



Universitat Autònoma de Barcelona

Modelos de calibración n-dimensional para lenguas electrónicas

Tesis presentada por:

Raúl Cartas Rosado

Para optar al grado de Doctor en Ingeniería Electrónica

Director: Dr. Manel del Valle Zafra

Tutor: Dr. Marc Porti Pujal

Departamento de Ingeniería Electrónica

Escuela de Ingeniería

Universidad Autònoma de Barcelona

Barcelona, 2012



Dr. Manel del Valle Zafra, Profesor titular del Departamento de Química Analítica de la Universidad Autónoma de Barcelona,

CERTIFICO:

Que la tesis doctoral intitulada ***Modelos de calibración n-dimensionales para lenguas electrónicas***, presentada por Raúl Cartas Rosado para optar al grado de Doctor por la Universidad Autónoma de Barcelona, se ha realizado bajo mi dirección en los laboratorios del Grupo de Sensores y Biosensores del Departamento de Química Analítica de la misma universidad.

Bellaterra, Junio del 2012

Dr. Manel del Valle Zafra
Grupo de Sensores y Biosensores
Unidad de Química Analítica
Universidad Autónoma de Barcelona
Edificio Cn,08193, Bellaterra

Dr. Marc Porti Pujal, Titular de Universidad Numerario del Departamento de Ingeniería Electrónica de la Universidad Autónoma de Barcelona, apruebo la publicación de la tesis de Raúl Cartas Rosado realizada con mi tutoría.

Bellaterra, Junio del 2012

Dr. Marc Porti Pujal
Departamento de Ingeniería Electrónica
Área de Tecnología Electrónica
Universidad Autónoma de Barcelona
08193, Bellaterra

A mi aguerrida compañera de batallas en el sinuoso camino que nos
ha tocado recorrer juntos. Con amor, a mi esposa Delia,

...y a ti Diego Alonso, nuestro hijo.

A quienes nunca me han dejado solo. A mi madre Ana María,
a mis hermanas María de los Ángeles, Reyna, Hermila y Ana Rocío,
y a mi hermano Jesús.

Agradecimientos

La voluntad para conseguir una meta no lo es todo, la financiación desempeña un papel primordial. En primer lugar quiero expresar mi gratitud al **Consejo Nacional de Ciencia y Tecnología (CONACYT)** de México por haberme otorgado la beca que me permitió realizar los estudios de doctorado en España.

Al **Dr. Manel del Valle Zafra**, por varias razones: permitirme realizar en el GSB los estudios de doctorado a pesar de que mi formación académica es diferente a la química, allanar el camino burocrático antes de mi llegada y apoyarme en una época difícil vivida durante mi estancia doctoral.

Al **Dr. Marc Porti Pujal** por aceptar la tutoría de este trabajo realizado en un departamento diferente al de Electrónica.

A mis amigos, en quienes siempre he hallado diversión, consuelo e intercambio de conocimientos y experiencias. En este punto, también agradezco que exista el alfabeto latino para poder listarles ordenadamente de la A a la Z. Van mis agradecimientos en Cataluña al malandro **Edgar Mixcoha**, al panzo **Daniel Ribas**, a la familia **Gil Caballero** (la iaia **María**, el señor marqués **Sergio, Isabel y Álvaro**), a **Kepa Koldo** (a quien por cierto no le gusta el vino) y a **Marc Nadal**, el nen-net.

A los amigos a distancia, por el recuerdo y contacto permanente en México con **Arturo Gutiérrez, Citlali Servín, Elsa Amelia Ronquillo, Filiberto Guzmán, Herlinda Araiza, Jorge Luis Leo, Mayra Vázquez, Roberto Muñoz** y **Sais Flores**.

Si olvidé mencionarte, lo siento, no ha sido intencional.

Índice

Resumen.....	xiii
Summary.....	xv
Introducción	xvii
Objetivos de la tesis.....	xxiii
Estructura de la tesis.....	xxvi
Sección 1	1
Generación de información y propuestas de solución	3
1.1 Sensores electroquímicos.....	4
1.1.1 Sensores amperométricos	5
1.1.2 Sensores voltamperométricos.....	5
1.1.3 Sensores potenciométricos	6
1.2 Sistemas de flujo.....	7
1.2.1 Sistemas de Análisis por Inyección en Flujo (FIA).....	8
1.2.2 Sistemas de Análisis por Inyección Secuencial (SIA).....	8
1.3 Generación de información.....	10
1.4 Propuestas de solución.....	14
Aspectos teóricos generales.....	19
2.1 Redes Neuronales Artificiales	19
2.1.1 Unidad básica de procesamiento.....	20
2.1.2 Taxonomía de las ANNs	21
2.1.3 Red feedforward.....	22
2.2 Retropropagación del error	24
2.3 Análisis multimodal de datos.....	29
2.3.1 PARAFAC y PARAFAC2	30
2.3.2 Tucker3	32
2.3.3 Mínimos Cuadrados Parciales Multi-modo (N-PLS).....	33
Sección 2	35
Redes Neuronales Wavelet	37

3.1 Transformación Wavelet	38
3.1.1 Función wavelet.....	39
3.1.2 Transformada Wavelet Continua.....	42
3.2 Red Neuronal Wavelet	43
3.2.1 Red Neuronal Wavelet con función de activación basada en la norma Euclidiana.....	48
3.2.2 Red Neuronal Wavelet con función de activación basada en el producto tensorial.....	54
Redes Neuronales B-spline.....	61
4.1 Técnicas de calibración univariable	62
4.1.1 Interpolación polinomial.....	62
4.1.2 Polinomios de Lagrange.....	63
4.1.3 Interpolación de Hermite	63
4.2 Funciones spline	64
4.2.1 Funciones B-splines	65
4.2.2 B-splines multivariable.....	69
4.3 Red B-spline mono-dimensional.....	73
4.4 Red B-spline multi-dimensional	77
Redes Neuronales Spline Catmull-Rom.....	83
5.1 Splines cardinales y Catmull-Rom.....	83
5.2 Redes Neuronales Spline Catmull-Rom.....	84
Splines y B-splines Adaptativas de Regresión Multivariable (MARS y B-MARS)	91
6.1 Particionamiento recursivo	91
6.2 MARS	93
6.3 B-MARS.....	97
Preprocesamiento y Modelado con Redes Neuronales	101
7.1 Transformada Wavelet Discreta.....	101
7.1.1 Bancos de filtros y análisis multiresolución	103
7.2 Preprocesamiento con DWT + ANNs.....	107
Sección 3.....	113
Pruebas y resultados.....	115
8.1 Determinación de compuestos oxidables usando una WNN con función de transferencia basada en la norma-2.....	116
8.2 Estructuras de WNNs paralelas en la determinación de compuestos fenólicos para la monitorización de contaminantes en agua	132

8.3 Redes neuronales Wavelet basadas en el producto tensorial	146
8.4 Modelos MARS para la predicción de parámetros de control en la producción de azúcar a partir de la remolacha azucarera.....	156
8.5 Modelos B-MARS en la predicción de parámetros de control en la producción de azúcar a partir de la remolacha azucarera.....	165
8.6 Aplicación del preprocesamiento con la Transformada Wavelet Discreta y modelado con Redes Neuronales Artificiales	176
8.6.1 Cuantificación de mezclas binarias de metales pesados a partir de la respuesta cinética de un sensor potenciométrico. Doble caso de estudio.....	177
8.6.2 Cuantificación de mezclas ternarias de metales pesados a partir de la respuesta cinética de una matriz de sensores potenciométricos.....	186
8.7 Construcción de redes neuronales B-spline y Catmull-Rom para determinar compuestos fenólicos en la monitorización de la calidad del agua.....	196
8.8 Modelos N-PLS2 en el procesamiento multimodo de datos	203
8.8.1 Lengua electrónica potenciométrica para la cuantificación de metales pesados.....	203
8.8.2 Resolución de mezclas de compuestos fenólicos antioxidantes usando una bio-lengua electrónica del tipo voltamperométrico	209
Conclusiones	217
Artículos publicados	223
Artículo 1	225
Artículo 2	236
Artículo 3	244
Apéndices	253
Apéndice 1	255
Apéndice 2	257
Apéndice 3	259
Referencias	261

Resumen

Las herramientas computacionales que se describen en esta tesis representan posibles alternativas de solución en la construcción de modelos de calibración multivariable a partir de datos obtenidos con arreglos de sensores electroquímicos. Tanto el trabajo experimental como las aplicaciones computacionales están dirigidos a la construcción de lenguas electrónicas de los tipos potenciométrico y voltamperométrico.

Las propuestas de solución que aquí se presentan están basadas en técnicas computacionales diseñadas para explorar grandes bases de datos en la búsqueda de patrones consistentes y/o relaciones sistemáticas entre variables, que permitan posteriormente aplicar estos modelos a nuevos datos con el fin de generar predicciones o estimaciones de resultados esperados.

Algunas de las herramientas se implementaron con redes neuronales tipo perceptrón multicapas y diferentes funciones de transferencia en las neuronas de la capa oculta. Las funciones de activación sigmoideas comúnmente usadas en las redes neuronales se sustituyeron por funciones más complejas y de poco (o nulo) uso en el área química. Para hacer compatible la estructura de la mayoría de los datos usados en esta tesis, con las entradas de las redes neuronales, se hizo un tratamiento previo de la información electroquímica usando técnicas de procesamiento mono- o multi-modales para reducir el número de variables y dimensiones.

