto representativo de pares entrada-salida. Una vez que la red alcanza su mínimo global en la superficie de error, es decir que se alcanzó la performance deseada, los valores son congelados, lo que significa que la red trabaja de modo estático.

6.5.2. Aprendizaje No Supervisado

También llamado aprendizaje auto organizado, en el que no existe el "maestro externo". Los ejemplos de la función que sirven para entrenar a la red en el aprendizaje supervisado, no forman parte del sistema del aprendizaje no supervisado. Una medición de la calidad de representación que requiere la red es la encargada de proveer la

información necesaria para el proceso de aprendizaje, los parámetros de la red se optimizan con respecto a esa medición también, que es además independiente de la tarea. La red será capaz de crear nuevas clases en forma autónoma, no supervisada, y de formar representaciones internas de los datos de entrada; todo esto como consecuencia del proceso de aprendizaje.

El aprendizaje no supervisado hace uso de reglas de aprendizaje competitivo, donde las neuronas de la capa competitiva pugnan entre ellas por la oportunidad de responder a las características de los datos de entrada, bajo la ley del "ganador lleva todo". Esto es, la neurona con la mayor cantidad de aciertos gana la competencia y se enciende, mientras que las demás se apagan.

El aprendizaje no supervisado aparece también como una solución a la complejidad y cantidad de operaciones computacionales que se realizan en el proceso de aprendizaje supervisado. Este último, cuyo mejor referente el es algoritmo de retro-propagación, consta de dos fases, una propagación de la señal entrada hacia delante y la otra, una propagación hacia atrás de las señales de error que se producen al comparar la respuesta real de la red con la deseada. Es decir, que cuanto mayor sea el tamaño de la red, más intensivo será su funcionamiento computacional y el tiempo requerido para el entrenamiento también aumentará de forma exponencial, lo que resulta ineficiente, (Haykin, 1996).

El aprendizaje no supervisado permite ser aplicado en cada capa de la red en forma secuencial, lo que facilita el aprendizaje en redes de múltiples capas. Su habilidad para formar representaciones internas que modelen la estructura de los datos de entrada resulta más simple o explícita. Un estructura híbrida de aprendizaje debería proveer una solución más acertada que la aplicación del aprendizaje supervisado como único proceso, (Haykin, 1996).

6.5.3. Tareas de clasificación de parámetros

La clasificación de parámetros es una de las tantas tareas que pueden ser resueltas por una RNA. Como entrada a la red se presenta un grupo de patrones que identifican a la categoría a la cual pertenece el dato o evento a clasificar. Se lleva a cabo el entrenamiento, ya sea supervisado o no, en el cual de fijan los pesos de las neuronas de la red. Posteriormente se presenta a la red un nuevo patrón, que no fue utilizado en el entrenamiento, es decir, desconocido para la red, pero que pertenece a alguna de las categorías definidas en el entrenamiento. La tarea de clasificación de la red es encontrar a qué categoría pertenece este nuevo evento. Descripto de esta manera, éste es un problema de aprendizaje supervisado.

La ventaja de utilizar redes neuronales en clasificación de patrones es que una red puede construir límites de decisiones no lineales entre las diferentes clases de modo no paramétrico y de todos modos ofrecer métodos prácticos para resolver problemas complejos de clasificación de eventos.

Cuando se desconocen las clases o categorías en las que se pueden agrupar los patrones, lo más óptimo es la utilización de aprendizaje no supervisado. Este tipo de aprendizaje podría ser usado en una etapa previa, en la cual se agrupan los parámetros que serán luego clasificados.

6.5.4. Teoría del aprendizaje

La teoría del aprendizaje, si bien he descripto algunos aspectos, encierra aún más conceptos que tienen que ver con los fundamentos matemáticos necesarios para formular los principios esenciales que nos permitirán alcanzar las soluciones más óptimas a los problemas planteados a la red neuronal. Los aspectos fundamentales a desarrollar se harán en el contexto de aprendizaje supervisado.

Tres elementos importantes forman parte de un sistema de aprendizaje supervisado y actúan relacionándose uno con otro, Figura 6.7. Ellos son, el *entorno*, que suplanta un vector $\mathbf{x}$ con una distribución de probabilidad fija pero desconocida $P(\mathbf{x})$; el *maestro*,

que provee el vector $\mathbf{d}$ de respuestas deseadas, o *targets*, para cada entrada del vector $\mathbf{x}$ de acuerdo con la distribución de probabilidad condicional $P(\mathbf{d}|\mathbf{x})$ también fija pero desconocida. La función, desconocida, que relaciona ambos vectores es

$$\mathbf{d} = g(\mathbf{x}) \tag{6.11}$$

Y por último, el *algoritmo de aprendizaje*, capaz de implementar funciones de mapeo de entrada-salida descriptas por

$$\mathbf{y} = \mathrm{F}(\mathbf{x}, \mathbf{w}) \tag{6.12}$$

Con $\mathbf{y}$ como el vector de salidas producido por el algoritmo en respuesta al vector de entradas $\mathbf{x}$; y $\mathbf{w}$ es el conjunto de pesos sinápticos del espacio de pesos W seleccionados por el algoritmo.

El punto esencial en el problema de aprendizaje de una red neuronal es encontrar una función $\mathrm{F}(\mathbf{x}, \mathbf{w})$ que, a partir de un conjunto dado de pares de entrada-salida, mejor aproxime la función $g(\mathbf{x})$. La selección está basada en un conjunto de N ejemplos de entrenamiento *independientes e idénticamente distribuidos (i.i.d.),*

$$(\mathbf{x}_1, \mathbf{d}_1), (\mathbf{x}_2, \mathbf{d}_2), ..., (\mathbf{x}_N, \mathbf{d}_N). \tag{6.13}$$

El aspecto que más importa en este punto es saber si el conjunto de ejemplos es realmente representativo y contiene la suficiente información para permitir que el algoritmo de aprendizaje logre la generalización buscada.

Sea $L(\mathbf{d}, \mathrm{F}(\mathbf{x}, \mathbf{w}))$ la medida de la diferencia entre la salida deseada $\mathbf{d}$ y la respuesta producida por el algoritmo de aprendizaje $\mathrm{F}(\mathbf{x}, \mathbf{w})$. También llamada *función de pérdida cuadrática,* definida por la distancia cuadrática euclídea entre el vector $\mathbf{d}$ y la aproximación $\mathrm{F}(\mathbf{x}, \mathbf{w})$,

$$L(\mathbf{d}; \mathrm{F}(\mathbf{x}, \mathbf{w})) = \|\mathbf{d} - \mathrm{F}(\mathbf{x}, \mathbf{w})\|^2. \tag{6.14}$$

El valor esperado de la pérdida está definido por la funcional de riesgo (Vapnik, 1992)

$$R(\mathbf{w}) = \int L\left(\mathbf{d}; \mathrm{F}(\mathbf{x}, \mathbf{w})\right) dP(\mathbf{x}, \mathbf{d}), \tag{6.15}$$

donde $P(\mathbf{x}, \mathbf{d})$ es la distribución de probabilidad conjunta de $\mathbf{x}$ y $\mathbf{d}$, y la integral está tomada en el sentido de Riemann-Stieltjes. El objetivo total del proceso de aprendizaje es minimizar la funcional de riesgo $R(\mathbf{w})$. La distribución de probabilidad conjunta, está definida por

$$P(\mathbf{x}, \mathbf{d}) = P(\mathbf{d}|\mathbf{x}) P(\mathbf{x}), \tag{6.16}$$

y como ya lo mencionamos, la distribución de probabilidad de $\mathbf{d}$ restringida a $\mathbf{x}$ y la distribución de probabilidad de $\mathbf{x}$ son desconocidas, por lo tanto con la única información con la que contamos está contenida en el conjunto de entrenamiento. Esto convierte a este problema en una cuestión compleja desde el punto de vista matemático. Para simplificar la búsqueda de una solución, se aplica el principio inductivo de minimización del riesgo empírico, perfectamente aplicable a las redes neuronales.

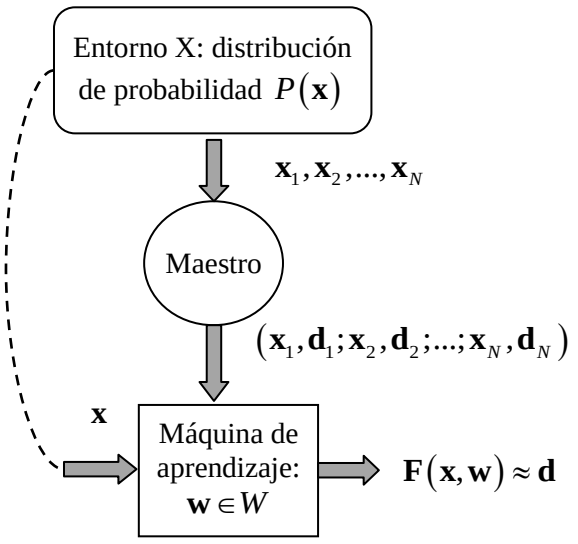

Figura 6.7. Los tres elementos de un sistema de aprendizaje supervisado.

Riesgo Empírico

La funcional del riesgo empírico está dada por

$$R_{\text{emp}}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} L\left(\mathbf{d}_i; \mathbf{F}(\mathbf{x}_i, \mathbf{w})\right), \tag{6.17}$$

que no depende la distribución de probabilidad $P(\mathbf{x},\mathbf{d})$, como puede observarse en (6.17). A diferencia de la funcional de riesgo original $R(\mathbf{w})$, la funcional de riesgo empírico $R_{\text{emp}}(\mathbf{w})$, puede ser minimizada con respecto al vector de pesos $\mathbf{w}$. El objetivo del uso de la funcional del riesgo empírico es encontrar bajo qué condiciones $R_{\text{emp}}(\mathbf{w})$ se aproxima a $R(\mathbf{w})$. Si esta aproximación es *uniforme* en $\mathbf{w}$ con una *precisión* ε, luego el mínimo de $R_{\text{emp}}(\mathbf{w})$ difiere del mínimo de $R(\mathbf{w})$ en un valor que no excede a 2ε. Es necesario entonces imponer una restricción, tal es que $\forall \mathbf{w} \in W, \forall \varepsilon > 0$ se cumple la siguiente relación probabilística (Haykin, 1996, Vapnik, 1992)

$$\text{Prob}\left\{\sup_{\mathbf{w}} \left|R(\mathbf{w}) - R_{\text{emp}}(\mathbf{w})\right| > \varepsilon\right\} \underset{N \to \infty}{\to} 0. \tag{6.18}$$

Si esta condición se cumple, entonces hay *convergencia uniforme* del vector $\mathbf{w}$ del riesgo empírico medio a su valor esperado. Equivalentemente se cumple la siguiente desigualdad, dado un $\varepsilon > 0$

$$\text{Prob}\left\{\sup_{\mathbf{w}} \left|R(\mathbf{w}) - R_{\text{emp}}(\mathbf{w})\right| > \varepsilon\right\} < \alpha, \tag{6.19}$$

para algún $\alpha > 0$. Luego, con $\mathbf{w}_0$ y $\mathbf{w}_{\text{emp}}$ los vectores de pesos de las funcional de riesgo real y empírico respectivamente, se cumple también la siguiente desigualdad

$$\text{Prob}\left\{R(\mathbf{w}_{\text{emp}}) - R(\mathbf{w}_0) > 2\varepsilon\right\} < \alpha. \tag{6.20}$$

Es decir que si se cumple la condición (6.19), con una probabilidad de al menos $(1-\alpha)$, la función $\mathrm{F}(\mathbf{x},\mathbf{w}_{\text{emp}})$ que minimiza $R_{\text{emp}}(\mathbf{w})$, tendrá un riesgo real $R(\mathbf{w}_{\text{emp}})$ que varía en no más de 2ε del valor mínimo posible de riesgo real $R(\mathbf{w}_0)$. Las siguientes desigualdades surgen como consecuencia del cumplimiento de la condición definida en (6.19), si consideramos una probabilidad de $(1-\alpha)$

$$R(\mathbf{w}_{\text{emp}}) - R_{\text{emp}}(\mathbf{w}_{\text{emp}}) < \varepsilon \tag{6.21}$$

$$R_{\text{emp}}(\mathbf{w}_0) - R(\mathbf{w}_0) < \varepsilon . \tag{6.22}$$

Los puntos mínimos de $R_{\text{emp}}(\mathbf{w})$ y $R(\mathbf{w})$ son $\mathbf{w}_{\text{emp}}$ y $\mathbf{w}_0$ respectivamente, por lo tanto se cumple que

$$R_{\text{emp}}(\mathbf{w}_{\text{emp}}) \le R_{\text{emp}}(\mathbf{w}_0) . \tag{6.23}$$

Entonces, sumando (6.21) y (6.22) y usando (6.23), es posible escribir

$$R(\mathbf{w}_{\text{emp}}) - R(\mathbf{w}_0) < 2\varepsilon , \tag{6.24}$$

también se satisface con $(1-\alpha)$. Si consideramos una probabilidad α, se cumple la siguiente desigualdad

$$R(\mathbf{w}_{\text{emp}}) - R(\mathbf{w}_0) > 2\varepsilon . \tag{6.25}$$

Teniendo en cuenta estas condiciones mencionada y estas desigualdades, podemos hacer mención del principio de minimización del riesgo empírico, (Vapnik, 1992).

La funcional de riesgo empírico aparece para simplificar la funcional de riesgo $R(\mathbf{w})$ y está dada por:

$$\mathrm{R}_{\text{emp}}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} L\left(\mathbf{d}_i ; \mathrm{F}(\mathbf{x}_i, \mathbf{w})\right) . \tag{6.26}$$

Sobre la base de N pares de ejemplos i.i.d

$$(\mathbf{x}_i, \mathbf{d}_i); \quad i = 1, 2, ..., N. \tag{6.27}$$

Sea $\mathbf{w}_{\text{emp}}$ el vector de pesos que minimiza la funcional del riesgo empírico $R_{\text{emp}}(\mathbf{w})$ sobre el espacio W. Luego, $R(\mathbf{w}_{\text{emp}})$ converge en probabilidad, al valor mínimo posible del riesgo real $R(\mathbf{w})$, $\mathbf{w} \in W$ cuando $N \to \infty$; siempre que la funcional del riesgo empírico $R_{\text{emp}}(\mathbf{w})$ converja uniformemente a la funcional de riesgo real $R(\mathbf{w})$. Ésta es condición necesaria y suficiente para la consistencia del principio de minimización empírica. La convergencia uniforme mencionada está dada por

$$\text{Prob}\left\{\sup_{\mathbf{w}} \left|R(\mathbf{w}) - R_{\text{emp}}(\mathbf{w})\right| > \varepsilon\right\} \to 0 \quad \text{para} \quad N \to \infty. \tag{6.28}$$

Además de los conceptos mencionados, resulta necesario un parámetro denominado *dimensión Vapnik-Chervonenkis* o simplemente *dimensión VC* en el que se basan los límites de la tasa de convergencia uniforme de $R_{\text{emp}}(\mathbf{w})$ a $R(\mathbf{w})$. Este parámetro es una medida de la capacidad de las familias de funciones de clasificación producidas por el algoritmo de aprendizaje.

Sea $\mathscr{F}$ la familia de funciones de clasificación $\mathbf{d}$ implementadas por el algoritmo de aprendizaje, con $\mathbf{d} \in \{0,1\}$ (suponemos esta dicotomía por simplicidad).

$$\mathscr{F} = \left\{\text{F}(\mathbf{x}, \mathbf{w}) : \mathbf{w} \in W, \text{F} : \mathbb{R}^p \to \{0,1\}\right\}, \tag{6.29}$$

y sea $\mathscr{L}$ el conjunto de N puntos en el espacio X de los vectores de entrada de dimensión p

$$\mathscr{L} = \left\{\mathbf{x}_i \in X : i = 1, 2, ..., N\right\}. \tag{6.30}$$

La dimensión VC de una familia de funciones de clasificación, $\mathscr{F}$ es el máximo número de ejemplos de entrenamiento que puede ser aprendido por el algoritmo de

aprendizaje sin errores, para todas las posibles "rótulos binarios" de las funciones de clasificación.

La dimensión VC es un concepto de combinatoria y nada tiene que ver con la idea geométrica de dimensión. En muchos casos la dimensión VC está determinada por los parámetros libres del algoritmo de aprendizaje. Determinarlo por medios analíticos no es tarea sencilla. Algunos autores, sin embargo, han publicado métodos para encontrar sus límites (Schmitt, 1999; Asian et al. 2009). *Un valor finito de dimensión VC implica la convergencia uniforme*.

Convergencia Uniforme

Sea $P(\mathbf{w})$ la probabilidad promedio del error de clasificación y de acuerdo a la ley de los grandes números, $\forall \mathbf{w} \in W, \forall \varepsilon > 0, \varepsilon \in \mathbb{R}$ y con N el tamaño del conjunto de entrenamiento, se cumple que (Vapnik, 1982)

$$\text{Prob}\left\{\left|P(\mathbf{w}) - v(\mathbf{w})\right| > \varepsilon\right\} \underset{N \to \infty}{\to} 0. \tag{6.31}$$

Esta condición no implica que la misma regla de clasificación que minimiza el error de entrenamiento $v(\mathbf{w})$ también minimice $P(\mathbf{w})$. Si N es suficientemente grande, la proximidad entre estos parámetros deviene de una condición más fuerte que también define que $\forall \varepsilon > 0$ se cumple que

$$\text{Prob}\left\{\sup_{\mathbf{w}}\left|P(\mathbf{w}) - v(\mathbf{w})\right| > \varepsilon\right\} \underset{N \to \infty}{\to} 0. \tag{6.32}$$

La convergencia uniforme hace mención a la convergencia uniforme de la frecuencia de errores de entrenamiento a su probabilidad promedio.

El límite de la tasa de convergencia uniforme está dado por la mencionada dimensión VC, a la que llamaremos h. Para el conjunto de funciones de clasificación de dimensión VC igual a h, se verifica la siguiente desigualdad

$$\text{Prob}\left\{\sup_{\mathbf{w}}\frac{\left|P(\mathbf{w})-v(\mathbf{w})\right|}{\sqrt{P(\mathbf{w})}}>\varepsilon\right\}<\left(\frac{2eN}{h}\right)^{h}\exp\left(-\frac{\varepsilon^{2}N}{4}\right). \tag{6.33}$$

El factor $\exp\left(-\varepsilon^{2}N/4\right)$ decae exponencialmente para N grande. El factor $\left(2eN/h\right)^{h}$ representa una función creciente. Si el espacio de entrada X tiene dimensión finita, entonces la familia $\mathscr{F}$ tendrá dimensión VC finita con respecto X, lo que asegura la convergencia uniforme, (Haykin, 1996). No siempre resulta verdadero que dada una familia $\mathscr{F}$ de dimensión finita, resulte el espacio de entrada X de dimensión también finita.

Para la tasa de convergencia uniforme surgen los siguientes límites que se detallan. El primero, **a)** es una versión general del límite de tasa de convergencia uniforme y **b)** y **c)** son casos especiales según el valor del error de entrenamiento $v(\mathbf{w})$.

a)
$$P(\mathbf{w})<v(\mathbf{w})+\varepsilon_{1}(N,h,\alpha,v), \tag{6.34}$$

con

$$\varepsilon_{1}(N,h,\alpha,v)=2\varepsilon_{0}^{2}(N,h,\alpha)\left(1+\sqrt{1+\frac{v(\mathbf{w})}{\varepsilon_{0}^{2}(N,h,\alpha)}}\right) \tag{6.35}$$

$$\varepsilon_{0}(N,h,\alpha)=\sqrt{\frac{h}{N}\left[\ln\left(\frac{2N}{h}\right)+1\right]-\frac{1}{N}\ln\alpha} \tag{6.36}$$

$$\alpha=\left(\frac{2eN}{h}\right)^{h}\exp\left(-\varepsilon^{2}N\right) \tag{6.37}$$

$\varepsilon_0(N,h,\alpha)$ y $\varepsilon_1(N,h,\alpha,v)$ son valores especiales de ε y que denotan intervalos de confianza. Ambos satisfacen (6.33).

b) Para $v(\mathbf{w})$ pequeño, próximo a cero, el valor a continuación provee un límite bastante preciso para casos reales de aprendizaje

$$P(\mathbf{w}) \le v(\mathbf{w}) + 4\varepsilon_0^2(N,h,\alpha) \tag{6.38}$$

c) Para $v(\mathbf{w})$ grandes, cercanos a uno, se tiene el límite

$$P(\mathbf{w}) \le v(\mathbf{w}) + \varepsilon_0(N,h,\alpha) \tag{6.39}$$

Riesgo Estructural

Sean $v_{\text{train}}(\mathbf{w})$ y $v_{\text{gene}}(\mathbf{w})$ los errores de entrenamiento y generalización respectivamente producidos por el algoritmo de aprendizaje $\left(v_{\text{train}}(\mathbf{w}) = v(\mathbf{w})\right)$, sea h la dimensión VC de una familia de funciones de clasificación $\{\mathrm{F}(\mathbf{x},\mathbf{w}); \mathbf{w} \in W\}$ con respecto al espacio de entrada X. Teniendo en cuenta los conceptos de las tasas de convergencia uniforme y con probabilidad $(1-\alpha)$ para una cantidad $N > h$ de ejemplos de entrenamiento y para todas las funciones de clasificación $\mathrm{F}(\mathbf{x},\mathbf{w})$, el error de generalización $v_{\text{gene}}(\mathbf{w})$ es menor que un riesgo garantizado definido por la suma de un par de términos de competencia (Vapnik, 1992)

$$v_{\text{garant}}(\mathbf{w}) = v(\mathbf{w}) + \varepsilon_1(M,h,\alpha,v) \tag{6.40}$$

con $\varepsilon_1(M,h,\alpha,v)$ según (6.35). Para un número fijo de ejemplos de entrenamiento N, el error de entrenamiento $v(\mathbf{w})$ decrece en forma monótona cuando la dimensión VC h aumenta, mientras el intervalo de confianza $\varepsilon_1(M,h,\alpha,v)$ crece también en forma

monótona. Como consecuencia de esto, tanto el error de generalización como el riesgo garantizado se aproximan a un mínimo, Figura 6.8.

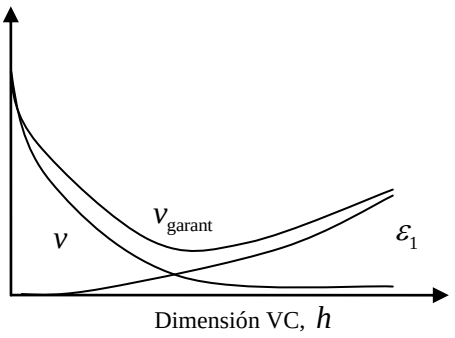

Figura 6.8. Variaciones de v, v_{train} y v_{garant} con la dimensión VC, h

Antes de alcanzar el mínimo, el problema de aprendizaje está magnificado, en el sentido que la capacidad del algoritmo todavía es pequeña en relación a la cantidad de detalles de entrenamiento. Similarmente, más allá del punto mínimo, el problema está menospreciado en el sentido que se tiene una gran capacidad del algoritmo para una cantidad de datos de entrenamiento determinada. Es claro que el objetivo es resolver el problema de aprendizaje logrando la mayor generalidad posible y eso se obtiene nivelando la capacidad del algoritmo con la cantidad de ejemplos de entrenamiento disponibles que describen el problema en cuestión. El método de minimización del riesgo estructural permite controlar el valor h y de esta manera llegar al mínimo óptimo (Vapnik, 1992).

Sea $\{\mathrm{F}(\mathbf{x},\mathbf{w});\mathbf{w}\in W\}$ una familia de funciones de clasificación binaria y sea el siguiente encaje de n subconjuntos,

$$\mathscr{S}_k = \{\mathrm{F}(\mathbf{x},\mathbf{w});w\in W_k\}; \qquad k=1,2,...,n \tag{6.41}$$

de tal modo que se verifica

$$\mathscr{S}_1 \subset \mathscr{S}_2 \subset ... \subset \mathscr{S}_n \tag{6.42}$$

a lo que corresponde que

$$h_1 < h_2 < ... < h_n \tag{6.43}$$

El método sistemático incluye minimizar el riesgo empírico, es decir el error de entrenamiento, y seleccionar el subconjunto $\mathscr{F}^*$ que contiene el menor de los mínimos. Para este subconjunto, se determina el mejor compromiso entre el error de entrenamiento y el intervalo de confianza, que son los dos términos involucrados en el riesgo garantizado. El objetivo es encontrar una estructura de red tal que la disminución de la dimensión VC ocurra a expensas del menor aumento posible del error de entrenamiento.

6.6. Modelos

Los modelos de redes neuronales artificiales o topologías de redes neuronales describen el modo en que están organizadas las neuronas y las capas formadas por estas neuronas, como así también las conexiones que entre ellas se producen. Tanto el algoritmo de aprendizaje como el modo en que se propaga la información en la red también forman parte del modelo. Podemos decir entonces la topología de la red neuronal describe no solo su arquitectura sino también su funcionamiento. Existen muchos modelos de redes neuronales que combinan variedad de estructura y mecanismos de aprendizaje. Entre las más conocidas por su aplicabilidad se destacan el perceptrón multicapa (MLP), LVQ, ADALINE, MEDALINE entre otros. Todos ellos de pertenecen al grupo cuyo proceso de aprendizaje es supervisado. Dentro del grupo de aprendizaje no supervisado, podemos citar las redes Hopfield, ART y SOM. Los modelos que describiré en este trabajo, son los utilizados en el desarrollo computacional. Ellos son el perceptrón multicapa, con reglas de aprendizaje nítidas y difusas y el modelo SOM de aprendizaje no-supervisado.

6.6.1. Perceptrón Multicapas

El *perceptrón multicapas* representa una generalización del perceptrón de una sola capa presentado en la sección 3 de este capítulo.

El perceptrón multicapas (MLP) es la topología que más ampliamente ha sido aplicada para resolver problemas de clasificación principalmente y de otras índoles. El entrenamiento supervisado es el que más se acomoda a esta arquitectura, con el *algoritmo de retro-propagación de errores*, basado en la *regla de corrección de errores*, como su principal exponente.

Una de las características distintivas de una red MLP es que a la salida de cada neurona ésta incluye una *no linealidad*, que además resulta ser diferenciable en todo punto. Este aspecto es importante debido a que la relación de entrada salida de la red no está reducida a la de un perceptrón de una única capa. La función no lineal que más comúnmente se utiliza es la sigmoidea

$$y_i = \frac{1}{1+\exp\left(-u_j\right)} \tag{6.44}$$

donde u_j es el nivel de actividad interna de la red de la neurona j y y_i es la salida de la neurona. Otra característica a destacar es que la red posee una o más capas de *neuronas ocultas* que no pertenecen a la entrada ni a la salida de la red. Esto permite que la red aprenda cuestiones complejas debido a que progresivamente extrayendo características significativas de los patrones de entrada. Y como último aspecto destacable, esta topología presenta un alto grado de *conectividad*, determinado por las sinapsis de la red.

La Figura 6.9 muestra la estructura clásica de una red MLP con 2 capas de neuronas ocultas

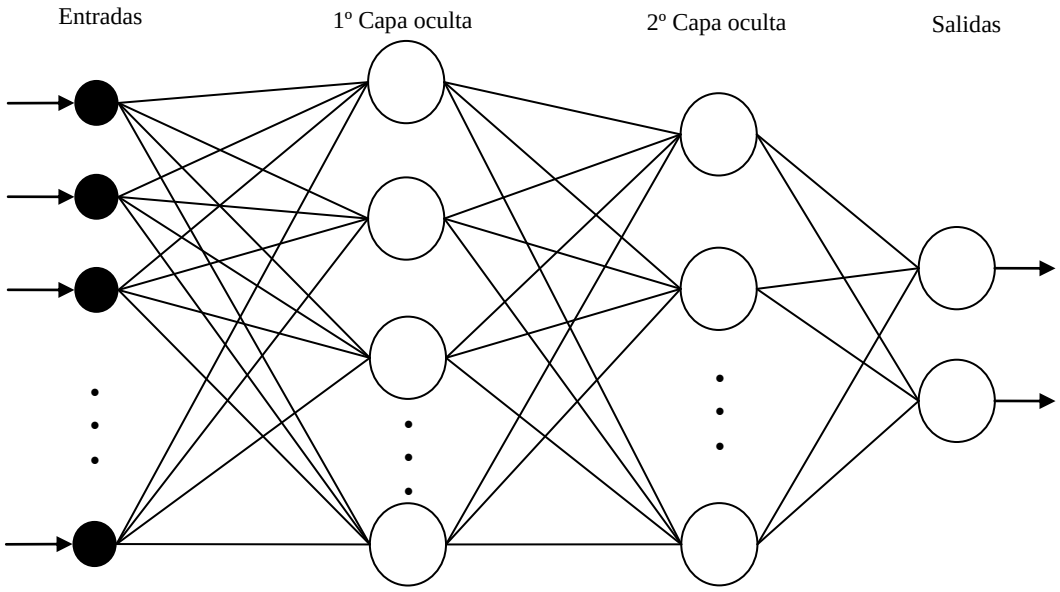

Figura 6.9. Arquitectura de una red neuronal MLP genérica totalmente conectada

La red de la Figura 6.9 está totalmente conectada, es decir que cada neurona en cada capa tiene conexión con todas las neuronas de la capa anterior. En esta topología la señal fluye hacia delante, de izquierda a derecha en una base de capa por capa. En esta red se producen dos tipos de señales: las *señales de función* que es una señal de entrada que ingresa a la red, se propaga hacia adelante a través de la red, pasando neurona a neurona y emerge a la salida de la red como una señal de salida. Se las denomina de esta manera debido a que realizan una función útil en la red y además la señal que pasa a través de cada neurona se calcula como una función de las entradas y los pesos asociados aplicados a cada neurona. El otro tipo de señales son las *señales de error* que se originan en una neurona de salida de la red y se propaga a través de la red en sentido contrario a las señales de función. Se las denomina de esta forma porque su cálculo involucra una función dependiente del error.

Cada neurona oculta o de salida tiene a su cargo el cálculo de dos operaciones. Una de las operaciones es calcular la señal de función, expresada como una función no lineal continua de la señal de entrada y los pesos sinápticos asociados a cada neurona. La otra operación es la estimación instantánea de un vector gradiente, que se necesita para la propagación hacia atrás a través de la red.

6.6.2. El algoritmo de Retro Propagación

Para comprender el funcionamiento del algoritmo de retro-propagación, es necesario describir su derivación de manera que los parámetros y variables que intervienen en su operación, estén definidos y no resulte arbitrario su cálculo.

Sea j una neurona de salida, y n la iteración. Si consideramos la señal de error a la salida de dicha neuronal en la iteración n

$$e_j = d_j(n) - y_i(n) \tag{6.45}$$

Teniendo determinado este valor, podemos definir el promedio del error cuadrático de la red como

$$\mathscr{E}_{\text{av}} = \frac{1}{N}\sum_{n=1}^{N}\mathscr{E}(n), \tag{6.46}$$

donde

$$\mathscr{E}(n) = (1/2)\sum_{j\in C} e_j^2(n), \tag{6.47}$$

con C como el conjunto de todas las neuronas en la capa de salida de la red y N el número total de pares de ejemplos del conjunto de entrenamiento.

La ecuación (6.46) representa la *función costo* como una medida de performance del conjunto de datos de entrenamiento. Es claro que el objetivo del algoritmo de aprendizaje en minimizar esta función mediante el ajuste de los pesos de la red. Esto se logra con la aplicación de un método de entrenamiento por el cual los pesos son ajustados en una base patrón por patrón. Los ajustes se realizan teniendo en cuenta los errores respectivos computados para cada patrón presentado a la red.

Consideremos ahora el nivel de actividad interno de la red, el que está dado por $v_j(n)$ producido a la entrada de la no linealidad asociado con la neurona j

$$v_j(n) = \sum_{i=0}^{p} w_{ji}(n)\, y_i(n) \tag{6.48}$$

donde p es el número total de entradas aplicadas a la neurona j. Por lo tanto la señal de función $y_i(n)$ aparece a la salida de la neurona j en la iteración n es

$$y_i(n) = \varphi_j\left(v_j(n)\right). \tag{6.49}$$

El algoritmo de retro propagación aplica una corrección $\Delta w_{ji}(n)$ al peso sináptico $w_{ji}(n)$, que es proporcional al gradiente instantáneo $\partial \mathscr{E}(n)/\partial w_{ji}$ y que haciendo uso de la regla de la cadena, se puede expresar este gradiente así

$$\frac{\partial \mathscr{E}(n)}{\partial w_{ji}(n)} = \frac{\partial \mathscr{E}(n)}{\partial e_j(n)} \cdot \frac{\partial e_j(n)}{\partial y_i(n)} \cdot \frac{\partial y_i(n)}{\partial v_j(n)} \cdot \frac{\partial v_j(n)}{\partial w_{ji}(n)}. \tag{6.50}$$

Este gradiente representa un factor de sensibilidad, que determina la dirección de búsqueda en el espacio W de pesos para el peso sináptico $w_{ji}(n)$.

Derivando ecuaciones (6.45), (6.47), (6.48) y (6.49) respecto de $y_i(n)$, $e_j(n)$, $v_j(n)$ y $w_{ji}(n)$ respectivamente y acomodando los términos según sea conveniente, podemos reescribir la ecuación (6.50)

$$\frac{\partial \mathscr{E}(n)}{\partial w_{ji}(n)} = -e_j(n) \cdot \varphi_j'\left(v_j(n)\right) \cdot y_i(n) \tag{6.51}$$

El $\Delta w_{ji}(n)$ aplicado a $w_{ji}(n)$ está definido por la *regla delta*

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathscr{E}(n)}{\partial w_{ji}(n)} \tag{6.52}$$

donde la constante η es el parámetro de la tasa de aprendizaje del algoritmo de retro-propagación. El signo negativo en (6.52) se relaciona al *descenso del gradiente* en el espacio W. Reemplazando (6.51) en (6.52), obtenemos

$$\Delta w_{ji}(n) = \eta \delta_i(n) y_i(n), \tag{6.53}$$

donde $\delta_j(n)$ es el gradiente local y está definido por

$$\delta_j(n) = -\frac{\partial \mathrm{E}(n)}{\partial e_j(n)} \cdot \frac{\partial e_j(n)}{\partial y_i(n)} \cdot \frac{\partial y_j(n)}{\partial v_j(n)} = e_j(n) \varphi_j'(v_j(n)). \tag{6.54}$$

Claramente se ve en (6.53) y (6.54) que la señal de error $e_j(n)$ a la salida de la neurona j es un factor importante en el cálculo de $\Delta w_{ji}(n)$.

El algoritmo de retro-propagación sigue las pautas que se mencionan a continuación, como pasos en un procedimiento. Sea el conjunto de datos de entrenamiento $\{[\mathbf{x}(n), \mathbf{d}(n)]; n = 1, 2, ...N\}$

a) Inicialización: Se inicia la red con valores de configuración razonables y todos los pesos sinápticos se fijan con valores aleatorios bajos, al igual que los umbrales. Estos valores aleatorios están además uniformemente distribuidos.

b) Entrada de los ejemplos de entrenamiento: los ejemplos de entrenamiento son ingresados a la red y por cada uno de ellos se realizan las secuencias de computación en las direcciones hacia adelante y hacia atrás, característicos de esta configuración. Los pasos que siguen detallan estos cálculos.

c) Computación hacia adelante: Sea $[\mathbf{x}(n), \mathbf{d}(n)]$, un ejemplo de entrenamiento en una iteración, con el vector de entrada $\mathbf{x}(n)$ aplicado a la capa de entrada de nodos sensores y el vector de respuesta deseado $\mathbf{d}(n)$ que fue presentado a la capa de salida como *target*. Se calculan los potenciales de activación y las

señales de función de la red avanzando a través de la red capa por capa. El nivel de actividad interno de la neurona j en la capa l está dado por

$$v_j^{(l)}(n) = \sum_{i=0}^{p} w_{ji}^{(l)}(n)\, y_i^{(l-1)}(n) \tag{6.55}$$

donde $y_i^{(l-1)}(n)$ es la señal de función de la neurona i en la capa previa $l-1$ en la iteración n y $w_{ji}^{(l)}(n)$ es el peso sináptico de la neurona j en la capa l, alimentada de la neurona i en la capa anterior, $l-1$. Para $i=0$, tenemos que $y_0^{(l-1)}(n) = -1$ y $w_{j0}^{(l)}(n) = \theta_j^{(l)}(n)$ que es el umbral aplicado a la neurona j en la capa l. Si consideramos una señal de función no lineal sigmoidea, la señal de función a la salida de la neurona j en la capa l es

$$y_j^{(l)}(n) = \frac{1}{1+\exp\left(-v_j^{(l)}(n)\right)}. \tag{6.56}$$

Si la neurona j pertenece a la capa oculta $l=1$, se determina que

$$y_j^{(0)}(n) = x_j(n), \tag{6.57}$$

donde $x_j(n)$ es el j-ésimo elemento del vector de entrada $\mathbf{x}(n)$. Si la neurona *j* está en la última capa, es decir en la capa de salida $(l=L)$, entonces

$$y_j^{(L)}(n) = o_j(n). \tag{6.58}$$

Por lo tanto, el cómputo de la señal de error está dado por

$$e_j(n) = d_j(n) - o_j(n) \tag{6.59}$$

donde $d_j(n)$ es el elemento j-ésimo del vector de respuestas deseadas $\mathbf{d}(n)$.

d) Computación hacia atrás: Se calculan los gradientes locales de la red, δ capa a capa en sentido reverso.

$$\delta_j^{(L)}(n) = e_j^{(L)}(n) o_j(n)\left[1 - o_j(n)\right]$$
$$\delta_j^{(L)}(n) = y_j^{(l)}(n)\left[1 - y_j^{(l)}(n)\right]\sum_k \delta_k^{(l+1)}(n) w_{kj}^{(l+1)}(n) \tag{6.60}$$

para la neurona j en la capa de salida L, en el primer caso y en el segundo, para la neurona j en la capa oculta k .

Siguiendo la regla delta de generalización, se ajustan los pesos sinápticos en la capa j de la red:

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \alpha\left[w_{ji}^{(l)}(n) - w_{ji}^{(l)}(n-1)\right] + \eta\delta_j^{(l)}(n) y_i^{(l-1)}(n), \tag{6.61}$$

donde η representa el parámetro de la tasa de aprendizaje y α la constante de momento o impulso.

e) Iteración: Se itera el proceso presentando nuevas *épocas* de ejemplos de entrenamiento a la red hasta que los parámetros libres de la red estabilicen sus valores y el error cuadrado medio $\mathscr{E}_{\text{av}}$ calculado sobre todo el conjunto de datos de entrenamiento llegue a un mínimo. El orden en que se presentan los datos de entrenamiento a la red debería ser aleatorio de época en época. Tanto η como α son ajustadas de manera decreciente a medida que aumenta en número de iteraciones.

6.6.3. Generalización

La generalización es la propiedad más buscada en una red neuronal artificial que se diseña con el objetivo de solucionar problemas complejos. Generalización es la capacidad que tiene la red de encontrar el mejor patrón de salida para un patrón de entrada dado, desconocido para la red. El término generalización está tomado de la

psicología pero en realidad lo que ocurre en la red es una búsqueda entre los valores posibles que fueron introducidos en el entrenamiento, de ahí la importancia de contar con un conjunto de ejemplos amplio y representativo que permitirá a la red lograr la generalización deseada.

El proceso de aprendizaje es ciertamente un problema de ajuste de curvas. Una visión apropiada de la red sería considerarla como un mapeo no lineal de entradas-salidas. Esto convierte al proceso de generalización en la consecuencia de una buena interpolación no lineal de datos de entrada-salida. La red puede realizar una interpolación útil debido a que los perceptrones multicapas con funciones de activación continuas producen funciones de salida también continuas.

Cuando una red fue diseñada para lograr una buena generalización, producirá un buen mapeo entrada-salida aunque los datos de entrada sean ligeramente diferentes a aquellos usados como ejemplos. Sin embargo, si la red establece excesivas relaciones entrada-salida, se produce un efecto no deseado, la red *memoriza* los datos de entrada, y por lo tanto es menos capaz de lograr una buena generalización.

6.6.4. Validación Cruzada

En forma genérica podríamos decir que el aprendizaje de retro-propagación, básicamente se trata de ingresar un conjunto de datos a la red MLP que sirven como ejemplo para el entrenamiento y mediante el algoritmo BP se calculan los valores de los pesos sinápticos. El objetivo primordial entonces es diseñar una red neuronal que logre una buena generalización. El aprendizaje de retro-propagación codifica las relaciones entrada-salida existentes en los datos que sirven para el entrenamiento representados por los pares $\{\mathbf{x},\mathbf{d}\}$, con una estructura MLP bien entrenada, de modo que a partir de lo aprendido en el pasado, pueda generalizar en el futuro. Viéndolo de este modo, se podría decir que el proceso de aprendizaje se resume a la parametrización de la red para un dado conjunto de datos.

Para lograr un proceso de entrenamiento más eficiente y encontrar la estructura del modelo que mejor resuelva el problema, se apela a la validación cruzada. Esta

herramienta estadística surge a partir del método de partición, que consiste en reservar parte de los datos de entrenamiento como datos de validación, lo que reduce la cantidad de datos para el entrenamiento y prueba. De allí surge la validación cruzada, pero con la ventaja de que es posible utilizar todos los datos disponibles para el entrenamiento. El proceso se inicia partiendo aleatoriamente el conjunto total de datos de ejemplo en dos subconjuntos, uno de entrenamiento y otro de testeo. El subconjunto de entrenamiento es a su vez particionado en dos nuevos subconjuntos: (a) un subconjunto, de menor tamaño, utilizado para el entrenamiento propiamente dicho de la red y (b) un subconjunto utilizado para validar el modelo. La idea es validar el modelo con un conjunto de datos distinto al usado para la estimación de parámetros.

En la validación cruzada de orden k, se realiza ente procedimiento k veces, intercambiando en cada iteración del proceso de entrenamiento los subconjuntos de datos, de manera tal que al final del entrenamiento, todos ellos han sido utilizados para entrenar, validar y testear el modelo.