Además de las propuestas basadas en estructuras de redes neuronales, también se ha planteado la construcción de modelos a partir de funciones base de los tipos spline truncada y B-spline. La primera se conoce como *Splines Adaptativas de Regresión Multivariable (MARS)* y la segunda como *B-splines Adaptativas de Regresión Multivariable (B-MARS)*. Adicionalmente a las herramientas anteriormente descritas e implementadas como propuestas de

solución, también se construyeron exitosamente modelos de calibración usando la regresión multimodo por mínimos cuadrados parciales (*N-PLS*).

Summary

The computational tools described in this thesis are meant to be alternative solutions to build multivariate calibration models from multi-way data obtained with arrays of electrochemical sensors. Both experimental and computational applications described herein are aimed to build electronic tongues of potentiometric and voltammetric types.

The solution proposals are based on computational techniques designed to explore large databases in search of consistent patterns and/or systematic relationships between variables, allowing then to apply these models to new data to predict or estimate expected results.

Some of the tools were implemented using multilayer perceptron neural networks with complex transfer functions (of little or no use in the chemical area) in the hidden layer neurons. To make compatible the type of structure of most of the data used in this thesis with the input of the neural networks, the electrochemical information was pretreated using mono- or multi-dimensional processing techniques in order to reduce the number of variables and dimensions.

In addition to the structures based on neural networks, we also propose to build models using base functions of the truncated spline and B-spline types. The first is known as *Adaptive Regression Splines Multivariable (MARS)* and the second as *B-splines Multivariate Adaptive Regression (B-MARS)*. In addition to the tools described above and implemented as proposed solutions, we also built successfully calibration models using multi-way partial least squares regression (*N-PLS*).

Introducción

Los sistemas bio-inspirados nacen de la aplicación de conceptos de inspiración biológica al diseño de sistemas analíticos. Los intentos de emulación del funcionamiento de los seres vivos se desarrollan en un entorno multidisciplinario que agrupa diferentes áreas de la ingeniería para aspirar a conseguir auténticos sistemas electrónicos dotados de sentidos artificiales que permitan facilitar un sinnúmero de tareas y resolver problemas hasta ahora no resueltos.

Una lengua electrónica es uno de tales sistemas bio-inspirados basado en el sentido del gusto que usa sensores de una manera novedosa para cuantificar, clasificar o identificar sustancias en medios líquidos [1]. Una definición ampliamente aceptada la describe como un instrumento analítico que consta de un arreglo de sensores químicos de baja selectividad y especificidad parcial (sensibilidad cruzada) a los diferentes compuestos de una solución, acoplado a una apropiada herramienta quimiométrica para el procesamiento de la información, capaz de reconocer de manera cuantitativa o cualitativa los componentes de soluciones sencillas y complejas [2, 3]. La confiabilidad de las lenguas electrónicas en tareas de reconocimiento (clasificación, identificación o discriminación) ha ido demostrándose durante los últimos años [4, 5]. Estos sistemas resultan apropiados para resolver problemas analíticos en donde exista traslape de señales debido a la presencia de diferentes especies y por tanto, no pueda obtenerse directamente un modelo de calibración.

El primer requisito en la construcción de una lengua electrónica es contar con un arreglo apropiado de sensores que, aunque genéricos, respondan a las diferentes especies químicas e intervalos dinámicos de concentraciones que se espera encontrar en el compuesto a analizar. Los sensores que se pueden usar en las lenguas electrónicas están basados en diferentes tipos de detección pero los más empleados son los electroquímicos y ópticos [1]. Debido a la falta de selectividad los sensores generan señales complejas de respuesta cruzada

que contiene información relacionada con los diferentes compuestos presentes en la sustancia más otras características adicionales, es por ello que la segunda parte importante en de una lengua electrónica es la etapa de procesamiento [5, 6, 7, 8].

En el análisis de las respuestas adquiridas con el arreglo de sensores se usan métodos de reconocimiento de patrones o técnicas de calibración multivariable debido a que las señales que se adquieren son producidas en soluciones con múltiples especies [9]. Como resultado de la naturaleza del medio en el que operan los sensores, las señales adquiridas son complejas en la mayoría de los casos y no pueden ser descritas usando ecuaciones teóricas como las usadas para describir la difusión o el flujo de corriente por estar limitadas a modelar un solo analito. Este esquema de operación de los sensores no representa un inconveniente ya que las soluciones reales de interés son de naturaleza multi-componente y las mediciones que se hacen en ellas pueden verse afectadas, además, por el comportamiento de los electrodos, las reacciones entre los analitos que la componen y otros interferentes [9, 10].

La complejidad de señales adquiridas con un conjunto de sensores de baja selectividad y sensibilidad cruzada puede resolverse con herramientas quimiométricas. El desarrollo de métodos quimiométricos de los últimos años en el área de la química analítica se ha enfocado al análisis multivariable, esto es debido a que se obtiene más información de un análisis cuando se toman en consideración múltiples variables de manera simultánea que cuando se analiza cada variable de manera independiente. Este incremento en la cantidad de información extraída se traduce como una ventaja que se conoce como *ventaja multivariable* [11, 12]. Cuando analizamos variables por separado despreciamos la correlación que existe entre la variable analizada y el resto de variables que afectan el fenómeno y por tanto perdemos información. En contraste con el análisis univariable, cuando analizamos múltiples variables de manera simultánea se obtiene información adicional que se presenta en forma de correlación entre ellas. Si comparamos los métodos de análisis univariable contra los métodos multivariable éstos últimos resultan más ventajosos ya que permiten reducir el nivel de ruido, discriminar una variable de interés a partir de medidas parcialmente selectivas e identificar falsas muestras, entre otras ventajas más [12].

Muchos de los avances en la quimiometría se pueden atribuir a la introducción de métodos de análisis y calibración desarrollados en otras áreas de la ciencia y conocidos por bastante tiempo, por ejemplo el método de Análisis por Componentes Principales (PCA) y la Regresión por Mínimos Cuadrados (PLSR). El primero de estos ejemplos se usa en el área química para realizar un análisis cualitativo de la solución mientras que el segundo para un análisis cuantitativo. Dependiendo del objetivo final del experimento desarrollado se elige uno u otro tipo de análisis, el primero se usa para identificar los constituyentes de la muestra analizada y el segundo para determinar la concentración del o los analitos de interés en la solución [13]. La relación matemática que pueda obtenerse entre las mediciones y la(s) concentración(es) de interés con estos tipos de análisis reciben los nombres de modelos de clasificación y calibración, respectivamente.

En el Grupo de Sensores y Biosensores (GSB) de la Universitat Autònoma de Barcelona se han construido lenguas electrónicas con sensores de los tipos potenciométrico y voltamperométrico que permitan realizar tareas de clasificación o cuantificación. Para conseguir estos objetivos también se necesita usar métodos de calibración que permitan la identificación de los analitos de interés en presencia de elementos interferentes.

Las herramientas de análisis multivariable que se explican en esta tesis para el tratamiento de datos adquiridos con arreglos de sensores electroquímicos están enfocadas a la construcción de modelos de calibración y son alternativas a las actuales herramientas usadas en la quimiometría. Las propuestas que se han hecho están basadas en la minería de datos, con orígenes en la neurofisiología o la estadística y que se han usado poco o nada en la quimiometría. La minería de datos es un procedimiento analítico basado en técnicas computacionalmente intensivas que ha sido diseñado para explorar grandes bases de datos en la búsqueda de patrones consistentes y/o relaciones sistemáticas entre variables, que permitan posteriormente aplicar estos modelos a nuevos datos con el fin de generar predicciones o estimaciones de resultados esperados. Cuando la minería se aplica en datos que contienen relaciones complejas es capaz de identificar relaciones que no son aparentes de otra manera [14]. Estas complejidades han sido un reto para los procedimientos analíticos tradicionales como la regresión lineal o el PLSR anteriormente mencionado.

Dentro de la minería de datos, las redes neuronales son una técnica muy popular debido a su habilidad de modelar funciones no-lineales, además que ha demostrado ser particularmente efectiva en el manejo de datos con interacciones complejas. Por mencionar un ejemplo, las redes perceptrón multicapa (junto con las redes con funciones base radiales) son arquitecturas populares en la aproximación de funciones e identificación de sistemas debido a sus propiedades de aproximación universal, generalización y capacidades de aprendizaje [15]. A pesar de sus bondades, las redes neuronales no son ideales para todos los conjuntos de datos. Algunas investigaciones han demostrado que los modelos obtenidos con las habituales herramientas de regresión son superiores a las redes neuronales cuando se conoce la relación funcional entre las variables independiente y dependiente [16, 17]. Quizás la mayor desventaja de las redes neuronales es que no es fácil para quienes las usan explicar las interacciones que existen entre las variables debido a las funciones complejas que la forman, es por ello que las redes neuronales son tratadas como cajas negras entre las variables de entrada y salida. Además de las redes neuronales, existen otras técnicas de minería de datos que permiten obtener modelos de predicción claros y entendibles.