En general, la validación cruzada se utiliza para redes neuronales grandes que manejan datos complejos y que necesitan obtener una buena generalización. Tal es el caso de las redes neuronales diseñadas para clasificar datos biológicos según lo menciona la bibliografía consultada.

6.7. Sistemas Difusos

6.7.1. Lógica Difusa

Surge debido a la necesidad de analizar matemáticamente conceptos ambiguos e imprecisos, usados naturalmente en el lenguaje coloquial. La propuesta original tiene como autor a Lofti A. Zadeh en 1965 y desde entonces ha cobrado importancia tanto en el desarrollo de aplicaciones tecnológicas de consumo masivo (lavarropas automáticos cuyos programas están basados en la lógica difusa, acondicionadores de aire con termostatos regidos por lógica difusa, etc.) como en la implementación de sistemas expertos de variada índole. Si bien la lógica difusa no es en sí mismo un modelo de redes neuronales, la caracterización como difusas de las neuronas que pertenecen a la

red son las que le dan la identidad de difusa. Para ello, presentaremos conceptos elementales de la teoría de lógica difusa y sus bases matemáticas.

6.7.2. Teoría de Conjuntos Difusos, Funciones de Membresía y Operaciones entre Conjuntos

El concepto que tenemos de conjuntos es muy sencillo, dado un conjunto y grupo de elementos, cada elemento pertenece o no a ese conjunto. Es como una regla lógica, una sentencia es verdadera o falsa, no hay lugar para ambigüedades. En un sistema de lógica difusa, los conjuntos de este tipo se llaman *nítidos*. Sin embargo, en este nuevo contexto, donde las ambigüedades pueden ser descriptas matemáticamente, los límites de los conjuntos no están establecidos. La pertenencia a un *conjunto difuso* está dada según grados de membresía, la que se mide de manera continua y en términos numéricos, en un rango de $[0,1]$, donde 0 representa el valor absolutamente falso y 1 absolutamente nítido. Sea X el conjunto universal y sea x un elemento cualquiera de X. Sea A un conjunto difuso, caracterizado por una *función de membresía* $\mu_A(x)$ que mapea el conjunto universal al intervalo real $[0,1]$. Cuanto más se acerque $\mu_A(x)$ a 1, mayor grado de pertenencia de x a A. La expresión formal está dada por:

$$A = \{(x, \mu_A(x)) | x \in X\}. \tag{6.62}$$

Cada elemento de A es un par ordenado formado por un elemento de X y una función aplicada a x, que define una asignación a los elementos de X, de valores comprendidos entre $[0,1]$. Esta función es la función de membresía, nombrada anteriormente.

Los conjuntos difusos poseen dos componentes primordiales, el *núcleo* y el *soporte*. Éste último se refiere al conjunto de elementos de X con grado de membresía no-nulo en el conjunto difuso. El núcleo es el conjunto de todos los elementos cuya membresía es absolutamente verdadera, es decir de grado 1 en el conjunto difuso. La Figura 6.10

ilustra estos subconjuntos (Monzón, 2004). El corte-α que se muestra en la Figura 6.10, son los valores de X con grado mínimo de membresía, dado por α. Gracias a este concepto, es posible definir la *convexidad* de un conjunto difuso, que establece $\forall x \in X$, $\forall \lambda \in [0,1]$ se verifica que

$$\mu_A\left(\lambda x_1 + (1-\lambda) x_2\right) \geq \min\left(\mu_A(x_1), \mu_A(x_2)\right) \tag{6.63}$$

A será convexo si todos sus cortes-α son convexos.

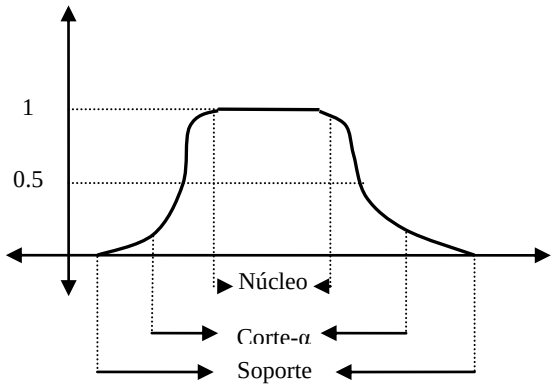

Figura 6.10. Características de un conjunto difuso

Resumiendo, podemos decir que una función de membresía asigna a cada punto en el espacio de entradas, un grado de pertenencia en el rango de $[0,1]$. Es además una función continua representada por una curva que puede ser de variadas formas, dependiendo de la conveniencia. Los tipos más usados están graficados en la Figura 6.11.

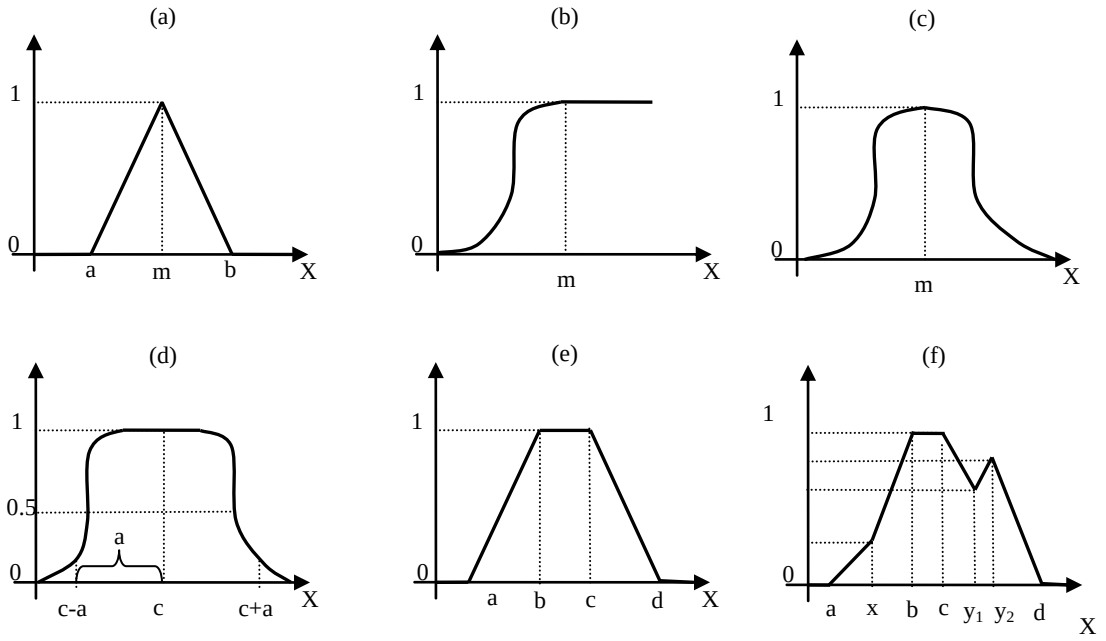

Figura 6.11. Tipos de funciones de membresía. (a) triangular. (b) gamma. (c) gaussiana. (d) campana general. (e) trapezoidal. (f) trapezoidal extendido

Las ecuaciones que describen están curvas son las siguientes

Triangular:
$$A(x)=\begin{cases}0 & \text{si } x\le a\\ (x-a)/(m-a) & \text{si } x\in(a,m]\\ (b-x)/(b-m) & \text{si } x\in(m,b)\\ 0 & \text{si } x\ge b\end{cases} \tag{6.64}$$

Gamma:
$$A(x)=\begin{cases}0 & \text{si } x\le a\\ 1-e^{-k(x-a)^2} & \text{si } x>a\end{cases} \tag{6.65}$$

Gaussiana:
$$A(x)=e^{-k(x-m)^2} \tag{6.66}$$

Campana General: $$A(x)=\frac{1}{\left(1+\left[\left(\left(x-c\right)/a\right)^{2}\right]^{b}\right)} \tag{6.67}$$

Trapezoidal: $$A(x)=\begin{cases}0 & \text{si } (x\le a)\text{or}(x\ge d)\\ (x-a)/(b-a) & \text{si } x\in(a,b]\\ 1 & \text{si } x\in(b,c)\\ (d-x)/(d-c) & \text{si } x\in(c,d)\end{cases} \tag{6.68}$$

Otro aspecto importante en la teoría de la lógica difusa es definir las operaciones entre conjuntos difusos que justamente están definidas por sus funciones de membresía. Tales operaciones son la *unión*, *intersección* y *complemento*. También la inclusión e igualdad de conjuntos pueden representarse con conjuntos difusos.

Sean A y B dos conjuntos difusos en X con funciones de membresía $\mu_A(x)$ y $\mu_B(x)$ respectivamente, se define la función de membresía de la unión entre A y B como

$$\mu_{A\cup B}(x)=\max\left(\mu_A(x),\mu_B(x)\right) \tag{6.69}$$

La función de membresía de la intersección está dada por

$$\mu_{A\cap B}(x)=\min\left(\mu_A(x),\mu_B(x)\right) \tag{6.70}$$

El complemento del conjunto A está definido por la siguiente función de membresía

$$\mu_{\bar{A}}(x)=1-\mu_A \tag{6.71}$$

La inclusión y la igualdad de conjuntos están dadas por

$$A\subseteq B\Leftrightarrow \mu_A(x)\le\mu_B(x) \tag{6.72}$$

y $$A = B \Leftrightarrow \mu_A(x) = \mu_B(x) \tag{6.73}$$

6.7.3. Reglas Difusas IF-THEN (Si-Entonces)

Las reglas de lógica difusa se usan en sistemas difusos y permite representar la relación de entradas del sistema con sus salidas en términos lingüísticos. Son un conjunto de proposiciones lógicas del tipo IF-THEN que proporcionan gran flexibilidad y que emulan el razonamiento humano.

En una regla difusa se asume lo siguiente:

Si x es A entonces y es B

IF x es A THEN y es B

A y B son valores lingüísticos definidos por los conjuntos difusos que pertenecen al universo X e Y respectivamente. Se llama premisa a la expresión "x es A" y consecuencia o conclusión a la expresión "y es B", como sucede en la lógica clásica. El antecedente representa las pruebas que deben realizarse sobre los datos existentes y el consecuente, representa las acciones a tomar en caso que del antecedente resulte un valor de verdad Es decir: IF (evaluar la condición) THEN (realizar la operación apropiada). La novedad en lógica difusa es que si el antecedente es una proposición difusa, verdadera con cierto grado de membresía, entonces el consecuente es también verdadero y en el mismo grado. Cualquier dato modificado a partir de esta acción o cualquier nuevo dato incorporado, puede servir como entrada a otras reglas. Estas reglas difusas pueden también combinarse mediante el operador lógico AND (conjunción) o bien OR (disyunción). Otra posibilidad es encadenar varias reglas difusas de manera que el consecuente de una de ellas sea el antecedente de la siguiente.

Para generar reglas difusas, se debería empezar por identificar las variables que intervienen con sus posibles valores. Luego identificar las restricciones que inducen las proposiciones y por último representar cada una de esas restricciones con una regla difusa (Jang 1999).

La salida de una regla difusa es un conjunto difuso. Éstos son agrupados (*aggregation*) en un único conjunto difuso de salida. Este proceso se cumple solo una vez para cada variable de salida. La entrada a este proceso está definida por las funciones de salida de las reglas IF-THEN, (Hanselman y Littlefield, 1995).

6.7.4. Razonamiento Difuso

Es un proceso de inferencia que saca conclusiones a partir de reglas difusas IF-THEN utilizando conjuntos difusos.

Sean $A \in X, A' \in X, B \in Y$ conjuntos difusos. Sea la implicación difusa $A \to B$ expresada como una relación $R \in X \times Y$. El conjunto difuso B inducido por "x es A" y la regla difusa "si x es A entonces y es B" están definidos según la expresión que sigue

$$\mu_{B'}(y) = \max_x \min\left[\mu_{A'}(x), \mu_R(x,y)\right] = \vee_x \left[\mu_{A'}(x) \wedge \mu_R(x)\right] \qquad (6.74)$$

6.8. Redes Neuronales Difusas (FNN -Fuzzy Neural Networks).

Una Red Neuronal Difusa o Red Difusa es una red neuronal que aplica los axiomas de la lógica difusa para resolver problemas complejos que involucran la toma de decisiones basada en información de tipo lingüística o bien cuando los datos de entrada que se tiene son del tipo numérico y se pretende aplicar la lógica difusa. En cualquiera de estos casos, este sistema híbrido inteligente, llamado también, sistema neuro-difuso, se convierte en una alternativa más que acertada para encontrar una solución apropiada.

Un sistema neuro-difuso combina el estilo de razonamiento humano de los sistemas difusos con el aprendizaje y la estructura interconectada de las redes neuronales. El mencionado estilo de razonamiento humano, consiste en el uso de los conjuntos difusos y el modelo lingüístico, definido por las reglas difusas IF-THEN. La robustez de este

híbrido se halla en que es en realidad una aproximación universal con la habilidad de solicitar reglas IF-THEN interpretables.

Un sistema difuso involucra bloques que, una vez ingresados los datos al sistema, los interpreta y sobre una base de un conjunto de reglas, asigna valores al vector de salida (Monzón, 2004). Su representación gráfica se muestra en la Figura 6.12. Las funciones de estos bloques pueden resumirse en:

Base de Reglas Difusas: contiene reglas difusas del tipo IF-THEN.

Máquina de Inferencia Difusa: es la encargada de la toma de decisiones mediante las operaciones de inferencia sobre las reglas.

Interfase de Disfuminación (fuzzification): transforma las entradas nítidas en grados de apareo con valores lingüísticos.

Interfase de Des-disfuminación (defuzzificación): transforma los resultados difusos de la inferencia en valores nítidos de salida.

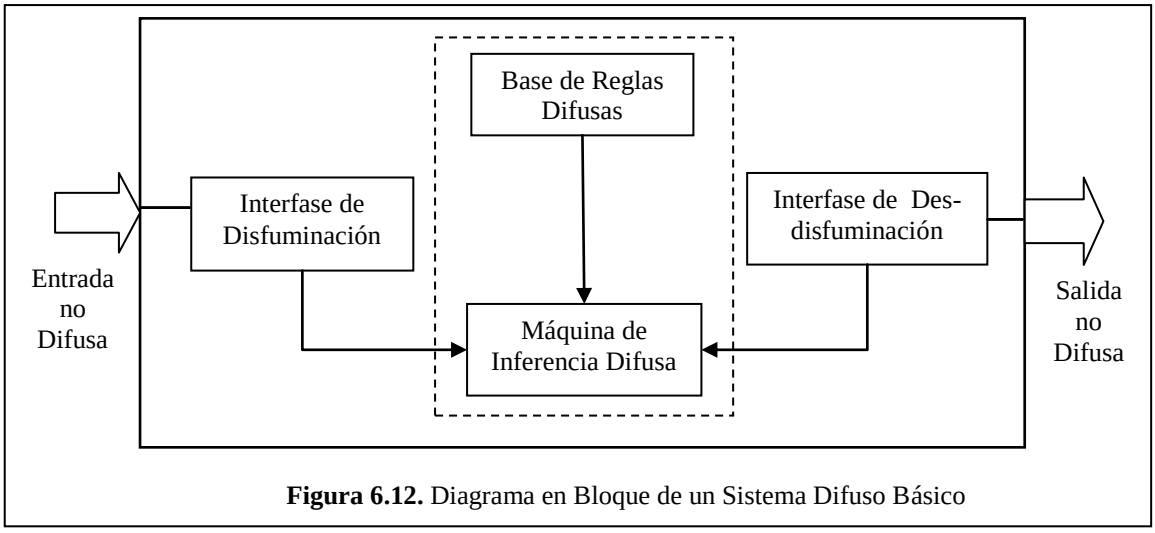

Figura 6.12. Diagrama en Bloque de un Sistema Difuso Básico

El proceso global de *inferencia* o proceso de razonamiento difuso comprende las siguientes etapas (Jang et al., 1993):

Disfuminación: Las variables que ingresan al sistema son comparadas con las funciones de membresía para establecer valores de pertenencia de cada etiqueta lingüística. Se determina el grado de verdad de cada premisa conocido como su *alfa.* Si la regla puede ser aplicada, entonces ésta se dispara.

Inferencia: etapa en la que se calcula el valor de verdad de la premisa de cada regla y se lo aplica entonces a la conclusión. Esto genera un subconjunto difuso que será asignado a cada variable de salida por cada regla. El operador **min** (*intersección*) es el más utilizado. La función de membresía sufre un *recorte* según el grado de verdad correspondiente que fue computado con la regla de la premisa (operación lógica AND difusa). La combinación de los valores de membresía en la premisa permite obtener la fuerza de disparo, equivalente al peso, de cada regla.

Composición: Los subconjuntos difusos asignados a cada variable de salida conforman un único subconjunto difuso para cada variable de salida. Se utiliza el operador **max** (*unión*), que toma el máximo sobre todos los conjuntos asignados a la variable por cada regla y produce la salida combinada (operación lógica OR difusa). Se generan los consecuentes de cada regla, según el peso de cada regla.

Des-disfuminación: esta etapa permite convertir un conjunto difuso de salida en valor matemático nítido. Los métodos más populares para llevar a cabo esta etapa son el método del *máximo*, en el que el valor nítido de salida que se elige es uno de los valores variables para el cual el subconjunto difuso alcanza su máximo valor de verdad; y el método del *centroide*, que como valor nítido de la variable de salida, halla el valor variable del centro de gravedad de la función de membresía para el valor difuso.

La **inferencia difusa** hace referencia al proceso de mapear a partir de datos de entrada, una salida utilizando la lógica difusa. Este mapeo provee una base a partir de cual pueden tomarse decisiones. Los sistemas de inferencia difusa han sido ampliamente aplicados en distintos campos como el control automático, clasificación de datos, análisis de decisiones, sistemas expertos, etc. (Sun et al., 2007).

Los sistemas de inferencia difusa pueden clasificarse en dos grandes grupos (Mitra y Hayashi, 2000):

1) Modelo Mamdani: incluye modelos lingüísticos basados en la colección de reglas IF-THEN, cuyos antecedentes y consecuentes utilizan valores difusos. El comportamiento del sistema puede ser descripto en términos *naturales*. Está representado por

$$R^i : \text{IF } x_1 \text{ es } A_1^i,\ x_2 \text{ es } A_2^i, ...,\ x_n \text{ es } A_m^i \text{ THEN } y^i \text{ es } B^i$$

donde $R^i;\ i = 1, 2, ..., l$ es la i-ésima regla difusa, $x_j;\ j = 1, 2, ..., n$ es la entrada, y^i es la salida de la regla difusa R^i y $A_1^i, A_2^i, ..., A_m^i, B^i;\ i = 1, 2, ...l$ son funciones de membresía asociadas con términos lingüísticos.

2) Modelo Sugeno: usa una estructura de regla que tiene un antecedente difuso y partes consecuentes funcionales. Está representado por:

$$R^i : \text{IF } x_1 \text{ es } A_1^i,\ x_2 \text{ es } A_2^i, ...,\ x_n \text{ es } A_m^i \text{ THEN } y^i = a_0^i + a_1^i x_1 + ... + a_n^i x_n$$

La propuesta aproxima un sistema no-lineal con una combinación de varios sistemas lineales. La salida es una combinación lineal de las variables de entrada.

Ambos modelos son capaces de representar tanto información cualitativa como cuantitativa y permiten la aplicación de técnicas de aprendizaje poderosas para su identificación. La optimización de parámetros se produce en forma iterativa utilizando algún método de optimización no lineal.

6.8.1. Modelos de Redes Difusas

Combinado con la habilidad de aprender de las redes neuronales, el sistema de inferencia difusa ha demostrado ser una construcción matemática poderosa, que permite también la expresión simbólica de los resultados de la máquina de aprendizaje. En la implementación de los sistemas neuro-difusos, Wang y Mendel (1992) propusieron

aplicar el algoritmo de retro propagación en el entrenamiento de sistemas difusos en la búsqueda de pares deseados de vectores de entrada-salida. La esencia del método es lograr un correcto mapeo no lineal, mediante el entrenamiento progresivo y lograr la optimización del entrenamiento gracias a la incorporación de reglas lingüísticas. De este modo, el sistema realmente combina las dos estrategias, utiliza información numérica (pares entrada-salida) e información lingüística (reglas IF-THEN).

6.8.2. Red Adaptiva

La ausencia de un método sistemático, que permita la transformación del conocimiento humano en una base de reglas difusas y la dificultad que presenta el ajuste de parámetros en las funciones de membresía para minimizar así errores en la salida, dieron lugar al desarrollo de un sistema adaptivo neuro-difuso que produce de manera automática reglas IF-THEN a partir de datos de entrada. El sistema es capaz de generar los pares de entrada-salida deseados identificando las funciones de membresía.

El sistema de inferencia basado en redes neuronales adaptivas (ANFIS - Adaptive Neuro-Fuzzy Inference System) es una propuesta interesante que implementa el sistema difuso de Takagi-Sugeno. El modelo presenta una estructura de cinco capas y construye un sistema de inferencia difuso. ANFIS puede emplear tres métodos de aprendizaje para actualizar los parámetros de las funciones de membresía. Ellos son: retro-propagación para todos los parámetros; un sistema híbrido que consiste en retro-propagación para todos los parámetros asociados con las funciones de membresía de entrada y la estimación de los mínimos cuadrados para los parámetros asociados a las funciones de membresía de salida; y por último, el método del agrupamiento sustractivo. En particular, en nuestros ensayos aplicamos la primera de las opciones, el algoritmo de retro-propagación para todos los parámetros

Una red de este tipo, adaptiva, está formada por varias capas de alimentación directa, donde cada nodo cumple una función de nodo, que le es particular sobre las señales que ingresan a él y con parámetros que le son propios. Estas funciones pueden variar de nodo a nodo. La Figura 6.13 esquematiza una red neuro-difusa adaptiva. Con el

objetivo de simplificar la gráfica, se muestra un sistema con 2 entradas x e y y 1 salida, con una base de reglas del tipo Takagi-Sugeno, de la forma

1) IF x es A_1 e y es B_1 THEN $f_1 = p_1 x + q_1 y + r_1$

2) IF x es A_2 e y es B_2 THEN $f_2 = p_2 x + q_2 y + r_2$

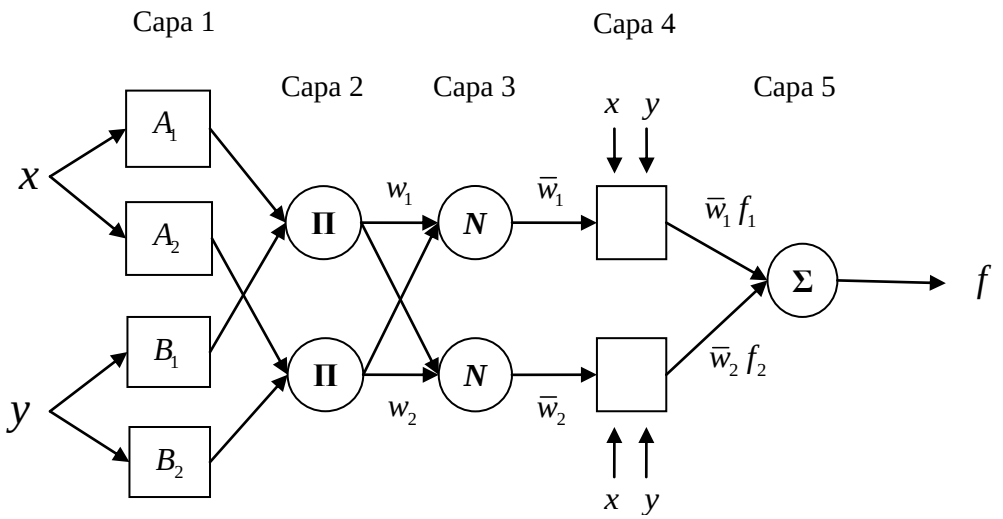

Figura 6.13. Sistema de Inferencia Difusa Basado en Redes Neuronales Adaptivas -ANFIS

Los nodos cuadrados representan nodos adaptables, con parámetros que pueden ser modificados y los nodos circulares, son los nodos fijos, sus parámetros permanecen invariables. Para los nodos de una misma capa, las funciones de membresía son del mismo tipo.

A continuación se describen cada una de las capas:

Capa 1: Sus nodos son adaptables, la función del nodo i viene dada por:

$$o_{i1} = \mu_{A_i(x)}, \tag{6.75}$$

donde A_i es el rótulo lingüístico asociado con la función de nodo y x es la entrada al nodo i. o_{i1} es la función de membresía de A_i que determina en qué grado x satisface al cuantificador A_i. La forma de la función que más se utiliza es de la forma de (6.67), campara generalizada. Esta función tiene un rango de variabilidad de $[0,1]$.

Capa 2: De nodo fijos. Tienen como rótulo **Π**, que multiplica las señales de entrada y produce el siguiente resultado:

$$w_i = \mu_{A_i(x)} \cdot \mu_{B_i(y)}; \qquad i = 1,2 \tag{6.76}$$

Cada nodo representa la fuerza de disparo de una regla.

Capa 3: También de nodos fijos, su rótulo ***N*** simboliza la fuerza de disparo normalizada. El nodo i calcula la razón entre de la fuerza de disparo de la regla i y la suma de las fuerzas de disparo de todas las reglas.

$$\bar{w}_i = \frac{w_i}{w_1 + w_2}; \qquad i = 1,2 \tag{6.77}$$

Capa 4: Adaptiva, con función de nodo dada por

$$o_{i4} = \bar{w}_i f_i = \bar{w}_i \left(p_i x + q_i x + r_i \right), \tag{6.78}$$

donde $\bar{w}_i$ es la salida de la capa 3 y $\{p_i, q_i, r_i\}$ es el conjunto de parámetros conocidos como consecuentes.

Capa 5: De parámetros fijos, es la encargada de computar la salida como la sumatoria, de allí su rótulo **Σ**, de todas las señales de entradas:

$$o_{i5} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{6.79}$$

Con esta estructura entonces queda conformada la red neuronal adaptiva.

7. Algoritmos Genéticos

7.1. Computación evolutiva

La computación evolutiva es el conjunto de algoritmos computacionales destinados a la búsqueda y la optimización de procesos inspirados en la evolución biológica. Sus características más significativas están relacionadas con el hecho de que la optimización o la búsqueda esta conducida:

1) basada en puntos de búsqueda múltiple o soluciones candidatas (búsqueda basada en la población),

2) usando operaciones inspiradas por evolución biológica, tales como cruzamiento y mutación,

3) basada en búsqueda probabilística y operaciones probabilísticas,

4) usando poca información del espacio de búsqueda.

Los paradigmas típicos basados en computación evolutiva son

AG (Algoritmos Genéticos): usualmente representan soluciones para cromosomas con codificación binaria y la búsqueda de soluciones candidatas en el espacio genotipo, usando operaciones de AG. Ellas son: selección, cruzamiento y mutación. De las tres operaciones, la de cruzamiento es la dominante.

EE (Estrategias de Evolución): representa soluciones como se expresan por los cromosomas con códigos de números reales (fenotipo) y búsquedas por la mejor solución en el espacio de fenotipo usando operaciones de EE, cruzamiento y mutación. La mutación de un número real se realiza mediante la suma de ruido gaussiano y controla los parámetros de una distribución gaussiana permitiéndole la convergencia a un óptimo global.

PE (Programación Evolutiva): similar a AG. Las diferencias primarias es que la mutación es solamente el operador PE. Utiliza codificación en números reales y la mutación a veces cambia la estructura, es decir la longitud del código PE.

Las diferencias y las similitudes entre AG y PE devienen de sus antecedentes. AG se inició a partir de la simulación de la evolución genética, mientras que la PE se inició a partir de la evolución ambiental.

PG (Programación Genética): utiliza estructura de árbol para representar un algoritmo computacional o crear nuevas estructuras de tareas. La operación de cruzamiento no se aplica a valores numéricos sino a ramas de la estructura de árbol. Si consideramos una relación de aplicación con redes neuronales, los AG y EE determinan los mejores pesos sinápticos, que es el aprendizaje de la red neuronal. Sin embargo, PG determina la mejor estructura de la red neuronal, es decir la configuración de la red, (Da Ruan, 1997).

7.2. El método de búsqueda basado en Algoritmos Genéticos

Antes de describir el modus operandi de los Algoritmos Genéticos como un método de búsqueda, resulta necesario describir el significado de los términos técnicos empleados en AG, algunos de los cuales ya fueron mencionados. La Tabla 7.1 lista un glosario de los términos más usados.

Tabla 7.1 Términos técnicos usados en los Algoritmos Genéticos

Cromosoma	Vector que representa soluciones de una tarea de aplicación.
Gen	Fragmento de un cromosoma que representa un parámetro del problema
Selección	Mecanismo por el cual las mejores soluciones tienden a ser elegidas para cruzarse y generar descendencias, y las soluciones no tan buenas tienden a desaparecer de la población.
Individuo	Cada vector solución que es cada cromosoma
Población	Conjunto de posibles soluciones. Es la estructura que contiene el material genético responsable de determinar las características de un individuo. Se compone de un conjunto de genes
Tamaño de la Población	Número de cromosomas
Función Fitness	Representa la función objetivo que debe ser optimizada. Proporciona la medida de lo que se quiere alcanzar con el AG.
Fenotipo	Es la estructura del individuo decodificada; es decir, es la solución en valores reales, no codificados. El fenotipo es codificado dentro del genotipo, mediante una codificación que depende del problema
Genotipo	Conjunto de valores que asumen los genes y que determinan una estructura única. Estructura de tamaño finita sobre un alfabeto finito. Ésta queda determinada de acuerdo a la iteración de los genes.

7.3. Funcionamiento General del AG

El primer paso en la aplicación de un algoritmo genético consiste en la generación de una población inicial. En general esta población se genera de manera aleatoria, y el tamaño de dicha población (la cantidad de individuos que la compone) es un parámetro que se define durante el diseño del algoritmo genético. Una vez generada esta población se debe evaluar la aptitud (fitness) de cada individuo.

El operador de selección es el encargado de decidir cuales individuos contribuirán en la formación de la próxima generación de individuos. Este mecanismo simula el proceso de selección natural, mediante el cual sólo los individuos más adaptados al ambiente se reproducen. El mecanismo de selección forma una población *intermedia*, que esta compuesta por los individuos con mayor aptitud de la generación actual.

La siguiente fase del algoritmo consiste en la aplicación de los operadores genéticos. El primero de ellos es la cruza, y su función es recombinar el material genético. Se toman aleatoriamente dos individuos que hayan sobrevivido al proceso de selección y se recombina su material genético creando uno o más descendientes, que pasan a la siguiente población. Este operador se aplica tantas veces como sea necesario para formar la nueva población. El último paso consiste en la aplicación del operador de mutación. Este operador, que en general actúa con muy baja probabilidad, modifica algunos genes del cromosoma, posibilitando de esta manera la búsqueda de soluciones alternativas.

Una vez finalizado el proceso de selección, cruza y mutación se obtiene la siguiente generación del algoritmo, la cual será evaluada, repitiéndose el ciclo descripto previamente. Tras cada iteración la calidad de la solución generalmente va incrementándose, y los individuos representan mejores soluciones al problema.

Existen varios tipos de algoritmos evolutivos, entre los cuales el algoritmo genético, o AG, es el más conocido. Básicamente, el AG realiza un ciclo consistente en un proceso iterativo para hacer que la población evolucione, a cada paso de este proceso se lo conoce como *generación*, y a la condición para su finalización se la llama *convergencia*. El funcionamiento del AG se puede esquematizar en el siguiente pseudocódigo.

Comienzo /* algoritmo genético */

Generar la población inicial;

Computar la aptitud de cada individuo;

Mientras no terminado *hacer ciclo*

Comenzar

Seleccionar individuos de las generaciones anteriores para reproducción;

Crear descendencia aplicando recombinación y/o mutación a los individuos seleccionados;

Computar la aptitud de los nuevos individuos;

Eliminar a los individuos antiguos para hacer lugar para los nuevos cromosomas e insertar la descendencia en la nueva generación;

Si la población ha convergido *Entonces* terminado:= VERDADERO;

Fin

Fin

7.4. Elementos de un Algoritmo Genético

7.4.1. El cromosoma

En los AGs cada cromosoma es una estructura de datos que representa una de las posibles soluciones del espacio de búsqueda del problema. Los Cromosomas son sometidos a un proceso de evolución que envuelve evaluación, selección, recombinación sexual (crossover) y mutación. Después de varios ciclos de evolución la población deberá contener individuos más aptos.

Los AG trabajan manipulando cromosomas, que son estructuras que codifican las distintas soluciones de un determinado problema. La forma en que se codifican los cromosomas es dependiente de cada problema en particular, y suele variar de problema en problema. Un cromosoma esta compuesto por un conjunto de genes. Cada gen representa una característica particular del individuo y ocupa una posición determinada en el cromosoma, llamada locus. A cada uno de los valores que puede tomar un gen se lo conoce como alelo.

Existen diversas formas de codificar los genes. Estos se pueden representar como cadenas binarias, números enteros, números reales, etc.

7.4.2. Evaluación y aptitud

La función de evaluación, o función objetivo, provee una medida de la performance de conjunto de parámetros. Indica que tan buena es la solución obtenida tras la aplicación de un conjunto de parámetros, independientemente de la aplicación de esta función sobre otro conjunto de parámetros. En general, esta medida se obtiene tras la decodificación de un cromosoma en el conjunto de parámetros que representa.

Por su parte, la función de adaptación siempre esta definida con respecto al resto de los individuos de la población. Indica que tan buena (o mala) es una solución comparada con el resto de las soluciones obtenidas hasta el momento. El valor obtenido por esta función se traduce, tras la aplicación del operador de selección, en oportunidades de reproducción. Aquellos que tengan mayor aptitud tendrán mayores posibilidades de reproducirse y transmitir su material genético a la próxima generación. Es la base para determinar qué soluciones tienen mayor o menor probabilidad de sobrevivir.

7.4.3. Población

Existen dos cuestiones fundamentales a resolver: primero, como generar la primera población, y luego, cual debe ser el tamaño de la población. La primera población debería tener un contenido genético tan amplio como sea posible para que pueda explorar la totalidad del espacio de búsqueda. En esta población deberían estar presentes todos los distintos alelos posibles de cada gen. Para lograrlo, en la mayoría de los casos se elige a la primera población de forma aleatoria. Sin embargo, a veces se puede usar alguna clase de heurística para establecer la población inicial. Así, la aptitud promedio de la población ya es alta y puede ayudar al algoritmo genético a encontrar soluciones más rápidamente. Pero al hacer esto hay que asegurarse de que el contenido genético todavía siga siendo lo suficientemente amplio. De lo contrario, si la población carece de diversidad, el algoritmo solo explorará una pequeña parte del espacio de búsqueda y nunca encontrará soluciones óptimas globales.

El tamaño de la población no es un problema. Si el tamaño de la población es muy grande más fácil es explorar el espacio de búsqueda. Sin embargo, el tiempo necesario para obtener una nueva generación a partir de la generación actual será mayor, dado que se estarán procesando individuos innecesarios. Si el tamaño de la población es muy chico podría llegar a darse el caso en que la población converja rápidamente a una solución que no sea lo suficientemente buena. En la práctica se suelen utilizar poblaciones de 20 individuos, pudiéndose aumentar el mismo en base a la complejidad del problema o a la calidad de la solución requerida. En algunos casos se utiliza un tamaño de población inicial, el cual se va incrementando a medida que la población tiende a ser homogénea.

7.4.4. Selección

Los individuos son copiados de acuerdo a su evaluación en la función objetivo (aptitud). Los más aptos tienen mayor probabilidad a contribuir con una o más copias en la siguiente generación (se simula selección natural).

Una elección adecuada para cada operador puede influir decisivamente en la eficiencia del proceso de búsqueda. A continuación se realiza una mención sobre las variantes más utilizadas de cada uno de los operadores.

a) Selección por ruleta

Es la más clásica y utilizada de las selecciones proporcionales, la cual simula una ruleta, donde cada cadena tiene un espacio en ella proporcional a su valor de aptitud.

$$\Pr(h) = \frac{Aptitud}{\sum_{j=1}^{N} Aptitud(h_j)} \tag{7.1}$$

Se pueden seleccionar individuos de la población actual, generar una nueva población y reemplazar con ella completamente a la población que se tenía. También a veces se mantienen los N mejores individuos de una población a la siguiente (esto parece ser la mejor opción).

b) Selección proporcional

La selección proporcional escoge a los individuos basada en sus valores de aptitud relativos a la aptitud de los otros individuos en la población.

c) Selección por torneo

Se seleccionan aleatoriamente dos individuos de la población, pasando a la siguiente generación aquel individuo que tiene mayor aptitud. Este proceso se repite N veces, donde N es el tamaño de la población. La selección por torneo no asegura que el número de copias que pasan a la próxima generación sea igual al esperado. Este operador se puede implementar tomando cualquier número de individuos.

7.4.5. Operadores Genéticos

a) Cruza

Tiene una alta probabilidad de ser utilizado y es considerado como el más importante dentro de los AG. Permite la generación de nuevos individuos tomando características de individuos padres. Consiste en seleccionar dos individuos después del proceso de selección, determinar una posición de cruce aleatoria e intercambiar las cadenas entre la posición inicial y el punto de cruce y el punto de cruce y la posición final. Existen diferentes tipos de cruza.

b) Cruza Uniforme

Este operador de cruza asigna aleatoriamente los pesos. Cada hijo tiene una probabilidad de 0.5 de recibir los genes de su padre y por ende de recibirlos de su madre.

c) Cruza Simple

En esta variante del operador de cruza se toma aleatoriamente un punto de cruza. Luego, a un hijo se le asignan todos los genes del padre ubicados a la izquierda del punto de cruza, y todos los genes de la madre ubicados a la derecha del punto de cruza. El segundo hijo es el complemento del primero. La figura 5.1 muestra un ejemplo de cruza simple en el cual se toma como punto de cruza el primero de ellos.

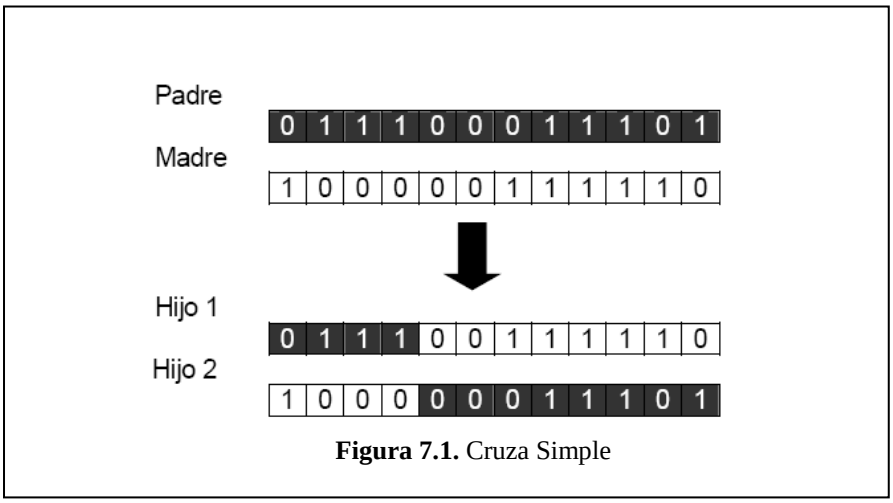

Figura 7.1. Cruza Simple

d) Cruza Multipunto

Esta variante del operador de cruza es similar al operador de cruza simple, sólo que la cantidad de puntos de cruza se determina aleatoriamente. Cada hijo recibe los genes entre dos puntos de cruza sucesivos de cada uno de los padres, de manera intercalada.

Cabe destacar que la cruza simple es un caso particular de la cruza multipunto, con un único punto de cruza. La figura 5.2 muestra un ejemplo de cruza multipunto, tomando dos puntos de cruza.

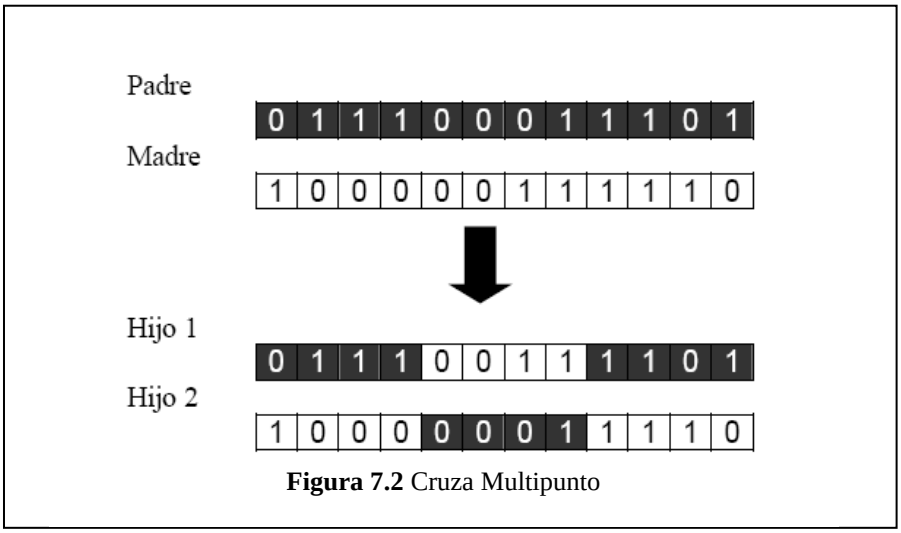

Figura 7.2 Cruza Multipunto

7.4.6. Mutación

La mutación es un operador de búsqueda simple consistente en cambios aleatorios en los valores de los genes en un cromosoma. Opera sobre un solo individuo, determina una posición y la invierte con cierta probabilidad. Permite salir de máximos locales.

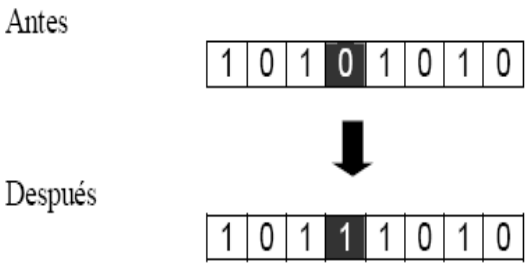

Figura 7.3 Ejemplo de Mutación

7.4.7. Parámetros y Criterios de Parada

En un algoritmo genético varios parámetros controlan el proceso de evolución:

- *Tamaño de la Población*: número de puntos del espacio de búsqueda siendo considerados en paralelo.
- *Tasa de Crossover*: probabilidad de un individuo de ser recombinado con otro.
- *Tasa de Mutación*: probabilidad de que el contenido de cada posición/gen del cromosoma sea alterado.
- *Número de Generaciones*: número total de ciclos de evolución de un algoritmo genético.