Algunas de las herramientas desarrolladas y explicadas a continuación están basadas en redes neuronales tipo perceptrón multicapa, con diferentes funciones de transferencia en las neuronas de la capa oculta. Las funciones de activación sigmoidales comúnmente usadas en las redes neuronales se han sustituido por funciones más complejas y de poco (o nulo) uso en el área química. La mayoría de los datos que se manejan en esta tesis son tensores de tercer orden, por lo que algunos de los modelos de redes que se han propuesto precisan de la reducción en la dimensión de los datos de entrada antes de ser entrenadas. Para alcanzar este objetivo se hizo un pretratamiento de la información electroquímica usando técnicas de procesamiento mono- o multimodales para reducir el número de variables. Los objetivos que se persigue con el pre-procesamiento son (i) extraer características relevantes de los registros y (ii) reducir la dimensión de las señales originales para evitar la *maldición de la dimensionalidad* [18].

Además de las propuestas basadas en estructuras de ANNs, también se ha planteado la construcción de modelos de calibración a partir de funciones base de los tipos spline truncada y B-spline. La primera se conoce como

Splines Adaptativas de Regresión Multivariable (MARS) y la segunda como *B-splines Adaptativas de Regresión Multivariable (B-MARS)*. Adicionalmente a las herramientas anteriormente descritas e implementadas como propuestas de solución, también se construyeron exitosamente modelos de calibración usando la regresión multimodo por mínimos cuadrados parciales (*N-PLS*).

Objetivos de la tesis

El objetivo que se ha perseguido en esta tesis es proponer e implementar herramientas alternativas de calibración multivariable para el tratamiento de datos de tercer orden recolectados usando arreglos matriciales multi-electrodo. El uso final de las herramientas es el desarrollo de lenguas electrónicas de los tipos potenciométrico y voltamperométrico.

Para hacer el tratamiento de los datos y obtener los modelos de calibración se ha propuesto el uso de la transformación wavelet, funciones spline, B-spline y redes neuronales artificiales. Estas herramientas no se usaron de manera independiente, sino fusionadas en diferentes estructuras que dieron origen a:

1. Redes Neuronales Wavelet. Son redes con estructura perceptrón multicapa donde las funciones de activación sigmoidales en la capa oculta son reemplazadas por funciones wavelet. En este rubro se construyeron dos subcategorías: una basada en wavelets multi-dimensionales basadas en el producto tensorial de wavelets mono-dimensionales y otra basada en la norma vectorial de funciones wavelet multi-dimensionales.
2. Redes Neuronales B-spline y spline. Estas redes son el resultado de usar funciones B-spline como funciones de activación en las neuronas de la capa oculta. Además de las funciones B-spline también se usaron funciones de activación spline Catmull-Rom.
3. Splines de Regresión Multivariable Adaptativa. Usados para construir modelos de regresión a partir de funciones spline truncadas. La obtención del modelo se hace de manera similar al particionamiento recursivo. Adicionalmente a las funciones spline truncadas también se usaron funciones B-spline de grados 1, 2 y 3.

Además de las herramientas anteriormente mencionadas, se construyeron también modelos de calibración usando redes neuronales entrenadas con datos obtenidos de las descomposiciones PARAFAC, Tucker3 y la Transformada Wavelet Discreta, además de modelos usando la regresión multimodo por mínimos cuadrados (N-PLS2, por sus siglas en inglés).

Estructura de la tesis

Esta tesis está estructurada en 8 Capítulos que forman parte de 3 secciones mayores.

La primera sección está formada por los Capítulos 1 y 2 y está dedicada a aspectos genéricos teóricos y de experimentación que son comunes a toda la tesis. En el Capítulo 1 se ofrece al principio un panorama general de los tipos de sensores y los sistemas de flujo que se emplean en los análisis electroquímicos con la finalidad de describir a continuación el sistema experimental usado en los laboratorios del Grupo de Sensores y Biosensores y la manera en como se genera la información de carácter multimodo; este panorama sirve a su vez de antesala a una breve descripción esquematizada de las propuestas de solución desarrolladas en esta tesis. En el Capítulo 2 se hace mención a la teoría generalizada de las redes neuronales artificiales, se describe el desarrollo de la técnica del gradiente descendiente usado en el entrenamiento de las redes, y se presentan técnicas de análisis multimodo desarrolladas por terceros que se usaron en esta tesis para construir modelos de calibración y reducir la dimensionalidad de los datos antes de entrenar algunas de las redes.

La segunda sección está formada por los Capítulos 3 a 8. En ellos se expone en detalle el desarrollo de las herramientas que se proponen para la solución de la problemática expuesta en el Capítulo 2. Cada uno de los desarrollos va precedido de las bases teóricas que las sustentan. El Capítulo 3 corresponde a las Redes Neuronales Wavelet, el Capítulo 4 a las Redes Neuronales B-spline, el Capítulo 5 a las Redes Neuronales Catmull-Rom, el Capítulo 6 a los modelos de regresión MARS y B-MARS y el Capítulo 7 a la combinación de preprocesamiento de la información seguida del modelado de la información procesada usando Redes Neuronales Artificiales.

La tercera sección comprende el Capítulo 9 y está enfocado a la presentación de los resultados que se obtuvieron aplicando las herramientas descritas en la Sección 2 a diferentes conjuntos de datos obtenidos con experimentos electroquímicos enfocados a la construcción de lenguas electrónicas. En este capítulo también se muestran los resultados obtenidos con la regresión multimodo por mínimos cuadrados parciales (N-PLS) presentada en el Capítulo 2.

Finalmente, se presentan como anexos los algoritmos usados en el desarrollo de esta tesis para que sirvan como complemento y referencia a trabajos futuros que continúen con la construcción de modelos de calibración.

Sección 1

1

Generación de información y propuestas de solución

Las lenguas electrónicas son sistemas apropiados para resolver problemas analíticos donde exista traslape de señales debido a la presencia de diferentes especies en la solución bajo análisis y por tanto, la calibración sea difícil. Cuando se construye una lengua electrónica, la parte que normalmente precisa de mayor atención e inversión de tiempo es la generación de la información que se usa en la construcción de los modelos de calibración. En este contexto, el uso de técnicas de flujo mecanizadas y automatizadas representan la mejor opción en la generación de los datos usados para modelar y calibrar una lengua electrónica, ya que permiten agilizar la preparación de las mezclas de analitos necesarios para el desarrollo experimental, el manejo de las muestras preparadas y la adquisición de las medidas. El alto rendimiento de las técnicas de flujo permite el procesamiento de un gran número de muestras en un pequeño periodo de tiempo.

A continuación se expone la teoría relacionada con los sensores químicos y las técnicas de flujo que se usan en la automatización de muestras, así como la naturaleza multimodal de los datos con los que se trabajan en esta tesis y las propuestas de solución para construir modelos de calibración. Los temas relacionados con los sensores químicos y técnicas de flujo no han sido desarrollados en profundidad, solamente se han tratado los aspectos relevantes al desarrollo de esta tesis por lo que se invita a consultar las referencias correspondientes en caso que se desee ahondar en los temas.

1.1 Sensores electroquímicos

Los sensores químicos son dispositivos formados por un elemento transductor cubierto por una capa de reconocimiento químico que entra en contacto con la sustancia química a analizar. Los cambios químicos cuantitativos o cualitativos resultantes de la interacción entre el analito de interés y el dispositivo sensor son transformados por el elemento transductor en una señal analítica útil, comúnmente del tipo eléctrico [19, 20, 21]. El propósito final de los sensores es proporcionar información confiable en tiempo real acerca de la composición química del entorno que lo rodea [20, 21].

Dentro de la familia de los sensores químicos, los sensores electroquímicos forman la sub-familia más grande y más antigua, la cual se caracterizan por usar electrodos metálicos como elementos de transducción [20, 22]. En esencia, estos sensores forman una celda electroquímica que funciona en configuración de dos o tres electrodos dependiendo de la técnica electroquímica empleada (Fig. 1.1) [23, 24]. La configuración básica de la celda siempre involucra un electrodo de trabajo (WE) y un electrodo de referencia (RE). Para el caso de la configuración de tres electrodos la celda incorpora además un electrodo auxiliar (AE) (Fig. 1.1). Los sensores electroquímicos pueden usarse para realizar mediciones tanto en estado estacionario como en estado transitorio. La corriente o voltaje que se les aplica varía de acuerdo al modo de operación, el cual se elige para mejorar la sensibilidad y selectividad de un sensor en particular [23].

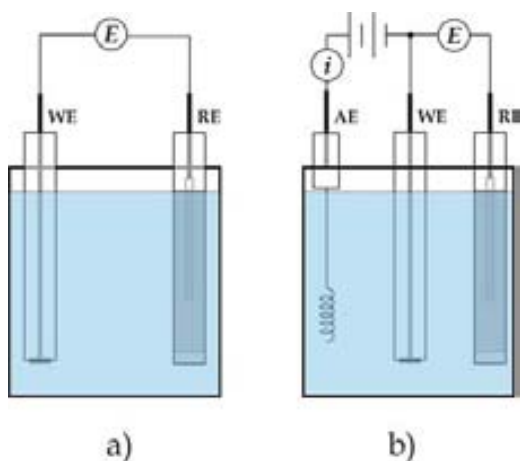


Fig. 1.1 Representación esquemática de las celdas de a) dos y b) tres electrodos.

En cuanto a la clasificación, los sensores electroquímicos pueden catalogarse como sensores de conductividad/capacitancia, potenciométricos, amperométricos y voltamperométricos [23], de los cuales, los más usados en soluciones electrolíticas líquidas son los tres últimos mencionados.

1.1.1 Sensores amperométricos

Los sensores amperométricos basan su funcionamiento en la detección de las especies electroactivas involucradas en el proceso de reconocimiento químico. La configuración más habitual para el uso de estos sensores es el arreglo de 3 electrodos mostrado en el apartado b) de la Fig. 1.1.

En este tipo de sensores se usa la técnica amperométrica de medición, la cual consiste en aplicar un potencial fijo al WE de la celda electroquímica (respecto al RE) y medir la corriente en función del tiempo debida al proceso de oxidación o reducción provocado. El voltaje aplicado entre los electrodos de trabajo y referencia es la fuerza motriz que provoca la transferencia de electrones de las especies electroactivas, y la corriente producida guarda relación con la concentración del analito mediante la ley de Faraday y la ley de transporte de masa [19, 24]. La característica clave para el uso de estos electrodos es que la corriente medida es proporcional a la concentración del analito, siempre y cuando el movimiento de las especies electroactivas sea constante [25].

1.1.2 Sensores voltamperométricos

La relación entre el potencial aplicado y la corriente que se genera en una celda electroquímica de tres electrodos es la base del funcionamiento de los sensores voltamperométricos. Los sensores amperométricos (que se basan en el mismo principio de funcionamiento) pueden considerarse como una subcategoría de los voltamperométricos [23]. La diferencia entre ambos es la técnica de uso, en los sensores amperométricos se aplica un potencial fijo al WE y se mide la corriente generada, en los voltamperométricos se aplica un barrido de voltaje al WE y se monitoriza la corriente resultante que fluye a través de la celda electroquímica [25, 26]. Ambas técnicas se consideran

activas porque se aplica un potencial que fuerza la oxidación o reducción de una especie electroactiva en la superficie del electrodo.

Los sensores voltamperométricos pueden operar con barridos de voltaje lineal o cíclico. En la voltamperometría de barrido lineal el voltaje aplicado se incrementa a velocidad constante desde un valor inicial hasta un límite máximo predefinido. La voltamperometría cíclica es similar a la de barrido lineal, excepto que el voltaje aplicado regresa al valor inicial. La diferencia entre las curvas obtenidas con ambos modos de operación radica en que en la primera se muestra un pico de corriente en el voltaje donde ocurre la oxidación o reducción, mientras que en la segunda se generan picos de corriente que corresponden a las reacciones de oxidación y reducción [23]. La sensibilidad de los métodos voltamperométricos es comúnmente muy alta, la selectividad, por el contrario, es pobre en la mayoría de los casos ya que todas las especies presentes en la solución que sean electroquímicamente activas para el potencial aplicado contribuirán a la corriente medida [27].

1.1.3 Sensores potenciométricos

En los sensores potenciométricos la información analítica se obtiene convirtiendo el proceso de reconocimiento en un potencial, el cual es proporcional a la concentración de la especie en el evento de reconocimiento. A diferencia de las mediciones voltamperométricas, las potenciométricas se hacen bajo condición de equilibrio electroquímico, es decir, sin flujo de corriente a través de los electrodos de trabajo y referencia que forman la celda [22, 25, 28].

Si existen múltiples especies oxidándose o reduciéndose durante la medición entonces el potencial medido no puede ser usado para cuantificar el analito de interés [23]. Para mejorar la selectividad iónica de los electrodos potenciométricos se recubre su superficie con una membrana funcional específica conocida también como membrana con permeabilidad selectiva [22]. El electrodo recubierto recibe el nombre Electrodo Selectivo a Iones (ISE). Estos electrodos son baratos, tienen respuesta rápida, un amplio intervalo lineal de trabajo, no son destructivos y son compatibles con el análisis en línea [21].

Los ISEs son capaces de medir la actividad de una especie iónica en particular, también conocida como ion principal, mediante el diseño de la membrana que permita de manera selectiva a los iones del analito de interés difundirse o migrar a través de la membrana hasta alcanzar el electrodo (Fig. 1.2). La membrana en cuestión es usualmente no porosa, insoluble en agua y mecánicamente estable, y de acuerdo al material usado para su fabricación los ISE pueden clasificarse en tres grupos: electrodos de vidrio, líquidos o sólidos [20]. Debe resaltarse que los ISEs sensan la *actividad* en lugar de la concentración de iones en una solución. El término *actividad* se refiere a la concentración efectiva de un ion en particular en una mezcla de sustancias. Es posible relacionar actividad y concentración si se fijan determinadas condiciones de operación, como temperatura y fuerza iónica.

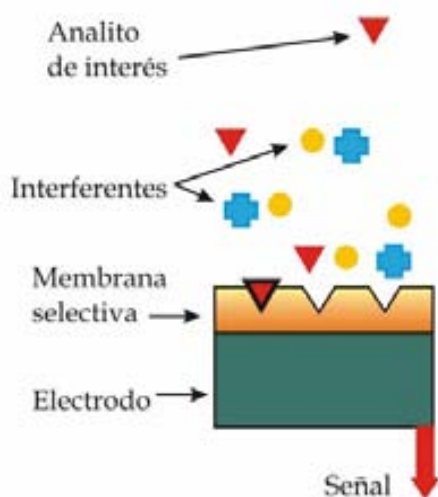


Fig. 1.2. La membrana selectiva a iones es permeable solamente al analito de interés, impidiendo la difusión de iones interferentes hacia el electrodo.

1.2 Sistemas de flujo

La primera técnica de flujo fue el Análisis de Flujo Segmentado (SFA) [32, 29] propuesta por Skeegs en 1957, pero debido a los inconvenientes que presentaba, la técnica fue gradualmente reemplazada por técnicas de flujo continuas como el *Análisis por Inyección en Flujo (FIA)* o el *Análisis por Inyección Secuencial (SIA)*.

Los componentes básicos de un sistema de análisis de flujo son: una bomba de líquidos (tipo peristáltica, pistón de dos vías o micro-bomba), un

dispositivo para inyectar la muestra, tubería plástica (serpentín de reacción) y un detector [29]. La técnica empleada en estos sistemas consiste en la inyección de una muestra en un tubo de diámetro pequeño, en donde se agregan también los reactivos, los cuales se mezclan con la muestra antes de ser transportada al detector. El uso de los sistemas de análisis de flujo aumenta la eficiencia de los laboratorios porque permite una mayor frecuencia de análisis y un mínimo manejo y consumo de muestras, lo que a su vez tiene como consecuencia la minimización en la generación de residuos [30].

El desarrollo experimental que se requiere para construir un modelo de calibración multicomponente en sistemas de flujo es complejo y tardado debido a la elevada cantidad de estándares o muestras que se requieren [31]. La necesidad de automatizar la preparación de muestras, lo que simplifica el proceso y mejora la reproducibilidad de los experimentos, lo que hizo posible el nacimiento de los *sistemas de análisis de flujo* [32].

1.2.1 Sistemas de Análisis por Inyección en Flujo (FIA)

La técnica de Análisis por Inyección de Flujo (FIA), conocida como primera generación de técnicas de flujo, fue creada en 1975 [33] por Ruzicka y Hansen. El esquema básico de un sistema FIA se muestra en la Fig. 1.3. El sistema está formado por una bomba peristáltica, una válvula de inyección, un serpentín de reacción y un detector final. En esta técnica la muestra se inyecta en el flujo de un líquido portador que la transporta por el sistema colector hasta un reactor tubular, donde se mezcla con el flujo continuo del reactivo antes de alcanzar el detector, donde se obtiene una respuesta dinámica.

El sistema FIA elimina las desventajas del sistema SFA, además de disminuir considerablemente el consumo de reactivos y muestras debido a la reducción en el diámetro de la tubería plástica utilizada [34].

1.2.2 Sistemas de Análisis por Inyección Secuencial (SIA)

En 1990, Ruzicka desarrolló la segunda generación de sistemas de flujo conocida como Análisis por Inyección Secuencial (SIA) [35]. Un sistema SIA está formado básicamente por una bomba bi-direccional tipo pistón de un solo

canal y elevada precisión, una válvula multi-posición, un tubo de retención, un serpentín de reacción y un detector. El esquema de implementación del sistema SIA se muestra en la Fig. 1.4. La técnica se basa en la aspiración secuencial de volúmenes precisos de muestras y reactivos que son mezclados por dispersión en el serpentín de retención; el flujo de la mezcla es posteriormente invertido y finalmente bombeado a través de la bobina de reacción hacia el detector [30, 36].