El último parámetro se emplea generalmente como criterio de parada de un GA. Un algoritmo genético puede ser descrito como un proceso continuo que repite ciclos de evolución controlados por un criterio de parada.

7.5. Funcionamiento de un Algoritmo Genético Simple

Se genera aleatoriamente la población inicial, constituida por un conjunto de cromosomas, o cadenas de caracteres que representan las soluciones posibles del problema. A cada cromosoma de esta población, se le aplicar la función de aptitud, a fin de saber la bondad de la solución que codifica. Sabiendo la aptitud de cada cromosoma, se procede a la selección de los que se cruzarán en la siguiente generación (presumiblemente, se escogerá a los "mejores"). Realizada la selección, se procede a la reproducción sexual o cruza de los individuos seleccionados.

En esta etapa, los sobrevivientes intercambiarán material cromosómico y sus descendientes formarán la población de la siguiente generación. Las 2 formas más comunes de reproducción sexual son:

- Uso de un punto único de cruza y
- Uso de 2 puntos de cruza.

Cuando se usa un solo punto de cruza, éste se escoge de forma aleatoria sobre la longitud de la cadena que representa el cromosoma, y a partir de él se realiza el intercambio de material de los 2 individuos.

Cuando se usan 2 puntos de cruza, se procede de manera similar. Normalmente, la cruza se maneja en la implementación del AG como un porcentaje que indica con qué frecuencia se efectuará.

Esto significa que no todas las parejas de cromosomas se cruzarán, sino que habrá algunas que pasarán intactas a la siguiente generación.

Parte II.

Aplicaciones

8. Introducción

En este capítulo describiremos los algoritmos computacionales diseñados e implementados con cada una de las metodologías descriptas en la Parte 1 de la tesis. Exhibiremos también los resultados obtenidos y los discutiremos.

Antes de iniciar la explicación de los algoritmos, es necesario describir los índices utilizados para la evaluación de los datos obtenidos. Tales índices son medidas estandarizadas especialmente diseñadas para el análisis de datos biomédicos.

Los índices utilizados evalúan, en el primer caso, la performance del algoritmo clasificador de arritmias, ya sea con Bancos de Filtros o Redes Neuronales. El segundo conjunto de índices, propuesto por Vincent y colaboradores, (Vince et al., 2006) evalúa la performance del algoritmo encargado de separar las fuentes en las aplicaciones de ICA, descriptas en el capítulo 5.

8.1 Índices de Performance

8.1.1 Índices para la evaluación del algoritmo clasificador

Para evaluar la performance de los algoritmos clasificatorios de latidos electrocardiográficos, recurrimos a los índices normalizados (AAMI, 1987). Los datos que participan en la construcción de los índices son:

TP (True Positive): Verdadero Positivo. Son aquellos latidos PVC detectados como tales.

FP (False Positive): Falso Positivo. Son los latidos Normales detectados como PVC.

FN (False Negative): Falso Negativo. Son los latidos PVC detectados como Normales.

TR (True Negative): Verdadero Negativo. Son los latidos Normales detectados como tales.

De la definición se desprende que son mutuamente excluyentes. Con estos datos se construyen los índices de performance del clasificador. Tales índices son:

Sensibilidad (S), representa la fracción de eventos reales correctamente detectados entre el total de los eventos reales. Su fórmula está dada por:

$$S = \frac{TP}{TP + FN} \tag{8.1}$$

Especificidad (E), es la fracción de no-eventos (latidos Normales en nuestro caso) correctamente rechazados. Su fórmula está dada por:

$$E = \frac{TN}{TN + FP} \tag{8.2}$$

Predictividad Positiva (PP), representa la fracción de eventos reales entre todos los eventos. Su fórmula está dada por:

$$PP = \frac{TP}{TP + FP} \tag{8.3}$$

8.1.2 Índices para la evaluación del algoritmo de separación de fuentes.

A los fines de evaluar cuantitativamente el comportamiento del método propuesto en la eliminación del ruido –o recuperación de fuentes-, adoptamos el criterio propuesto por Vincent y colaboradores, basado en la comparación de cada fuente verdadera s_j con la estimada $\hat{s}_j$. El principio de la medida de la performance del algoritmo de separación es descomponer

$$\hat{s}_j = s_{\text{def}} + e_{\text{interf}} + e_{\text{ruido}} + e_{\text{artef}} \tag{8.4}$$

donde:

s_{def} : representa una deformación de la fuente s_j;

e_{interf} : representa la interferencia de las fuentes no deseadas;

e_{ruido} : representa la deformación por el ruido de perturbación;

e_{artef} : representa los artefactos del propio algoritmo de separación

La ponderación de cada uno de los términos indicados en la ecuación (8.4) puede ser evaluada numéricamente, calculando algunas relaciones, definidas como sigue:

Relación de Fuente a Interferencias (SIR)

$$SIR := 10\log_{10}\frac{\left\|s_{\text{def}}\right\|^2}{\left\|e_{\text{interf}}\right\|^2}. \tag{8.5}$$

Relación Señal Ruido (SNR)

$$SNR := 10\log_{10}\frac{\left\|s_{\text{def}} + e_{\text{interf}}\right\|^2}{\left\|e_{\text{ruido}}\right\|^2}. \tag{8.6}$$

Relación de Fuentes a Artefactos (SAR)

$$SAR := 10\log_{10}\frac{\left\|s_{\text{def}} + e_{\text{interf}} + e_{\text{ruido}}\right\|^2}{\left\|e_{\text{artef}}\right\|^2}. \tag{8.7}$$

Relación de Fuente a Distorsión (SDR)

$$SDR := 10\log_{10}\frac{\left\|s_{\text{def}}\right\|^2}{\left\|e_{\text{interf}} + e_{\text{ruido}} + e_{\text{artef}}\right\|^2}. \tag{8.8}$$

Para la evaluación de estos índices, utilizamos la herramienta BBS_EVAL (Févotte et al., 2005). Evaluamos la performance para los casos en que la única distorsión sobre $\hat{s}_j$ eran la ganancia invariante en el tiempo.

Usamos para los cálculos la herramienta BBS_EVAL, desarrollada en MATLAB® por Févotte y colaboradores. BBS_EVAL se puede aplicar cuando cabe la comparación entre las fuentes originales y los ruidos perturbadores. Utilizamos esta herramienta para compara nuestro método de separación basado en los criterios de estabilidad de Liapunov y el algoritmo FastICA desarrollados en MATLAB® por Gävert y colaboradores, (Gävert, 2005).

9. Banco de Filtros.

9.1. Elección del Banco de Filtros

Un Banco de Filtros diseñado para el procesamiento del ECG debería cumplir ciertas características. La fase lineal de los filtros del Banco de Análisis y del Banco de Síntesis nos asegura que puntos tales como el pico R en la señal electrocardiográfica tienen el mismo retardo de muestras a lo largo de todos los filtros. Otra característica incorporada es la de Reconstrucción Perfecta, ya que luego de realizar todas las tareas en el bloque Proceso (Figura 9.1), suele ser necesario reconstruir la señal, por ejemplo en el caso en que una de las tareas realizadas involucre la eliminación de ruido interferente.

Cumplir con estas características durante el diseño de un Banco de Filtros no es trivial (Strang y Nguyen, 1996; Soman et al., 1993). Afortunadamente, existen métodos de diseño de bancos de filtros que aseguran los rasgos recién destacados y que producen filtros de análisis y síntesis con anchos de banda de filtros uniforme.

La elección del número de canales en el banco de filtros también depende del método de diseño. Algunos algoritmos de diseño de banco de filtros sugieren un número potencia de dos de canales y otros simplemente un número par de canales. Luego de varias pruebas de laboratorio para definir la estructura del banco de filtros que mejor se ajuste a nuestras necesidades, es decir, detectar el pico R en la señal de ECG, clasificar los latidos en normales y PVC, eliminar ruido blanco y por último reconstruir la señal lo más parecida a la señal original, optamos por implementar un banco de filtros con una estructura de 4 filtros en el banco de análisis, 2 filtros pasa bajos y 2 filtros pasa altos y 4 filtros en el banco se síntesis, de iguales características a los del banco de análisis, $(M = 4)$. Cada uno de los filtros cuenta con 8 coeficientes $(L = 8)$, es decir $L=2M$, según recomendaciones (Vaidyanathan, 1993; Strang y Nguyen, 1996; Soman *et al*., 1993; Malvar, 1992)

9.2. Implementación del Banco de Filtros

Hemos implementado un banco de filtros digitales máximamente diezmado para el procesamiento de señales de ECG. Los procesos que realizamos sobre las bioseñales incluyen la eliminación de ruido blanco, generalmente presente en estas señales de baja frecuencia, la detección del complejo QRS y la clasificación de latidos del tipo PVC (*Premature Ventricular Contraction*). Todos los procesos mencionados, se realizaron en el bloque **PROCESO** de la Figura 9.1 y sobre un solo canal, es decir sobre una de las sub-bandas resultantes luego de aplicar el banco de análisis a la señal de ECG.

El diseño del Banco de Filtros utilizado se describe en la figura 9.1. Como puede verse, en dicha figura, implementamos el tipo II de representación polifásica, ver capítulo 3, por ser la de mejor implementación computacional (Afonso, 1997; Tran, 1999).

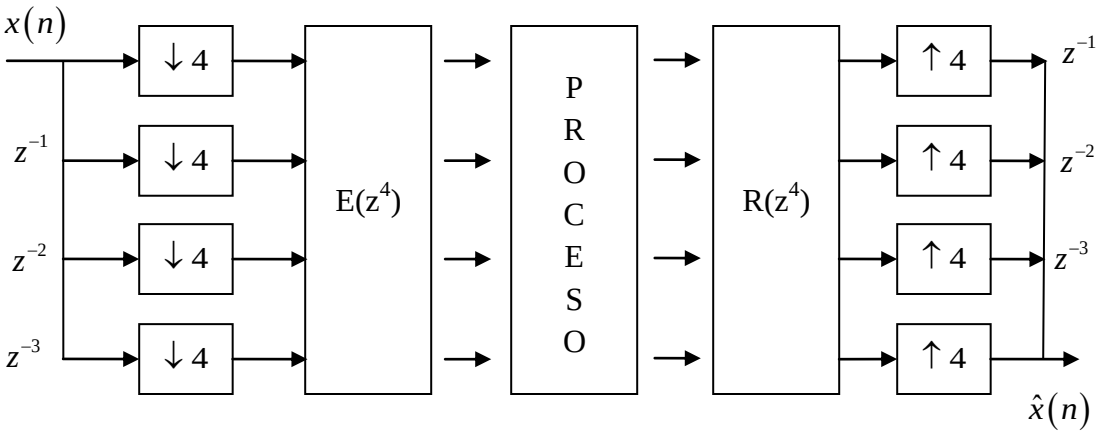

Figura 9.1. Diagrama en bloque del Banco de Filtros implementado. Representación Polifásica Tipo II

9.3.Diseño de los Filtros

Hemos diseñado los filtros que forman parte del banco utilizando la Transformada Ortogonal Superpuesta (LOT –*Lapped Orthogonal Transform*), que, como mencionamos en el capítulo 3, brinda ventajas significativas sobre otras transformadas de bloque, pues permite la transformación de cada uno de los bloques en forma superpuesta, eliminando discontinuidades al reconstruir la señal (Pereira et al., 1999; Vetterli, 2001).

Con la aplicación de la transformada LOT para obtener los coeficientes de los filtros del banco, nos aseguramos que el banco cumpla con los requerimientos mencionados en el punto 9.1.

Diseñamos una matriz LOT P, de 4 columnas y 8 filas, donde cada columna representa un filtro de análisis con 8 coeficientes. Como lo mencionamos previamente, los filtros así diseñados son filtros FIR de fase lineal. La matriz P además cumple con la propiedad de ser ortogonal no sólo a ella misma, sino a funciones en las dos superposiciones adyacentes de los filtros que están operando sobre la señal. La matriz P está listada en la Tabla 9.1.

Las Figura 9.2 y 9.3 muestran la respuesta en frecuencia del banco de filtros implementado. La Figura 9.2 muestra los 4 canales, presentando la amplitud del canal para cada canal.

Tabla 9.1 **Coeficientes de los filtros diseñados mediante la Transformada LOT**

0.149864045922457	-0.326640741219094	0.356666636616021	-0.04407171892975
-0.503417436515731	0.326640741219094	-0.542396006915731	-0.256768107237054
0.312075720333186	-0.13529902503655	0.279066655336785	0.194470777435732
0.312075720333186	-0.135299025036549	0.279066655336785	0.194470777435732
0.312075720333186	-0.135299025036549	-0.279066655336785	-0.194470777435732
0.312075720333186	-0.13529902503655	-0.279066655336785	-0.194470777435732
-0.503417436515731	0.326640741219094	0.542396006915731	0.256768107237054
0.149864045922457	-0.326640741219094	-0.356666636616021	0.04407171892975

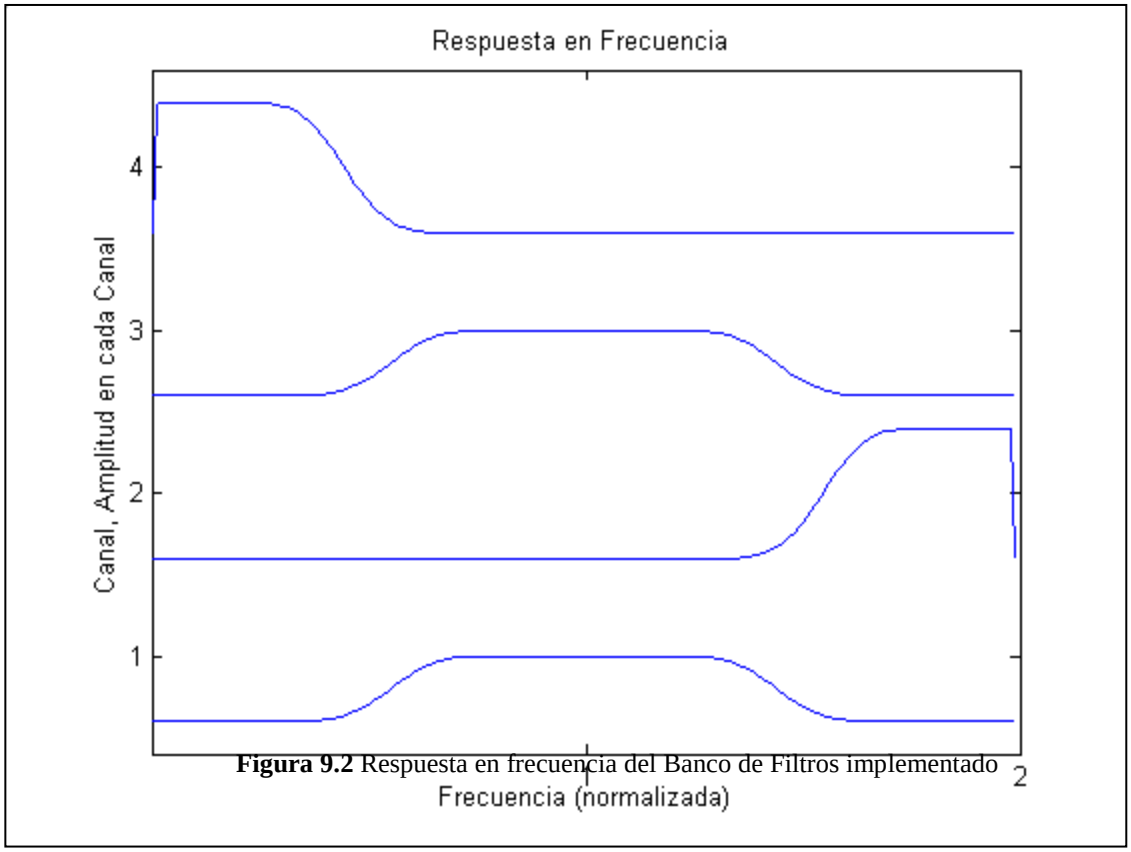

Figura 9.2 Respuesta en frecuencia del Banco de Filtros implementado

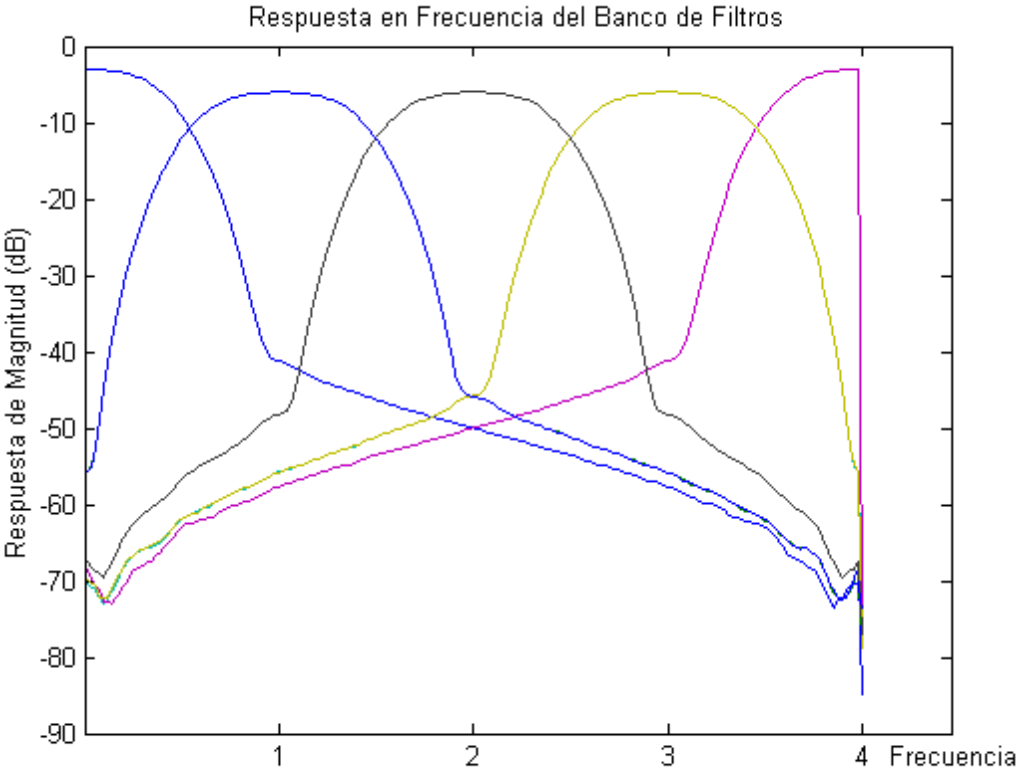

Figura 9.3. Respuesta al impulso del Banco de filtros implementado**.**

9.4.Eficiencia Computacional

Por las características propias del banco de filtros, los procesos se realizan a una tasa de muestreo menor que la de la señal original, pues se realizan sobre las sub-bandas de frecuencia. Así, tanto la detección del pico R, la clasificación de latidos y la eliminación de ruido de 50 ciclos se realiza a una tasa 1/*M* el valor de la tasa original de la señal.

9.5. Submuestreo

La Figura 9.4 muestra las salidas de los filtros de análisis del banco diseñado. Debido a que el contenido de frecuencias de ECG clínico de entrada está limitado a 100 Hz, cada uno de los filtros tiene un ancho de banda de 25Hz, en cuatro rangos iguales distribuidos entre 0 y 100Hz. Las abscisas en la figura referenciada representan el número de muestras procesadas. Adviértase el sub-muestreo en los cuatro canales del banco. La operación de submuestreo nos asegura que los procesos realizados sobre las sub-bandas tendrán una tasa de $\frac{1}{M}$ la velocidad de la señal original, siendo *M* el número de sub-bandas o canales.

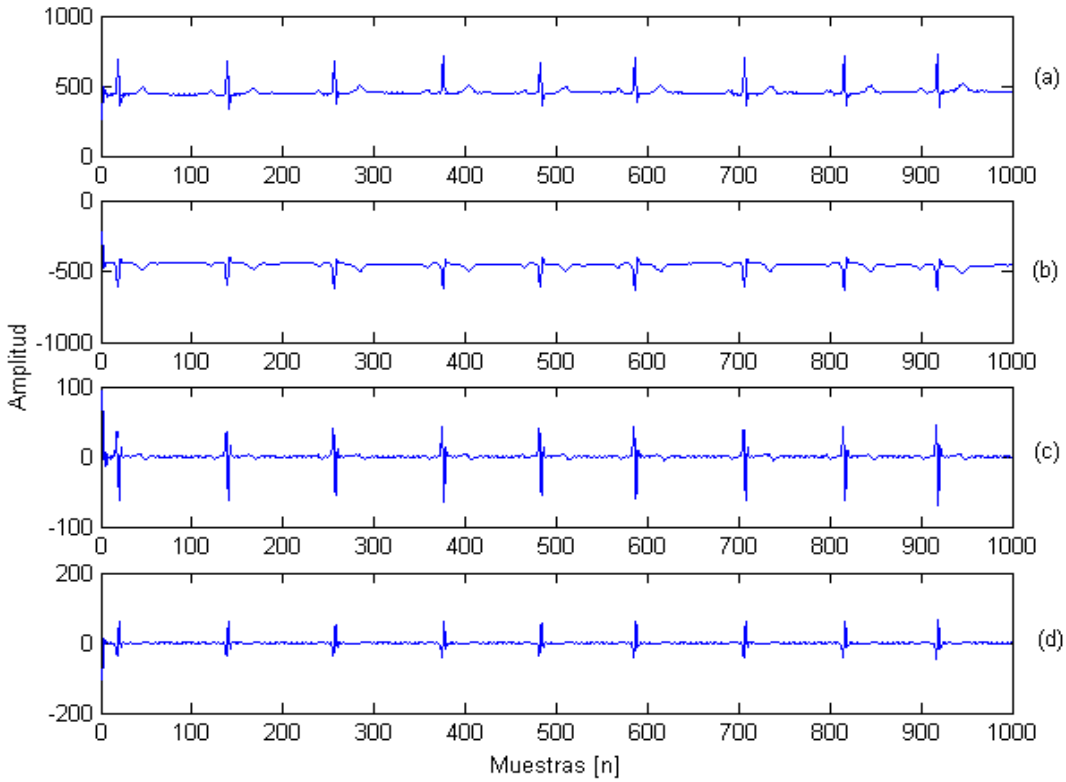

Figura 9.4 Sub-bandas de frecuencia de la señal de ECG. (a) Frecuencia 0-25Hz. (b) Frecuencia de 25-50Hz. (c) Frecuencia de 50-75Hz. (d) Frecuencia de 75-100Hz

9.6. Procesos

La señal del ECG es la señal biológica más estudiada con algoritmos digitales a través de los tiempos. Sus características morfológicas y su condición periódica, hacen que esta señal resulte atractiva para los diseñadores de algoritmos de procesamiento de señales. Más allá del atractivo matemático que encierra la señal electrocardiográfica, su análisis clínico es de interés para la comunidad médica, ya que mediante su inspección visual, el especialista puede determinar ciertas patologías que tienen que ver con sus características de forma y de ritmo. Sin embargo, no siempre la inspección visual resulta en un diagnóstico acabado, sumado al hecho de que la señal, de baja frecuencia, puede traer consigo señales espurias que contaminan la señal de interés y que puede producir un diagnóstico poco certero. Más aún, si pensamos en que la señal es capturada en un ambiente hospitalario, donde encontramos otros instrumentos médicos y posiblemente algunos conectados al paciente, estas interferencias seguramente estarán presentes en la toma del ECG. De allí que el procesamiento de señales para la eliminación de ruido interferente resulta de vital importancia para un diagnóstico acertado.

Otra tarea importante en el procesamiento de la señal de ECG, es la detección del complejo QRS. Esta etapa es crucial en los sistemas de monitoreo básico del ECG y resulta de interés para otras aplicaciones sobre la señal.

La clasificación de arritmias es también una tarea importante en un sistema interpretativo que provea una clasificación de diagnóstico del ECG.

Las tres tareas descriptas se realizaron utilizando el mismo Banco de Filtros, y esta es una ventaja significativa del método, pues podemos realizar varias tareas en forma paralela utilizando los mismos filtros. Esto no sucedería así, si filtráramos la señal de forma convencional, es decir, con un filtro diseñado específicamente para que atenúe ciertas frecuencias, y realizar algún proceso sobre la señal, por ejemplo la detección de un punto característico. Para realizar cualquier otra tarea sobre la señal, como detectar otro punto de interés o bien eliminar frecuencias indeseadas, deberíamos diseñar otro filtro, adecuado para ese fin particular. Es decir, deberíamos diseñar un filtro para cada una de las tareas que

queremos realizar sobre la señal de interés. Sin embargo, con el diseño de un Banco de Filtros, podemos realizar varias tareas con la implementación de un único banco de filtros.

Los procesos realizados en el Banco de Filtros están esquematizados en la Figura 9.5 Ellos son:

- la eliminación de ruido de 50Hz
- la detección del pico R
- la clasificación de latidos en normales y PVC

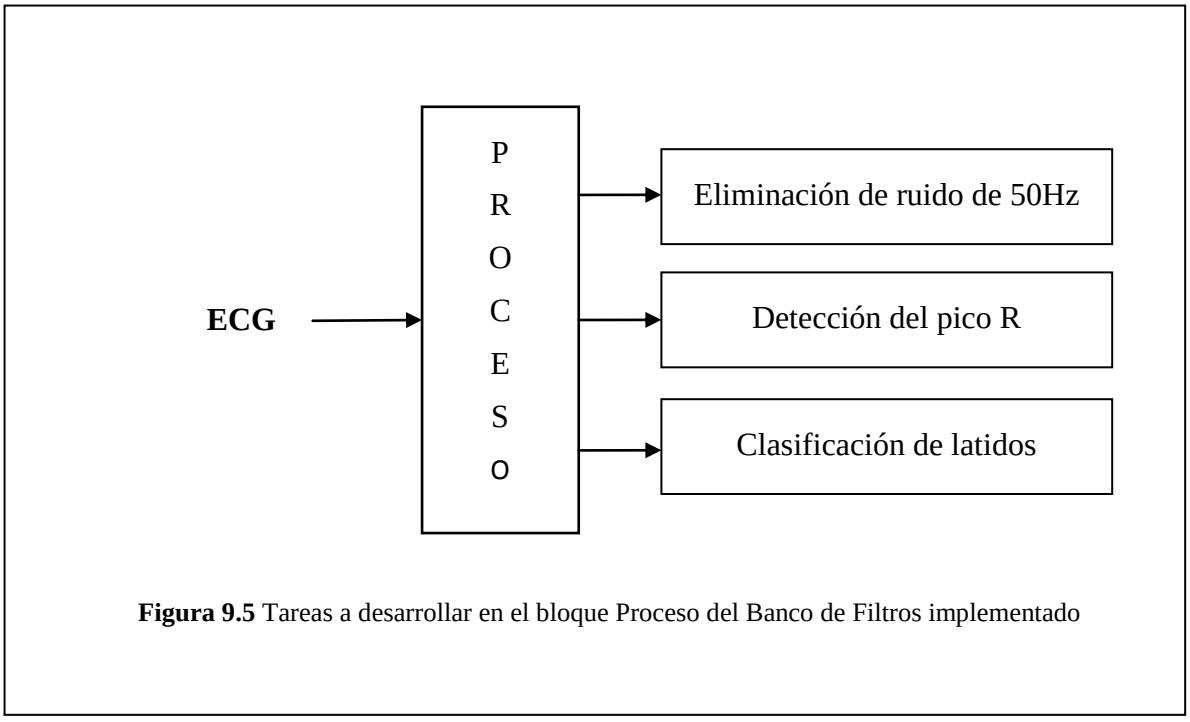

Figura 9.5 Tareas a desarrollar en el bloque Proceso del Banco de Filtros implementado

9.6.1. Eliminación de Ruido de 50 Hz

Una de las técnicas utilizadas para la eliminación de ruido superpuesto a la señal es el método de promediación (signal averaging), que permite la separación de la señal y del ruido sin introducir distorsión.

La Figura 9.6 muestra en detalle el ECG original con un 30% de ruido agregado. En la Figura 9.6(b) se ilustran los resultados obtenidos con un filtro de promediación de la señal basado en el cálculo de la media aritmética, con 9 iteraciones, y que mejora la relación señal–ruido (SNR –*signal to noise ratio*) en un factor de tres.

La implementación de este método de eliminación de ruido se basa en encontrar un punto característico, en este caso el pico R y tomar un número definido de muestras alrededor de este punto. Nuestro algoritmo toma 50 muestras a izquierda y 50 muestras a derecha. Luego se centran 9 segmentos consecutivos tomando como referencia el pico R detectado según el algoritmo descripto en la sub-sección anterior. Debido a que el ruido gaussiano es aleatorio, al realizar el promedio estadístico de los segmentos centrados en el pico R, el ruido se anula.

La elección del nivel de ruido (30% la amplitud de la señal original), no fue al azar. Corroboramos varios niveles de ruido, por encima y por debajo de ese umbral, comprobando que para una señal cuyo enmascaramiento con ruido blanco superaba el 30%, la efectividad del algoritmo de eliminación de ruido, se veía disminuida y en casos extremos, por encima del 70%, los resultados no devolvían una señal realmente limpia. Cabe destacar que para niveles de ruido por debajo del 30%, el algoritmo funciona realmente muy bien. A modo de ejemplo, la Figura 9.6, muestra la señal #100. Figura 9.6(a), señal ruidosa con 30% de ruido agregado; Figura 9.6(b), señal limpia, luego de la aplicación del proceso de eliminación de ruido.

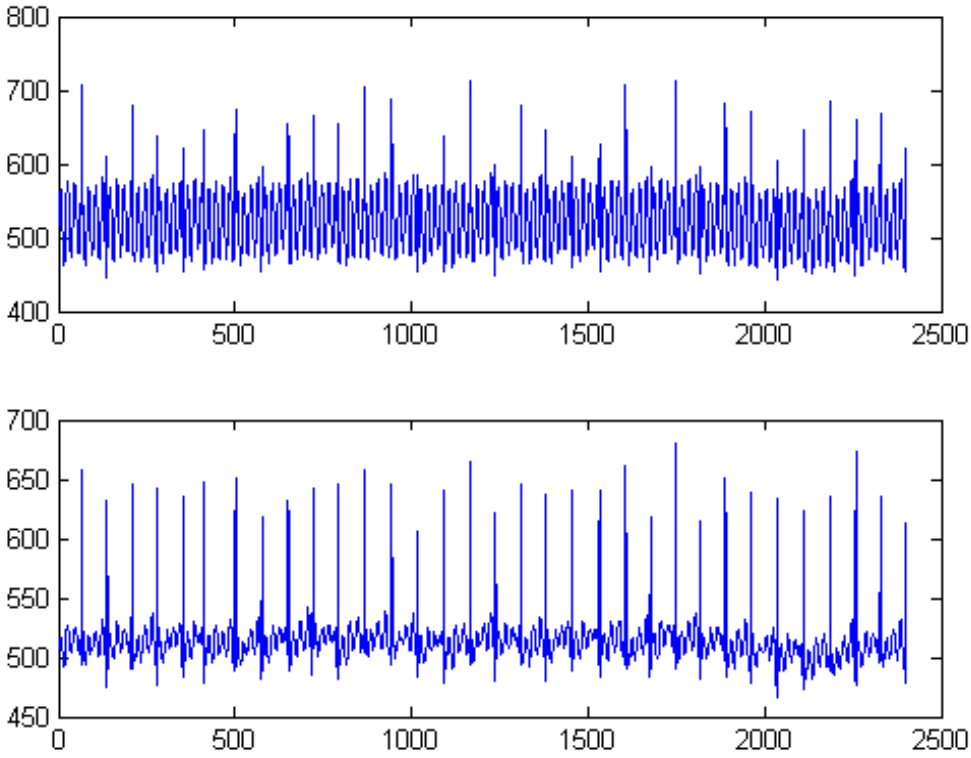

Figura 9.6. Ejemplo de eliminación de ruido blanco sobre una de las sub-bandas

Realizamos el proceso de eliminación de ruido gaussiano sobre una de las sub-bandas de la señal. La sub-banda elegida fue la que se encuentra en el rango de 0-25Hz, es decir la sub-banda Nº1. La elección tiene que ver con que esta sub-banda es la que conserva la forma de la señal de ECG y visualmente queríamos mostrar el nivel de ruido con respecto al pico R.

9.6.2. Detección de la onda R.

Una parte crucial en cualquier algoritmo de procesamiento del ECG es la detección de latidos. Los algoritmos de detección de latidos por lo general involucran un filtro de preprocesamiento de la señal que descompone la señal en una señal que maximiza el cociente señal-ruido del complejo QRS. Una etapa de procesamiento por lo general no lineal y luego la aplicación de un integrador de ventana deslizante se utilizan para procesar una señal que enfatice la energía del complejo QRS, (Pan y Tompkins, 1985). El uso de umbrales también es una técnica muy utilizada y que produce buenos resultados, (Hamilton y Tompkins, 1986; Li et al., 1995).

En nuestro Banco de Filtros, implementamos el algoritmo conocido y ampliamente probado de Pan y Tompkins, (Tompkins, 1993) para detectar los picos R. La novedad en nuestro caso fue que adaptamos el proceso a una sub-banda de frecuencia, la comprendida entre los 75 y 100 Hz, pues es, a nuestro criterio, donde se maximiza la diferencia con otros picos presentes en la señal de ECG. Analizamos todas las señales de la base de datos de arritmias del MIT-BIH. La **¡Error! No se encuentra el origen de la referencia.** muestra las etapas del algoritmo aplicado sobre una de las sub-bandas de frecuencia.

Aplicamos el algoritmo sobre la totalidad de la base de datos del MIT-BIH. Los 48 registros fueron evaluados. Los resultados son muy alentadores. Sobre un total de 87951 latidos evaluados, es decir el total de latidos de las señales que procesamos, algunas de ellas con una duración de 30 minutos y otras solamente extractos de 10 minutos aproximadamente, detectamos el total de los picos R, ya sean normales o patológicos. Es decir, no se produjeron falsos negativos, sin embargo, debemos destacar, que obtuvimos un total 403 falsos positivos distribuidos a lo largo de toda la base de datos. Estos resultados generales fueron tabulados en la siguiente tabla:

Tabla 9.2. Detección de latidos

Latidos	TP	FP	FN	Se[%]	PP[%]
87951	87951	403	0	100%	99,54%

Los datos obtenidos resultan alentadores y demuestran que en el bloque de proceso del banco de filtros, se puede implementar un algoritmo conocido y ya probado, pero con la ventaja de procesar los datos a una velocidad menor, debido a que trabajamos solo sobre una sub-banda de frecuencia. De la bibliografía consultada (Afonso, 1996), donde se describe un algoritmo con la implementación de un banco de 32 filtros con 64 coeficientes cada uno, y la detección del complejo QRS que involucra la utilización de varias sub-bandas de frecuencia, podemos afirmar, que si bien sus datos publicados alcanzan buenos índices de performance, PP: 99,56% y S: 99,59%, encontramos nuestro algoritmo más sencillo de implementar y más eficiente, si medimos los tiempos computacionales, pues no realizamos un proceso recursivo, que sí lo aplica Afonso. Planteado de esta manera, las ventajas de un método sobre otro caen en una decisión más bien subjetiva. El diseñador deberá definir entonces cuál criterio resulta de mayor interés particular para aplicar uno u otro método.

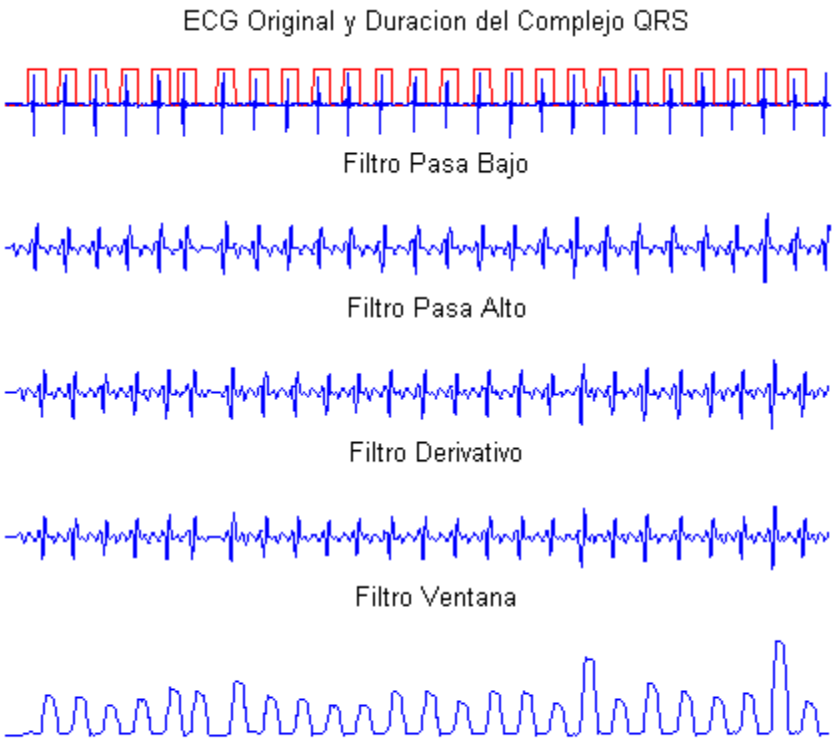

Figura 9.7. Etapas del algoritmo que detecta los complejos QRS. Señal #100

9.6.3. Detección de Arritmias

El algoritmo diseñado para la detección de arritmias se basa en características temporales. La duración entre dos intervalos R-R de latidos normales es significativamente diferente a la duración del intervalo R-R de un latido normal y una PVC, ya sea la PVC previa o posterior al latido normal. Este fue el criterio tenido en cuenta para la elaboración del algoritmo, que además cuenta con la ventaja de procesar los datos a una tasa menor que la original, pues trabajamos sobre una única sub-banda de frecuencia.

En la Tabla 9.3(a) Detección de PVC **con Bancos de Filtros y sin Banco de Filtros**resumimos los resultados obtenidos con las señales patológicas seleccionadas, correspondientes a la derivación II modificada (MLII). En el lado izquierdo de la tabla se

indica el desempeño del algoritmo de detección de PVC; sobre la derecha, se listan los resultados con el algoritmo que incorpora el banco de filtros. PP representa el índice de Predictividad Positiva, que es el porcentaje de PVC detectadas que realmente son PVC, mientras que S es el porcentaje de PVC clasificas como PVC. El total de PVC anotadas en la base de datos es la suma de los verdaderos positivos TP (True Positive), es decir aquellas PVC que el algoritmo reconoce como tal y los falsos negativos FN (False Negative), aquellas PVC que se "escapan" de la detección. Cabe destacar que algunas de las señales pertenecientes a la base de datos del MIT-BIH no fueron evaluadas por nuestro algoritmo y es porque o bien no cuentan con PVC o bien porque poseen un número no significativo de PVCs (menor a 20) y consideramos que no son señales representativas de esta patología. Las tablas fueron separadas según las señales corresponden a la serie 100 o 200 para una mejor visualización.

Tabla 9.3(a) Detección de PVC con Bancos de Filtros y sin Banco de Filtros. Serie 100

Señal	Sin FB					Con FB				
	TP	FP	FN	PP[%]	S[%]	TP	FP	FN	PP[%]	S[%]
105	38	15	3	71,70	92,68	40	8	1	83,33	97,56
106	495	87	25	85,05	95,19	493	53	27	90,29	94,81
107	49	19	10	72,05	83,05	52	12	7	81,25	88,14
111	76	14	5	84,44	93,83	81	10	0	89,01	100,00
114	34	19	0	64,15	100,00	34	16	0	68,00	100,00
116	31	14	9	68,88	77,5	36	10	4	78,26	90,00
119	441	71	3	86,13	99,32	438	0	6	100,00	98,65
124	16	10	4	61,54	80,00	18	7	2	72,00	90,00

Tabla 9.4(b) Detección de PVC con Bancos de Filtros y sin Banco de Filtros. Serie 200

Señal	Sin FB					Con FB				
	TP	FP	FN	PP[%]	S[%]	TP	FP	FN	PP[%]	S[%]
200	809	72	19	91,83	97,71	807	12	21	98,53	97,46
201	196	470	4	29,43	98,00	200	433	0	31,60	100,00
203	412	41	32	90,95	92,79	432	27	12	94,12	97,30
205	69	20	2	77,52	97,18	65	19	6	77,38	92,00
207	94	20	11	82,46	89,52	89	22	31	80,18	74,16
208	941	207	424	81,97	68,94	1120	291	245	79,38	82,05
210	176	31	18	85,02	90,72	180	21	14	89,55	92,78
213	192	29	28	86,88	87,27	204	31	16	86,81	92,73
214	231	33	25	87,50	90,23	245	29	11	89,42	95,70
215	148	28	16	84,10	90,24	150	22	14	87,21	91,46
217	138	35	23	79,77	85,71	154	31	8	88,00	95,06
219	58	25	6	69,88	90,62	58	19	6	75,32	92,06
221	112	27	48	80,57	73,20	120	18	40	86,96	75,00
223	45	19	13	70,31	77,58	49	12	9	80,33	84,48
228	351	59	11	85,61	96,96	350	13	12	96,42	96,69
233	214	52	44	80,45	82,95	221	45	37	83,08	85,66

De las Tabla 9.3(a) y (b) se desprende que la incorporación de un banco de filtros al algoritmo, mejora la predictibilidad positiva, y la sensibilidad en la detección de PVC. En algunos casos, como las señales 106 y 119 la sensibilidad tiene valores de índices más altos en el caso en que no se aplica un banco de filtros. Esto se debe a que con FB se detecta un menor número de falsos positivos, pero con mayor incidencia de falsos negativos. Notamos también que esta tendencia no se mantiene para las señales 201 y 208, que exhiben índices de performance bajos. Una explicación posible radica en el hecho de que ambas son señales que incluyen otras arritmias, además de aquellas identificadas y anotadas como PVC. La señal 201 registra episodios de bloqueos y fibrilación auricular. En el caso particular de la señal 208, de las 1365 PVC, sólo 992 corresponden a PVC típicas mientras que 373 son de las denominadas *PVC por fusión.*

En general, los índices de predictividad positiva y sensibilidad se ven mejorados con la incorporación de un banco de filtros. Cabe aclarar que para la detección del pico R y de arritmias del ECG, hemos implementado los mismos algoritmos de detección y clasificación, para establecer una comparación más objetiva.