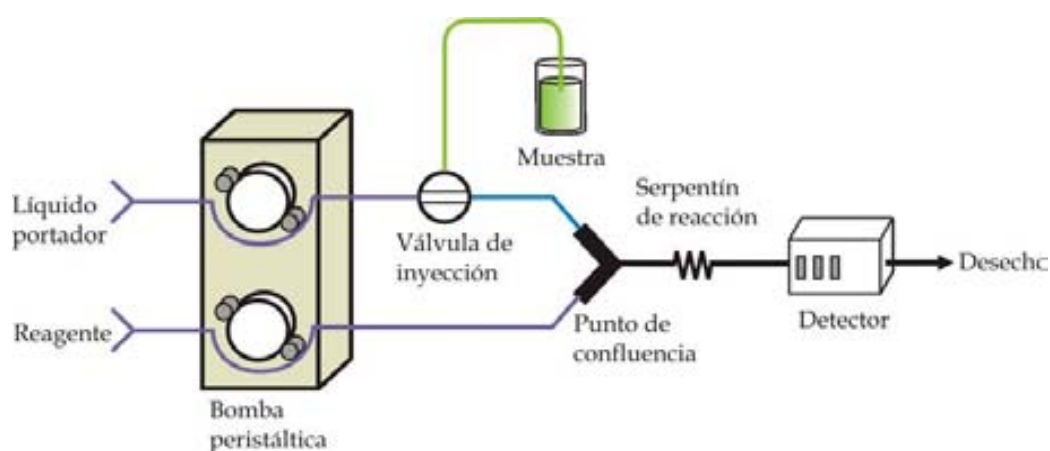


Fig. 1.3. Esquema básico de implementación de un sistema FIA.

La configuración básica del sistema SIA también puede tener otros componentes que permiten pre-tratamientos tales como separaciones de las muestras, preconcentraciones (por ejemplo, extracción líquido-líquido, precipitación/co-precipitación en reactores auxiliares o extracción de fase sólida en columnas empaquetadas) [37].

El sistema FIA elimina las desventajas del sistema SFA, además de disminuir considerablemente el consumo de reactivos y muestras debido a la reducción en el diámetro de la tubería plástica utilizada [38].

En comparación con la técnica FIA, SIA permite usar la misma tubería para realizar una amplia variedad de ensayos, tiene dispositivos de bombeo más robustos y la precisión en los volúmenes aspirados es la principal ventaja de esta técnica, ya que disminuye el consumo de reactivos y muestras. La principal desventaja es que la frecuencia de muestreo es menor que la del FIA [39].

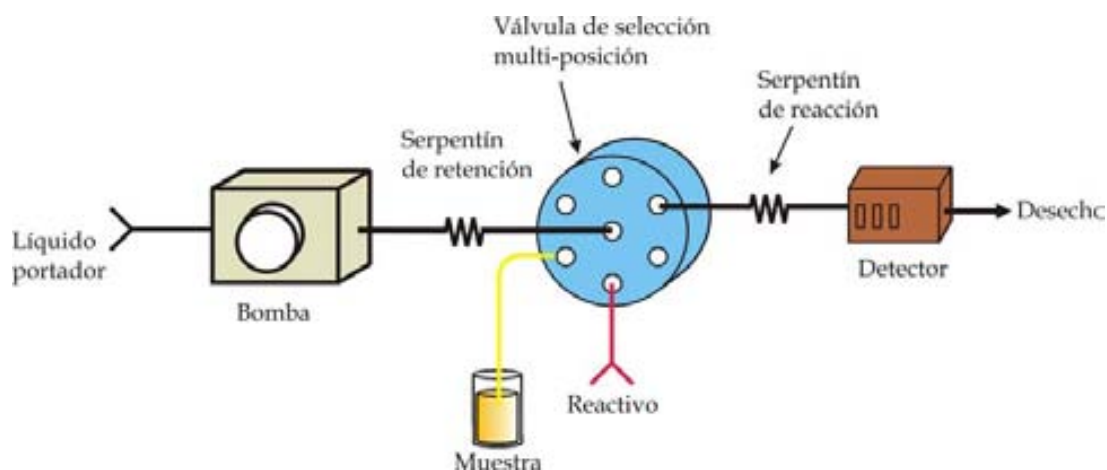


Fig. 1.4. Esquema básico de implementación de un sistema SIA.

Los sistemas FIA y SIA no se contraponen, por el contrario, pueden considerarse complementarios, los dos son adecuados para usarse con una amplia variedad de técnicas de detección tales como espectroscopia atómica, espectrofotometría, conductimetría, potenciometría o voltamperometría [40, 41, 42, 43, 44, 45, 46].

El acoplo de los sistemas de flujo con sensores electroquímicos de baja especificidad y selectividad cruzada, junto con el posterior tratamiento quimiométrico de las señales adquiridas con estos sensores se ha usado en el GSB para construir lenguas electrónicas.

1.3 Generación de información

La información extraída del análisis electroquímico de muestras en el laboratorio depende en gran parte del equipo y la metodología de medición usados [47]. En el *Grupo de Sensores y Biosensores (GSB)* de la UAB se cuenta con un sistema SIA que automatiza el proceso de preparación de muestras y las mediciones. Dicho sistema, esquematizado en la Fig. 1.5, está formado de dos partes principales: el sistema de flujo y el sistema de medición y adquisición de datos. El sistema de flujo está formado por una microbureta que tiene acoplada una jeringa de 5ml (este conjunto asegura la exactitud en el manejo de las soluciones y su repetibilidad), una celda de mezcla usada para homogeneizar las soluciones, un serpentín de retención y otro de reacción. El

sistema de medición y adquisición está formado por una celda que porta los sensores electroquímicos y al electrodo de referencia, además de un sistema de adquisición de datos de 8 canales construido en el laboratorio del GSB. Tanto la microburetra como la válvula y el sistema de adquisición están controlados por ordenador. Los detalles relacionados con las marcas, modelos y tamaño de los elementos hidráulicos que forman al sistema SIA se pueden encontrar en publicaciones previas del grupo [48, 49]. En cuanto a los sensores electroquímicos usados en el GSB para la construcción de lenguas electrónicas, éstos son de los tipos potenciométrico y voltamperométrico y de naturaleza variada. Los detalles relacionados con sus técnicas de fabricación quedan fuera del alcance de esta tesis, sin embargo, en la sección experimental se hará una breve descripción de los mismos.

Cuando se hace el análisis electroquímico de una muestra de laboratorio se puede medir una variable de manera puntual (e.g. el potencial de equilibrio que se relaciona con la concentración de iones), una variable evolutiva dependiente de un parámetro de control (e.g. un voltamperograma) o un conjunto de variables evolutivas usando un sistema multi-sensor (e.g. un conjunto de voltamperogramas medidos con varios electrodos). Como resultado de estas mediciones obtenemos un valor escalar, un vector o una matriz de valores, a los cuales se les conoce como *datos de orden cero, uno y dos*, respectivamente [50]. Esta clasificación no es exclusiva de la electroquímica y se usa también para identificar al instrumento de medición y las herramientas matemáticas usadas para construir los modelos de clasificación y/o calibración [51]. La terminología usada para referirnos a estos datos adquiridos con los sensores está directamente relacionada con el concepto de tensores. En un enfoque clásico, los tensores son definidos como objetos matemáticos que se pueden representar por arreglos n -dimensionales de valores escalares, vectoriales o matriciales. El orden del tensor es el número de modos o espacios generados por él y la dimensionalidad es el número de elementos en cada orden del tensor. Usando esta definición se puede concluir que un escalar es un tensor de orden cero, un vector es un tensor de orden uno y una matriz es un tensor de orden dos (Fig. 1.6) [52]. Los escalares serán identificados en esta tesis con letras minúsculas itálicas (x), los vectores con letras minúsculas en negritas (\mathbf{x}) y las matrices con letras mayúsculas en negritas (\mathbf{X}).

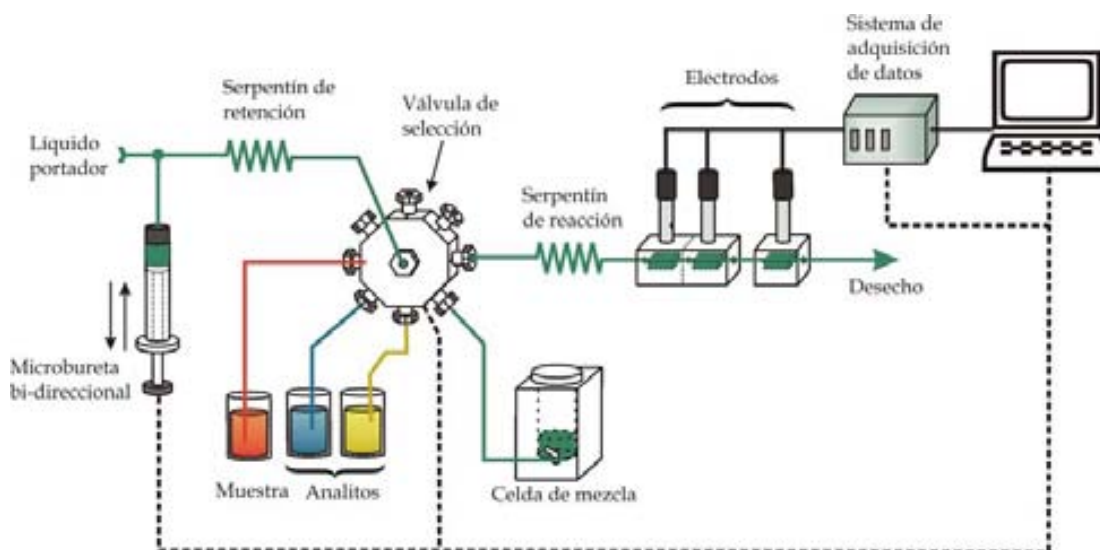


Fig. 1.5. Esquema del sistema SIA existente en el laboratorio del GSB. Los elementos que forman los sistemas flúidico y de medición están descritos en el texto. El sistema se controla por ordenador, lo que añade robustez en la reproducibilidad de los experimentos.

Cuando se usan sensores potenciométricos, las medidas que se realizan son del tipo unipolar y se realizan sobre el electrodo de trabajo con el electrodo de referencia puesto a tierra. Las lecturas de potencial que se hacen en la celda electrolítica tienen un comportamiento que se rige por la ecuación de Nernst y son proporcionales a las actividades de todos los iones presentes en la solución [23, 25-27, 29]. Los registros potenciométricos son comúnmente tensores de orden cero (valores puntuales) que se miden una vez que se ha alcanzado el estado de equilibrio electroquímico en la solución. Recientemente se ha reportado el uso de registros potenciométricos adquiridos a partir de la respuesta transitoria de los sensores cuando son sometidos a la inyección de una muestra de la solución a medir usando un sistema automático de flujo [53, 54, 55]. Esta característica representa una ventaja debido a que el contenido de información de un sensor se ve enriquecido en su componente dinámica, lo que mejora la capacidad de discriminación de un sensor para una especie primaria en presencia de interferentes. Para estos casos, los registros son medidas evolutivas de potencial respecto al tiempo. Debido a la naturaleza de la medición, los registros potenciométricos de señales dinámicas se clasifican como tensores de primer orden.

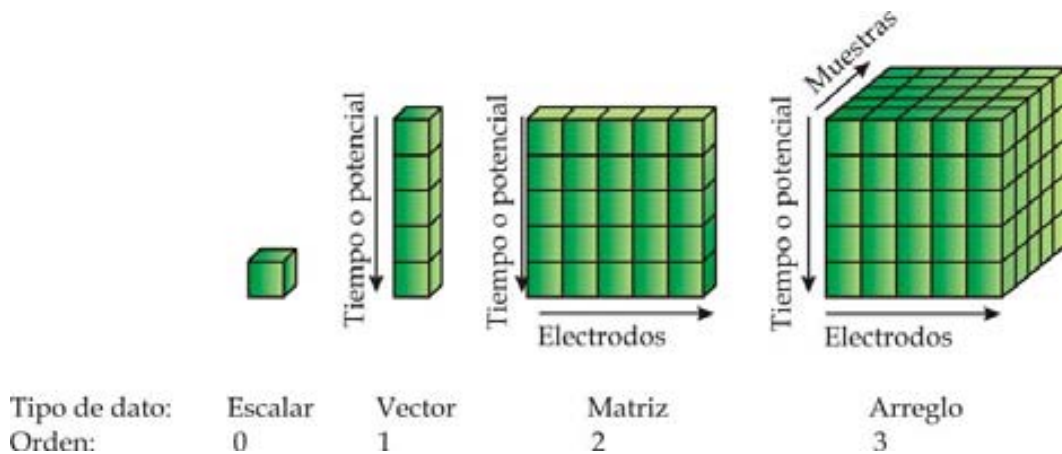


Fig. 1.6. Las primeras tres figuras son la representación esquemática de los arreglos de orden cero hasta dos para datos obtenidos con una sola muestra. El orden de los datos aumenta en una unidad cuando se miden varias muestras en un experimento, lo que da origen a un cubo de datos como el mostrado en la cuarta figura en el extremo derecho.

En el caso de la voltamperometría, existen varias técnicas que pueden usarse para excitar a los electrodos, pero entre ellas la voltamperometría de pulsos es la técnica más recurrida porque aumenta la sensibilidad y resolución de las mediciones [23, 56]. Cuando se usan técnicas voltamperométricas de pulsos en una celda electroquímica de tres electrodos, lo que se hace es aplicar una serie de escalones de voltaje entre los electrodos de trabajo y referencia y medir las corrientes que circulan entre los electrodos de trabajo y auxiliar como resultado de la aplicación de cada uno de los pulsos de voltaje. En el caso de la voltamperometría, los registros adquiridos con un sensor por cada muestra corresponden a tensores de primer orden. Comúnmente, un experimento consiste de la medición de varias muestras. Cuando el número de muestras experimentales es mayor a 1, entonces el orden de los datos medidos aumenta una unidad; así, los tensores de orden cero se convierten en tensores de primer orden, los de orden uno en tensores de segundo orden y los tensores de orden dos en tensores de tercer orden. En general, un conjunto de datos de orden N medidos para varias muestras crea una estructura de orden $N+1$ que recibe el nombre de *tensor de orden- N* o *arreglo multi-modo* [55, 57]. Los tensores de orden mayor a dos serán identificados a lo largo de esta tesis con una letra mayúscula en negrita y subrayada (**X**).

La aplicación de las técnicas electroquímicas anteriormente mencionadas en el análisis de muestras multicomponente genera dos conjuntos de datos apareados, uno formado por los registros eléctricos y otro formado por las concentraciones de analitos en solución de los cuales se obtuvo la información eléctrica. Cuando en un experimento electroquímico se analizan N muestras se genera un número igual de conjuntos de datos apareados $\{\mathbf{x}^{(n)}, \mathbf{y}^{(n)}\}_{n=1}^N$, donde cada n -ésima pareja de datos está formada por un tensor $\mathbf{x}^{(n)}$ de longitud variable que está relacionado con una concentración o conjunto de concentraciones de analitos $\mathbf{y}^{(n)}$. Los conjuntos de datos $\{\mathbf{x}^{(n)}, \mathbf{y}^{(n)}\}_{n=1}^N$ obtenidos del trabajo experimental se usan en la construcción de la lengua electrónica para hallar un modelo de calibración

$$f : \mathbf{x}^{(n)} \rightarrow \mathbf{y}^{(n)} \quad (1.1)$$

La función $f(\cdot)$ de la expresión (1.1) que representa al modelo de calibración puede verse como una función de mapeo que expresa la relación entre las concentraciones de analitos $\mathbf{y}^{(n)}$ y los registros electroquímicos $\mathbf{x}^{(n)}$. Si postulamos la existencia de un modelo similar a la expresión $f : \mathbf{x}^{(n)} \rightarrow \mathbf{y}^{(n)}$ que explique dicha relación entonces nos estaremos enfrentando a un problema que equivale a encontrar una función $f(\cdot)$ a partir de las parejas de datos $\{\mathbf{x}^{(n)}, \mathbf{y}^{(n)}\}_{n=1}^N$ que permita modelar al sistema y realizar tareas de predicción. Esta descripción del problema sugiere conexiones con la teoría de aproximación de funciones [58], redes neuronales [59] y estadística [60].

1.4 Propuestas de solución

A partir de la anterior descripción plantearemos el desarrollo de las herramientas descritas en este trabajo de tesis, las cuales pretenden encontrar un modelo de calibración para la relación $\mathbf{y}^{(n)} = f(\mathbf{x}^{(n)})$. Las implementaciones computacionales que se reportan están basadas en:

- El procesamiento independiente de cada matriz de datos adquirida con los sensores que forman el arreglo tridimensional mediante redes neuronales wavelet y su convergencia a la salida de la estructura (Fig. 1.7).

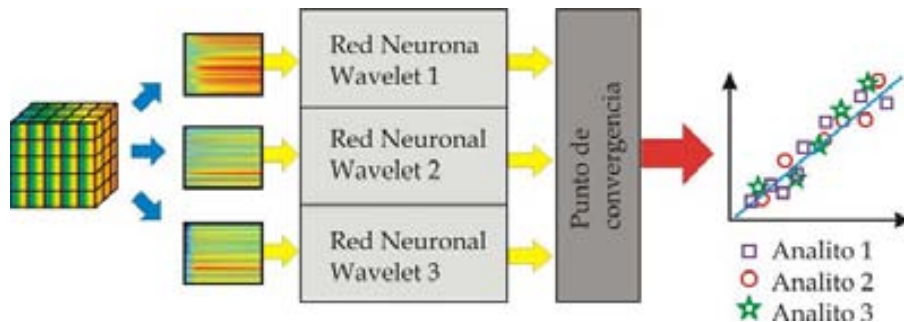


Fig. 1.7. Esquema de la aproximación con redes neuronales wavelet.

- El tratamiento de una sola matriz de datos mediante redes neuronales wavelet para intentar modelar simultáneamente múltiples analitos (Fig. 1.8).

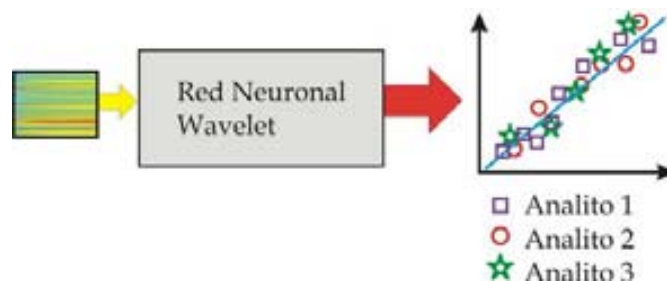


Fig. 1.8. Esquema de aproximación donde se intenta modelar tres analitos con la información de una sola matriz.

- La reducción del número de variables de cada uno de los modos que forman el arreglo de tercer orden mediante Tucker3 o PARAFAC2 y el posterior tratamiento con redes neuronales Spline y B-spline multi-dimensionales de las variables reducidas. La metodología es similar al preprocesamiento con PCA seguida del modelizado con ANNs (Fig. 1.9).