Podemos mejorar la eficacia del algoritmo aplicado a las señales atípicas 201 y 208 cambiando el criterio de clasificación. Como ejemplo ilustrativo, mostramos en la Tabla 9.5 los resultados obtenidos utilizando la derivación precordial 1 (V1) y considerando como PVC a aquel latido con una amplitud de onda R que supera en un 50% a la amplitud de la onda R normal. Con este criterio de clasificación basado en umbrales (parámetro morfológico y no temporal), el algoritmo con FB aumenta la PP y mantiene los porcentuales de sensibilidad. Pero en el caso de la señal #119, el parámetro morfológico de clasificación y detección no mejora la eficacia de los algoritmos.

Tabla 9.5 Detección de arritmias

Señal	Sin FB					Con FB				
	TP	FP	FN	PP[%]	S[%]	TP	FP	FN	PP[%]	S[%]
119	432	85	12	83,53	97,30	432	44	12	90,76	97,30
201	200	0	0	100	100	200	0	0	100	100
208	1354	98	11	93,25	99,19	1355	19	10	98,62	99,27

9.7.El algoritmo clasificador en entornos ruidosos

En la Tabla 9.6 hemos tomado sólo la señal #119, como ejemplo representativo de un registro de arritmia ventricular, ya que tiene anotados 444 latidos ventriculares prematuros y 1543 latidos ventriculares normales.

Sobre la base de la señal sin interferencias, generamos señales con ruido de base (bw), de movimiento de electrodos (em) y de contracciones musculares (ma), con relaciones señal-ruido que van desde los 24dB hasta los -6dB, siendo ésta la más ruidosa. Obtuvimos estas señales con estos niveles de ruido mediante la utilización de la función *nstgen* provista por PhysioToolkit, el paquete de programas que conforma Physionet (www.physionet.org), de distribución libre y gratuita.

De manera similar a lo expuesto para las Tabla 9.3, indicamos en la Tabla 9.6 los índices de eficacia del algoritmo detector de PVC implementado con y sin Banco de Filtros, y que opera sobre estas señales ruidosas.

Tabla 9.6 Detección de arritmias en entornos ruidosos.

Señal	Sin FB					Con FB				
	TP	FP	FN	PP[%]	S[%]	TP	FP	FN	PP[%]	S[%]
119bw_6	432	67	13	86,55	97,07	395	26	49	93,82	88,96
119bw00	432	67	12	86,57	97,30	404	31	40	92,87	90,99
119bw06	441	72	3	86,96	99,32	412	23	32	94,71	92,79
119bw12	442	72	2	85,99	99,55	419	12	25	97,22	94,37
119bw18	443	63	1	87,55	99,77	423	1	21	99,76	95,27
119bw24	444	86	0	83,77	100,00	428	2	16	99,53	96,40
119em_6	390	60	54	86,67	87,84	401	0	43	100,00	90,32
119em00	396	79	48	83,37	89,19	407	0	37	100,00	91,67
119em06	397	69	47	85,19	89,41	420	29	24	93,54	94,59
119em12	431	60	13	87,78	97,07	426	5	18	98,84	95,95
119em18	434	53	10	89,12	97,75	434	27	10	94,14	97,75
119em24	436	41	8	91,40	98,20	422	19	22	95,69	95,05
119ma_6	426	75	18	85,03	95,95	411	4	33	99,04	92,57
119ma00	430	72	14	85,66	96,85	415	43	29	90,61	93,47
119ma06	438	71	6	86,05	98,65	421	2	23	99,53	94,82
119ma12	439	71	5	86,08	98,87	423	4	21	99,06	95,27
119ma18	438	56	6	88,66	98,65	426	9	18	97,06	95,95
119ma24	438	45	6	90,68	98,65	434	1	10	99,77	97,75

La detección de PVC en entornos ruidosos depende del tipo y nivel de ruido, Notamos que para señales ruidosas por movimiento de electrodos (em), la sensibilidad se mantiene en la mayoría de los casos y solo baja, siguiendo la tendencia general de las otras señales, en el caso de señales con menor ruido superpuesto (SNR=12, 18 y 24), La característica discutida en la detección de arritmias plasmada en la tabla anterior, se mantiene en el caso de estas señales ruidosas, esto es que la incorporación de un banco de filtros al algoritmo mejora la predictibilidad positiva, aunque disminuye la sensibilidad de la detección de PVC, con la diferencia que a medida que el nivel de ruido disminuye, la sensibilidad del algoritmo con banco de filtros, alcanza altos valores de performance, que se diferencian de los del algoritmo sin banco de filtros en menos de 7 puntos,

Si tenemos en cuenta los niveles de ruido, podemos decir que a medida que el nivel de ruido disminuye, los índices de performance mejoran, excepto en el caso del ruido de movimiento de electrodos (em) que presenta una predictividad positiva del 100% para los niveles de ruido más altos, (-6dB y 0dB) y luego para los demás niveles de ruido, éste índice se mantiene con valores altos pero por debajo del 100%,

9.8.Reconstrucción

Los bancos de filtros digitales diseñados tienen la propiedad de la Reconstrucción Perfecta, sin embargo, dicha reconstrucción no es ideal en el sentido que si bien se recupera el número de muestras original, previo al proceso de sub-muestreo, el banco de filtros identidad, incorpora una distorsión en la amplitud de la señal reconstruida, Figura 9.10*,* **¡Error! No se encuentra el origen de la referencia.**Para comprobar de forma objetiva la diferencia entre la señal original que ingresa al Banco de Filtros y la señal reconstruida, la salida del Banco de Filtros, y que no se resuma simplemente a la inspección visual, recurrimos al índice de correlación de Pearson, descripto en el capítulo 8 entre señales que evalúa la igualdad o diferencia entre dos series de datos, Para facilitar la lectura, convertimos los datos a porcentajes, así un valor de 100% para el coeficiente de

correlación, significará una similitud exacta entre la señal original y la recuperada por el Banco de Filtros diseñado, luego la propiedad de Reconstrucción Perfecta se satisface,

Analizamos la totalidad del banco de datos con el que contamos, Los resultados están listados en la Tabla 9,7,

Como puede verse en la tabla, la similitud entre la señal original y la reconstruida por el Banco de Filtros varía desde un 55,060% para la señal #220 y un 96,691% para la señal #207, Ambas graficadas en las Figura 9.8 y Figura 9.9, Además, cabe destacar que 35 de las 48 señales presentan índices por encima del 80%,

Si bien, del índice numérico deducimos que existe una gran distorsión para señales con índices por debajo del 60%, de la gráfica podemos concluir que la diferencia entre una señal y otra se debe principalmente a una distorsión de amplitud y no de forma,

Podemos decir entonces que, debido a que la propiedad de Reconstrucción Perfecta es ideal, nuestro Banco de Filtros cumple con el objetivo de obtener, luego de la implementación del Banco, una señal reconstruida que difiere sólo en amplitud de la señal origina, Luego, nuestro Banco de Filtros cumple con la propiedad de Reconstrucción Perfecta,

Tabla 9,7 Coeficiente de correlación entre la señal original y la reconstruida por el Banco de Filtros

Serie 100		Serie 200	
Señal	Coeficiente de Correlación [%]	Señal	Coeficiente de Correlación [%]
100	58,964	200	90,308
101	88,162	201	82,598
102	87,751	202	89,550
103	67,896	203	92,760
104	87,691	205	64,374
105	88,212	207	96,691
106	82,057	208	90,664
107	94,866	209	62,407
108	92,363	210	88,496
109	91,753	212	77,366
111	91,910	213	84,443
112	87,612	214	92,255
113	85,009	215	74,757
114	86,801	217	94,839
115	69,702	219	86,754
116	78,701	220	55,060
117	87,862	221	85,602
118	84,676	222	80,781
119	90,333	223	86,471
121	96,165	228	95,876
122	86,270	230	74,214
123	73,353	231	73,309
124	91,623	232	82,287
		233	90,381
		234	67,207

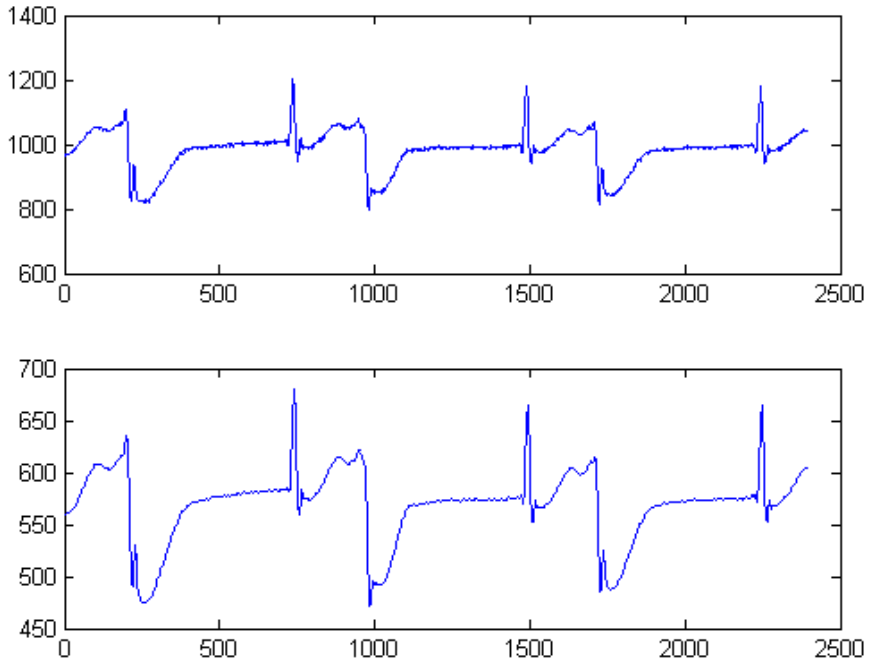

Figura 9.8 Reconstrucción de la señal 207.

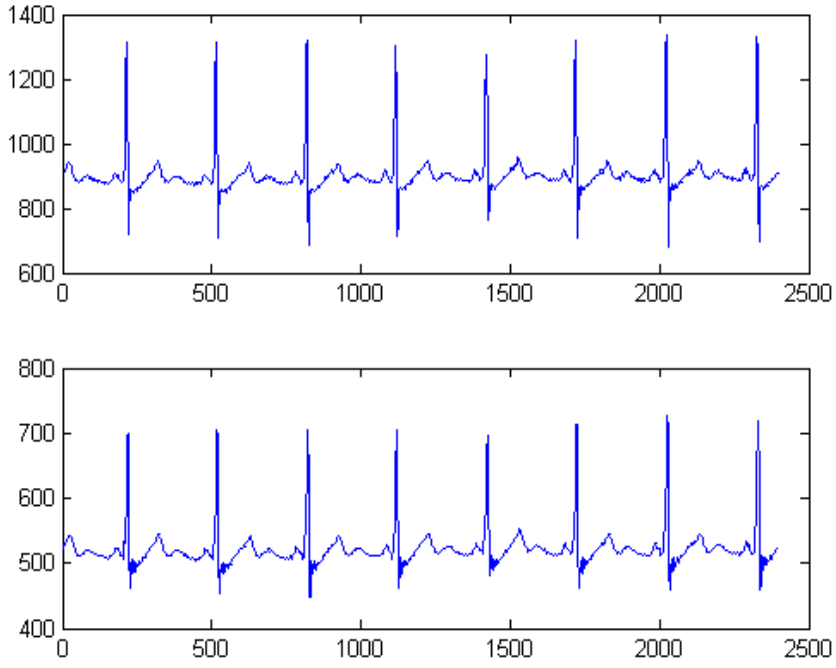

Figura 9.9 Reconstrucción de la señal 220

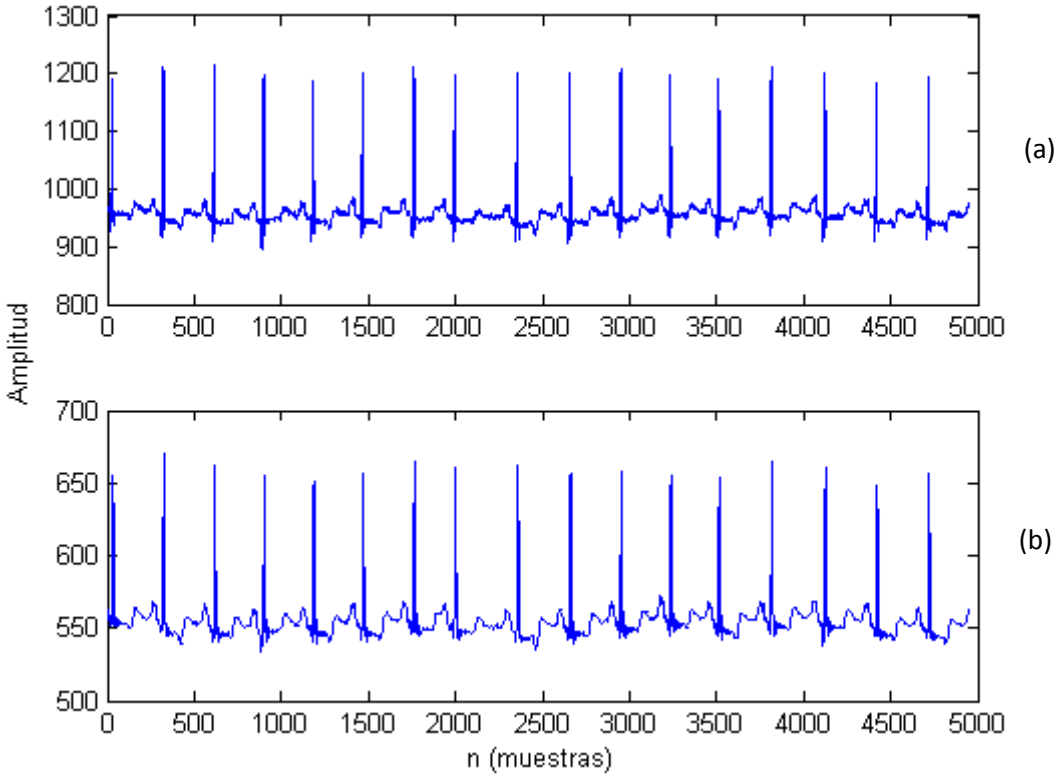

Figura 9.10. Reconstrucción de la señal 100. (a) ECG original; (b) ECG reconstruido

10. Análisis de Componentes Principales

10.1. Motivación

Las características morfológicas y temporales de la señal cardíaca resultan de interés particular para su análisis. Ya sea que se detecten ondas relevantes en la señal o se caractericen segmentos de interés para colaborar con el diagnóstico médico, la acumulación de datos que complete la información necesaria, es casi inevitable. Como solución a esto, recurrimos al Análisis de Componentes Principales, que logra reducir las dimensiones de una matriz de datos original, y obtener así una transformación a partir de ésta que la represente de la mejor manera posible con la ventaja de eliminar aquellos componentes que contribuyen menos a la variación del conjunto de datos.

El objetivo particular de este capítulo es extraer los puntos característicos del complejo QRS para su análisis morfológico y obtener así una matriz de datos que será luego transformada mediante el Análisis de Componentes Principales. El resultado de esta transformación producirá:

- una matriz de dimensiones reducidas en comparación con la original;
- vectores de entrada ortogonalizados que aseguran la no-correlación entre ellos y
- ordenación de los componentes principales, de manera que en los primeros lugares estés aquellos que más aportan a la variabilidad a los datos.

La matriz de datos así transformada será luego incorporada a una red neuronal capaz de clasificar latidos cardíacos.

10.2. Procesos

Los procesos incluyen:

- Construir una matriz de datos que contenga los valores de amplitud de cada complejo QRS de la señal cardíaca.
- Obtener los componentes principales de la matriz construida
- Analizar los resultados obtenidos mediante un diagrama de Pareto.

10.3. Construcción de la matriz de datos originales

La base de datos de Arritmias del MIT-BIH contiene 48 registros electrocardiográficos que corresponden a sujetos sanos y también sujetos con afecciones cardiacas. Cada señal en la base de datos es un complejo de tres señales, un archivo con extensión *.dat* que contiene los datos propiamente dichos, otro con extensión *.atr* que es un archivo de atributos, donde sólo aparecen como datos los complejos QRS detectados con el algoritmo de Jiapu Pan (Pan y Tompkins, 1985) y un tercer archivo con la extensión *.hea* (header), que permite visualizar la señal de ECG con el programa *GTKwave* provisto por *PhysioToolkit*.

El algoritmo que diseñamos utiliza tanto la señal del electrocardiograma como el archivo de anotaciones extraído también de la base de datos de Arritmias del MIT-BIH correspondiente a esa señal. Este archivo de atributos contiene, en la primera columna, el tiempo en que sucede la onda Q en cada latido ventricular (lo que marca el comienzo del complejo QRS); en otra columna se indica el orden de la muestra que corresponde a ese dato; también se asienta el código de clasificación del latido (“N” si se trata de un latido normal). A continuación se ilustran las primeras dos filas de un archivo.

0:00.194 70 N

0:01.005 362 N

. . . . . .

Con los datos extraídos del archivo de datos propiamente dicho de la señal (.dat) generamos una matriz con los valores de muestras que corresponden al complejo QRS, un registro por cada latido. La novedad que presenta el algoritmo es que utilizamos archivos provistos por la base de datos de *Physionet* que ya fueron fuertemente testeados. Por cada registro leído en el archivo de anotaciones, mediante un apareo con el archivo de datos de la señal, obtenemos los valores de amplitud de cada punto del complejo QRS. Estos valores de amplitud son cada una de las variables de nuestra matriz de datos. Con esto nos aseguramos que todos los latidos de la señal son detectados y por lo tanto, todos ellos serán caracterizados en una matriz de datos.

La matriz que contiene los datos que describen cada complejo QRS tiene un total de 50 columnas que son las variables. En la columna 30 se ubica el pico R de cada latido. La elección de las dimensiones de la matriz no fue al azar. La bibliografía consultada (Moody y Mark, 1989; Moody GB, 2000) propone una matriz de 30 columnas para caracterizar el complejo QRS. Nuestra opción nos parece más acertada, debido a que si bien incluimos puntos más allá de la onda S, con esta cantidad de datos nos aseguramos que todos los complejos QRS de toda la base de datos están perfectamente completos y caracterizados, pues sucede que algunos registros poseen rasgos patológicos que se reflejan en complejos QRS más “deformados” o de mayor longitud en el tiempo.

La Figura 10.1 muestra las primeras 6 filas de la matriz obtenida con el algoritmo. La señal mostrada corresponde al registro #100 de la base de datos del MIT-BIH.

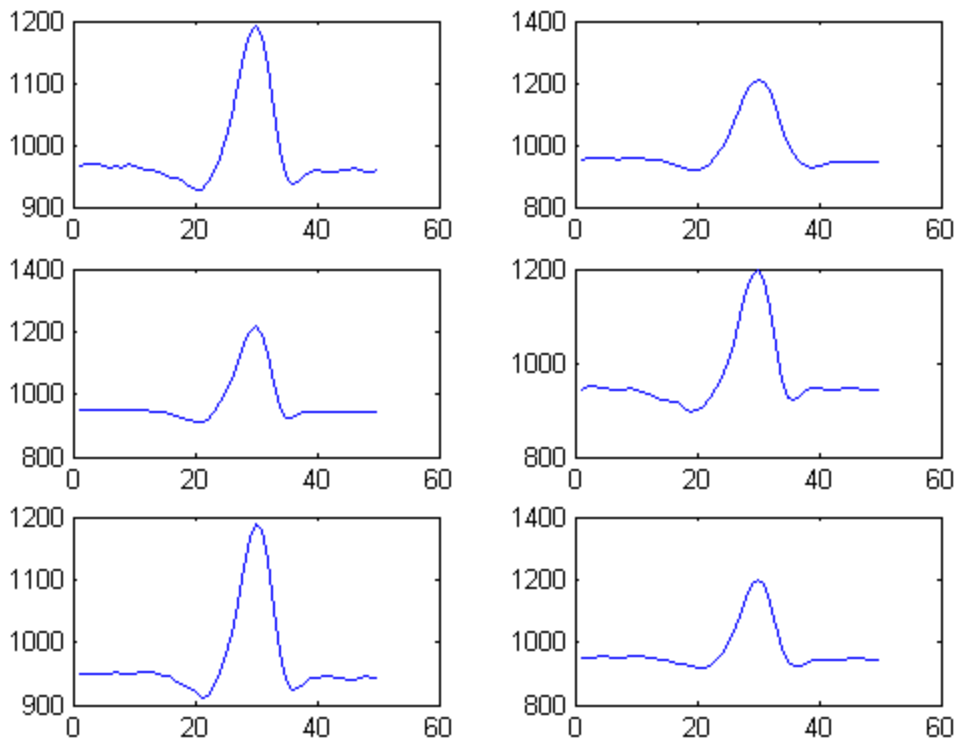

Figura 10.1 Complejos QRS de la señal #100. Matriz de datos original

10.4. Implementación de PCA a la matriz de datos original

El análisis de Componentes Principales fue llevado a cabo utilizando MATLAB®.

Previa a la aplicación de los componentes principales, estandarizamos los datos, dividiéndolos cada columna de la matriz por su por su desviación estándar.

Aplicamos la función *princomp* que devuelve las salidas que serán descriptas a continuación:

a) La primera salida de la función, es una matriz 50x50, pues nuestra matriz original contiene 50 variables que son cada uno de los puntos del complejo QRS. Esa matriz cuadrada, que llamamos **P,** contiene las combinaciones lineales de las

variables originales, es decir los componentes principales. Dichas combinaciones lineales generan las nuevas variables. Los elementos de los componentes principales tienen el mismo signo. Son los promedios ponderados de todas las variables. La ortogonalidad de **P** se demuestra fácilmente: $\mathbf{I} = \mathbf{P} \cdot \mathbf{P}^T$.

b) La segunda salida de la función *princomp* son los datos transformados, en el nuevo espacio, definidos por los componentes principales. La matriz que se genera tiene las mismas dimensiones que la matriz de datos originales.

c) La tercera salida son las varianzas. Es un vector que contiene las varianzas de sondeo de las columnas correspondientes de los nuevos datos.

10.5. Análisis de Pareto

Antes de exponer los resultados obtenidos, haremos una breve mención al Principio de Pareto, que nos ayudará a identificar cuáles de todos los componentes principales son los que más contribuyen con información no-redundante y que nos permitirá de esta manera decidir cuáles de los factores integrarán ahora los datos.

El Principio de Pareto afirma que en todo grupo de elementos o factores que contribuyen a un mismo efecto, unos pocos son responsables de la mayor parte de dicho efecto.

El Análisis de Pareto es una comparación cuantitativa y ordenada de elementos o factores según su contribución a un determinado efecto.

El objetivo de esta comparación es clasificar dichos elementos o factores en dos categorías: Las "Pocas Vitales" (los elementos muy importantes en su contribución) y los "Muchos Triviales" (los elementos poco importantes en ella). Es decir que el análisis de Pareto es una comparación ordenada de factores relativos a un problema. Esta comparación ayuda a identificar y enfocar los pocos factores vitales diferenciándolos de los muchos factores útiles.

La aplicación del mismo permite exhibir visualmente en orden de importancia, la contribución de cada elemento en el efecto total. El Diagrama de Pareto comunica de forma clara, evidente y de un "vistazo", el resultado del análisis de comparación y priorización.

Para demostrar gráficamente el aporte de cada uno de los componentes principales utilizaremos entonces el diagrama de Pareto, de fácil interpretación.

10.6. Resultados y Discusión

Luego de aplicar las herramientas que proporciona MATLAB® para la determinación de los Componentes Principales obtuvimos la matriz de correlaciones que tiene una dimensión de 50 x 50; otra salida obtenida es la matriz que contiene los nuevos datos transformados. Las dimensiones de esta matriz coinciden con las dimensiones de la matriz original. Un tercer dato obtenido es la varianza para cada uno de los latidos. Es decir, un vector columna, donde cada valor representa la varianza de todas las variables del latido.

Para evaluar el aporte que cada componente principal hace a la información total acerca del latido cardíaco utilizamos el análisis de Pareto, mencionado previamente.

La Figura 10.2 muestra tres diagramas de Pareto obtenidos en función de los Componentes Principales (vectores propios) seleccionados. Hemos seleccionado estos tres esquemas como variantes de los resultados obtenidos. En general la técnica arrojó un resultado similar para las señales #100, #101, #102, #103, #104, #105, #106, #107, #109, #111, #112, #113, #115, #116, #118, #119, #122, #123, #124, #200, #201, #202, #203, #205, #207, #208, #209, #210, #212, #213, #214, #215, #217, #219, #220, #221, #223, #228, #230, #232, #233, #234.

En la Figura 10.2(a) se observa el diagrama de la señal #119. Esta es una señal que contiene 444 PVC (latidos ventriculares prematuros), todos regulares. Según el esquema, en las primeras 10 columnas de la nueva matriz de componentes principales se acumula más del 95% de la información que caracteriza cada complejo QRS.

Lo mismo ocurre con la Figura 10.2(b), donde evaluamos la señal #208, aunque la distribución de los pesos es diferente debido a que no sólo existen PVC sino que cuando aparecen en pares o triples, uno es PVC de fusión. Además no se alcanza el 95% de la información, es decir que podríamos considerar una o dos columnas más.

Por último queremos mostrar en la Figura 10.2(c) el comportamiento de la técnica de PCA cuando la aplicamos a la señal #121. Una explicación de esta diferencia podría ser que esta señal dura sólo diez minutos mientras que las dos primeras, treinta minutos con lo cual hay menos probabilidad de diversidad en los datos y otra explicación posible sería que es una señal en la que los picos R no tienen una espiga pronunciada y esto provoca poca variabilidad en los datos. Por ello en las primeras tres columnas se almacena más del 95% de la información necesaria para describir los complejos QRS. Este comportamiento también lo apreciamos en las señales #114, #108, #117, #231, #222; todas extractos de diez minutos.

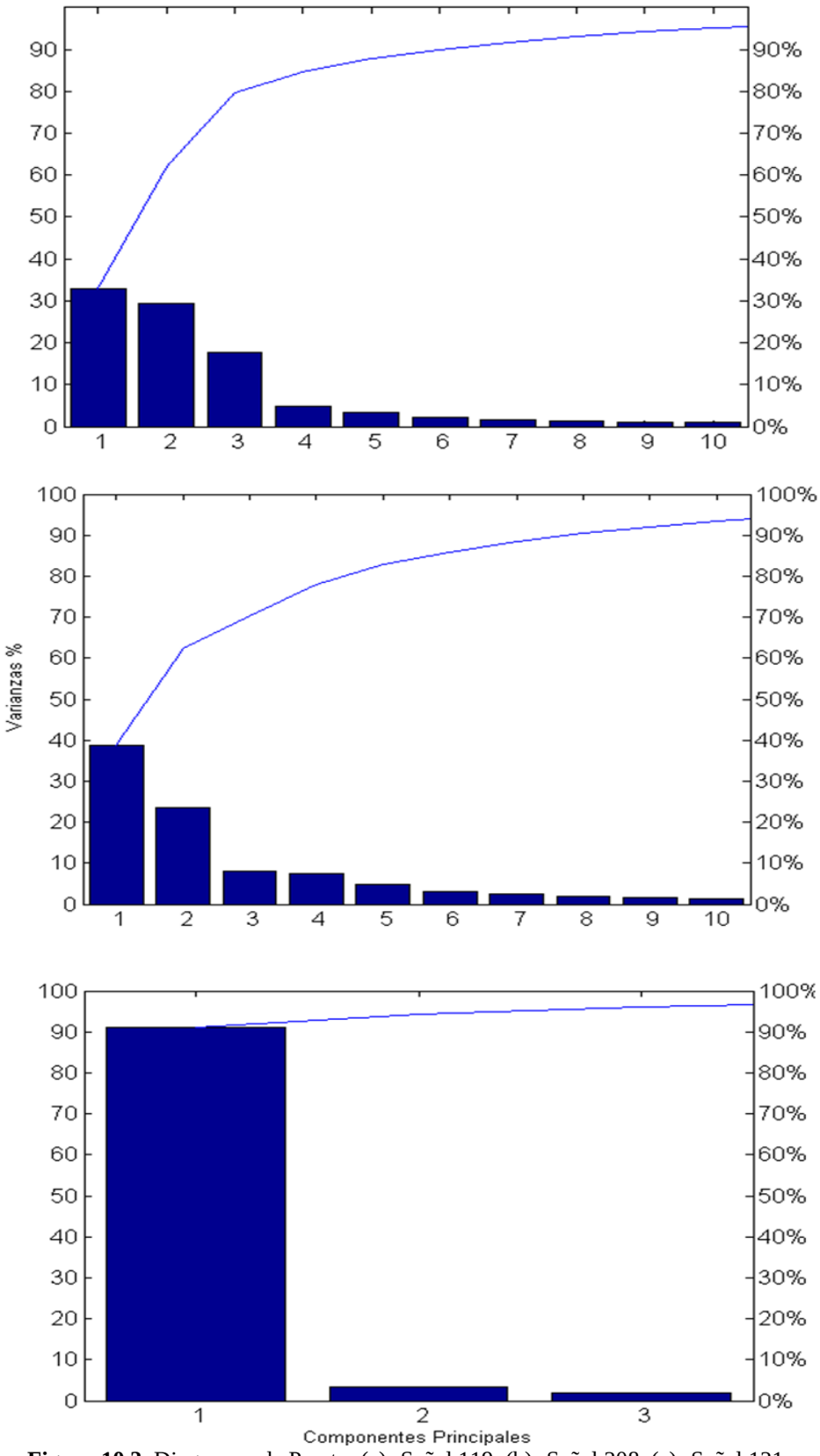

Figura 10.2. Diagramas de Pareto. (a)- Señal 119. (b)- Señal 208. (c)- Señal 121

11. Análisis de Componentes Independientes

11.1. Motivación

Dentro de los procesos posibles aplicables a una señal eléctrica cardíaca, resulta ineludible la eliminación -o al menos minimización- de interferencias producidas por equipos cercanos al paciente, la red eléctrica o las mismas señales producidas por el cuerpo que enmascaran la señal de interés. El análisis y la interpretación de los registros eléctricos de la actividad cardíaca, se han visto siempre limitados por la presencia de tales interferencias, que son de distinta naturaleza y magnitud, obligando a los diseñadores de algoritmos a la búsqueda constante de procedimientos computacionales y desarrollos tecnológicos que permitan obtener señales más limpias.

El análisis de componentes independientes (ICA), es una técnica que se aplica para separar, en una mezcla de señales estadísticamente independientes, las fuentes de señales que la componen. Este método resulta útil para eliminar señales interferentes que enmascaran la señal de interés.

11.2. Señales experimentales

Para la separación ciega de Fuentes con ICA utilizamos las señales electrocardiográficas de la base de datos MIT Arrhythmia Database descriptas en el capítulo 2. En particular usamos los siguientes registros, todos con una duración de 30 minutos:

- #100, representa 2239 latidos de ritmo sinusal normal, complejos QRS regulares y un única PVC; (Figura 11.1).

- #101, contiene 1860 latidos normales y 3 APC.

- #102, fue seleccionado por presentar 2028 latidos de marcapaso, 99 latidos

normales y 4 PVC.

- #119, contiene 1543 latidos ventriculares normales y 444 contracciones ventriculares prematuros. Las PVCs de esta señal abarcan latidos ventriculares prematuros, latidos de escape ventricular, latidos ventriculares ectópicos y latidos ventriculares R-en-T; (Figura 11.2).

- #200, contiene 1743 latido normales, 826 PVC y 30 APC

- #201, contiene un total de 1625 latidos normales, 198 PVC y 30 APC, además presenta PVC de fusión y latidos aberrantes.

- #202, contiene 2061 latidos normales, no posee una cantidad significativa de PVC

- #208, presenta 1369 latidos anómalos sobre un total 2955 latidos; (Figura 11.3).

Así como todos los registros de la base de datos en estudio, estos registros constan de dos señales, capturadas en simultáneo. Tal como se las puede ver en la aplicación para la visualización, Figura 11.1, Figura 11.2 y Figura 11.3, la señal ubicada en la parte superior, a la que llamaremos señal 1, es de la derivación II modificada (MLII), con electrodos ubicados en el pecho y no en los miembros. La señal ubicada debajo de ésta, señal 2, es una derivación precordial modificada V1. Usualmente, los latidos normales son más prominentes en la señal 1, sin embargo, las PVC son más prominentes en la señal 2. Tomamos ambas derivaciones en razón de que el número de fuentes debe ser igual al número de observaciones, para llegar a un resultado cierto.

Para generar las señales ruidosas que luego serán separadas mediante nuestro algoritmo, utilizamos otra base de datos perteneciente también al *Physiobank*, la base de datos de ruidos, *MIT-BIH Noise Stress Test Database*, descripta también en el capítulo 2. Ésta incluye tres registros de ruido frecuentemente encontrado en el ambiente hospitalario. Estos tres registros de ruido se refieren a artefactos provocados por movimiento de electrodos (em), artefactos por electromiografía (ma) y ruido de lineal de base (bw).

Con la herramienta *nstdbgen*, provista desde la página web por el MIT en forma gratuita y utilizando el registro limpio 119, creamos los registros ruidosos con estas tres

señales. Agregamos seis diferentes niveles de ruido por cada clase de ruido: em, ma y bw que representan los siguientes cocientes señal-ruido (SNR) en dB: 24, 18, 12, 6, 0 y -6. De esta manera completamos una base de datos de 18 señales.

Otro ruido presente muy frecuentemente en la captura del ECG es el ruido gaussiano, o ruido blanco, producido por la interferencia que causa la red eléctrica. Para probar las bondades de nuestro algoritmo frente a esta clase de ruidos y debido a que en la base de datos original del MIT-BIH, ninguno de estos 3 registros muestra ruido interferente de 50 Hz, emulamos tal interferencia. Para lograrlo, superpusimos al ECG original, es decir, limpio, señales de ruido generadas aleatoriamente con MATLAB®.

Las señales así generadas poseen cada una los niveles 10%, 30%, 40%, 50% y 65% de la amplitud promedio de la onda R para los registros #100 y #119. Amplitudes de 10%, 30% y 37% fueron las agregadas en el caso del registro #208.

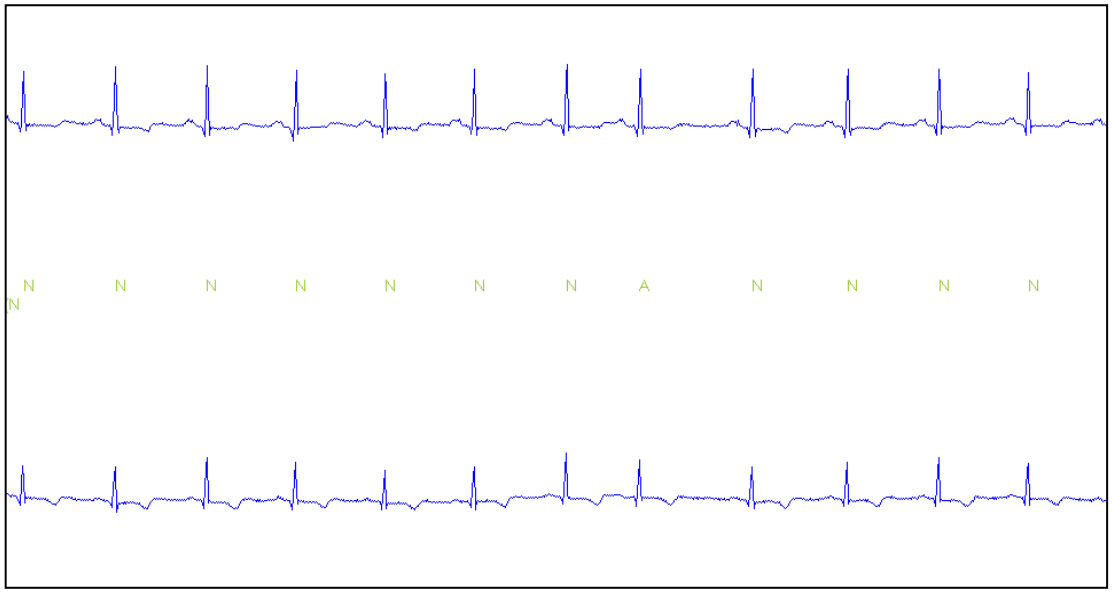

Figura 11.1 Registro #100 original. Señales 1 y 2

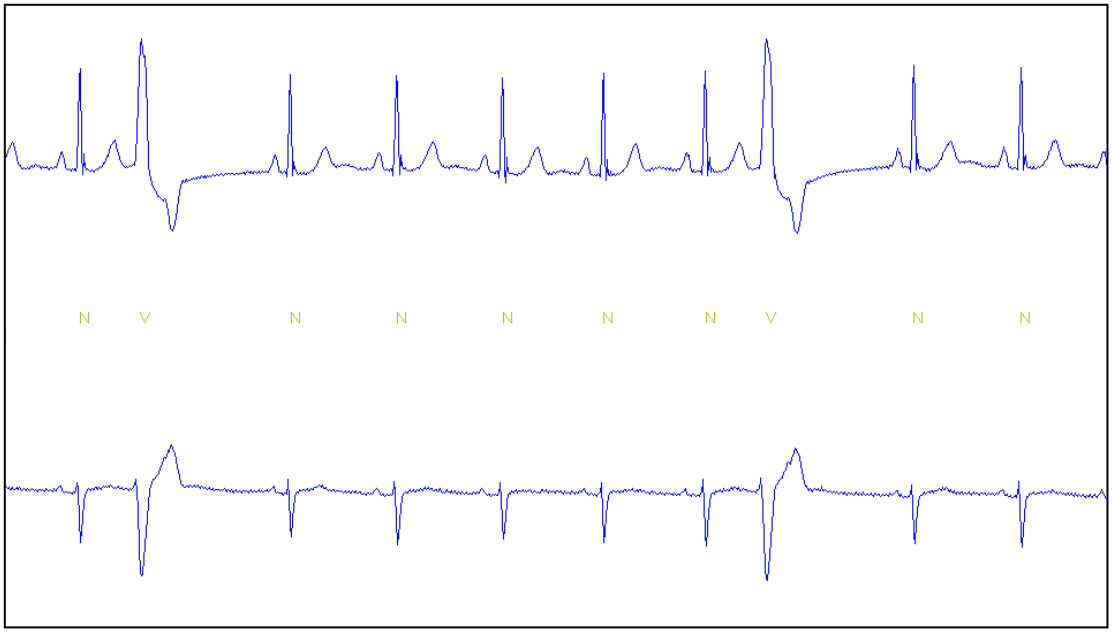

Figura 11.2 Registro #119 original. Señales 1 y 2

Figura 11.3 Registro #208 original. Señales 1 y 2

11.3. Procesos

Los procesos que realizamos sobre las señales, como ya lo mencionamos, incluyen:

- La generación de una base de datos con señales ruidosas. Tales ruidos incluyen los tres mencionados en el punto 11.2 (*ma*, *bw*, *em*) con sus seis niveles SNR en dB (-6, 0, 6, 12, 18, 24) y ruido blanco gaussiano en porcentajes con respecto a la amplitud del pico **R** (10%, 30%, 40%, 50% y 65%). También separamos fuentes en el dominio de las frecuencias. Las señales ruidosas para tal fin, contienen ruido gaussiano en 3 frecuencias distintas, 30Hz, 50Hz y 60Hz.

- Obtención de la matriz de des-mezcla, mencionada en el capítulo 5, que nos permitirá separar la señal limpia de la ruidosa mediante la resolución de la ecuación de Liapunov.

- Aplicación del algoritmo FastICA (que será luego descripto) a la misma base de datos ruidosa a fin de comparar nuestros resultados con un algoritmo ampliamente utilizado y testeado, para los casos en que trabajamos en el dominio del tiempo.

- Evaluación de los dos conjuntos de datos con índices de performance mencionados en el capítulo 8 para sacar conclusiones más objetivas.

11.4. Algoritmo Computacional

El algoritmo computacional toma la señal en crudo tal como es capturada por el sistema correspondiente. Esta señal está corrompida por los ruidos antes mencionados. Como existen 2 señales que intervienen en la mezcla (señal 1 corrompida y señal 2 corrompida), hemos definido la carga de dichas señales en una matriz bidimensional.

En el capítulo 5 mencionamos la importancia de la independencia estadística de las señales que serán “separadas” para que el Análisis de Componentes Independientes se pueda llevar cabo con éxito. Una de las medidas de independencia estadística mencionada

es la kurtosis, que evalúa la no-gaussianidad de una serie de datos. Optamos por este índice para evaluar si era posible separa las fuentes, por su simplicidad para implementar. Cuando la kurtosis es cero, la distribución es normal, por lo tanto, la des-mezcla no se produce. Cuanto más por encima de cero sea el valor de kurtosis, más independientes estadísticamente hablando son las señales entre sí y la gráfica de la distribución de probabilidad de la señal tiene una cresta más “pronunciada”.

Para hallar la resolución de la ecuación matricial de Liapunov, que nos permitirá encontrar la matriz de mezcla desconocida, elegimos las matrices que forman parte del sistema y que son conocidas, capítulo 5. En este caso, y luego de varias pruebas de laboratorio, definimos matrices aleatorias, que fueron las que mejores resultados arrojaron, pues con matrices identidad, no se produjo la des-mezcla.

El algoritmo diseñado se vale de la función de Liapunov para hallar la matriz incógnita y fue desarrollado haciendo uso de las herramientas de MATLAB® destinadas al diseño de sistemas en el espacio de estados. Debido a que la ecuación de Liapunov maneja matrices cuadradas, aplicamos el análisis de componentes principales (capítulo 4) a los datos de entrada para la extracción de componentes principales y obtuvimos así una matriz que contiene más del 98% de la información y que cumple con las exigencias matemáticas (la de ser cuadrada) para la aplicación de este método.