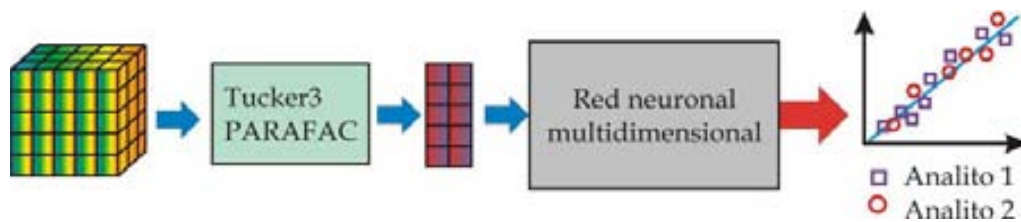


Fig. 1.9. Esquema de la aproximación con preprocesamiento seguida de modelado con red neurona multidimensional.

- El uso de minería de datos para obtener los modelos de calibración a partir de los arreglos de tercer orden, sin reducción de modos ni matricación, aunque con tratamiento independiente y secuencial del conjunto de registros obtenidos con la matriz de sensores por cada muestra analizada (Fig. 1.10).



Fig. 1.10. Esquema de la aproximación basada en el modelado con MARS o B-MARS.

- El preprocesamiento mediante la *DWT* de los registros adquiridos por cada sensor para extraer características relevantes de ellos y reducir el tamaño de las matrices antes de realizar el modelado (Fig. 1.11).

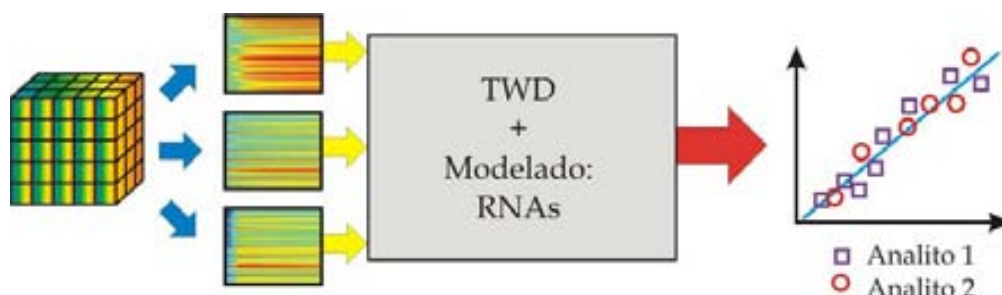


Fig. 1.11. Esquema de la aproximación basada en el preprocesamiento basado en la transformada wavelet y el modelado con redes neuronales.

Las anteriores descripciones no formulan un verdadero tratamiento multi-modo de la información sino un tratamiento bi-modal precedido de la matrización de los arreglos tri-dimensionales obtenidos de los sensores. De manera adicional a las propuestas de solución que se acaban de mencionar también se construyeron modelos de calibración de algunos de los datos experimentales usando la herramienta quimiométrica de *Regresión Multilineal por Mínimos Cuadrados Parciales (N-PLS)* desarrollada por Rasmus Bro de la Universidad de Copenhague (Fig. 1.12).

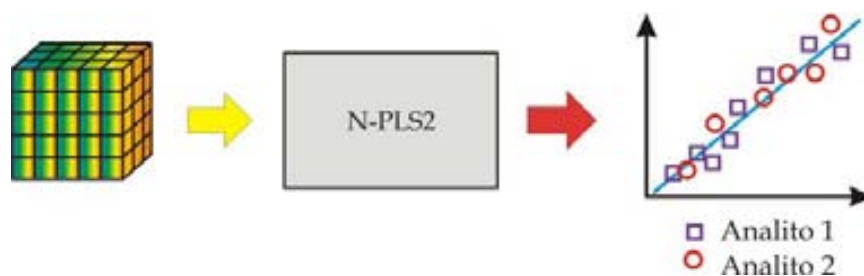


Fig. 1.12. Esquema de la aproximación multi-modo usando N-PLS2.

2

Aspectos teóricos generales

Las Redes Neuronales Artificiales y las técnicas de análisis multimodal son elementos comunes en algunas de las herramientas de calibración que se desarrollaron en esta tesis. La teoría relacionada con las redes neuronales será tratada a continuación haciendo énfasis en la estructura Perceptrón Multicapa. Además de la descripción de las redes neuronales también se mencionarán dos técnicas de entrenamiento: la regularización Bayesiana y el gradiente conjugado; la primera de estas dos técnicas será tratará brevemente por haber sido tomada directamente del Toolbox de Redes Neuronales de Matlab, la segunda será expuesta en detalle por haber sido implementada en las herramientas descritas en los Capítulos 3, 4 y 5. En cuanto a las técnicas de análisis multimodal se describen los modelos PARAFAC y Tucker3. La descomposición de datos de tres modos fue usada para extraer información relevante y reducir la dimensionalidad de la información antes de entrenar las redes neuronales.

2.1 Redes Neuronales Artificiales

Las *Redes Neuronales Artificiales* (ANNs) son sistemas computacionales que emergieron como modelos matemáticos simplificados para entender mejor la neurobiología y la psicología cognitiva [61]. La definición más ampliamente aceptada la describe como *una red masiva de elementos simples (usualmente adaptativos) interconectados paralelamente en una estructura con organización jerárquica que va a interactuar con los objetos del mundo real de la misma manera en que lo hace el sistema nervioso biológico* [62]. A pesar del objetivo

principal para el que fueron concebidas, las áreas biológicas y del conocimiento no fueron las únicas beneficiadas de las ANNs [61, 63], las redes neuronales también han sido usadas en la ingeniería debido a su óptimo funcionamiento en tareas de clasificación, regresión y predicción [62, 63, 64, 65].

2.1.1 Unidad básica de procesamiento

Los elementos más simples que forman las ANNs son unidades básicas de procesamiento con múltiples entradas ponderadas y una sola salida llamadas neuronas artificiales, que se conectan para formar estructuras más grandes que intentan imitar el comportamiento del cerebro [61, 62, 64]. El trabajo más antiguo relacionado con el desarrollo de las ANNs es el de McCulloch y Pitts, que combina la neurofisiología y la lógica matemática con la propiedad de todo o nada de disparo de una neurona para modelar a las neuronas artificiales como elementos binarios discretos (Fig. 2.1) [62, 64, 65, 66]. El modelo desarrollado por McCulloch y Pitts fue el primero en vincular el estudio de las redes neuronales a la idea de la computación en su sentido moderno [67, 68].

En una neurona artificial las entradas se ponderan con valores positivos o negativos antes de entrar en la neurona. Un valor positivo representa una conexión excitatoria mientras que un valor negativo representa una conexión inhibitoria. La suma de estos valores ponderados produce una salida que depende de si la suma ponderada ha alcanzado o no el umbral de disparo predefinido. En la actualidad la neurona de McCulloch-Pitts está en desuso debido a sus limitaciones en reflejar el comportamiento de una neurona biológica, aunque fue la base de los posteriores avances [62, 63, 64, 68].

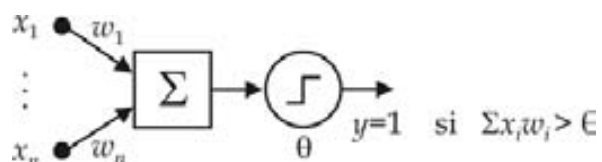


Fig. 2.1. La neurona artificial McCulloch-Pitts funciona como un elemento binario. La neurona dispara si la suma ponderada de las entradas rebasa un valor de umbral θ .

En su estructura básica una ANN está construida por capas de neuronas artificiales, las cuales se identifican de acuerdo a su posición en la red: la capa

de entrada reciben datos del exterior de la red y transmite las salidas al interior de la misma, la capa de salida envía datos fuera de la red, y la capa oculta (o capas ocultas) recibe y envía datos entre capas. Las conexiones entre las neuronas de las diferentes capas tienen un valor de ponderación que representa la fuerza de la conexión sináptica de las neuronas biológicas. La estructura específica de una red neuronal dependerá del número de neuronas de entrada, salida y ocultas, las funciones de activación usadas y la dirección del flujo de información. Las referencias [64-73] brindan un panorama amplísimo sobre topologías y áreas de aplicación.

2.1.2 Taxonomía de las ANNs

Básicamente, se puede considerar que existen dos tipos de redes neuronales: las *redes feedforward* (o *unidireccional*) y las *redes recurrentes* [63, 68]. En una red feedforward las señales se propagan en una dirección, de la etapa de entrada hacia la etapa de salida a través de las neuronas intermedias. En las redes recurrentes las señales pueden propagarse en ambas direcciones, de la entrada hacia la salida y de la salida de cualquier neurona a la entrada de cualquier otra neurona.