La ecuación de Liapunov logra hallar la matriz incógnita, que es para nosotros la matriz de mezcla. A partir de allí, obtuvimos la inversa de dicha matriz, la matriz de des-mezcla y la aplicamos a los datos originales, es decir a las señales ruidosas.

Con el algoritmo así diseñado obtuvimos los resultados que serán luego mencionados.

11.5. El Algoritmo FastICA

Es un algoritmo de punto fijo que utiliza la kurtosis para evaluar la independencia estadística de las series de datos. Fue generalizado para funciones de contraste general. La

ecuación que rige el algoritmo involucra derivadas de funciones de contraste general y el algoritmo, iterativo, ajusta los pesos normalizados un vector de norma 1 en cada iteración.

Los valores esperados se estiman utilizando los promedios de muestras sobre un número lo suficientemente grande de los datos de entrada.

El algoritmo de FastICA es neural y es paralelo y distribuido pero no adaptivo. En lugar de utilizar cada dato inmediatamente para el aprendizaje, utiliza promedios de muestras computadas sobre grandes cantidades de datos. La velocidad de convergencia es alta, de allí su nombre.

11.6. Resultados y Discusión.

Dividimos los procesos en tres desarrollos independientes, cada uno con sus resultados y discusión:

- En el primer desarrollo procesamos y evaluamos los registros con los ruidos agregados provistos por *PhysioBank* (ma, em y bw) con todos sus niveles de ruido. Evaluamos la performance con índices diseñados específicamente para la evaluación de algoritmos de separación ciega de fuentes y comparamos nuestros resultados con los obtenidos luego de aplicar FastICA al mismo conjunto de datos.

- En el segundo desarrollo procesamos y evaluamos los registros con ruido gaussiano agregado mediante un algoritmo computacional diseñado por nosotros para tal fin. Evaluamos la performance de manera similar al la realizada en el desarrollo 1.

- En el tercer desarrollo procesamos y evaluamos los registros con ruido gaussiano agregado mediante un algoritmo computacional diseñado por nosotros para tal fin, y separamos las frecuencias deseadas de las no-deseadas. Evaluamos los resultados con índices estadísticos.

11.6.1. Desarrollo 1

En el primer desarrollo, procesamos y evaluamos los registros con los ruidos agregados provistos por PhysioBank (ma, em y bw) con todos sus niveles de ruido. Los resultados se encentran listados en la tabla 11.1

Evaluamos la performance para el caso solamente en que la distorsión sobre $\hat{s}_j$ son invariantes en el tiempo, ver capítulo 8.

Utilizamos los índices de performance propuestos por Vincent y colaboradores (Vincent *et al*., 2006). Valiéndonos de estas herramientas, comparamos nuestro algoritmo de separación ciega de fuentes –basado en las condiciones de estabilidad de Liapunov mencionadas en el capítulo 5- y el bien conocido algoritmo FastICA, brevemente descripto en el punto 11.5, también desarrollado en MATLAB® por Gävert y colaboradores (Gävert *et al.*, 2005).

Tabla 11.1 Medida de la Performance de los algoritmos de Separación Ciega de Fuentes.

Señal	Nuestro Algoritmo			FastICA		
	SNR [dB]	SAR [dB]	SDR [dB]	SNR [dB]	SAR [dB]	SDR [dB]
119em_6	21.40	0.28	0.22	11.03	-9.00	-10.16
119em00	32.32	5.51	5.50	38.76	10.32	10.31
119em06	16.64	-11.65	-11.75	24.60	-3.87	-3.89
119em12	19.43	-11.78	-11.84	26.81	-4.27	-4.28
119em18	25.09	8.29	8.19	28.43	10.28	10.21
119em24	22.23	0.78	0.72	22.18	0.73	0.67
119ma_6	31.35	-5.60	-5.61	27.64	-9.31	-9.32
119ma00	19.69	-0.47	-0.55	21.49	-2.76	-2.80
119ma06	25.27	1.09	1.06	25.11	-2.88	-2.90
119ma12	29.57	9.35	9.30	26.60	6.38	6.33
119ma18	20.45	2.05	1.95	20.39	-2.74	-2.80
119ma24	25.61	1.83	1.80	33.76	11.21	11.19
119bw_6	21.93	-1.94	-1.98	27.76	1.57	1.54
119bw00	27.22	2.37	2.35	27.28	2.43	2.41
119bw06	29.63	3.22	3.20	32.63	5.40	5.39
119bw12	35.02	5.24	5.23	33.24	3.60	3.59
119bw18	24.45	-10.73	-10.75	38.27	1.54	1.53
119bw24	22.98	-3.42	-3.45	25.47	0.06	0.04

La Tabla 11.1 muestra nuestros resultados. En la columna 1 listamos las señales y los tipos y niveles de ruido asociado. De las columnas 2 a la 4 presentamos los cocientes para cada señal experimental obtenidos con nuestro método de separación. Los cocientes de energía calculados para las señales separadas usando FastICA están listadas en las columnas 5 a la 7.

Para una comparación más apropiada, nuestro algoritmo, basado en criterios de estabilidad, y el método FastICA fueron aplicados sobre los mismos segmentos de los registros de ECG.

Cabe destacar que nuestro método no realiza pre-procesamiento de los datos, sin embargo, FastICA utiliza, previo a la extracción de los componentes independientes, el Análisis de Componentes Principales para limpiar los datos si fuera necesario.

SNR representa el ruido residual de la señal fuente recuperada. Como lo mencionamos previamente, agregamos tres tipos de ruido a nuestro registro de ECG: movimiento de electrodo (em), artefactos musculares (ma) y movimiento de línea de base (bw). Un valor elevado del índice SNR indica menor ruido residual y por consiguiente una mejor performance en la extracción. Esta tendencia es válida para todas las señales experimentales y para ambos algoritmos, excepto para el caso de señales o muy limpias o muy ruidosas (las que presentan niveles de dB 24 y -6).

Nuestro algoritmo presenta SNR más elevados en 7 de las 18 señales que integran este grupo. Tanto para las señales 119em24, 11ma06 y 11bw00, la diferencia entre un método y otro no supera los 0,16 puntos. El valor más alto de SNR lo exhibe el algoritmo FastICA para la señal 119em00, con 38,76. El máximo valor de SRN alcanzado por nuestro algoritmo es de 35,02 para la señal 119bw12, Figura 11.4. La Figura 11.5 muestra la performance para la misma señal del algoritmo FastICA. Las Figura 11.6Figura 11.7 muestran la performance para la señal 119em00 del algoritmo basado en el criterio de estabilidad y del FastICA respectivamente.

De las figuras se desprende que si bien hay una diferencia de varios puntos en los valores de los índices de SNR para ambos métodos, en un caso a favor de nuestro algoritmo

y en otro, a favor del FastICA, visualmente las performances demuestran similitudes. Por ejemplo, el hecho de arrastrar el ruido interferente a la derivación V1 (señal 2): Figura 11.4(d), Figura 11.5(d) Figura 11.6(d) y Figura 11.7(d). Otro aspecto similar es que la señal separada, la fuente de interés, ubicada en la cuadro (c) en las cuatro figuras, está estandarizada para que su línea de base se mantenga con valores de amplitud cercanos a cero.

Con respecto a los otros índices listados en la Tabla 11.1 Medida de la Performance de los algoritmos de Separación Ciega de Fuentes.Tabla 11.1, observamos que los valores de SAR son bajos para ambos métodos. Tanto SAR como SDR exhiben valores similares. Esto se debe al hecho que consideramos e_{interf} en la evaluación -que se refiere a la interferencia de fuentes no deseadas- con valor cero.

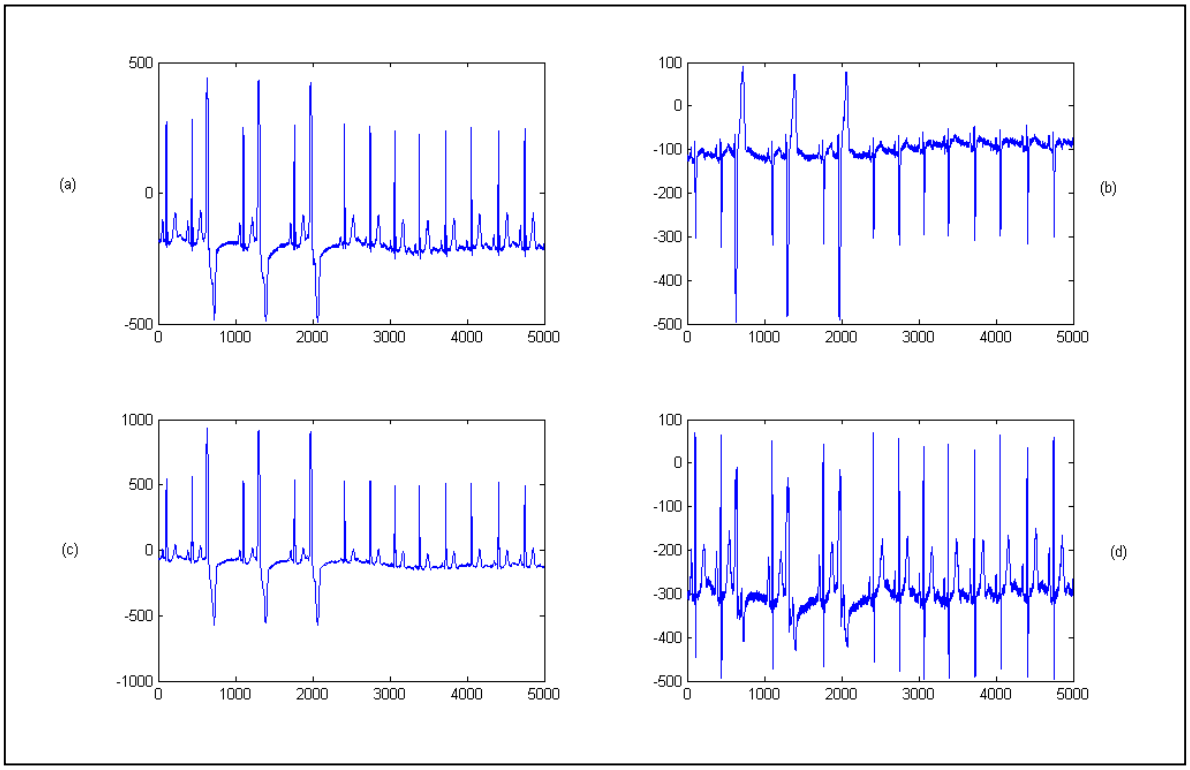

Figura 11.4 Señal 119 con ruido bw de nivel 12dB (119bw12 en tabla). Performance obtenida con algoritmo de estabilidad. (a) señal 1 ruidosa. (b) señal 2 ruidosa. (c) fuente limpia, separada. (d) fuente ruidosa, separada

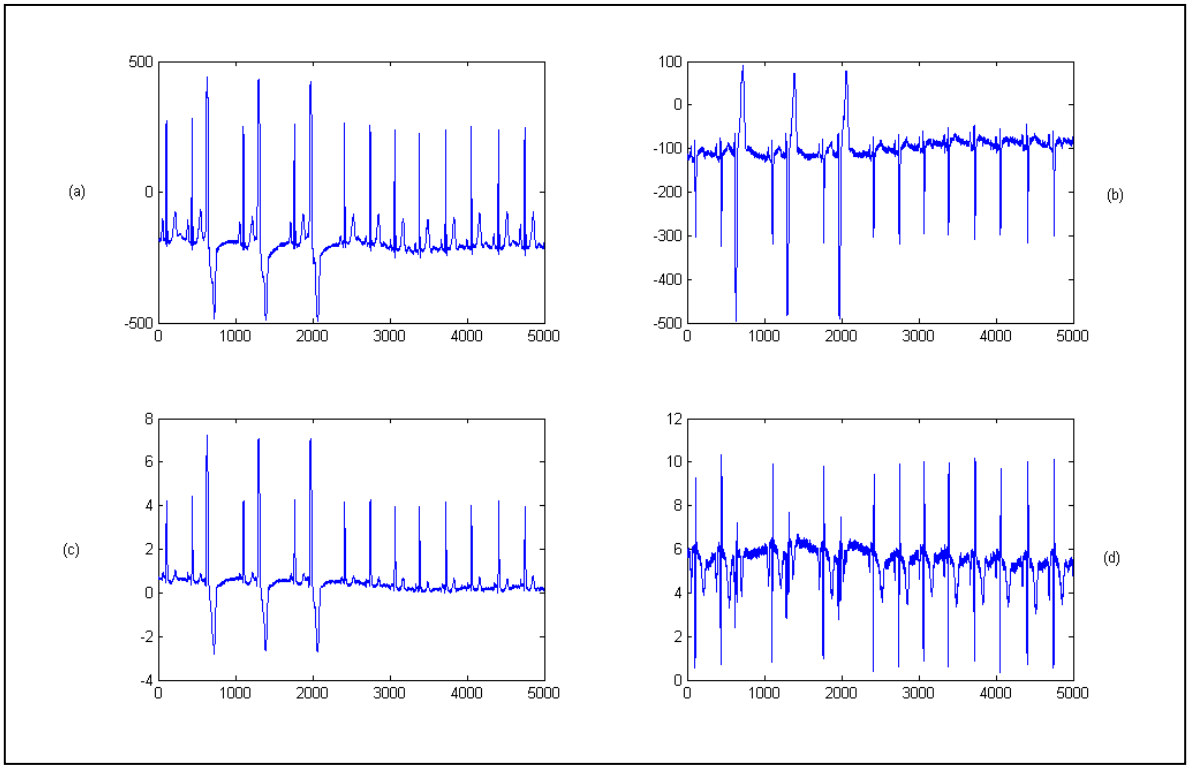

Figura 11.5 Señal 119 con ruido bw de nivel 12dB (119bw12 en tabla). Performance obtenida con algoritmo FastICA. (a) señal 1 ruidosa. (b) señal 2 ruidosa. (c) fuente limpia, separada. (d) fuente ruidosa, separada

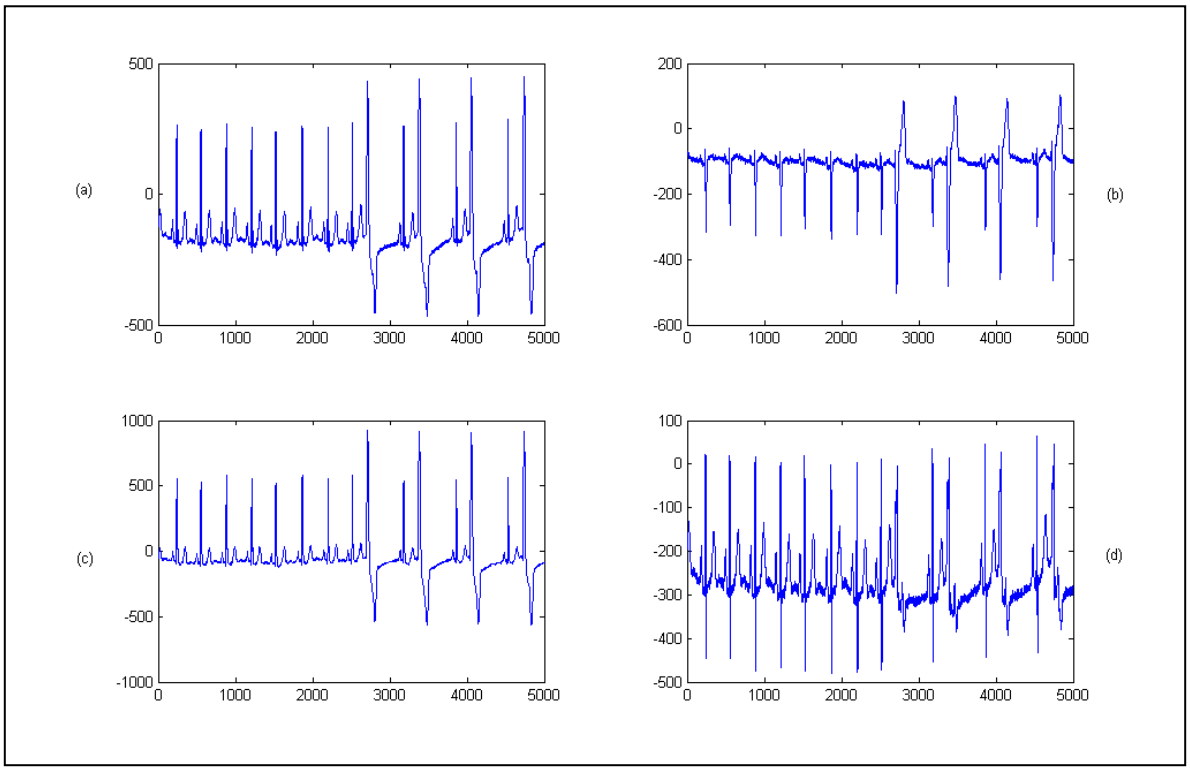

Figura 11.6 Señal 119 con ruido em de nivel 0dB (119em00 en tabla). Performance obtenida con algoritmo de estabilidad. (a) señal 1 ruidosa. (b) señal 2 ruidosa. (c) fuente limpia, separada. (d) fuente ruidosa, separada

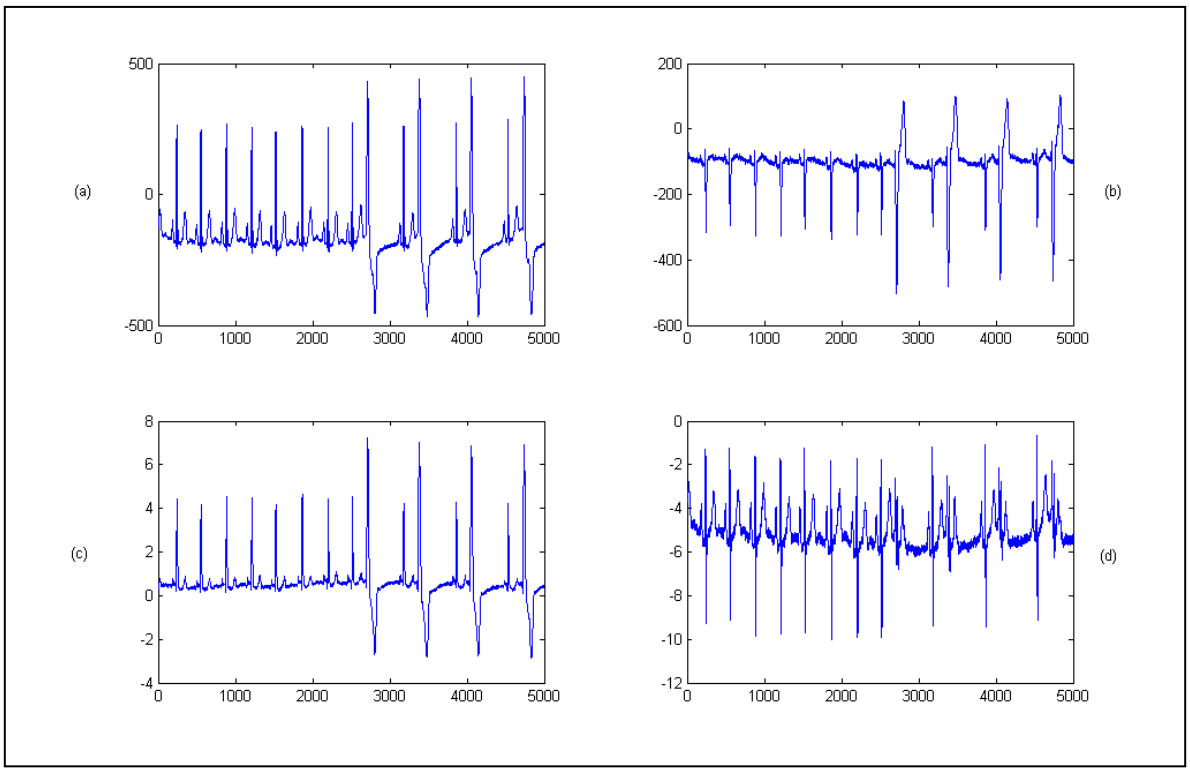

Figura 11.7 Señal 119 con ruido em de nivel 0dB (119em00 en tabla). Performance obtenida con algoritmo FastICA. (a) señal 1 ruidosa. (b) señal 2 ruidosa. (c) fuente limpia, separada. (d) fuente ruidosa, separada

11.6.2. Desarrollo 2

En la segunda etapa, procesamos y evaluamos los registros con ruido gaussiano agregado mediante un algoritmo computacional diseñado por nosotros para tal fin. Las señales a las que se les agregó ruido son las señales #100, #119 y #208. Los niveles de ruido están en porcentajes con respecto a la amplitud del pico R y son 10%, 30%, 40%, 50% y 65% para las señales #100 y #119. Para la señal #208 los niveles son 10%, 30% y 37%. En reiteradas pruebas testeamos tanto nuestro algoritmo como el FastICA para niveles de ruido por encima del 37% para esta señal en particular, sin obtener, en ninguno de los casos, resultados esperados.

La Figura 11.8 ejemplifica el caso particular de la señal #119, con ruido agregado de 40%. Las dos derivaciones analizadas, MLII y V1, se muestran en su forma original en (a) y (b) respectivamente, y en (c) y (d) se indican con el ruido agregado. La Figura 11.8(e)

ilustra la señal MLII recuperada por aplicación de la Ecuación Algebraica de Liapunov, mientras que en (f) se muestra la otra fuente separada, que consiste en la mezcla del ruido agregado con la señal V1. De manera análoga, la Figura 11.9 ilustra el caso de la señal #208, con 20% de ruido blanco agregado.

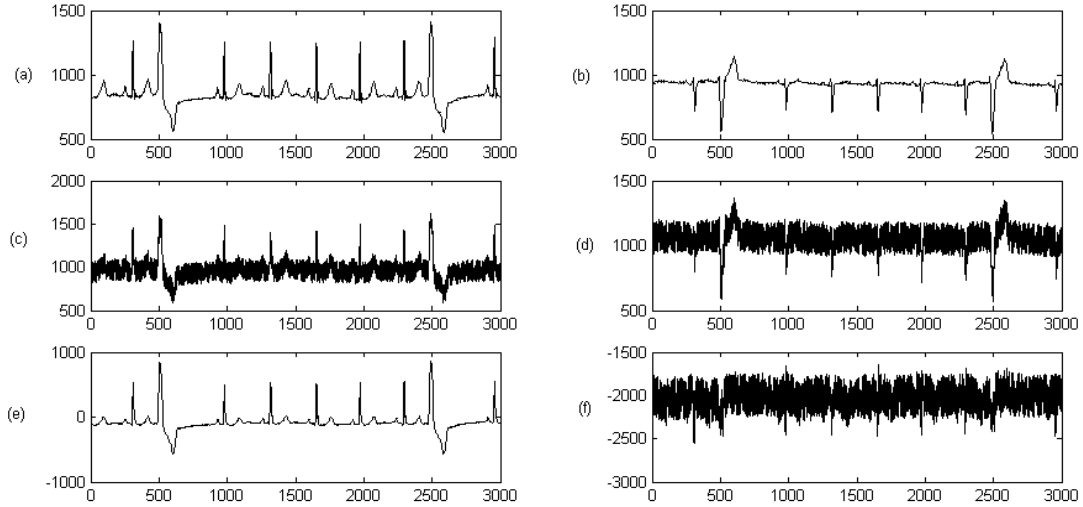

Figura 11.8. Señal 119 con ruido agregado al 40%. Columna 1 derivación MLII. Columna 2 derivación V1. (a) y (b) Señal original. (c) y (d) Señal con ruido agregado. (e) y (f) Señales separadas algoritmo basado en criterios de estabilidad.

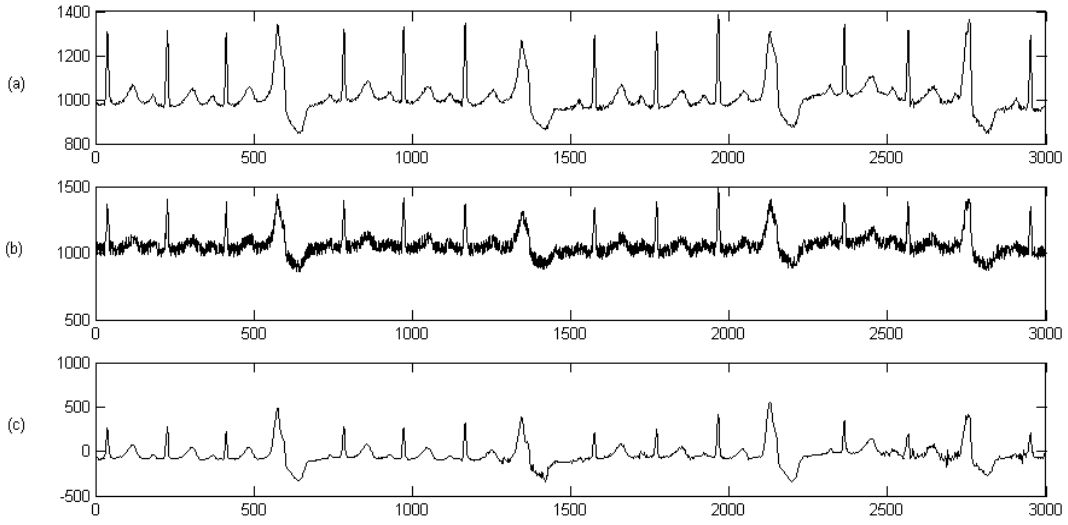

Figura 11.9 Señal 208 derivación MLII con ruido agregado del 20%. (a) Señal original. (b) Señal con ruido agregado. (c) Señal separada luego de aplicado el método.

Como en el desarrollo 1, a los fines comparativos, procedimos también a separar las señales utilizando la herramienta FastICA, Figura 11.10.

Para una comparación cuantitativa, que sirviera de complemento a la evaluación visual del comportamiento de ambos métodos, Liapunov y FastICA, calculamos los índices indicados en el capítulo 8, al igual que en el desarrollo 1. Obtuvimos para el caso de la señal #119 con Liapunov un SAR de 234.69 y de 262.74 con FastICA. Los índices SDR y SIR para la separación por Liapunov fueron iguales a –8.86, mientras que ambos índices, para FastICA arrojaron valores de –0.59 y –0.58 respectivamente.

La relación señal-ruido (SNR) por aplicación de Lyapunov fue de 246.37, y para FastICA fue de 30.16.

Con respecto a la señal #208, cabe mencionar que por encima del 37% de ruido agregado, el método Liapunov no logró separar las fuentes, como tampoco logró el método FastICA. Sin embargo, por debajo de ese nivel de ruido, el método propuesto de Lyapunov exhibe una mejor performance que el FastICA.

En la Figura 11.11 se muestra el comportamiento de la kurtosis en la señal #100 con los distintos niveles de ruido evaluados. La señal #100 se caracteriza por un comportamiento muy regular del ritmo sinusal normal, como ya lo mencionamos. La kurtosis disminuye bruscamente hasta un ruido agregado del 40%, manteniéndose en las proximidades de 0 para valores de ruido mayores. A partir de un ruido agregado del 65%, el método propuesto de Liapunov no lograría eliminar el ruido.

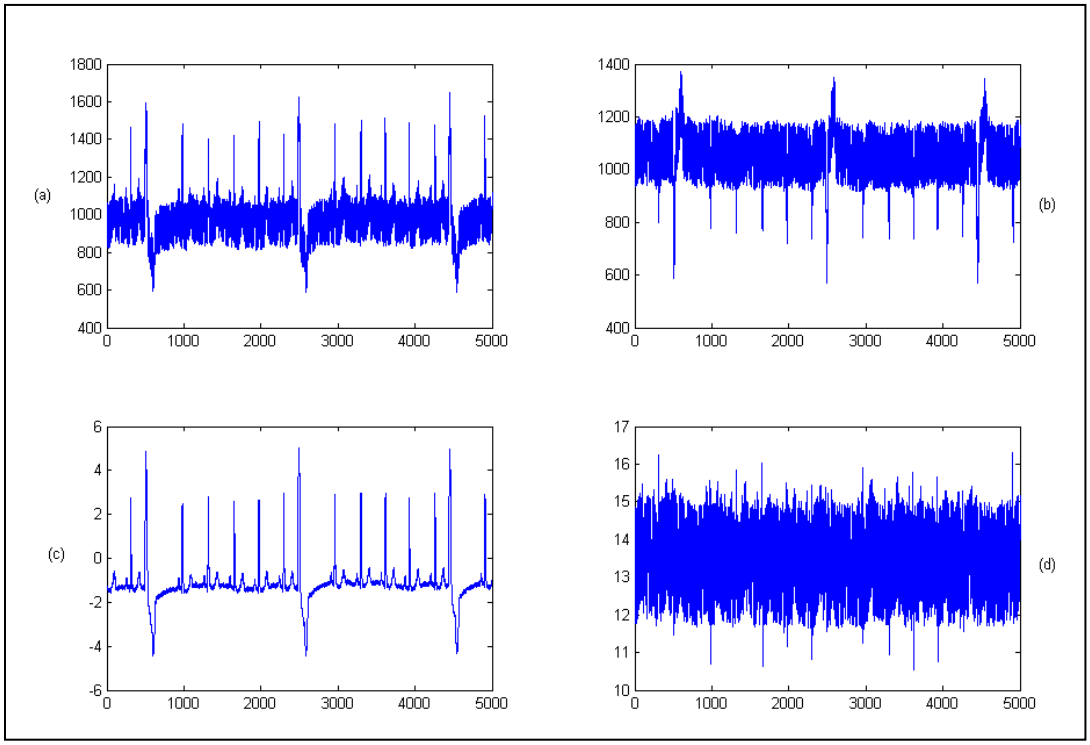

Figura 11.10. Señal 119 con 40% de ruido agregado. Algoritmo de separación utilizado: FastICA. (a) y (b) señales ruidosas, entradas del sistema. (c) y (d), señales separadas, salida del sistema.

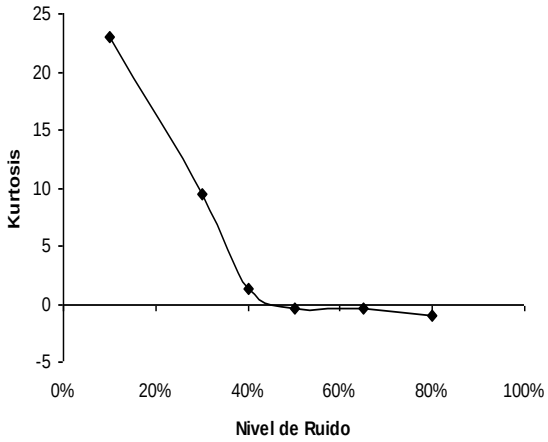

Figura 11.11 . Índice de Separación. Evolución de la kurtosis en la señal 100 en función de los niveles de ruido definidos.

11.6.3. Desarrollo 3

En este tercer desarrollo utilizamos nuestro método de separación de fuentes basado en los criterios de estabilidad de Liapunov para separar señales de ECG que contienen ruido gaussiano agregado a diferentes frecuencias. La diferencia existente entre este desarrollo y el desarrollo 2 es que tratamos a las señales en el espacio de las frecuencias. Mediante la separación de fuentes en el espacio de las frecuencias podremos identificar las frecuencias asociadas a las señales no deseadas.

Las señales a utilizar en este desarrollo pertenecen a la base de datos descripta en el capítulo 2. Seleccionamos tres señales de la serie 100 y tres señales de la serie 200. Utilizamos en particular los registros identificados como #100, #101 y #102, digitalizados a 11 bits y 360 Hz de velocidad de captura.

De la serie 200 utilizamos 3 señales. Ellas son la #200, #201 y #202. Estas señales fueron elegidas por contener episodios de fibrilación auricular, trigemia ventricular y taquicardia ventricular entre otras anomalías.

Estas 6 señales fueron obtenidas utilizando la derivación precordial V1 y una modificación de la derivación II, denominada MLII, con electrodos pectorales. Se toman ambas derivaciones en razón de que el número de fuentes debe ser igual al número de observaciones, para llegar a un resultado cierto.

En la base de datos original del MIT-BIH, ninguno de estos 6 registros muestra ruido interferente. Para lograrlo, hemos superpuesto al ECG normal señales de ruido de distintas frecuencias generadas mediante un algoritmo desarrollado por nosotros para tal fin, utilizamos para ello MATLAB®.

Las señales de ruido generadas contenían frecuencias de 30, 50 y 60 Hz. Las cuales fueron elegidas debido a que en el ECG encontramos frecuencias menores a los 100 Hz.

Para la evaluación cuantitativa del comportamiento del método propuesto para la eliminación del ruido –o recuperación de fuentes-, adoptamos el criterio de la Covarianza y el Coeficiente de Correlación Lineal de Pearson, descriptos en el capítulo 8.

La Covarianza entre 2 variables es un estadístico resumen indicador de si las puntuaciones están relacionadas entre sí. Se la utiliza para medir el grado de relación de 2 variables. Este índice refleja la relación lineal que existe entre 2 variable x e y.

A modo ejemplo, mostramos los resultados para la señal #100 con un ruido blanco de 50Hz. En la Figura 11.12 graficamos la señal #100 limpia y la misma con un ruido agregado de 50Hz. Ésta última, en sus dos derivaciones, componen la matriz de entrada del sistema.

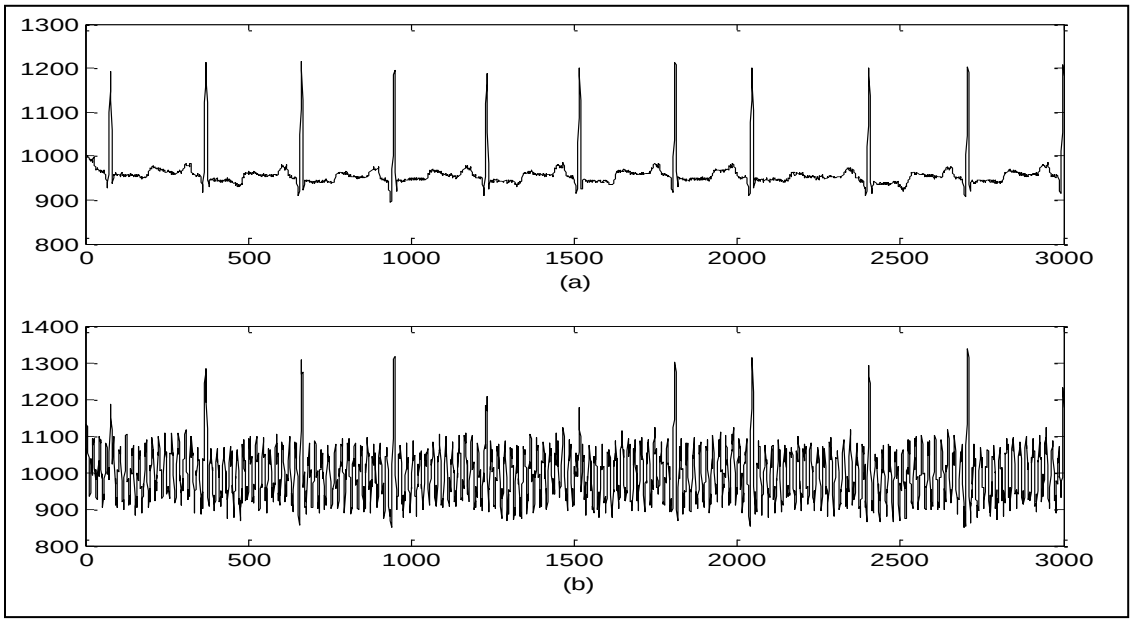

Figura 11.12 Señal #100 limpia; 3000 muestras. (a) Señal limpia. (b) Señal corrompida con ruido blanco de 50Hz.

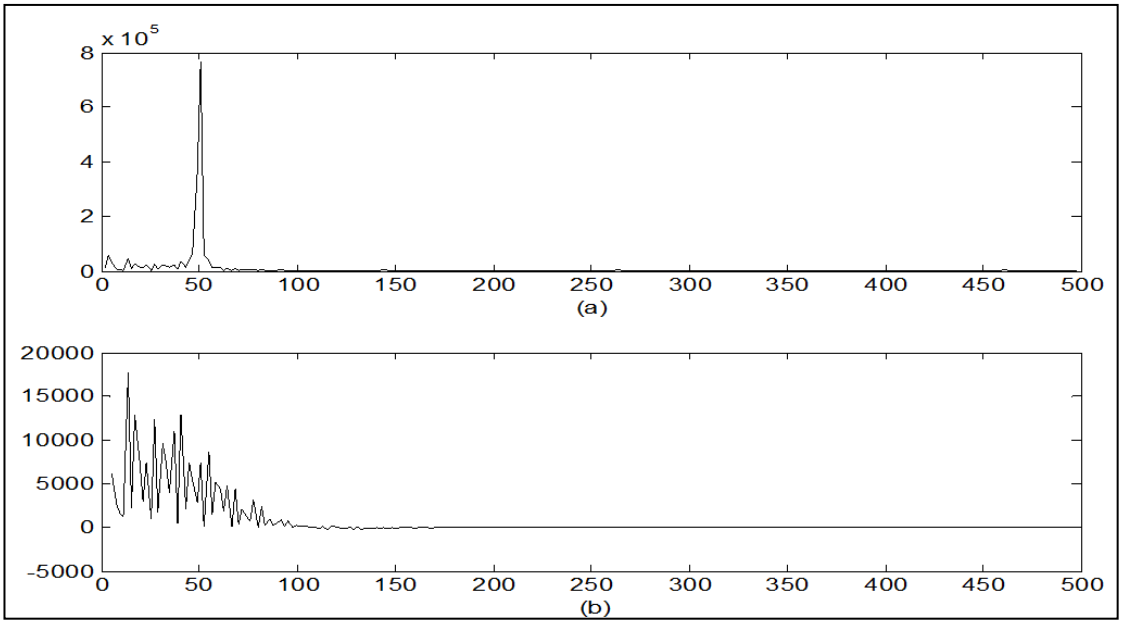

Figura 11.14 Salida del proceso. (a) Espectro de frecuencia de la señal no deseada. (b) Espectro de frecuencia de la señal de interés (#100).

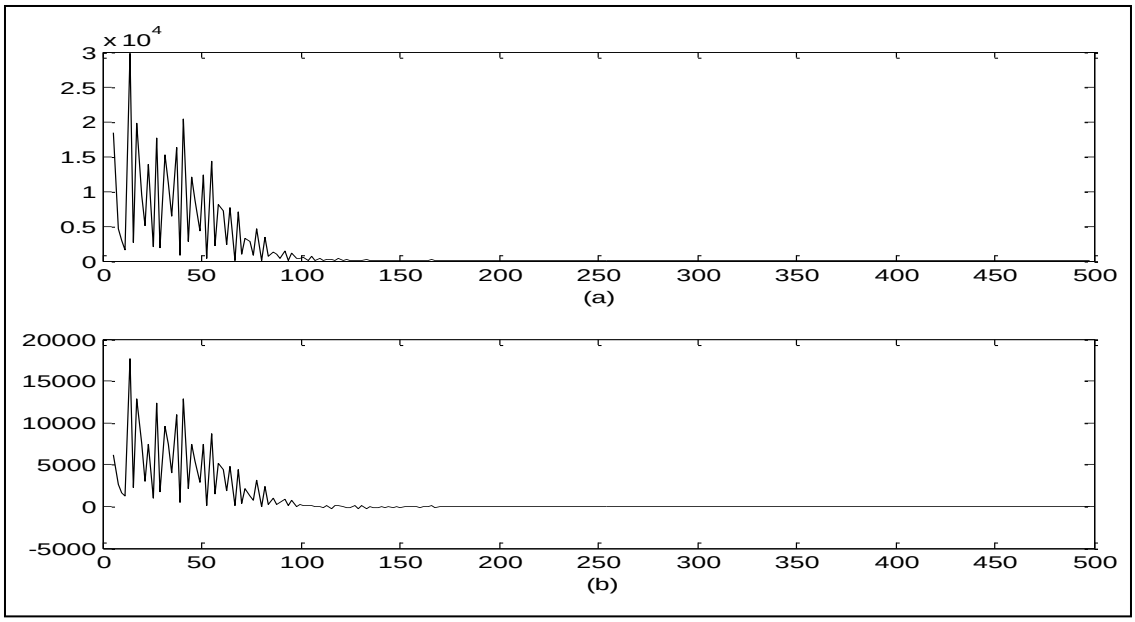

Figura 11.13 (a) Espectro de frecuencia de la señal #100 sin corromper. (b) Espectro de frecuencia de la señal (#100) luego se la separación de fuentes.

Durante el proceso utilizamos la Transformada de Fourier Rápida para señales discretas (FFT) (capítulo 3) para obtener los espectros de frecuencia de las señales estudiadas, que se encuentran mezcladas de una forma desconocida. Además de la FFT, aplicamos el proceso de Análisis de Componentes Independientes propuesto en conjunto con técnicas de Análisis de Componentes Principales que permiten obtener una salida de la forma de la Figura 11.12.

En la Figura 11.12(a) podemos observar el espectro de frecuencia del ruido de 50Hz, ubicado en su componente de frecuencia correspondiente. La Figura 11.12(b) muestra el espectro de frecuencia de la señal de ECG, en este caso la #100.

Para poder comparar visualmente los resultados, agregamos Figura 11.13, donde el lector puede observar en (a) el espectro de frecuencias de la señal #100 limpia, es decir sin ruido agregado y en (b), puede observar el espectro de frecuencias de dicha señal que se obtuvo luego de separar las observaciones con el método propuesto. A simple vista ambas señales guardan analogía morfológica, distinguiéndose en sus valores de amplitud.

La comparación cuantitativa la llevamos a cabo haciendo uso de los índices estadísticos mencionados con anterioridad. La Tabla 11.2 exhibe los resultados obtenidos.

Los índices evalúan la correlación existente entre el espectro de frecuencias de la señal de interés limpia Figura 11.13(a) y el espectro de frecuencias de la señal de interés luego de la separación Figura 11.13(b).

Tabla 11.2 Índices de Covarianza y Coeficiente de Correlación de Pearson. Señales #100, #101, #102, #200, #201, #202

Serie 100				**Serie 200**			
Señal	**Ruido en Hz**	**Covarianza**	**Coeficiente de Correlación**	**Señal**	**Ruido en Hz**	**Covarianza**	**Coeficiente de Correlación**
100	30Hz	8949058,28	0,996	200	30Hz	313794816,81	0,969
	50Hz	8949058,28	0,996		50Hz	292948777,45	0,967
	60Hz	8949058,28	0,996		60Hz	290363108,58	0,946
101	30Hz	60650883,55	0,979	201	30Hz	35842190,15	0,972
	50Hz	59315682,65	0,980		50Hz	35842190,15	0,972
	60Hz	59563475,86	0,971		60Hz	35842190,15	0,972
102	30Hz	83760641672,04	1,000	202	30Hz	89709075,73	0,984
	50Hz	83760641672,04	1,000		50Hz	88343323,58	0,985
	60Hz	83760641672,04	1,000		60Hz	88047050,92	0,976

De la Tabla 11.2 se deduce que para todas las señales analizadas en este desarrollo, el índice de covarianza da valores numéricos realmente altos, lo que nos da una idea de la gran relación que existe entre el espectro de frecuencias de la señal original, libre de interferencias y la señal recuperada luego de aplicado el método para la separación de fuentes. Los índices del Coeficiente de Correlación, también con buenos resultados, son todos muy próximos a uno o bien uno, como en el caso de la señal #102.

Como realizamos en los dos desarrollos anteriores, evaluamos la kurtosis previo a la aplicación de nuestro algoritmo. La kurtosis mide la no-gaussianidad de los datos y representa la elevación o achatamiento de una distribución comparada con la distribución normal. Una kurtosis positiva indica una distribución relativamente elevada mientras que una kurtosis negativa indica una distribución relativamente plana. En todos los casos probados, en este desarrollo, obtuvimos valores de kurtosis por superiores a 5.

12. Redes Neuronales

12.1. Motivación

El reconocimiento y la clasificación de señales electrocardiográficas han llamado la atención de los investigadores durante las últimas décadas. Muchos algoritmos dedicados a la detección y clasificación de latidos cardíacos han sido reportados, (Ince et al. 2009; Yu y Chou, 2006; Osowski y Linh, 2000; Ghongade y Ghatol, 2008). El amplio uso de las redes neuronales para estas tareas se debe principalmente a sus características de adaptabilidad, no linealidad y su habilidad para generalizar datos incompletos.

De todas las tareas que una red neuronal puede realizar, sin dudas, su habilidad para clasificar conjuntos de datos es lo que las ha hecho tan útiles y populares. Valiéndonos de estas ventajas, recurrimos a las redes neuronales, en la topología MLP (perceptrón multicapas) con reglas de aprendizaje nítidas y difusas, a fin de clasificar contracciones ventriculares prematuras.

Las grandes variaciones de morfología de la señal eléctrica cardíaca representan un problema para el análisis automático de la actividad cardiovascular, pues resulta esencial que los sistemas instrumentales detecten y clasifiquen en forma precisa distintas características de las ondas del ECG. Con el objetivo de identificar y clasificar patrones anómalos, el análisis rutinario del ECG clínico depende casi con exclusividad de la inspección visual. Las formas y los intervalos de las ondas componentes del ECG proveen de importante información sobre el estado del corazón. La variación en el tiempo de estas ondas, por ejemplo, es una de las características morfológicas destacadas del electrocardiograma.

Aunque el sistema de conducción eléctrica esté intacto, el corazón puede generar latidos denominados aberrantes, entre los que se cuentan las extrasístoles auriculares (EA) y las extrasístoles ventriculares (EV). Estas últimas se cuentan entre las arritmias más frecuentes y potencialmente más peligrosas, y pueden aparecer en personas sanas.

Su peligrosidad radica en que eventualmente degeneran en taquicardia o fibrilación ventriculares.

Hemos diseñado e implementado redes neuronales para la detección de latidos ventriculares prematuros en el conjunto de señales descriptas en el capítulo 2. La topología de la red neuronal diseñada es *MLP* (Multiple Layer Perceptron –Perceptron Multi-Capa) con el algoritmo de retro-propagación. Diseñamos además una *red neuronal con lógica difusa*, que logra diferenciar latidos ectópico de los normales.

La estructura, el entrenamiento, las entradas y los resultados son descriptos a continuación, diferenciándose entre la red nítida y difusa.

12.2. La Red Nítida

12.2.1. Entradas

Las entradas a la red están compuestas por

a) la frecuencia instantánea, distancia R-R;
b) los valores de amplitud de cada complejo QRS, que forman un vector fila de 50 muestras, donde en el orden 30, se encuentra el valor de amplitud del pico R;
c) la transformada rápida de Fourier de cada complejo QRS;
d) la longitud en muestras del complejo QRS;
e) la detección o no de la onda P.

Luego de numerosos ensayos, con otros parámetros característicos de las PVC, escogimos esta configuración, por ser la que mejor caracteriza a los latidos cardíacos y en particular a los latidos ventriculares prematuros.

Cabe destacar que no creímos necesario aplicar algún tipo de pre-procesamiento, como un filtrado, ya que las señales de la base de datos están libres de ruidos.

El entrenamiento, prueba y validación se hicieron siguiendo el método de validación cruzada de orden 3, descripto en el capítulo 6.

El punto clave en la conformación del vector de entradas a la red es la matriz que contiene los valores de amplitud del complejo QRS, pues los demás datos, con excepción de la detección de la onda P, se deducen de dicha matriz. En el capítulo 10 de aplicaciones del análisis de componentes principales, realizamos una descripción detallada de la matriz que contiene los datos de cada complejo QRS de las señales a clasificar. A partir de esta matriz, obtenemos los valores de amplitud de cada onda característica Q, R y S y el momento exacto en que cada una de ellas se produce. Con estos dos datos de cada pico, podemos generar los datos a) y d) mencionadas en la descripción de las entradas. Para hallar la transformada rápida de Fourier, y obtener así el espectro de frecuencias de cada complejo QRS, aplicamos la FFT, descripta en el capítulo 3 a cada fila de la matriz que contiene los datos del complejo QRS. Luego, por cada latido, obtenemos 50 valores en frecuencia de los latidos cardíacos.

Un párrafo aparte merece la detección de la onda P. Incluimos esta entrada debido a que, como los latidos a clasificar son los latidos ventriculares prematuros, cuando éstos ocurren, ocurren antes de lo esperado, de allí el nombre prematuro. Cuando se produce una PVC, la onda P no se distingue, como puede apreciarse en la Figura 12.1.

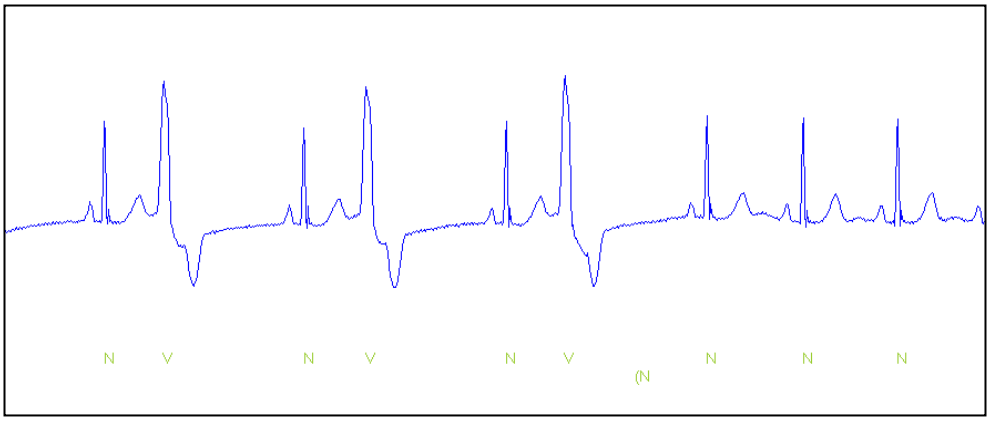

Figura 12.1 Señal 119. Nótese la ausencia de la onda P al producirse una PVC

La aparición o no de la onda P se convierte en un rasgo interesante a considerar para la detección o clasificación de arritmias ventriculares prematuras. En el vector de entradas, incorporamos un campo binario para determinar si hay onda P o no. Si ocurre la onda, entonces estamos en presencia de un latido normal y el campo tiene el valor 0, de lo contrario, estamos el latido es una PVC y el campo tiene un valor 1.

El algoritmo que diseñamos para la detección de la onda P aplica un filtro pasa bajos, luego un filtro derivativo seguido de una ventana que toma las 50 muestras

previas al complejo QRS. Haciendo uso de umbrales, establecemos el valor máximo dentro de esa ventana y lo comparamos con el pico R de la matriz de los complejos QRS ya conformada. Si el valor máximo es positivo y además es, en proporción con respecto al pico R, menor o igual a un umbral elegido convenientemente, entonces estamos en presencia de una onda P, y el dato se establece en cero, de lo contrario, el campo vale 1, como ya lo mencionamos.

A continuación, exhibimos en las Tabla 12.1, Tabla 12.2 y Tabla 12.3 los resultados obtenidos con la aplicación de este algoritmo, separadas por series por conveniencia. Cabe aclarar que listamos únicamente las señales que luego serán analizadas por la red neuronal y que son aquellas que contienen PVC. Las señales restantes de la base de datos fueron descartadas por no poseer PVC.

Tabla 12.1 Detección de la onda P. Serie 100

Onda P. Serie 100					
Señales	TP	FP	FN	S [%]	PP [%]
100	2117	153	1	99,95	93,26
104	2090	137	2	99,90	93,85
105	2529	34	39	98,48	98,67
107	2111	0	26	98,78	100
108	560	4	6	98,94	99,29
109	746	99	11	98,55	88,28
111	697	0	0	100	100
114	523	1	32	94,23	99,81
116	795	0	1	99,87	100
118	2219	56	12	99,46	97,54
119	1987	0	0	100	100
123	504	0	1	99,80	100
124	503	0	20	96,18	100

Tabla 12.2 Detección de la onda P. Serie 200

Onda P. Serie 200					
Señales	TP	FP	FN	S [%]	PP [%]
200	1805	0	795	69,42	100,00
201	1801	0	198	90,10	100,00
202	2113	2	19	99,11	99,91
203	2541	54	409	86,14	97,92
205	2586	0	70	97,36	100,00
207	2217	41	71	96,90	98,18
208	1960	11	991	66,42	99,44
209	2915	96	0	100,00	96,81
210	2460	2	188	92,90	99,92
213	3092	118	40	98,72	96,32
214	2004	21	241	89,27	98,96
215	3228	3	133	96,04	99,91
217	1900	143	156	92,41	93,00
221	665	0	160	80,61	100,00
223	782	0	55	93,43	100,00
228	680	2	36	94,97	99,71
231	673	0	2	99,70	100,00
233	745	2	75	90,85	99,73

Tabla 12.3. Detección de la onda P. Señal 119 con ruidos agregados

Señal	Onda P				
	TP	FP	FN	S [%]	PP [%]
119bw_6	937	1029	19	98,01	47,66
119bw00	961	1018	3	99,69	48,56
119bw06	1044	939	2	99,81	52,65
119bw12	1172	813	0	100,00	59,04
119bw18	1280	705	0	100,00	64,48
119bw24	1575	410	0	100,00	79,35
119em_6	1057	835	93	91,91	55,87
119em00	1170	803	12	98,98	59,30
119em06	1273	709	3	99,76	64,23
119em12	1479	506	0	100,00	74,51
119em18	1636	349	0	100,00	82,42
119em24	1702	283	0	100,00	85,74
119ma_6	1031	762	192	84,30	57,50
119ma00	1254	728	3	99,76	63,27
119ma06	1148	754	83	93,26	60,36
119ma12	1481	504	0	100,00	74,61
119ma18	1618	367	0	100,00	81,51
119ma24	1618	367	0	100,00	81,51

Los índices obtenidos demuestran una muy buena performance del algoritmo diseñado cuando analizamos señales libres de ruido. La serie 100 en particular exhibe los valores más altos de sensibilidad y predictividad positiva. De la serie 200, si bien los índices están por encima del 66% para el caso de la S y por encima del 93% para la PP, las señales #200 y #208 son las que menores valores de Sensibilidad arrojaron y es que son señales difíciles de examinar pues sus PVCs son multiforme; también contienen segmentos de ruido de alta frecuencia lo que dificulta su análisis. Además la señal #208 presenta mayormente PVC de fusión que se presentan en modo de bigeminismo. Con

respecto a los resultados obtenidos con la señal #119 con ruido agregado, podemos decir que el algoritmo produce, en casi todos los casos altos números de FP y es por el tipo de proceso que se realiza sobre la señal para detectar la onda P. Observamos que en entornos ruidosos es necesario ajustar los umbrales definidos para evitar la acumulación de Falsos Positivos y lograr así una detección más eficiente de la onda P.

12.2.2. Diseño de la Red Neuronal

Nuestro modelo de red MLP fue diseñado con MATLAB® (The MathWorks Inc., Natick, Massachusetts, USA). Basándonos en resultados obtenidos en procedimientos de prueba y error, optamos por una arquitectura 25-40-1, es decir, 25 nodos en la capa de entrada, 40 en la capa oculta y 1 en la salida.

Las funciones de ajuste para cada una de las capas también fueron elegidas mediante la prueba y el error, y es que de la bibliografía consultada se desprende que no existe una configuración establecida para definir el número de capas, neuronas y funciones de la red. Todo dependerá de los datos que se procesen y de la complejidad del problema que la red tendrá que resolver. Nuestro algoritmo en particular, maneja muchos datos en la matriz de entrada y además la clasificación de latidos no es una tarea trivial. De hecho, para cada paciente en nuestra base de datos, y en general sucede esto, las señales varían, inclusive si tomamos dos señales en dos momentos diferentes del mismo sujeto, no son iguales. Trabajar con datos biológicos conlleva un reto más aún si la tarea involucra clasificación.

Tanto en la primera como en la última capa, elegimos la función "*tansig*" que toma valores entre [-1, 1] para abscisa y ordenada. En la capa oculta, la función de activación fue "*logsig*", que toma valores entre [0,1] para ordenada y [-1,1] para la variable independiente.

La Figura 12.2 muestra el esquema de la red neuronal implementada.

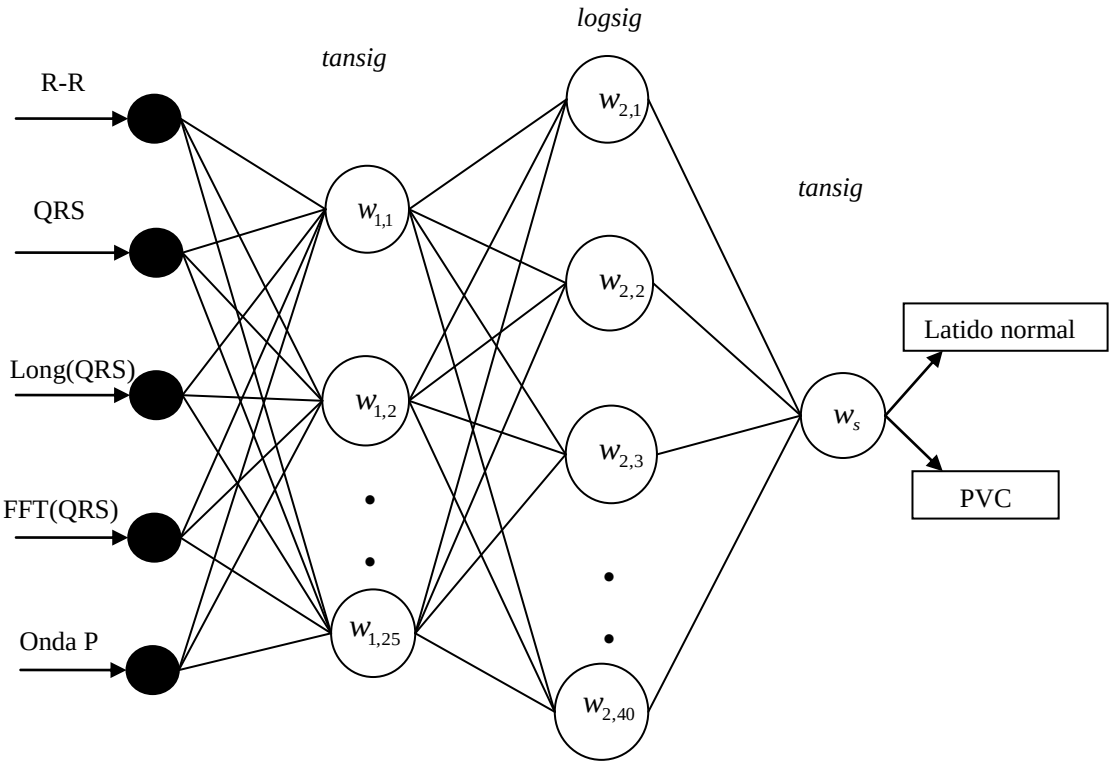

Figura 12.2 Representación gráfica de la red neuronal MLP implementada.

12.2.3. Entrenamiento

La red diseñada del modo descripto previamente fue entrenada con la señal #106 de la base de Arritmias del MIT-BIH (Capítulo 2), pues presenta un número representativo de PVCs y suficientes datos para el entrenamiento, prueba y validación. Del total de 2027 latidos cardíacos que presenta esta señal, 1507 son normales y 520 son anómalos.

Para la etapa de entrenamiento, aplicamos el método de validación cruzada de orden 3, descripto en el capítulo de redes neuronales. Este método divide en tres partes iguales el conjunto de datos para el entrenamiento y utiliza cada tercio del conjunto total para las tareas de entrenamiento prueba y validación. La característica particular que posee el método es que cruza los tres subconjuntos de datos, para que una vez que un subconjunto fue utilizado para el entrenamiento, en las siguientes dos iteraciones, sea utilizado como validación una vez y prueba otra vez. Es decir que todos los subconjuntos son utilizados como entrenamiento, validación y prueba. La iteración hace

que los pesos de las neuronas se ajusten de manera más eficiente y el proceso completo alcance un mínimo global y no caiga en un mínimo local.

Dentro de la definición de los parámetros para el entrenamiento de la red, se encuentra el número de veces que se producirán las iteraciones hasta alcanzar un mínimo global previamente establecido o bien hasta completar la cantidad de iteraciones. En el diseño de la red neuronal multicapas establecimos en 1000 el número de iteraciones. Otro parámetro a definir es el error aceptable entre los datos de entrada y la salida, fijamos este valor en cero.

La Figura 12.3 muestra la evolución del entrenamiento de nuestra red con validación cruzada. Podemos observar en la figura que si bien definimos en 1000 el número de iteraciones, el algoritmo de entrenamiento iteró menos de 35 veces, pues llegó a un error mínimo pre-establecido.

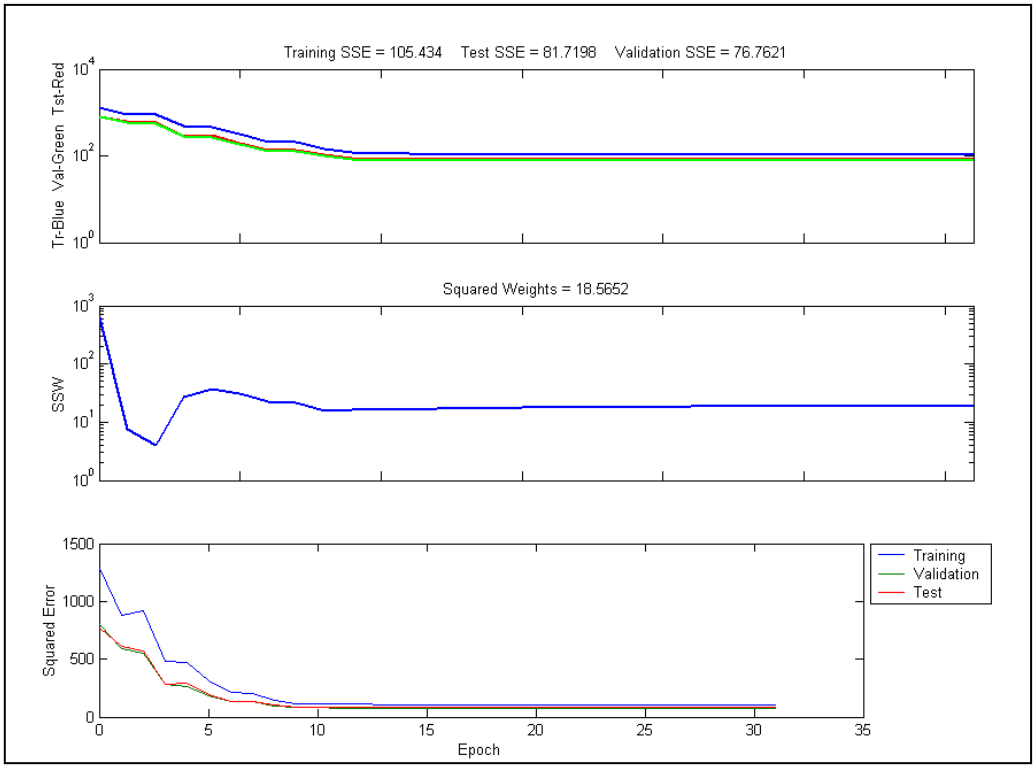

Figura 12.3 Evolución del entrenamiento con validación cruzada

12.2.4. Aplicación del Análisis de Componentes principales al vector de entradas

Un aspecto importante que incluimos durante la etapa de entrenamiento, es que todo el vector de entradas sufre una transformación previa al ingreso de la red. Los datos son transformados mediante el análisis de componentes principales, descripto en el capítulo 4, para obtener así, un vector de datos normalizados, y cuyas dimensiones están reducidas, como una consecuencia indirecta, el proceso de entrenamiento acorta sus tiempos. La decisión fue tomada debido a que trabajamos con un vector de datos de 103 columnas por cada latido y como lo explicamos en el capítulo 10 de aplicaciones de Análisis de Componentes Principales, hay mucha redundancia de información en la variable que contiene los valores de amplitud del complejo QRS, como así también la variable que contiene la transformada rápida de Fourier.

Otro aspecto importante es que los datos tienen distintas unidades de medida, algunas miden amplitud, otras tiempo y otras frecuencias, luego, nos pareció coherente aplicar algún tipo de estandarización para obtener datos normalizados.

Previo al entrenamiento propiamente dicho y una vez definido el vector de entradas a la red, los datos son pre-procesados de modo que su media sea cero y su desvío estándar uno. Es decir, pre-procesamos el conjunto de entrenamiento normalizando las entradas y las salidas para que tengan las características recién mencionadas. Como segundo paso, aplicamos el análisis de componentes principales. Este análisis transforma los datos de entrada de manera que los elementos del vector de entradas estén no-correlacionados. Como consecuencia directa, el tamaño de los vectores de entrada se ve reducido pues retiene sólo aquellos componentes que contribuyen más que una fracción especificada de la variación total en el conjunto de datos, 0,001 en nuestro caso.

El algoritmo, desarrollado en MATLAB®, utiliza la descomposición de valores singulares para obtener los componentes principales. Los vectores de entrada son multiplicados por una matriz cuyas filas consisten en los autovectores de la matriz de covarianza. Esto produce vectores de entrada transformados cuyos componentes están no-correlacionados y ordenados de acuerdo a la magnitud de su varianza. Los componentes que menos contribuyen a la varianza total son eliminados. En nuestro caso

particular, de un vector de entradas de 103 columnas, obtuvimos un vector de 31 valores.

Una vez que la red ha sido entrenada, la matriz transformación de los componentes principales, se usa para transformar cualquier entrada futura aplicada a la red. La matriz forma parte de la estructura de la red, al igual que los pesos y las conexiones entre neuronas.

Para evaluar la respuesta del proceso de entrenamiento, hicimos un post-proceso de los datos mediante una función que realiza una regresión lineal entre cada elemento de la respuesta de la red y su correspondiente target. La Figura 12.4 muestra la regresión lineal de nuestra red.

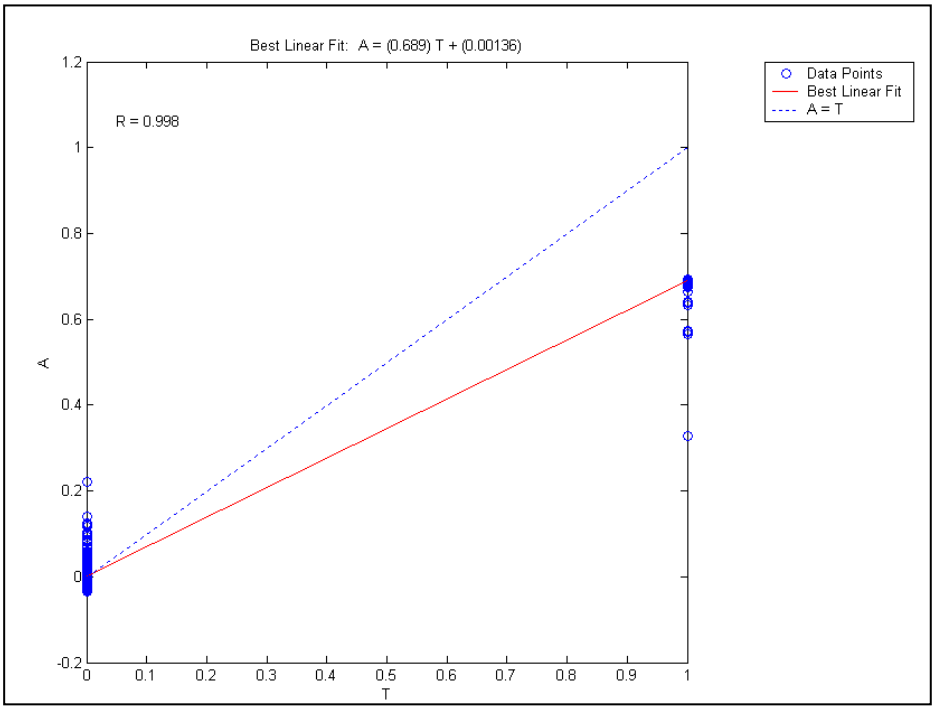

Figura 12.4. Gráfica de la regresión lineal entre cada elemento de respuesta de la red y su correspondiente target.

12.2.5. Resultados

Las Tabla 12.4 yTabla 12.5, separadas por series para una mejor visualización, muestran los resultados obtenidos con el perceptrón multicapa diseñado como clasificador de arritmias, sobre el conjunto las bioseñales experimentales presentadas en el capítulo 2.

Tabla 12.4 Resultados de la prueba de MLP con la serie 100 de MIT-BIH

Serie 100 Señales	MLP								
	Total Normales	Total PVC	TP	FP	FN	TN	S [%]	E [%]	PP [%]
100	2339	1	1	4	0	2173	100,00	99,82	20,00
104	2229	2	2	7	0	1702	100,00	99,59	22,22
105	2526	41	41	18	0	2423	100,00	99,26	69,49
107	2078	59	35	39	24	1938	59,32	98,03	47,30
108	555	6	4	9	2	494	66,67	98,21	30,77
109	840	11	3	12	8	716	27,27	98,35	20,00
111	639	81	81	0	0	697	100,00	100,00	100,00
114	519	34	34	20	0	502	100,00	96,17	62,96
116	756	40	40	9	0	747	100,00	98,81	81,63
118	2312	16	7	13	9	2195	43,75	99,41	35,00
119	1543	444	444	0	0	1543	100,00	100,00	100,00
123	505	1	1	4	0	465	100,00	99,15	20,00
124	502	20	18	24	2	479	90,00	95,23	42,86

Tabla 12.5 Resultados de la prueba de MLP con la serie 200 de MIT-BIH

Serie 200. Señales	MLP.								
	Total Normales	Total PVC	TP	FP	FN	TN	S [%]	E [%]	PP[%]
200	1775	826	705	14	121	1760	85,35	99,21	98,05
201	1802	198	187	53	20	1678	90,34	96,94	77,92
202	2117	19	15	8	4	2014	78,95	99,60	65,22
203	2536	444	415	10	29	2429	93,47	99,59	97,65
205	2585	71	71	8	0	2576	100,00	99,69	89,87
207	2227	105	82	48	23	2070	78,10	97,73	63,08
208	1961	992	300	105	692	1764	30,24	94,38	74,07
210	2455	194	184	43	10	2412	94,85	98,25	81,06
213	3031	220	188	151	32	2878	85,45	95,01	55,46
214	2003	256	236	20	36	1926	86,76	98,97	92,19
215	3199	164	164	52	0	3047	100,00	98,32	75,93
217	1942	162	146	103	16	1942	90,12	94,96	58,63
219	2223	64	58	21	6	1921	90,63	98,92	73,42
221	667	160	160	8	0	657	100,00	98,80	95,24
223	767	58	57	37	1	742	98,28	95,25	60,64
228	541	160	153	17	3	545	98,08	96,98	90,00
231	503	2	2	13	0	610	100,00	97,91	13,33
233	745	132	119	57	13	613	90,15	91,49	67,61

De los resultados obtenidos, podemos decir que el algoritmo tiene una performance aceptable, y que, en comparación con otros algoritmos publicados, está dentro de lo esperado. De la serie 100 destacamos señales como la #109 y #118 con performances más deficientes, sus índices de S y PP no superan el 50%, sin embargo el índice Especificidad alcanza valores por encima del 98,40%. Destacamos también señales como #111, #116 y #119 como buenos ejemplos del rendimiento del algoritmo clasificador. Con una exactitud del 100% para las señales #111 y #119.

El índice PP fue el que arrojó valores más bajos para todas las señales de esta serie, es decir que detectamos muchos falsos positivos, (latidos normales detectados como PVC). Una explicación a este comportamiento es el hecho de que en las señales analizadas no sólo se producen latidos normales y PVC, sino otro tipo de afecciones que se ven reflejadas en la señal cardíaca y que en este trabajo las consideramos como latidos normales, de lo contrario hubiéramos tenido mayor número de falsos negativos. Una solución a este inconveniente sería o bien ampliar el abanico de latidos a clasificar, es decir, incorporar nuevos parámetros que caractericen las demás afecciones y realizar una clasificación más ajustada sobre cada tipo de latido. La otra solución, más sencilla, sería probar nuestro algoritmo sólo con señales que contengan PVCs y latidos normales, de esta manera evaluaríamos la clasificación específica de los latidos ventriculares prematuros. De hecho, las señales #106 (utilizada para el entrenamiento), #111 y #119 tienen esta característica recién mencionada. Como fuera discutido, las tres señales tienen un rendimiento del 100% en los tres índices analizados, lo que nos da una idea de la buena clasificación de PVC de nuestra red neuronal.

Los resultados de la serie 200 de la base de datos son exhibidos en la Tabla 12.5. Comparando con la serie 100, vemos que obtuvimos mejores resultados en general. Destacamos las señales #200, #201, #203, #205, #210, #214, #221 y #228 como buenos ejemplos de performance de la red. Todos los valores de Especificidad para esta serie dieron por encima del 91%, lo que significa que el algoritmo rechaza correctamente los latidos normales. La señal #208 con un valor de Sensibilidad bajo, alrededor del 30% y la señal #231 con un índice de Predictividad Positiva alrededor del 13%, son las señales con performance más deficiente. Cabe destacar que la señal #208 es una señal difícil de analizar, hasta para el ojo humano, por la variedad de afecciones que presenta.

El Algoritmo en entornos ruidosos

Al igual que con Bancos de Filtros Digitales, hemos tomado sólo la señal #119 como ejemplo representativo de un registro de arritmia ventricular para testear el algoritmo en entornos ruidosos, con interferencias frecuentes en la captura de bioseñales. Las señales de ruido comprenden ruido de base (bw), de movimiento de electrodos (em) y de contracciones musculares (ma), con variadas relaciones señal-ruido (24 dB a -6 dB), como ya lo explicáramos con anterioridad. En la Tabla 12.6 indicamos

los índices de performance del algoritmo detector de PVC implementado con la misma red neuronal de arquitectura MLP descripta en el punto 12.2 (Diseño de la Red Neuronal Nítida).

Tabla 12.6. Resultados de la prueba de MLP con la señal #119 con ruido agregado

Señales	MLP						
	TP	FP	FN	TN	S [%]	E [%]	PP [%]
119bw_6	131	58	313	1483	29,50	96,24	69,31
119bw00	215	41	229	1500	48,42	97,34	83,98
119bw06	300	18	144	1523	67,57	98,83	94,34
119bw12	333	8	111	1533	75,00	99,48	97,65
119bw18	366	0	78	1541	82,43	100,00	100,00
119bw24	423	0	21	1541	95,27	100,00	100,00
119em_6	136	132	308	1409	30,63	91,43	50,75
119em00	179	106	265	1435	40,32	93,12	62,81
119em06	230	71	214	1470	51,80	95,39	76,41
119em12	350	29	94	1512	78,83	98,12	92,35
119em18	425	3	21	1538	95,29	99,81	99,30
119em24	441	0	3	1541	99,32	100,00	100,00
119ma_6	68	103	376	1438	15,32	93,32	39,77
119ma00	142	94	302	1447	31,98	93,90	60,17
119ma06	244	61	200	1480	54,95	96,04	80,00
119ma12	288	25	156	1516	64,86	98,38	92,01
119ma18	382	2	62	1539	86,04	99,87	99,48
119ma24	437	7	0	1541	100,00	99,55	98,42

Los valores de Especificidad para todas las señales ruidosas demuestran una muy buena tasa de rechazo de latidos que no son considerados PVCs. Observamos también

de la Tabla 12.6, que a medida que el nivel de ruido decrece, la performance de la red clasificadora mejora muy notablemente. Los valores de falsos positivos se reducen al igual que los valores de falsos positivos.

En entornos muy ruidosos, con niveles de ruido de -6dB a 6dB el algoritmo omite la detección de PVCs.

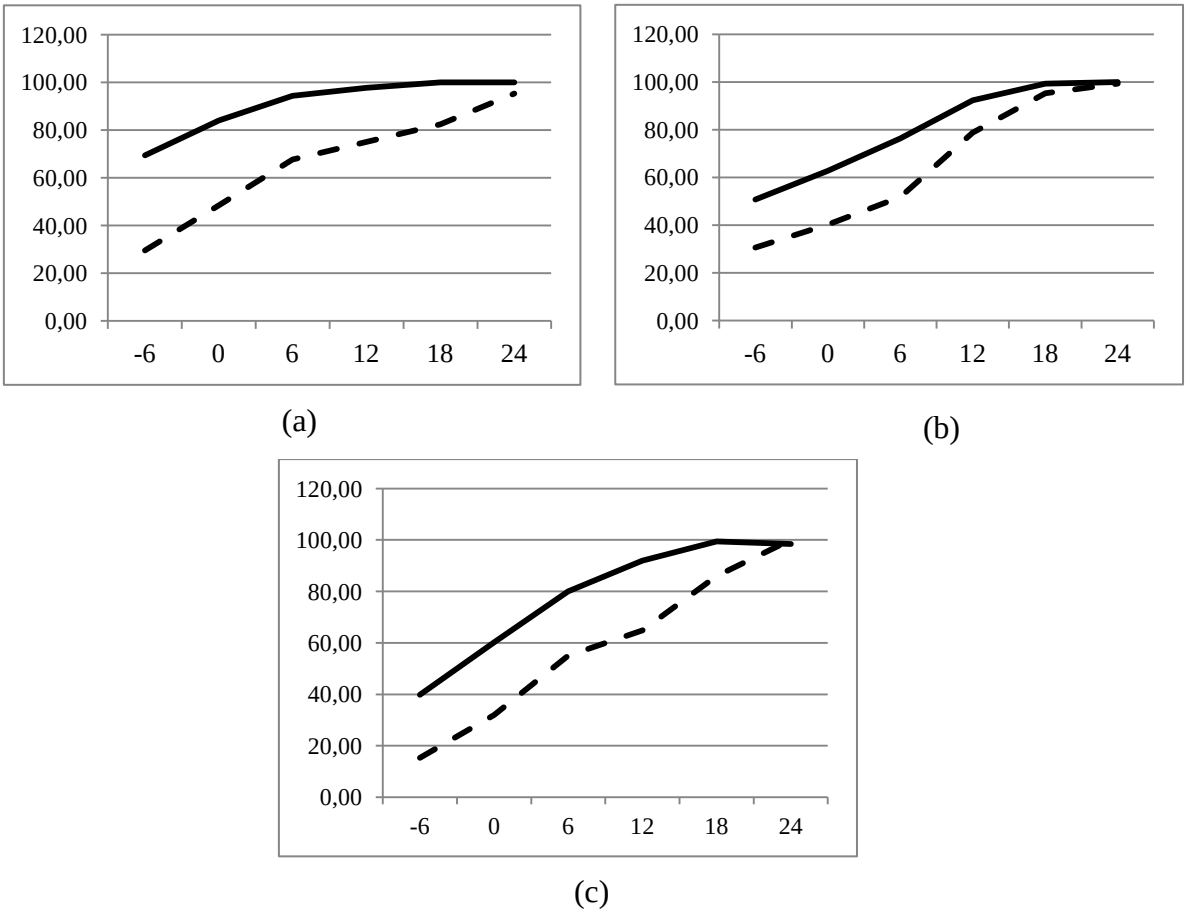

Figura 12. 5. Evolución de la performance de la Red diseñada en entornos ruidosos. Línea sólida: PP. Línea Punteada: S. (a) bw, (b) em, (c) ma.

En la Figura 12. 5 graficamos las distintas performances del algoritmo clasificador en entornos ruidosos. Un gráfico por cada tipo de ruido. Los índices representados son: en línea de puntos, la Sensibilidad y en línea continua, la Predictividad Positiva. En el eje de las abscisas aparecen los niveles de dB, de más ruidoso (-6dB) a menos ruidoso (24dB). En los tres gráficos observamos la misma evolución, lo que era de esperarse, a medida que el ruido disminuye, los índices que evalúan el rendimiento de la red neuronal, aumentan. Todos ellos y con todos los ruidos, alcanzan el 100% o bien están

muy próximos a este valor, con niveles de ruido bajo. El algoritmo entonces, se ve afectado con la presencia de ruido.

12.3. La Red Difusa

12.3.1. Diseño, arquitectura y entradas.

Al igual que los otros procesos descriptos, implementamos nuestro modelo de Red Neuronal difusa con MATLAB®. El diseño se basa en la arquitectura ANFIS, descripta en el capítulo 6 y representada en la Figura 6.13.

En el caso de la red difusa, incorporamos las siguientes entradas:

a) la frecuencia instantánea, distancia R-R;
b) la longitud en muestras del complejo QRS;
c) la detección o no de la onda P.

Estas entradas fueron seleccionadas del conjunto de entradas de la red neuronal nítida. La diferencia entre un conjunto y otro, se basa en las limitaciones propias de ANFIS, que permite manejar un vector de entradas de pocas dimensiones, de lo contrario el algoritmo diverge.

Por cada función de entrada elegimos 2 funciones de membresía del tipo campana. Por cada función, dispusimos de tres reglas difusas. Esta arquitectura así conformada es el resultado de experimentaciones con varias combinaciones de funciones de membresía, tipos de funciones y número de reglas por función. Como el algoritmo debe clasificar entre un latido normal y una PVC, optamos una salida constante, que es el modelo del tipo Sugeno de orden cero, descripto en el capítulo 6.

Al igual que con la Red nítida, utilizamos la señal #106 como señal de entrenamiento.

Los resultados son mostrados a continuación y luego discutidos.

12.3.2 Resultados y Discusión.

Las Tabla 12.7 y Tabla 12.8, correspondientes a la serie 100 y serie 200 respectivamente, muestran los resultados obtenidos con la Red Neuronal Difusa diseñada como clasificadora de arritmias, sobre el conjunto las bioseñales experimentales presentadas en el capítulo 2.

Tabla 12.7 Resultados de la prueba de FNN con señales del MIT-BIH. Serie 100

Serie 100. Señal	FNN								
	Total Normales	Total PVC	TP	FP	FN	TN	S [%]	E [%]	PP [%]
100	2339	1	1	29	0	2310	100,00	98,76	3,33
104	2229	2	2	9	0	2220	100,00	99,60	18,18
105	2526	41	41	87	0	2439	100,00	96,56	32,03
107	2078	59	54	72	5	2006	91,53	96,54	42,86
108	555	6	4	45	2	510	66,67	91,89	8,16
109	840	11	8	61	3	779	72,73	92,74	11,59
111	639	81	1	69	0	650	100,00	90,40	1,43
114	519	34	29	12	5	507	85,29	97,69	70,73
116	756	40	21	25	19	731	52,50	96,69	45,65
118	2312	16	8	13	8	2299	50,00	99,44	38,10
119	1543	444	374	18	70	1525	84,23	98,83	95,41
123	505	1	1	91	0	414	100,00	81,98	1,09
124	502	20	15	37	5	465	75,00	92,63	28,85

Tabla 12.8 Resultados de la prueba de FNN con señales del MIT-BIH. Serie 200

Serie 200. Señal	FNN								
	Total Normales	Total PVC	TP	FP	FN	TN	S [%]	E [%]	PP [%]
200	1775	826	590	145	236	1630	71,43	91,83	80,27
201	1802	198	141	21	57	1781	71,21	98,83	87,04
202	2117	19	13	21	16	2086	44,83	99,00	38,24
203	2536	444	428	119	16	2417	96,40	95,31	78,24
205	2585	71	59	51	12	2534	83,10	98,03	53,64
207	2227	105	81	29	24	2198	77,14	98,70	73,64
208	1961	992	891	357	101	1604	89,82	81,80	71,39
210	2455	194	175	65	19	2390	90,21	97,35	72,92
213	3031	220	203	63	17	2968	92,27	97,92	76,32
214	2003	256	240	49	16	1954	93,75	97,55	83,04
215	3199	164	144	72	20	3127	87,80	97,75	66,67
217	1942	162	121	97	41	1845	74,69	95,01	55,50
219	2223	64	55	69	9	2154	85,94	96,90	44,35
221	667	160	150	123	10	544	93,75	81,56	54,95
223	767	58	43	41	17	724	71,67	94,64	51,19
228	541	160	136	27	20	518	87,18	95,05	83,44
231	503	2	2	5	0	498	100,00	99,01	28,57
233	745	132	248	103	23	503	91,51	83,00	70,66

Tanto en la serie 100 como en la serie 200 observamos una tendencia similar entre los valores de índices de performance. Para ambas tablas, el índice de Especificidad tiene valores cercanos al 100%, es decir que la tasa de rechazo de los latidos normales ha alcanzado prácticamente su máximo. Observamos también, al igual que con la red nítida, una alto número de falsos positivos detectados, lo que afecta directamente al índice Predictividad Positiva. Con respecto al índice sensibilidad que evalúa el número

de falsos negativos detectados, observamos que la serie 200 ha tenido una mejor performance, con un rango de variabilidad que va desde el 71,21% para la señal #201 hasta el 100% para la señal #231. Sin embargo, si bien en la serie 100 aparecen señales como #108, #116 y #118 con índices del 66,67%, 52,50% y 50% respectivamente, señales como #100, #104, #105, #111 y #123 tiene un índice de sensibilidad del 100%, característica que observamos sólo en la señal #231 de la serie 200.

El Algoritmo en entornos ruidosos

De igual modo que con la red nítida, probamos nuestra red difusa con la señal #119 con ruido agregado. Los ruidos y niveles de ruidos han sido ya ampliamente descriptos en secciones anteriores.

En la Tabla 12.6 indicamos los índices de performance del algoritmo detector de PVC implementado con la red difusa.

Tabla 12.9 Resultados de la prueba de FNN con la señal #119 ruidosa

Señales	FNN						
	TP	FP	FN	TN	S [%]	E [%]	PP [%]
119bw_6	51	225	393	1328	11,49	85,51	18,48
119bw00	55	257	389	1296	12,39	83,45	17,63
119bw06	62	151	382	1402	13,96	90,28	29,11
119bw12	290	114	154	1439	65,32	92,66	71,78
119bw18	314	86	130	1467	70,72	94,46	78,50
119bw24	357	21	87	1532	80,41	98,65	94,44
119em_6	149	301	295	1252	33,56	80,62	33,11
119em00	203	283	241	1270	45,72	81,78	41,77
119em06	251	256	193	1297	56,53	83,52	49,51
119em12	344	201	100	1352	77,48	87,06	63,12
119em18	357	81	87	1472	80,41	94,78	81,51
119em24	369	32	75	1521	83,11	97,94	92,02
119ma_6	87	295	357	1258	19,59	81,00	22,77
119ma00	93	175	351	1378	20,95	88,73	34,70
119ma06	199	95	245	1458	44,82	93,88	67,69
119ma12	269	95	175	1458	60,59	93,88	73,90
119ma18	353	88	91	1465	79,50	94,33	80,05
119ma24	366	37	78	1621	82,43	97,62	90,82

En entornos ruidosos, nuestra red difusa tiene un comportamiento similar a la red MLP descripta en el punto 12.2, esto es, a medida que el ruido disminuye, los índices de performance aumentan, demostrando un mejor rendimiento del algoritmo. Esta tendencia se mantiene independientemente del ruido evaluado.

13. Algoritmos Genéticos

13.1. Motivación

La rápida expansión de la inteligencia artificial, en particular las redes neuronales y los algoritmos genéticos, está provocando una nueva revolución en el área de la informática.

Los algoritmos computacionales que involucran inteligencia artificial, han sido de gran interés para la comunidad científica, en especial aquellos útiles en la clasificación y detección de latidos cardíacos. Entre ellos, las mencionadas redes neuronales (capítulo 6) y los algoritmos genéticos.

Al igual que las redes neuronales, los algoritmos genéticos están inspirados en la biología: se basan en la teoría de la evolución genética y en el concepto de la supervivencia del más apto. Se utilizan fundamentalmente en la resolución de problemas de búsqueda y de optimización.

Los algoritmos evolutivos representan algoritmos de búsqueda robusta y efectivos que permiten rápidamente localizar áreas de soluciones de alta calidad aun si el espacio de búsqueda es amplio y complejo. Esta cualidad hace que estos algoritmos se ajusten bien al diseño y entrenamiento de una RNA, donde el espacio de búsqueda es infinito y de grandes dimensiones.

El uso de algoritmos evolutivos como un procedimiento de búsqueda global, que asiste al diseño y entrenamiento de RNA es uno de los sistemas híbridos que más interesan en la investigación actual, especialmente debido a que el cerebro humano es el resultado de la evolución biológica, (Branke, 1995).

La aplicación desarrollada y descripta en este capítulo se basa en la combinación de dos técnicas utilizadas en la construcción de sistemas inteligentes, los algoritmos evolutivos y las redes neuronales, para crear un sistema donde, tras definir una topología de red neuronal, los primeros sean utilizados para entrenar a las segundas, en reemplazo de los

tradicionales algoritmos de entrenamiento, con el propósito de obtener un algoritmo que pueda distinguir entre latidos normales y latidos ectópicos prematuros.

13.2. El Algoritmo Computacional

El algoritmo computacional global fue diseñado e implementado en MATLAB® y consiste en:

- Partir de la red neuronal MLP descripta en el capítulo 12, con la misma topología y con las mismas entradas.
- Utilizar los pesos obtenidos de la red entrenada como cromosomas de los algoritmos genéticos.
- Los operadores genéticos que intervienen para lograr la supervivencia del más apto, ajustan los cromosomas para obtener resultados más acertados en la simulación de datos desconocidos.
- Simular la red optimizada con las señales de ECG seleccionadas para tal fin.
- Obtener los índices de performance

13.3. El Proceso de Optimización

Para entrenar la red neuronal utilizamos codificación de valores reales, lo que significa que cada peso de la red representa un cromosoma.

Cada cromosoma produce un error sobre todos los patrones de entrenamiento. El objetivo es minimizar este error.

El proceso se inicia con una población aleatoria; y se construyen generaciones sucesivas utilizando operadores genéticos de selección, cruza y mutación.

- El operador de selección elige los individuos de la población que tengan mayor grado de adaptación.

- El operador de cruza recombina el material genético de los individuos de la población para producir nuevos individuos.

- El operador de mutación altera las características de algunos individuos para garantizar la diversidad de la población.

Realizamos el entrenamiento en función del tamaño de la población, la cual fue variando entre 20, 25 y 30; y también en el número de generaciones y una vez encontrado al conjunto de pesos óptimo, el reconocimiento se realiza rápidamente.

Este proceso se repite hasta que alcanzamos el número de generaciones, al final obtendremos a un individuo que tendrá el conjunto de pesos para la red neuronal que mejor reconocerá los valores de entrenamiento.

13.4. Resultados y Discusión.

Los resultados obtenidos se exhiben en las Tabla 13.1Tabla 13.2 y Tabla 13.3. Los índices utilizados para la evaluación de la performance del algoritmo son los ya descriptos en el capítulo 8 y utilizados en la evaluación de las redes neuronales diseñadas en el capítulo 12.

Tabla 13.1 Resultados de la prueba de MLP con Algoritmos Genéticos. Serie 100

Serie 100. Señal	MLP con Algoritmos Genéticos. Serie 100								
	Total Normales	Total PVC	TP	FP	FN	TN	S [%]	E [%]	PP [%]
100	2339	1	1	4	0	2336	100,00	99,83	20,00
104	2229	2	2	6	0	2225	100,00	99,73	25,00
105	2526	41	41	10	0	2557	100,00	99,61	80,39
107	2078	59	40	21	19	2116	67,80	99,02	65,57
108	555	6	5	3	1	558	83,33	99,47	62,50
109	840	11	9	7	2	844	81,82	99,18	56,25
111	639	81	81	0	0	720	100,00	100,00	100,00
114	519	34	34	13	0	540	100,00	97,65	72,34
116	756	40	40	5	0	791	100,00	99,37	88,89
118	2312	16	11	7	5	2321	68,75	99,70	61,11
119	1543	444	444	0	0	1987	100,00	100,00	100,00
123	505	1	1	2	0	504	100,00	99,60	33,33
124	502	20	20	14	0	508	100,00	97,32	58,82

Tabla 13.2 Resultados de la prueba de MLP con Algoritmos Genéticos. Serie 200

Serie 200. Señales	MLP con Algoritmos Genéticos. Serie 200								
	Total Normales	Total PVC	TP	FP	FN	TN	S [%]	E [%]	PP [%]
200	1775	826	782	8	44	2593	94,67	99,69	98,99
201	1802	198	201	36	-3	1964	101,52	98,20	84,81
202	2117	19	19	5	0	2131	100,00	99,77	79,17
203	2536	444	430	7	14	2973	96,85	99,77	98,40
205	2585	71	71	3	0	2653	100,00	99,89	95,95
207	2227	105	92	30	13	2302	87,62	98,71	75,41
208	1961	992	403	97	589	2856	40,63	96,72	80,60
210	2455	194	190	26	4	2623	97,94	99,02	87,96
213	3031	220	203	104	17	3147	92,27	96,80	66,12
214	2003	256	248	13	8	2246	96,88	99,42	95,02
215	3199	164	164	20	0	3343	100,00	99,41	89,13
217	1942	162	152	81	10	2023	93,83	96,15	65,24
219	2223	64	63	10	1	2277	98,44	99,56	86,30
221	667	160	160	2	0	825	100,00	99,76	98,77
223	767	58	58	16	0	809	100,00	98,06	78,38
228	541	160	156	6	4	695	97,50	99,14	96,30
231	503	2	2	4	0	501	100,00	99,21	33,33
233	745	132	131	38	1	839	99,24	95,67	77,51

Tabla 13.3 Resultados de la prueba de MLP con Algoritmos Genéticos con la señal #119 ruidosa

Señales	MLP con Algoritmos Genéticos						
	TP	FP	FN	TN	S [%]	E [%]	PP [%]
119bw_6	138	45	306	1942	31,08	97,74	75,41
119bw00	238	12	206	1975	53,60	99,40	95,20
119bw06	325	1	119	1986	73,20	99,95	99,69
119bw12	444	0	0	1987	100,00	100,00	100,00
119bw18	444	0	0	1987	100,00	100,00	100,00
119bw24	444	0	0	1887	100,00	94,97	81,62
119em_6	186	100	258	1902	41,89	95,72	68,63
119em00	201	85	243	1934	45,27	97,33	79,13
119em06	250	53	194	1971	56,31	99,19	93,98
119em12	326	16	118	1987	73,42	100,00	100,00
119em18	444	0	0	1987	100,00	100,00	100,00
119em24	444	0	0	1903	100,00	95,77	84,09
119ma_6	201	84	243	1915	45,27	96,38	73,63
119ma00	259	72	185	1937	58,33	97,48	83,82
119ma06	316	50	128	1971	71,17	99,19	95,18
119ma12	386	16	58	1987	86,94	100,00	100,00
119ma18	444	0	0	1987	100,00	100,00	100,00
119ma24	444	0	0	1987	100,00	100,00	100,00

De las tablas resultados se desprende que la tendencia de nuestra red neuronal MLP de ser eficiente en la detección de latidos ventriculares prematuros se mantiene. Gracias a la incorporación de algoritmos genéticos para ajustar los pesos de la red neuronal una vez entrenada la red los índices de performance han mejorado. La optimización más notoria se refleja en la Tabla 13.3 de la señal #119 con ruido agregado, es decir que para entornos

ruidosos, la incorporación de los algoritmos evolutivos para el ajuste de pesos de una red neuronal podría mejorar la eficiencia de la red.

La tendencia de detectar muchos falsos positivos de nuestro algoritmo clasificador presente tanto en la red MLP nítida como en la red difusa, también se refleja en la red optimizada con algoritmos genéticos.

Si bien la mejora de los índices no es substancial, gracias a la incorporación de la computación evolutiva, todas las señales mostraron un incremento en sus valores de indicadores de rendimiento del algoritmo, incluso algunas de ellas alcanzaron el 100% de performance.

Tabla 13.4 Comparación de la performance de los Sistemas Neuronales con Algoritmos Genéticos. Serie 100

Serie 100. Señales	Sin AG			Con AG		
	S [%]	E [%]	PP [%]	S [%]	E [%]	PP [%]
100	100,00	99,82	20,00	100,00	99,83	20,00
104	100,00	99,59	22,22	100,00	99,73	25,00
105	100,00	99,26	69,49	100,00	99,61	80,39
107	59,32	98,03	47,30	67,80	99,02	65,57
108	66,67	98,21	30,77	83,33	99,47	62,50
109	27,27	98,35	20,00	81,82	99,18	56,25
111	100,00	100,00	100,00	100,00	100,00	100,00
114	100,00	96,17	62,96	100,00	97,65	72,34
116	100,00	98,81	81,63	100,00	99,37	88,89
118	43,75	99,41	35,00	68,75	99,70	61,11
119	100,00	100,00	100,00	100,00	100,00	100,00
123	100,00	99,15	20,00	100,00	99,60	33,33
124	90,00	95,23	42,86	100,00	97,32	58,82

Tabla 13.5 Comparación de la performance de los Sistemas Neuronales con Algoritmos Genéticos. Serie 200

Señales	Sin AG			Con AG		
	S [%]	E [%]	PP [%]	S [%]	E [%]	PP [%]
200	85,35	99,21	98,05	94,67	99,69	98,99
201	90,34	96,94	77,92	101,52	98,20	84,81
202	78,95	99,60	65,22	100,00	99,77	79,17
203	93,47	99,59	97,65	96,85	99,77	98,40
205	100,00	99,69	89,87	100,00	99,89	95,95
207	78,10	97,73	63,08	87,62	98,71	75,41
208	30,24	94,38	74,07	40,63	96,72	80,60
210	94,85	98,25	81,06	97,94	99,02	87,96
213	85,45	95,01	55,46	92,27	96,80	66,12
214	86,76	98,97	92,19	96,88	99,42	95,02
215	100,00	98,32	75,93	100,00	99,41	89,13
217	90,12	94,96	58,63	93,83	96,15	65,24
219	90,63	98,92	73,42	98,44	99,56	86,30
221	100,00	98,80	95,24	100,00	99,76	98,77
223	98,28	95,25	60,64	100,00	98,06	78,38
228	98,08	96,98	90,00	97,50	99,14	96,30
231	100,00	97,91	13,33	100,00	99,21	33,33
233	90,15	91,49	67,61	99,24	95,67	77,51

Las Tabla 13.4 y Tabla 13.5 muestran la comparación de las 3 series de señales analizadas. Claramente se ve el aumento de los índices de performance con la incorporación de los algoritmos genéticos para el ajuste de pesos de una red neuronal.

Siguiendo la metodología empleada en las aplicaciones de Redes Neuronales y Banco de Filtros, probamos el algoritmo en entornos ruidosos. La señal elegida fue la #119 para continuar con el mismo criterio y poder entonces realizar una comparación lo más objetiva posible.

Tabla 13.6 Comparación de la performance de los Sistemas Neuronales con Algoritmos Genéticos. Señal #119 ruidosa

Señales	Sin AG			Con AG		
	S [%]	E [%]	PP [%]	S [%]	E [%]	PP [%]
119bw_6	29,50	96,24	69,31	31,08	97,74	75,41
119bw00	48,42	97,34	83,98	53,60	99,40	95,20
119bw06	67,57	98,83	94,34	73,20	99,95	99,69
119bw12	75,00	99,48	97,65	100,00	100,00	100,00
119bw18	82,43	100,00	100,00	100,00	100,00	100,00
119bw24	95,27	100,00	100,00	100,00	94,97	81,62
119em_6	30,63	91,43	50,75	41,89	95,72	68,63
119em00	40,32	93,12	62,81	45,27	97,33	79,13
119em06	51,80	95,39	76,41	56,31	99,19	93,98
119em12	78,83	98,12	92,35	73,42	100,00	100,00
119em18	95,29	99,81	99,30	100,00	100,00	100,00
119em24	99,32	100,00	100,00	100,00	95,77	84,09
119ma_6	15,32	93,32	39,77	45,27	96,38	73,63
119ma00	31,98	93,90	60,17	58,33	97,48	83,82
119ma06	54,95	96,04	80,00	71,17	99,19	95,18
119ma12	64,86	98,38	92,01	86,94	100,00	100,00
119ma18	86,04	99,87	99,48	100,00	100,00	100,00
119ma24	100,00	99,55	98,42	100,00	100,00	100,00

14. Conclusiones

14.1. Análisis Estadístico

Para evaluar la performance de los algoritmos implementados a fin de identificar latidos ectópicos, además de los índices ya mencionados, Sensibilidad, Especificidad y Predictividad Positiva, realizamos un análisis estadístico descriptivo que clarifica los resultados obtenidos y nos permitirá una comparación cuantitativa de ellos.

La estadística utilizada para tal fin tiene en cuenta medidas de tendencia central y de dispersión. Comparamos también gráficamente el desempeño de cada índice de performance en cada una de las señales. A continuación, las figuras y la discusión correspondiente.

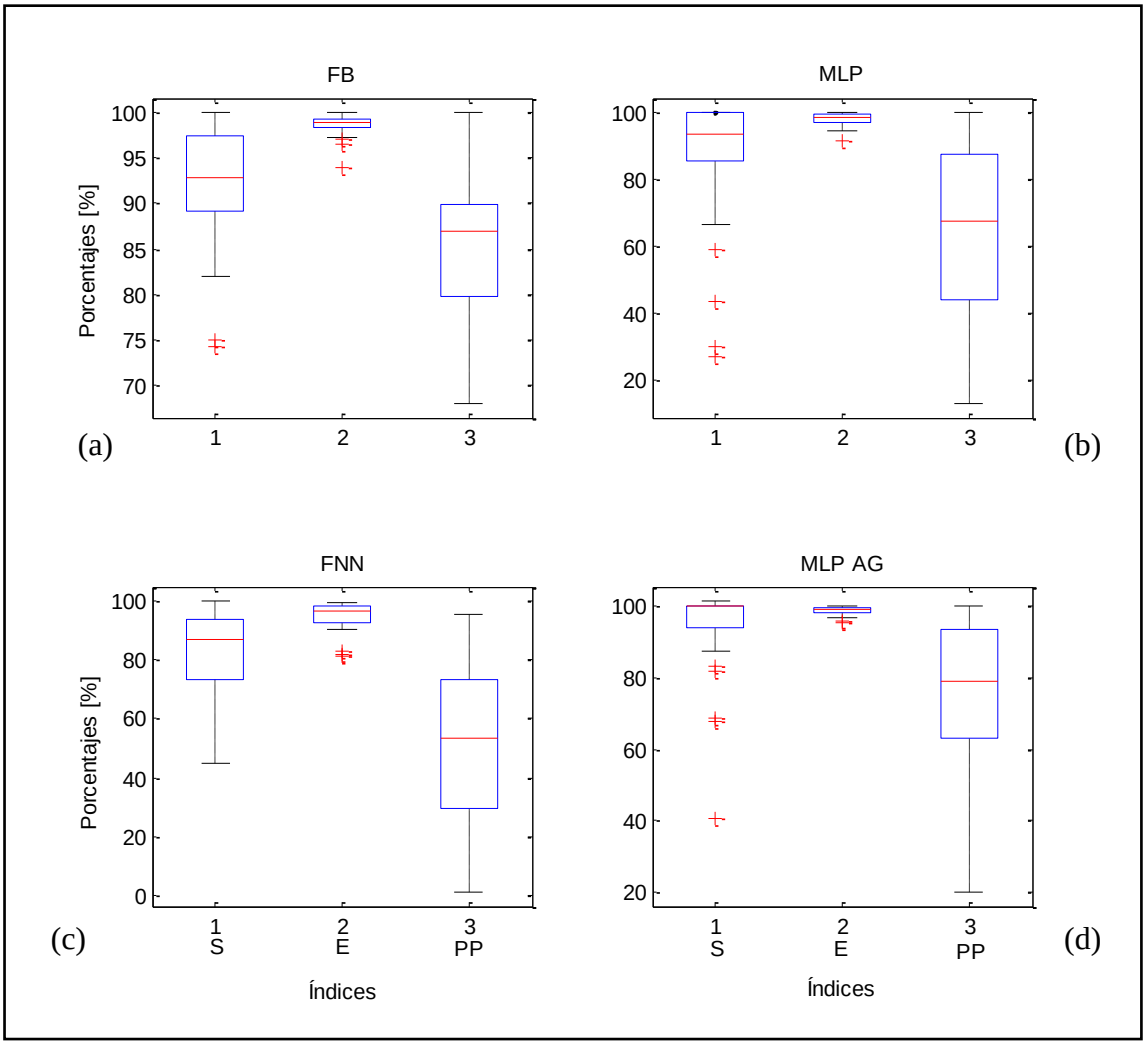

Figura 14.1 Boxplot base de datos completa. (a) Banco de Filtros Digitales. (b) Red Neuronal MLP. (c) Red Neuronal Difusa. (d) Red Neuronal MLP con Algoritmos Genéticos.

La Figura 1 muestra los gráficos boxplot de los índices de performance para cada uno de los algoritmos implementados. Este gráfico asocia las cinco medidas de dispersión, mediana, primer cuartil, tercer cuartil, valor máximo y valor mínimo que en general se trabajan por separado.

En la Figura 14.1(a), se observa la distribución de los valores para el Banco de Filtros implementado. El índice PP (Predictividad Positiva) presenta la mayor dispersión de los datos, sin valores atípicos, sin embargo el largo de los bigotes, en ambos extremos, representa un rango amplio de valores. La mediana, representada por la línea roja que divide en dos el rectángulo, nos indica que el 50% de los datos está por encima del 87% de performance. Los límites superior e inferior del rectángulo representan los cuartiles 3 y 1 respectivamente. Claramente podemos observar que el

75% de los resultados obtenidos con este método tienen entre un 80% y 90% de PP, lo cual es altamente aceptable si consideramos que de los tres índices, es el menos eficiente. Un comportamiento similar se observa en todos los métodos evaluados, sin embargo, lo que los distingue es el rango de valores que toman los datos, siendo la red neuronal difusa (FNN) la más deficiente con un rango de valores que van desde 1% para la señal #123 a 95% para la #119.

Para todos los algoritmos implementados, el índice especificidad (E), es el que mejor performance alcanzó, con más del 90% de efectividad en todos los casos, siendo los más eficientes el Banco de Filtros y la Red Neuronal MLP con Algoritmos Genéticos.

Los valores más altos de sensibilidad (S) fueron alcanzados por las redes neuronales nítidas, la MLP y la MLP con AG, Figura 14.1(b) y Figura 14.1(c). Ambas muestran poca dispersión de los datos, si bien existen algunos valores atípicos. Otro aspecto positivo a destacar es que el 75% de los datos está por encima del 85% en el primer caso y por encima del 95% en el segundo.

La Figura 14.2 muestra los índices de performance de los cuatro paradigmas para la base de datos ruidosa, esto es, la señal #119 con ruido agregado. Claramente, el Banco de Filtros Digitales, Figura 14.2(a), es el algoritmo que mejor desempeño demostró en entornos ruidosos. Los tres índices de performance tuvieron valores por encima del 94% de efectividad, salvo en un 25% de los datos para la S, que resultó ser por encima del 88% en el límite inferior que es el caso del ruido de línea de base con un nivel de -6db.

De las tres redes neuronales implementadas, podemos observar que la más eficiente resultó ser la MLP con AG, Figura 14.2(d) con mejores índices de performance, la E dio valores de más del 90% de efectividad y la PP más del 80% de efectividad para más del 75% de las señales de la tabla, lo cual es altamente aceptable. El índice S resultó tener mayor dispersión de los datos para este caso, con un rango que va desde el 30% de efectividad, aproximadamente, hasta el 100% para las señales menos ruidosas. Tanto la red MLP, Figura 14.2(b), como la red difusa, Figura 14.2(c), demostraron tener mucha dispersión en los datos cuando se evalúan la Sensibilidad y la Predictividad.

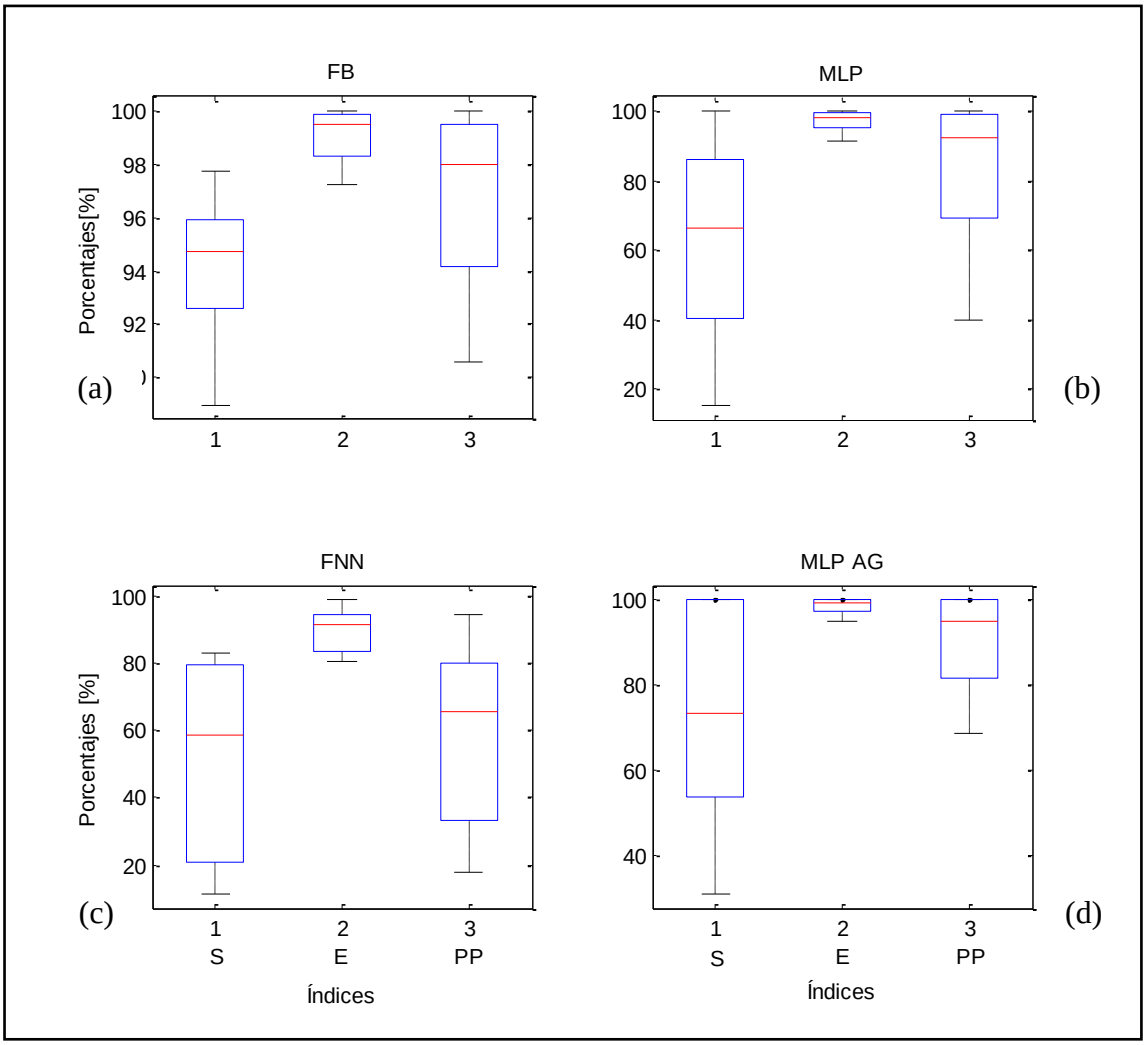

Figura 14.2 Boxplot señal #119 con ruido agregado. (a) Banco de Filtros Digitales. (b) Red Neuronal MLP. (c) Red Neuronal Difusa. (d) Red Neuronal MLP con Algoritmos Genéticos.

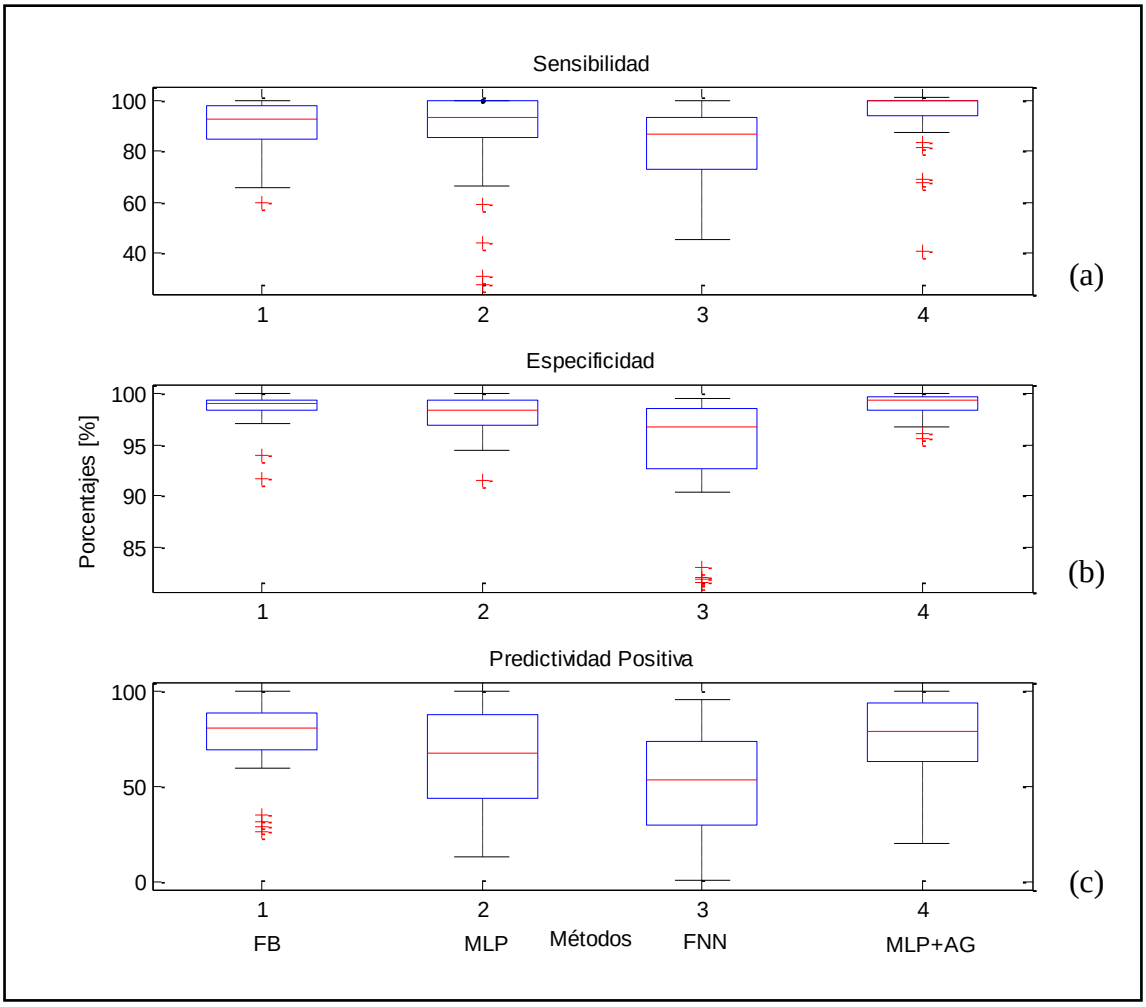

Figura 14.3 Boxplot base de datos completa. (a) Sensibilidad. (b) Especificidad. (c) Predictividad Positiva

Las Figura 14.3 y Figura 14.4 exhiben los mismos resultados que las Figura 1 y Figura 14.2, agrupando en cada una los cuatro métodos analizados. Estos gráficos fueron agregados para una mayor clarificación de los valores de performance de cada uno de los algoritmos. En estas figuras puede verse que tanto el Banco de Filtros Digitales como la Red MLP con AG han tenido un desempeño similar, poca variabilidad de los datos e índices con valores en el mismo rango. Aquí también puede observarse que el Banco de Filtros diseñado produce menos falsos positivos que cualquiera de las Redes Neuronales.

En la Figura 14.4 se destaca la supremacía del Banco de Filtros en la detección de PVC por sobre las redes neuronales. Si bien la red MLP con AG ha tenido una muy buena performance, el FB resultó ser más robusto en casos de señales ruidosas, con más del 90% de efectividad en la gran mayoría de los casos.

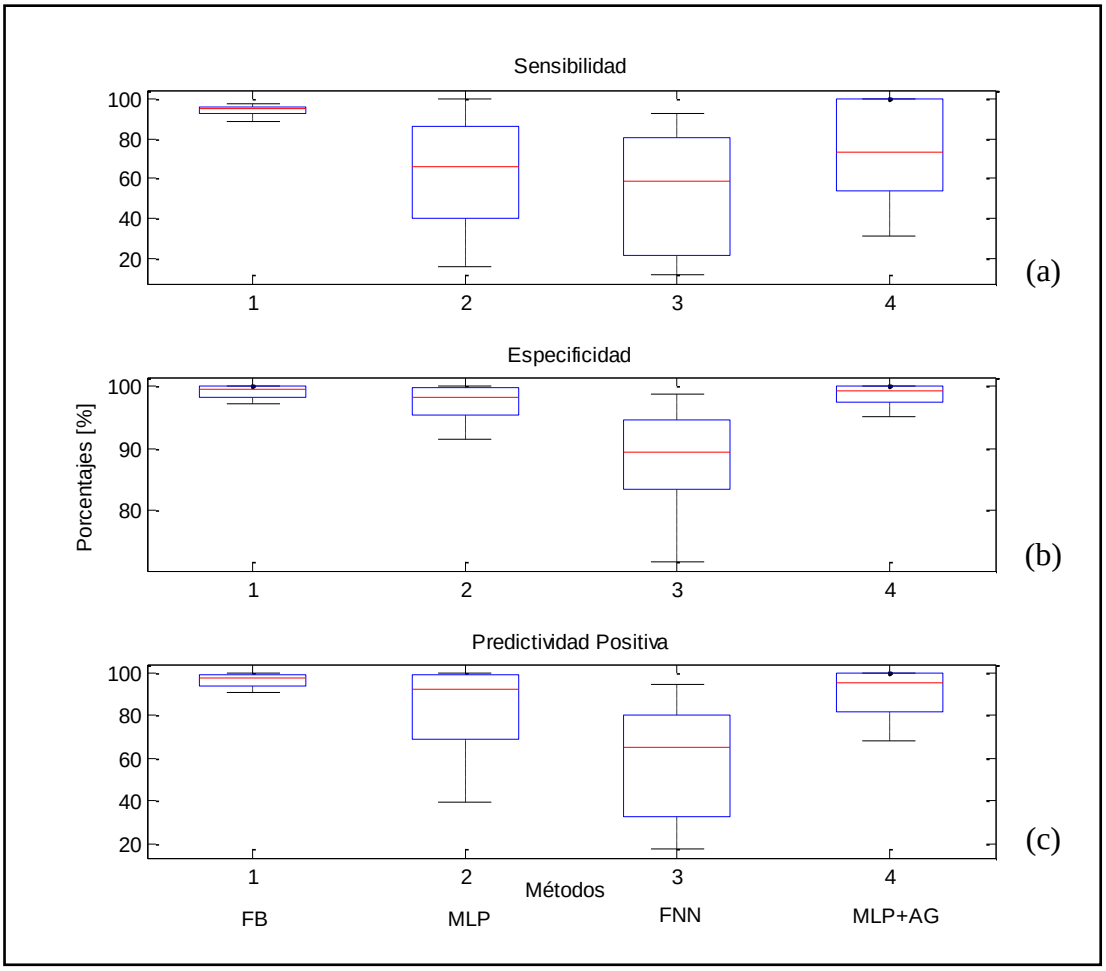

Figura 14.4 Boxplot Señal #119 con ruido agregado. (a) Sensibilidad. (b) Especificidad. (c) Predictividad Positiva

Otras medidas de tendencia central y dispersión.

Tabla 14.1 Media Aritmética y Desvío Estándar. Base de Datos completa.

Base de Datos completa				
	FB	MLP	FNN	MLP + AG
S[%]	91,82 ± 7,24	86,38 ± 20,34	83,25 ± 15,21	**93,52 ± 12,9**
E[%]	98,53 ± 1,03	97,87 ± 2,05	94,61 ± 5,48	**98,88 ± 1,21**
PP[%]	**85,62 ± 8,58**	63,92 ± 27,13	**50,56 ± 28,03**	74,57 ± 22,61

Tabla 14.2 Media Aritmética y Desvío Estándar. Señal #119 con ruido agregado

Señal #119 con ruido agregado				
	FB	MLP	FNN	MLP + AG
S[%]	**94,11 ± 2,47**	63,75 ± 27,02	**53,68 ± 28,76**	74,25 ± 24,74
E[%]	**99,14 ± 0,87**	97,27 ± 2,81	90 ± 6,17	98,51 ± 1,79
PP[%]	**96,95 ± 3**	83,15 ± 19,15	57,83 ± 26,95	90,57 ± 11

En la Tabla 14.1 Media Aritmética y Desvío Estándar. Base de Datos completa. se listan los valores de media aritmética más menos el desvío estándar. Si analizamos los valores extremos, aquellos que están resaltados, podemos observar que para el índice S, la mejor performance exhibe la red MLP con AG, con una media aritmética elevada, pero también con un desvío estándar alto. La E también alcanza su valor más alto para la MLP con AG¸ con un valor muy próximo al de FB. Los índices más bajos resultaron para la PP, en el peor de los casos, observamos que la FNN resultó ser el algoritmo que más falsos positivos detecta, con un rendimiento, en promedio de apenas un poco más alto del 50%, lo que la convierte en una red poco eficiente para la clasificación de latidos. La mayor PP la exhibe el FB con un 85,62% de efectividad. En general todos los métodos presentan altos índices de E y una PP bastante menor.

La

Tabla 14.2 lista la media aritmética y el desvío estándar de la base de datos ruidosa. Nuevamente nos concentraremos en los valores extremos de la tabla. Como se puede observar, el Banco de Filtros digitales ha resultado ser el método más apropiado para la detección de PVC en entornos ruidosos, con un promedio de más de 94% de efectividad en todos los índices evaluados y con un desvío estándar menor o igual al 3%. Cabe destacar que la red neuronal con algoritmos genéticos también obtuvo buenos índices de performances, sin embargo, no alcanzaron los niveles de eficacia del Banco de Filtros. Observamos también que la red difusa fue la menos acertada en la clasificación de latidos cardíacos, con una sensibilidad de apenas el 53% de aciertos y un desvío estándar de más del 50% de su valor, lo que disminuye aún más su eficiencia.

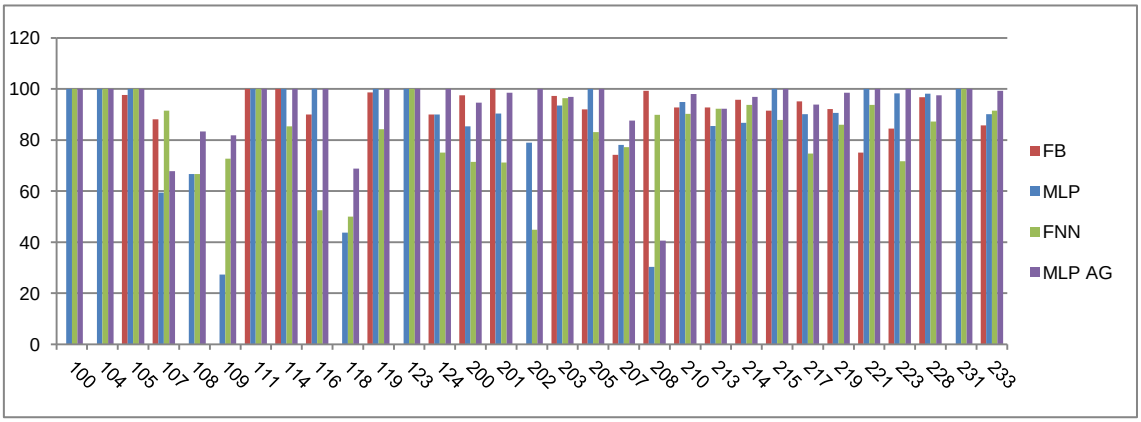

Figura 14.5. Índice de Sensibilidad para la Base de datos completa

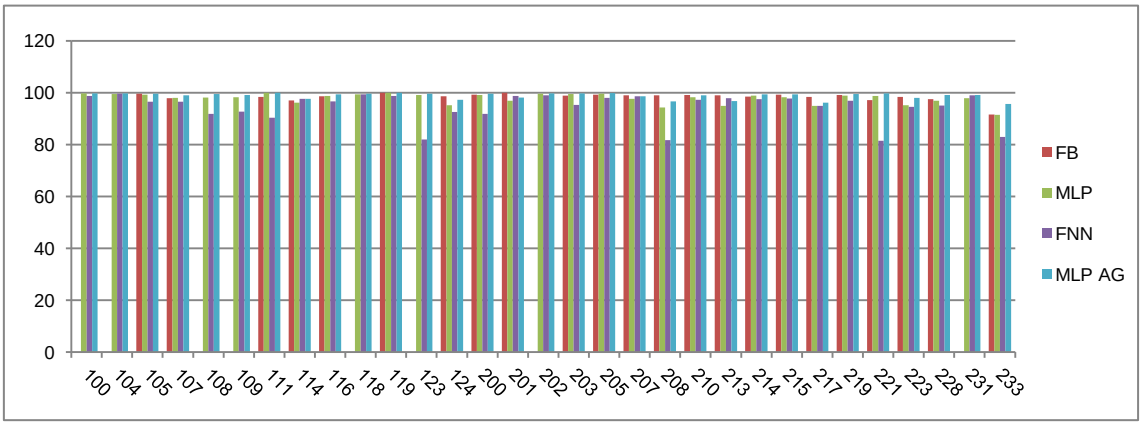

Figura 14.6 Índice de Especificidad para la Base de datos completa

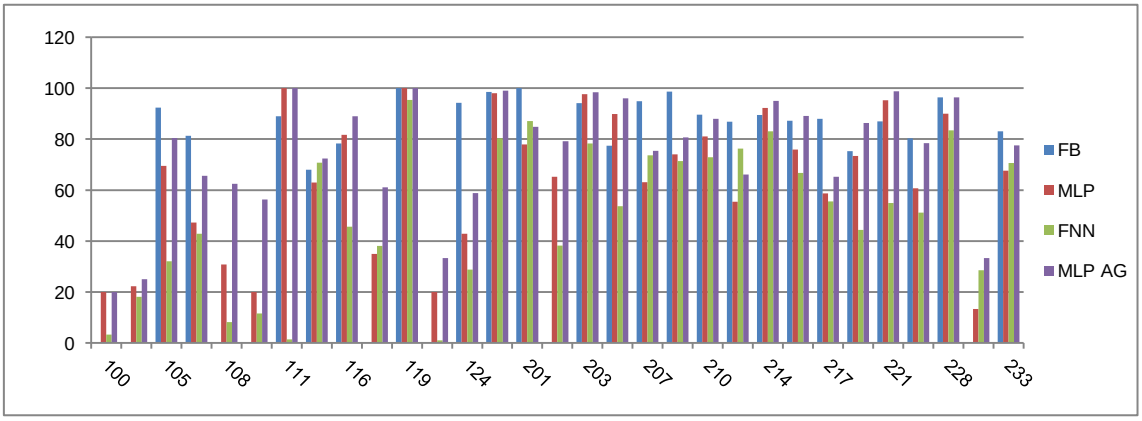

Figura 14.7 Índice de Predictividad Positiva para la Base de datos completa

En las Figura 14.5, Figura 14.6 y Figura 14.7 graficamos los índices de Sensibilidad, Especificidad y Predictividad Positiva respectivamente de cada una de las señales de la base de datos. Nótese que algunas señales están omitidas para el Banco de Filtros, debido a que no fueron evaluadas. Claramente puede observarse que el índice de Especificidad (Figura 14.6) arroja resultados más parejos para los cuatro métodos de estudio. Un comportamiento similar puede observarse en la Figura 14.9 que grafica dicho índice para la base de datos ruidosa.

Si observamos la Figura 14.5, podemos distinguir algunas señales con comportamiento disímil para cada uno de los algoritmos, tal es el caso de las señales #109, #202 y #208. Tanto en la señal #109 como en la #202 puede destacarse que la implementación de los Algoritmos Genéticos para el ajuste de pesos de la red MLP ha sido un gran acierto, con una mejora del más del 50% en el primer caso y más del 20% en el segundo, alcanzando el 100% de efectividad. Para este índice en discusión, la FNN tiene la particularidad de presentar un buen desempeño en la señal #208, con más del 20% de efectividad que las redes nítidas, sin embargo no alcanza para igualar la alta performance que se obtuvo con el FB. En particular podemos decir que esta señal es de morfología muy irregular, lo que dificulta su análisis con medios automáticos. Para las señales restantes, los cuatro métodos se comportaron de manera similar al evaluar el índice de sensibilidad.

En la Figura 14.7 se observa la Predictividad Positiva, con valores disímiles para las señales agrupadas en la serie 100, esto se debe al bajo número de PVCs que, en general, aparecen en estas señales, por lo que tiene mucha incidencia la detección de falsos positivos en este índice de performance. En la gran mayoría de las señales, el FB ha sido el algoritmo más certero, seguido por la red MLP con AG.

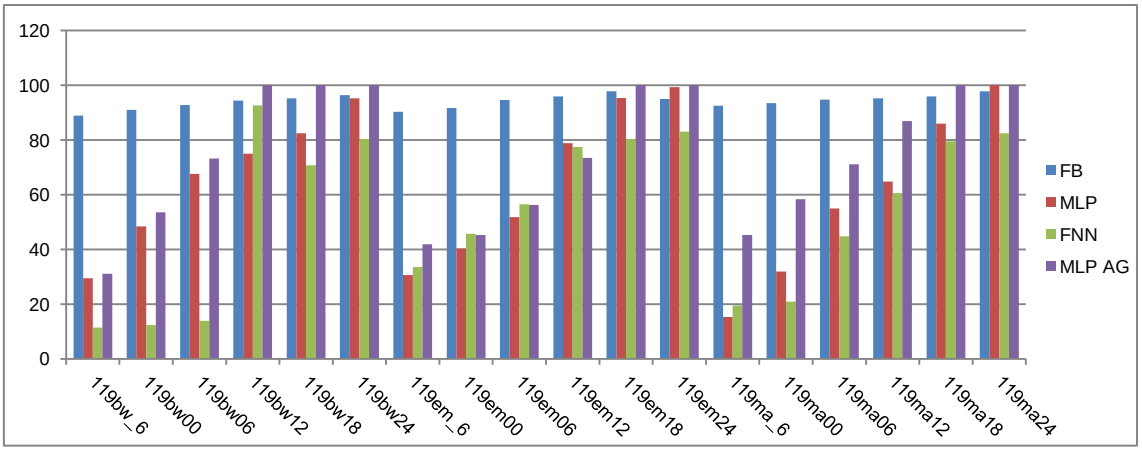

Figura 14.8 Índice de Sensibilidad para la señal #119 con ruido agregado

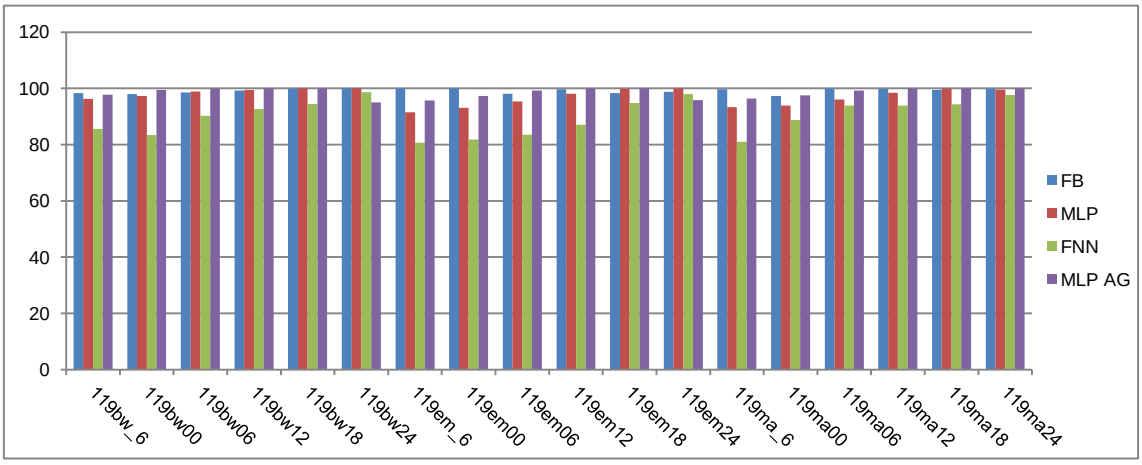

Figura 14.9 Índice de Especificidad para la señal #119 con ruido agregado

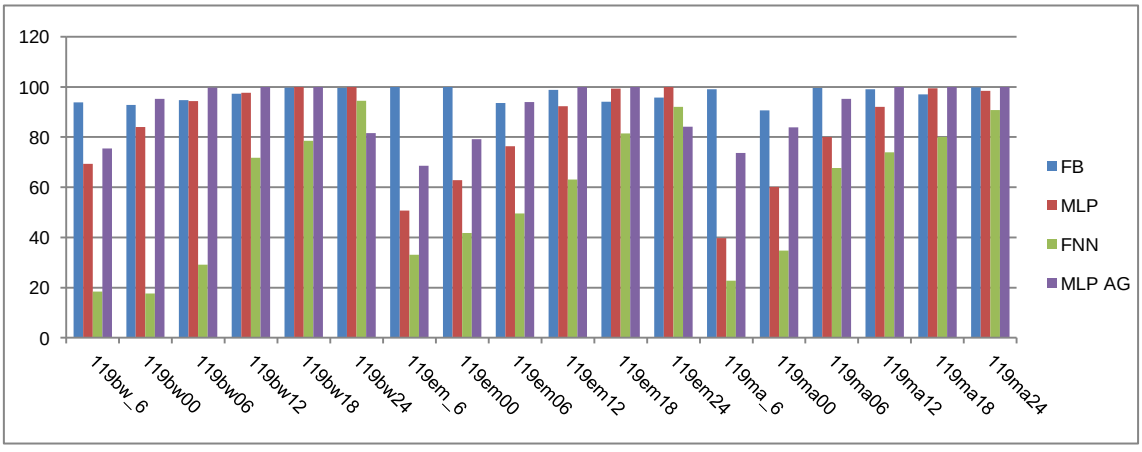

Figura 14.10 Índice de Predictividad Positiva para la señal #119 con ruido agregado

Las Figura 14.8, Figura 14.9 y Figura 14.10 muestran el desempeño de los cuatro métodos evaluados para la clasificación de latidos sobre la base de datos ruidosa. Puede observarse en el índice de Sensibilidad, Figura 14.8, como a medida que el nivel de ruidos en db disminuye, el valor del índice aumenta. Si bien esta característica se observa para las redes neuronales implementadas, no podemos dejar de mencionar que el FB ha tenido un desempeño muy regular a lo largo de toda la base de datos. Lo mismo sucede con el índice de Predictividad Positiva, como es de esperarse, cuanto menos ruido trae consigo la señal, más efectivo es el método. También destacamos la eficacia del Banco de Filtros que alcanzó valores por encima del 90% de efectividad en todas las señales ruidosas evaluadas.

Acerca de la clasificación de latidos cardíacos.

Del análisis estadístico surge la clara conclusión, que en entornos ruidosos el Banco de Filtros Digitales aquí diseñado resulta ser muy robusto para la clasificación de latidos ectópicos prematuros. Cuando evaluamos la base de datos completa, podemos ver que este paradigma además no incurre en la falsa detección de PVCs, esto es falsos positivos, lo que ocurre en las redes neuronales, ya sean estas nítidas o difusas. Si bien los índices de Sensibilidad y Especificidad alcanzan sus valores máximos con la red neuronal MLP con AG, podemos decir que en general, el Banco de Filtros Digitales con las características descriptas resulta ser un método altamente eficiente para la detección de PVCs.

De las tres redes neuronales diseñadas e implementadas, la red MLP con AG ha sido la más eficiente. La incorporación de los algoritmos genéticos en el proceso de ajuste de los pesos produjo una mejora de hasta un 50% en la precisión para algunos casos. Como desventaja, podemos citar la alta detección de falsos positivos detectados con este método. Una explicación a este fenómeno puede ser el hecho de que las señales evaluadas presentan otras patologías además de las PVCs, algunas con características similares a estas. Esto produce un falso positivo pues la red identifica a este latido que no es normal pero tampoco es una PVC como tal, al contabilizar las PVC, aparece un alto número de falsos positivos. Esta explicación puede ser fácilmente comprobada, pues en señales que únicamente presentan PVC y latidos normales, la eficacia de la red

ha sido del 100% o próxima al 100%. Este dato es sumamente alentador, así como la alta tasa de rechazo de latidos normales, plasmada en el índice de Especificidad, que también se cumple en el Banco de Filtros Digitales.

En la bibliografía actual consultada (Ahmad et al., 2012), (Mar et al., 2011), (Jokit et al., 2010), encontramos un nuevo índice de performance denominado Índice de Precisión, que evalúa la totalidad de aciertos sobre la totalidad de errores y que no había sido calculado por nosotros en trabajos previos. Para el Banco de Filtros Digitales aquí diseñado obtuvimos un índice precisión promedio de **95,93%** de efectividad en la clasificación de latidos. Para las redes neuronales descriptas obtuvimos, un **93,50%** de precisión para la red MLP, un **88,04%** para la red difusa y un **96,62%** para la red híbrida, MLP con AG. Si consideramos que este índice de performance evalúa todos los resultados obtenidos con cada algoritmo, podemos concluir que **todos los paradigmas aquí diseñados han tenido una alta tasa de aciertos, si bien la red MLP con AG ha sido la más precisa**. Cuando evaluamos las señales ruidosas, obtenemos un 96,01% para el FB contra un 88,23% de la red MLP+AG, reafirmando nuestra primera conclusión: **los bancos de filtros digitales son un método robusto para la clasificación de latidos en entornos ruidosos**.

14.2. Aportes genuinos de la tesis

Los desarrollos realizados en esta tesis demuestran que:

Con respecto a la implementación de un **Banco de Filtros Digitales**, máximamente diezmado, que tiene la propiedad de la Reconstrucción Perfecta y filtros de fase lineal, resultan útiles en el análisis del ECG. En general están vinculados a transformadas Wavelet. Son eficientes computacionalmente hablando dado que operan sobre un número reducido de muestras en la distintas sub-bandas de frecuencias en las que la señal está dividida. Esta particularidad permite aplicar diferentes procesos, en paralelo, sobre cada una de las sub-señales que son el resultado de la aplicación del Banco de Análisis y el proceso de sub-muestreo.

En esta tesis realizamos tres procesos diferentes sobre la señal de ECG, con un único grupo de filtros, ellos son: (i) la detección de puntos característicos en la señal,

(ii) la eliminación de ruido blanco y (iii) la clasificación de latidos. Lo novedoso del método viene dado por el hecho de la utilización en simultáneo de algunas o todas las sub-bandas de frecuencias en las que, en cada una de ellas, es posible obtener diferentes características morfológicas y de contenido espectral de la señal de ECG. Esto es lo que permite realizar los procesos en paralelo con una tasa de muestreo menor que la original, lo que resulta ventajoso pues acelera los tiempos computacionales. Posterior al proceso reconstruimos la señal de ECG, con una distorsión despreciable gracias a las características del Banco de Filtros y al diseño del conjunto de filtros. En este sentido, las transformadas superpuestas ortogonales juegan un papel preponderante, pues la pérdida de datos es mínima. Los Bancos de Filtros representan una metodología alternativa para el estudio de las bioseñales y constituyen una herramienta adicional para el diagnóstico médico.

Los procesos de extracción de **Componentes Principales** y **Componentes Independientes** realizados en esta tesis, demuestran la gran utilidad de estas técnicas para el proceso de señales biológicas. Su capacidad para reducir matrices de datos que acumulan gran cantidad de información, muchas veces redundante, permite que los procesos computacionales se agilicen y resulten más efectivos debido a la estandarización de los datos que realiza el PCA. Nuestro aporte en esta área está dado por la extracción de los Componentes Independientes, mediante la resolución de la matriz de mezcla resolviendo la ecuación matricial de Liapunov. Éste demostró ser un algoritmo adecuado para la separación ciega de fuentes. La utilización de esta ecuación, a diferencia de los métodos citados en la bibliografía, tiene dos ventajas puntuales, (i) no es necesario filtrar los datos previamente para estandarizarlos, pues utilizamos PCA durante el proceso de separación de fuentes; (ii) se trata de un proceso no recursivo, lo que lo convierte en un proceso rápido y sin arrastre de errores: (iii) incorporamos conceptos de estabilidad, atribuibles a nuestro sistema, lo que nos permitió utilizar las herramientas de la teoría de control moderno, esto imprime en nuestros resultados la aplicabilidad concreta de la matemática. Los resultados obtenidos fueron validados con el algoritmo FastICA, ampliamente conocido y utilizado.

Acerca de las **Redes Neuronales** implementadas para la clasificación de latidos ventriculares prematuros, podemos concluir que los resultados obtenidos son muy buenos, siempre que planteemos a la RNA como una herramienta más en la cual el cardiólogo puede apoyarse en casos de dudas para tomar decisiones correctas. La incorporación del Análisis de Componentes Principales a los datos de entrada a la red resultó ser sumamente beneficioso y mejoró los índices previamente obtenidos. Esto también permitió que la convergencia de los procesos de aprendizaje se realizara a una velocidad más alta.

El uso de los algoritmos genéticos para asistir a las redes neuronales puede considerarse una de las áreas más prometedoras en la combinación de paradigmas de sistemas expertos.

Para mejorar aún más los índices de performance obtenidos con las redes neuronales, aplicamos **Algoritmos Genéticos** para lograr un ajuste más estrecho en los valores de los pesos de las neuronas. En la mayoría de los casos observamos una mejora en los resultados cuando utilizamos los algoritmos genéticos, lo que podría resultar en una mejora general de la performance si incluimos datos que tengan en cuenta la variabilidad entre señales, esto es, incluir nuevos datos de entrada, o bien, entrenar la señal con una composición de señales formada por aquellas más representativas de ambas series.

Entonces, en cuanto a las Redes Neuronales podemos concluir que (i) la integración de las técnicas de PCA a la red neuronal permitieron no sólo la reducción de la matriz de datos de entrada, sino la estandarización de todo el sistema. Esto le imprimió a la red un carácter novedoso, con resultados muy satisfactorios y procesos más ágiles. La convergencia de la red también se vio mejorada, con pocas iteraciones fue posible alcanzar el mínimo global. Como resultado adicional, no hubo signos de “sobre-entrenamiento” de la red, por lo tanto, se logró la generalización buscada. (ii) el uso de los Algoritmos Genéticos para la optimización del ajuste de los pesos de la red, si bien, complejiza el proceso, mejora sustancialmente los índices de performance.

14.3. Perspectivas Futuras

En relación a los **Bancos de Filtros**, una línea de investigación a seguir es obtener una localización en tiempo-frecuencia mediante nuestro Banco de Filtros Digitales con decimación a partir de la transformada ondita de la señal *s(t)*. Una discretización diádica de los parámetros, según la bibliografía consultada, el análisis con los filtros correspondientes a la Daubechies de 4 puntos, permiten una buena localización en las bandas de frecuencias que quedan determinadas a partir de esa ondita $\psi(t)$ y del intervalo de muestreo del equipo de adquisición utilizado. Algunos trabajos actuales sobre el diseño de Bancos de Filtros con características de ortogonalidad, fase lineal y reconstrucción perfecta, mencionan métodos iterativos de eigen-filtros, (Chaphekar et al., 2012). Ésta podría ser también una línea a seguir, ya que los autores proponen como desafío la aplicación de este método para Bancos de Filtros de *M* canales y la extensión a bancos multidimensionales.

Tanto en el proceso de diseño de **Redes Neuronales** como de **Algoritmos Genéticos**, todavía la heurística juega un papel protagónico. Sería de interés el desarrollo de técnicas capaces de definir una arquitectura "ideal" que nos permita alcanzar resultados realmente satisfactorios.

Los trabajos más actuales proponen métodos híbridos, similares a los propuestos en esta tesis. Algunos autores combinan las técnicas de PCA con Redes Neuronales, con buena tasa de aciertos, poco más del 98%, así lo plantean Martis y colaboradores en (Martis et al., 2012). Lo mismo hacen Wang y colaboradores, (Wang et al., 2012) que reducen la dimensión de la matriz de entrada combinando PCA con LDA (análisis discriminante lineal) para luego clasificar los latidos con dos clases de redes neuronales, probabilística en un caso y difusa en otro. Sus resultados también se encuentran por encima del 98% de aciertos. Si bien otros autores proponen otras arquitecturas de redes neuronales para la clasificación de latidos, utilizan mezclas de sistemas expertos para mejorar sus tasas de aciertos, (Javadi et al., 2012), que propone un aprendizaje de correlación negativa. Lo cierto es que, todos los autores aquí mencionados utilizan la base de datos de arritmias de MIT-BIH para probar sus resultados y testear sus redes. Los índices de performance que hallan son los mismos estándares que calculamos en

este trabajo, es decir, es válida la comparación con cualquiera de los sistemas mencionados. Los resultados de nuestra propuesta están dentro de los rangos de eficiencia de métodos formulados por otros autores.

La tendencia en el desarrollo de algoritmos que involucran Inteligencia Artificial, demuestran que la SVM (Support Vector Machine –Máquina de Soporte Vectorial), que es un método novedoso basado en la teoría del aprendizaje estadístico y que es una herramienta poderosa para resolver problemas con pequeñas muestras de datos, no linealidad y grandes dimensiones, características que encontramos en las señales biomédicas, están desplazando a las redes neuronales, debido principalmente a sus métodos de multi-clasificación que incluyen uno versus el resto y uno versus uno. Sus procedimientos han sido ampliamente aplicados a diferentes áreas, incluida la ingeniería biomédica. Recientes trabajos mencionan este sistema experto para la clasificación de señales intracraneales en combinación con PCA para la reducción de la dimensión de la matriz de datos, tal es el caso de Uğuz, (Uğuz, 2012).

La aplicación de esta novedosa técnica es una línea a seguir en la clasificación de latidos ventriculares prematuros.

Otra línea de investigación a profundizar, es ampliar el abanico de clasificación de latidos anómalos presentes en la base de datos objeto de estudio, e incluir todas las patologías presentes, no solamente las PVC. Esto conlleva un gran desarrollo matemático y computacional, debido a que cada anomalía posee sus propias características que la identifican, y como lo mencionáramos a lo largo de la tesis, los parámetros de entrada a las redes son un elemento fundamental en el diseño para poder lograr niveles de performance óptimos.

Bibliografía

AAMI, 1987. *Recommended Practice for Testing and Reporting Performance Results of Ventricular Arrhythmia Detection Algorithms,* Publication ECAR-1987. Arlington, Virginia: Association for the Advancement of Medical Instrumentation.

Afonso VX, 1996**.** "Improving ECG processing algorithms using Filter Banks". PhD Thesis. *University of Wisconsin.* Madison, Wisconsin.

Afonso VX, Tompkins WJ, Nguyen TQ, Michler K y Luo S, 1996**.** "Comparing stress ECG enhancement algorithms: With an introduction ti filter bank based approach", *IEEE Eng. In Med. And Biol. Mag*., vol. 15 no. 3, pp. 37-44

Afonso VX, Tompkins WJ, Nguyen TQ, Trautmann S y Luo S, 1995**.** "Filter bank-based processing of the stress ECG". *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*

Ahmad Shukri MH, Ali MSAM, Noor MZH, Jahidin AH, Saaid MF, Zolkapli M, 2012**. "**Investigation on Elman Neural Network for Detection of Cardiomyopathy". *Control and System Graduate Research Colloquium (ICSGRC), 2012 IEEE.* Pp 328 – 332.

Álvarez Picaza C, Monzón JE, Veglia, JE, 2006. "Aplicación de ICA con conceptos de estabilidad para separar señales". *Reunión de Comunicaciones Científicas SGCyT-UNNE.*

Anderson TW, 1963. "Asymptotic Theory for Principal Components Analysis." *The Annals of Mathematical Statistics.* Vol 34, Num 1, Pp 122-148.

Asian O, Yildiz OT, Alpaydin E, 2009. "Calculating the VC-dimension of decision trees 24th International Symposium on *Computer and Information Sciences".* Pp 193–198.

Berne RM, Levy MN, 1983. *Physiology*. Sain Louis Missouri: C. V. Mosby

Branke J, 1995. "Evolutionary Algorithms for Neural Network Design and Training", *Proceedings of the First Nordic Workshop on Genetic Algorithms and its Applications.*

A.P. Calderón AP, 1962. "Intermediate spaces and interpolation, the complex method". *Studia Math*. Vol 24. Pp113–190.

Calhoun VD y Adali T, 2006. "Unmixing fMRI with Independent Component Analysis". *IEEE Engineering In Medicine and Biology Magazine*. Vol 25. No 2 Pp 79-90.

Cardoso JF, Laheld B, 1994. "Equivariant adaptive source separation*". IEEE Transactions on Signal Processing.*

Chaphekar P, Gune OA, Gadre VM, 2012**.** "Construction of 4-band linear-phase orthogonal perfect-reconstruction filter banks using iterative eigenfilters method". *Signal Processing and Communications.* Pp 1-5.

Chui C, 1997. *Wavelets: A Mathematical Tool for Signal Analysis*, Editorial Siam, Primera Edición. EE.UU.,

Comon P, 1994. "Independent component analysis, A new concept?," *Signal Processing*, Vol. 36, pp. 287–314,

D'Attellis CE y Fernández Berdaguer E, 1995. *Harmonic Analysis and Wavelet Theory in Applied Sciences*. Birkhauser, Boston.

D'Attellis CE, 1985. *Distribucional de sistemas lineales, cursos y seminarios*. FCEyN, UBA**.**

Daubechies I, 1992. *"Ten lectures on wavelets"*. Society for Industrial and Applied Mathematics. Philadelphia, Pennsylvania.

de la Fuente DA, Corona AV, 2010. "Redes Neuronales Artificiales". Tutorial. *Universidad Politécnica de Madrid-UPM.* España. http://www.gc.ssr.upm.es/inves/neural/ann2/concepts/biotype.htm. FUC: Mar 2010.

de Queiroz RL, 1996. "On Lapped Transforms". *Ph.D Dissertation, The University of Texas at Arlington.*

Févotte C, Gribonval R y Vincent E, 2005. "BSS_EVAL Toolbox User Guide – Revision 2.0," *French GdR-ISIS/CNRS, Paris, France, Internal Report 1706*, pp. 1–19.

Fuster R, Giménez I, 2006. *Variable Compleja y Ecuaciones Diferenciales.* Editorial Reverté. Madrid, España.

Ganong WF, 1986. *Fisiología Médica.* 19na Edición. Ed. El Manual Moderno, Méjico D. F.

Gävert H, Hurri J, Särelä J y Hyvärinen A, 2005. *FastICA for Matlab, version 2.5*. Natick: Massachusetts: The MathWorks Inc.

Ghongade R y Ghatol A, 2008. "A robust and reliable ECG pattern classification using QRS morphological features and ANN". *TENCON 2008 - 2008 IEEE Region 10 Conference* Pp 1 - 6

Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE, 2000**.** "PhysioBank, PhysioToolkit, and Physionet: Components of a New Research Resource for Complex Physiologic Signals". *Circulation* Nº 101, Vol. 23, Pp 215-220 [Circulation Electronic Pages; http://circ.ahajournals.org/cgi/content/full/101/23/e215]; FUC: Nov2012, (en línea).

Guisande González C, 2006. *Tratamiento de datos.* Ediciones Díaz de Santos. Madrid, España.

Hamilton PS, Tompkins WJ, 1986. "Quantitative investigation of QRS detection rules using the MIT/BIH arrhythmia database". *IEEE Transactions on Biomedical Engineering*, vol. BME-35, pp. 1157-1165

Hammerstrom D, 1992. "Neural networks at work". *IEEE Spectrum*, pp. 26-32. 1993

Hanselman D, Littlefield B, 1995. *Matlab User's Guide*. Englewood Cliffs, New Jersey: Prentice-Hall.

Hao J, Lee I, Lee TW, Sejnowski TJ, 2010. "Independent Vector Analysis for Source Separation Using a Mixture of Gaussians Prior". *Neural Computation*. Vol 22. No 6. Pp 1646-1673. PMCID: In Process

Haykin S, 1996. *"Neural Networks. A comprehensive Foundation"* IEEE Press. NJ.

Hernandez Rodriguez O, 1998. *Temas de Análisis Estadístico Multivariado*. Editorial Universidad de Costa Rica

Hongyu S, Yan X, 2009. "Classification Method of EEG Signals Based on Wavelet Neural Network". *3rd International Conference on Bioinformatics and Biomedical Engineering. IEEE-ICBBE*. Pp. 1-4.

Hyvärinen A, Oja E, 2000. "Independent Component Analysis: Algorithms and Applications". *Neural Networks,* Num 13, Vol (4-5). Pp 411-430.

Hyvärinen A., Karhunen J, Oja E, 2001. *Independent Component Analysis.* John Wiley and Sons Editors.

Hyvärinen A., 1999. "Survey on Independent Component Analysis". *Neural Computing Surveys*. vol 2, Pp. 94-128

Ince T, Kiranyaz S, Gabbouj M, 2009. "A Generic and Robust System for Automated Patient-Specific Classification of ECG Signals". *IEEE Transactions on Biomedical Engineering.* Vol 56. No 5. Pp 1415 - 1426

Jang JSR, Sun CT, Mizutani E, 1990. *Neuro-Fuzzy and Soft Computing. A Computational Approach to Learning and Machine Intelligence*. Prentice Hall, New York.

Javadi M, Arani SAAA, Sajedin A, Ebrahimpour R, 2012. "Classification of ECG arrhythmia by a modular neural network based on Mixture of

Experts and Negatively Correlated Learning". *Biomedical Signal Processing and Control*; (en prensa).

Jokic S, Krco A, Delic V, Sakac D, Jokic I, Lukic Z, 2010. "An Efficient ECG Modeling for Heartbeat Classification". *10th Symposium on Neural Networks Applications in Electrical Engineering*. Pp 73-76.

Jolliffe T, 2002. *Principal Component Analysis (2nd ed.)***.** Journal of the American Statistical Association. New York: Springer-Verlag,

Jutten C, Herault J, 1991. "Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture". *Signal Processing*, vol 24 Pp1-10.

Kharabian S, Shamsollahi MB, Sameni R, 2009. "Fetal R-wave detection from multichannel abdominal ECG recordings in Low SNR". *Proceedings of the IEEE EMBS Conference*. Pp 344-347

Kunz D, 2008. "An Orientation-Selective Orthogonal Lapped Transform". *IEEE Transactions on Image Processing*, vol. 17, no. 8, pp. 1313-1322.

Lan Z, Zheng Z y Li Y, 2010. "Toward Automated Anomaly Identification in Large-Scale Systems". *IEEE Transactions On Parallel And Distributed Systems*, Vol. 21, No. 2. Pp 174-187

Langley P, Rieta JJ, Stridh M, Millet J, Sörnmo L, Murray A, 2006 " Comparison of Atrial Signal Extraction Algorithms in 12-Lead ECGs With Atrial Fibrillation". *IEEE Transactions on Biomedical Engineering*, Vol. 53, No. 2, Pp 343-346

Li, SZ, Huang YH, and Fu J, 1995. "Convex energy functionals in the DA model". *In Proceedings of IEEE International Conference on Image Processing, Washington, D.C.*

Mallat S, 2009**.** A wavelet tour on signal processing. The sparse way. Tercera edición. Elsevier Press. Estados Unidos.

Malvar H, 1992. *Signal processing with lapped transforms*. Artech House. Universidad de Michigan.

Malvar HS, 1990. "Lapped Transforms for Efficient Transform/Subband Coding" *IEEE Transaction on Acustics, Speech, and Signal Processing*. Vol 38, No. 6, pp 969-978.

Malvar HS, Sataelin, DH, 1989. "The LOT: Transform Coding Without Blocking Effects". *IEEE Transaction on Acustics, Speech, and Signal Processing*. Vol 37, No. 4, pp 553-559.

Mar T, Zaunseder S, Martínez JP, Llamedo M, Poll R, 2011. "Optimization of ECG Classification by Means of Feature Selection". *IEEE Transactions On Biomedical Engineering*, Vol. 58, No. 8, pp 2168-2177.

Marcano, C. 2005, "Propuesta de resolución del sistema matricial en el segundo método de estabilidad de lyapunov". *Universidad, Ciencia y Tecnología".* Vol 9. No 35.Pp. 167-171.

Martis RJ, Achaya UR, Mandana KM, Ray AK, Chakraborty C, 2012. "Application of principal component analysis to ECG signals for automated diagnosis of cardiac health". *Expert Systems with Applications.* Vol 39, Pp 11792–11800.

MIT-BIH, 2004. Massachusetts Institute of Technology – Beth Israel Hospital Database Distribution, MIT, 77 Massachusetts Avenue, Room 20A-113, Cambridge MA.

Mitra S, Hayashi Y, 2000. "Neuro–Fuzzy Rule Generation: Survey in Soft Computing Framework". *IEEE Transactions On Neural Networks*, Vol.11, No.3, Pp 748-768.

Moody GB, 2000. "Morphology Representation Using Principal Components" http://physionet.org/tutorials/pr-comp. FUC: Nov 2012, (en línea).

Moody GB, 1999. *ECG Database Applications Guide*, 10th Edition. Boston, Massachusetts: Harvard-MIT Division of Health Sciences, Biomedical Engineering Center.

Moody GB, Mark RG, 1989. "QRS Morphology Representation and Noise Estimation using the Karhunen-Loève Transform". *Computers in Cardiology*, Pp. 269-272

Moody GB, Mark RG, 1994. "The MIT-BIH Arrhythmia Database CD-ROM". *Physics in Medicine and Biology*. Vol 39a Part 1, Pp 388.

Moody GB, Mark RG, 2001. "The Impact of the MIT-BIH Arrhythmia Database". *IEEE Engineering in Medicine & Biology Magazine*. Vol 20, No 3, Pp 45-50.

Moody GB, Mark RG, Goldberger AL, 2001. "PhysioNet: A Web-Based Resource for the Study of Physiologic Signals". *Engineering in Medicine & Biology Magazine, IEEE.* Vol 20, No 3, Pp 70-75.

Monzón JE, 2004. "Análisis del Electrocardiogramas con Redes Neuronales Basadas en Lógica Borrosa". PhD Thesis. *Universidad Nacional del Nordeste*. Corrientes, Argentina.

Monzón JE, Pisarello MI, 2005. "Cardiac Beat Classification using a Fuzzy Inference System" *Proceedings of the 27th Annual International Conference IEEE-EMBS.* ISBN 0-7803-8741-4), Pp. 5582-5584.

Monzón JE, Pisarello MI, 2005. "Evaluación del ECG con sistemas neuronales nítidos y difusos". *XV Congreso de Bioingeniería. IV Jornadas de Ingeniería Clínica. SABI.* (ISBN 950-698-156-6).

Monzón JE, Pisarello MI, 2004. **"**Identificación de Latidos Cardíacos Anómalos con Redes Neuronales Difusas". *Reunión de Comunicaciones Científicas y Tecnológicas. Secretaría General de Ciencia y Técnica. Universidad Nacional del Nordeste.*

Monzón JE, Pisarello MI, Álvarez Picaza C, 2007. "Blind Source Separation of electrocardiographic signals using system stability criteria" *Proceedings of the 29th Annual International Conference IEEE-EMBS.* Pp 3.493-3.495

Nakamura W , Anami K, Mori T, Saitoh O, 2006. "Removal of Ballistocardiogram Artifacts From Simultaneously Recorded EEG and fMRI Data Using Independent

Component Analysis", *IEEE Transactions on Biomedical Engineering,* Vol. 53, No. 7, Pp1294-130

Nesbit A, Vincent E, Plumbley MD, 2009. "Benchmarking flexible adaptive time-frequency transforms for underdetermined audio source separation". *IEEE International Conference on Acoustics Speech and Signal Processing*, Pp. 37-40.

Osowski S, Linh TH, 2000. "Fuzzy clustering neural network for classification of ECG beats". *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. Vol 5.* Pp 26 - 30

Pan JP, Tompkins WJ, 1985. "A real-time QRS detection algorithm"**.** *IEEE Transactions on Biomedical Engineering,* Vol 32. No 3. Pp: 230-236.

Pisarello MI, Álvarez Picaza C, Monzón JE, 2007. "Eliminación de ruido en bioseñales utilizando la ecuación algebraica de Lyapunov", *IV Congreso Latinoamericano de Ingeniería Biomédica, CLAIB.* Vol 18 – Pp 78-83.

Pisarello MI, Álvarez Picaza C, Monzón JE, 2008. "Separación de Frecuencias No Deseadas en la señal cardíaca utilizando ICA*", XII Jornadas Internacionales de Ingeniería Clínica y Tecnología Médica.*

Pisarello MI, Álvarez Picaza C, Monzón JE, 2006**.** "Análisis de Componentes Principales para la Compresión del ECG" *5ta Conferencia Iberoamericana en Sistemas, Cibernética e Informática: CISCI*. *CISCI*. ISBN: 980-6560-89-2. Vol 1, Pp. 48-54.

Pisarello MI, Monzón JE, 2002. "Utilización de Bancos de Filtros en el Análisis Digital del ECG", *Revista Argentina de Bioingeniería*, (ISBN 0329-5257). Volumen 8, N°2, Diciembre 2002.

Pisarello MI, Kleisinger GH, Monzón JE, 2001 Procesamiento Digital del ECG con Bancos de Filtros. *Reunión de Comunicaciones Científicas y Tecnológicas. Secretaría General de Ciencia y Técnica. Universidad Nacional del Nordeste*. Corrientes, 22 al 26 de Octubre de 2001.

Poungponsri S, Yu X-H, 2009. "Electrocardiogram (ECG) Signal Modeling and Noise Reduction Using Wavelet Neural Networks". *Proceedings of the IEEE International Conference on Automation and Logistics*. Pp. 394-398.

Rautenberg CN, D'Attellis CE, 2004. *Control Lineal Avanzado y Control Óptimo*. AADECA Asociación Argentina de Control Automático. Buenos Aires, Argentina

Rojo AO, 1983, *Algebra II*. Editorial El Ateneo.4ta Edición. Buenos Aires, Argentina

Ruan D, 1997. *Intelligent hybrid systems: fuzzy logic, neural networks, and genetic algorithms.* Springer Editor. Kluwer Academic Publisher, Massachusett, USA.

Sánchez Ruiz LM, Legua Fernández MP, 2006. *Fundamentos de Variable Compleja y Aplicaciones.* Editorial de la Universidad Politécnica de Valencia. Valencia, España.

Sánchez-Sinencio E, Lau C, 1992. *"Artificial Neural Networks, Paradigms, Applications and Hardware Implementations"*. IEEE Press. NJ.

Sandryhaila A, Chebira A, Milo C, Kovaˇcevic´ J, Püschel M, 2010. "Systematic Construction of Real Lapped Tight Frame Transforms". *IEEE Transactions on Signal Processing*, Vol. 58, No. 5, Pp 2556-25.

Schmitt M, 1999. "VC dimension bounds for higher-order neurons". *9th International Conference on Artificial Neural Networks (Conf. Publ. No. 470).* Vol 2. Pp 563-568.

Sejnowski TJ.1998. "A Preface to Independent Component Analysis". *Computational Neurobiology Laboratory at the Salk Institute.*

Simpson, PK, 1992. "Foundations of Neural Networks". Invited Paper. *Artificial Neural Networks, Paradigms, Applications and Hardware Implementations"*. IEEE Press, pp. 3-24. NJ

Smith MJ y Barnwell TP, 1984. "A procedure for designing exact reconstruction filter banks for tree structured sub-band coders". *Proceedins of the IEEE International Conference on Acoustic, Speech, and Signal Processing* .

Smith LI, 2002. "A tutorial on Principal Components Analysis". http://csnet.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf Fecha de acceso: Marzo 2010. [en línea]

Soman K, Vaidyanathan PP, Nguyen TQ, 1993. "Linear phase paraunitary filter banks: theory, factorizations and designs". *IEEE Transaction on Signal Processing.*

Strang G, Nguyen T, 1996. *"Wavelets and Filter Banks"*, Wellesley-Cambridge Press

Sun Z-L, Au K-F, Choi T-M, 2007. "A Neuro-Fuzzy Inference System Through Integration of Fuzzy Logic and Extreme Learning Machines". *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, Vol. 37, No. 5. Pp, 1321-1331.

Tompkins WJ (editor): 1993 *Biomedical Digital Signal Processing*. Prentice-Hall, Englewood Cliffs, New Jersey.

Tompkins WJ, Webster JG, 1998. *Interfacing Sensors to the IBM PC.* Englewood Cliff, New Jersey: Prentice Hall.

Tran TD, 1999. "Linear phase Perfect Reconstruction Filter Banks: Theory, Structure, Design, and Application in Image Compression". PhD Thesis. *University of Wisconsin.* Madison, Wisconsin.

Uğuz H, 2012. "A hybrid system based on information gain and principal component analysis for the classification of transcranial Doppler signals". *Computer methods and programs in biomedicine*. Vol 107, Pp 598–609.

Vaidyanathan PP, 1987. "Quadrature mirror filter banks, M-band extensions and perfect reconstruction techniques". *IEEE ASSP Magazine.Vol* 4. No 3. Pp 4–20

Vaidyanathan PP, 1993; *"Multirate Systems And Filter Banks"*. Prentice Hall signal processing series Electrical engineering. Electronic and digital design. Pearson Education.

Vapnik VN, 1992. "Principles of risk minimization for learning theory". Advances in Neural Information Processing Systems 4. *J.E. Moody, S.J. Hanson and R.P.Lippmann, eds*. Pp. 831-838. San Mateo.

Vetterli M, 1986. "Filter banks allowing perfect reconstruction". *Signal Processing.* Vol 10. No 3. Pp 219–244

Vigário R, 1997. "Extraction of ocular artifacts from EEG using independent component analysis". *Electroenceph. clin. Neurophysiol*., Nº103, Vol 3, Pp 395-404.

Vigário R, Särelä J, Jousmäki V, Hämäläinen M, y Oja E, 2000. "Independent Component Approach to the Analysis of EEG and MEG Recordings". *IEEE Transactions On Biomedical Engineering*, Vol 47, No. 5, Pp 589-593

Vigário R, Jousmäki V, Hämäläinen M, Hari R, Oja E, 1998. "Independent component analysis for identifcation of artifacts in magnetoencephalographic recordings". *In Advances in Neural Information Processing 10*. Pp 229-235.

Vigário R, Särelä J, Jousmäki V, Hämäläinen M, Oja E, 2000. "Independent Component Approach to the Analysis of EEG and MEG Recordings" *IEEE Transations on Biomedical Engineering*, Vol. 47, No 5, Pp 589-593

Vincent E, Gribonval R and Févotte C, 2006. "Performance Measurement in Blind Audio Source Separation," *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 14, No. 4, Pp. 1462–1469.

Vose MD, 1999. *The Simple Genetic Algorithm. Foundations and Theory*.MIT – Massachusett Institute of Technology.

Wang J-S, Chiang W-C, Hsu Y-L, Yang Y-TC, 2012 "ECG arrhythmia classification using a probabilistic neural network with a feature reduction method" *Neurocomputing*; (en prensa).

Weston H. 2010. "A note on standard deviation". Department of Mathematics and Statistics. University of Regina. (en línea) http://mathcentral.uregina.ca/RR/database/RR.09.95/weston2.html. FUC: Nov 2012; (en línea).

Yu S-N, Chou K-T, 2006. Combining Independent Component Analysis and Backpropagation Neural Network for ECG Beat Classification. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Pp: 3090 - 3093

Yuehui C, Yang B, Dong J, 2006. "Time-series prediction using a local linear wavelet neural network" *Neurocomputing*. Vol. 69, pp. 449–465.

Zarzoso V, y Comon P, 2010. "Robust Independent Component Analysis by Iterative Maximization of the Kurtosis Contrast With Algebraic Optimal Step Size". *IEEE Transactions On Neural Networks*, Vol. 21, No. 2, Pp 248-261.

Páginas Web consultadas:

http://www.electrocardiografia.es/derivaciones.html (en línea)

http://www.clinicadam.com/salud/5/001100.html (en línea)