Además de la clasificación basada en la dirección del flujo de información, las RNAs también pueden clasificarse de acuerdo al proceso de aprendizaje [63, 68]. El aprendizaje, también conocido como entrenamiento, se define como la optimización de los pesos en las conexiones entre neuronas para modelar correctamente una relación entrada-salida. El entrenamiento puede hacerse mediante un *procedimiento supervisado* o *sin supervisión*, la elección depende del tipo de red que será entrenada y de la información disponible para el entrenamiento. En el entrenamiento supervisado la red ajusta los valores de los pesos en base a la diferencia entre los valores de salida de la red y los valores esperados para un determinado patrón de entrada. En el entrenamiento sin supervisión la red aprende a agrupar los valores de entrada sin recibir información adicional de los grupos esperados. Esta breve descripción de la clasificación servirá para identificar el tipo de redes que se han implementado en este trabajo y para agrupar a las redes existentes dentro del esquema representado en la Fig. 2.2.

En esta tesis nos enfocamos en la estructura feedforward multi-capa, también conocida como Perceptrón Multi-Capa (MLP) [69], con retro-propagación del error y diferentes funciones de transferencia de la capa oculta. Se ha usado la estructura MLP por ser un poderoso sistema capaz de modelar relaciones complejas entre variables de entrada y salida [66, 70]. De manera teórica, una red perceptrón con una sola capa oculta puede aproximar cualquier función con cualquier grado de exactitud, por esta razón, las redes MLP son conocidas como aproximadores universales y pueden usarse cuando se sabe poco de la relación subyacente entre las variables de entrada y salida [71].

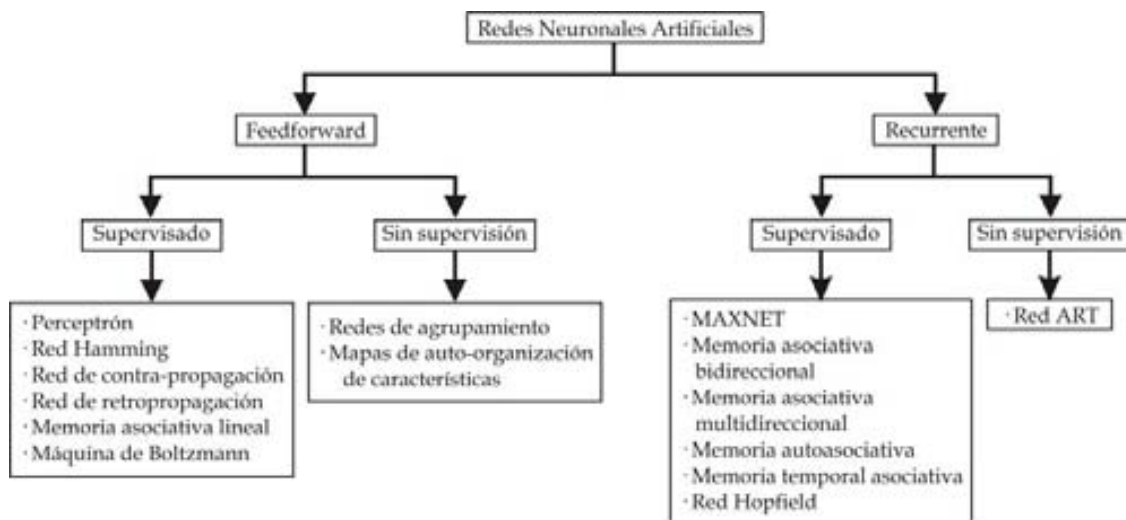


Fig. 2.2. Taxonomía de las Redes Neuronales Artificiales de acuerdo al flujo de información y el tipo de entrenamiento.

2.1.3 Red feedforward

Para ejemplificar el funcionamiento y flujo de información en una red feedforward usaremos la estructura de la Fig. 2.3. En ella, las neuronas en la capa de entrada reciben las señales de entrada x_k , calculan un valor de salida que depende de la función de activación y transmiten el resultado a las entradas de todas las neuronas en la capa intermedia a la que están conectadas. Estos valores son ponderados de manera positiva o negativa por los pesos w_{jk} antes de entrar a las neuronas de la capa intermedia, las cuales reciben dichas entradas, las suman y aplican una función $f_{L_1}(\cdot)$ a esta suma

para conocer el valor de sus salidas. Estas nuevas salidas son ponderadas por los pesos w_j antes de ser transmitidas a las neuronas de salida o a una segunda capa de neuronas ocultas, las cuales realizan la misma función de sumar y aplicar una función para determinar los nuevos valores de salida. El proceso se repite por cada nueva capa de neuronas ocultas que exista en la red hasta alcanzar la capa de neuronas de salida, donde se aplica una función final $f_{L2}(\cdot)$ a la suma ponderada para obtener la salida \hat{y}^n .

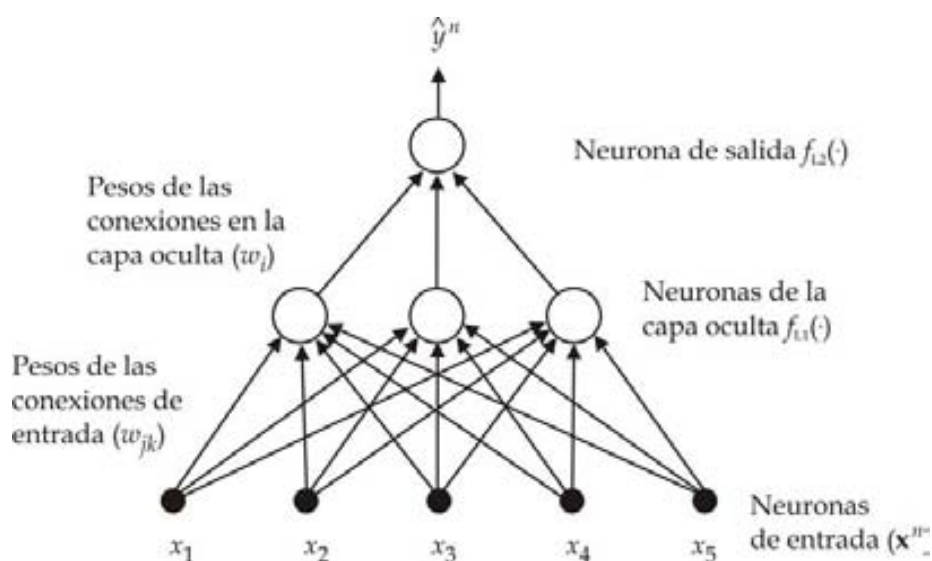


Fig. 2.3. Arquitectura de una red feedforward de una capa oculta mostrando las neuronas de entrada, las de la capa intermedia y las de salida, junto con las conexiones entre capas. En el esquema también se muestra la notación usada en el texto para describir el proceso de entrenamiento.

La dirección del flujo de información en una red feedforward hace que ésta se clasifique como un sistema jerárquico debido a que las capas están organizadas en niveles de menor a mayor jerarquía y la comunicación entre los miembros sólo puede realizarse de un nivel inferior a uno superior pero no en la dirección opuesta [71]. Las redes del tipo MLP aprenden la relación entre las variables de entrada y salida usando un algoritmo de entrenamiento supervisado conocido como retro-propagación del error. La descripción detallada del procedimiento puede consultarse en las referencias [63-65, 69-72, 73].

2.2 Retropropagación del error

La retro-propagación del error es un método de aprendizaje supervisado usado para ajustar los pesos de las conexiones en las ANNs y conseguir que desarrollen una tarea específica [72, 74]. Técnicamente hablando, la retro-propagación del error se usa para calcular el gradiente de cambio en los pesos de las conexiones entre neuronas. En este algoritmo las señales de entrada se presentan de manera iterativa a la red. Por cada vez que se introducen los valores de entrada la red calcula la salida y la compara contra el valor esperado. La diferencia resultante se retroalimenta como un valor de error que se usa para ajustar los pesos de las conexiones y minimizar el error hasta alcanzar un valor mínimo deseado [75].

De manera básica se puede considerar que la retropropagación del error consta básicamente de dos fases: la *propagación* y la *adaptación*. Durante la fase de propagación se presentan simultáneamente los patrones de entrada en la primera capa de la red y se propagan a través de las capas intermedias hasta que alcanzan la capa de salida, produciendo una respuesta por cada patrón de entrada. Las salidas generadas por la red se comparan contra los valores esperados y se obtiene un error a partir de las diferencias resultantes de la comparación, el cual se *retro-propaga* de la capa de salida hacia las demás capas de la red. La segunda fase del algoritmo consiste en el cálculo de los gradientes que se usarán para actualizar los pesos en las conexiones de las neuronas

El algoritmo de retro-propagación del error hace uso de una regla de aprendizaje por gradiente descendiente llamada *regla delta* para ajustar cada uno de los pesos de las conexiones en una cantidad proporcional al negativo de la derivada parcial (dirección de máximo gradiente) de una función de error a minimizar, y en la dirección de máximo gradiente. El objetivo de la regla delta es la minimización de la mencionada función de error, descrita por

$$E_r = \frac{1}{2} \sum_i (y^n - \hat{y}^n)^2 \quad (2.1)$$

donde y^n y \hat{y}^n corresponden al valor real y al valor de salida estimado por la red para el n -ésimo conjunto \mathbf{x}^n de entrada. Usando como referencia la Fig. 2.3

par el desarrollo de la regla delta observamos que la salida \hat{y}^n es función de $\{\mathbf{x}^n, w_{jk}, w_i\}$, es decir, $\hat{y}^n = f\{\mathbf{x}^n, \Theta\}$ donde Θ es una variable ficticia que contiene a w_{jk} y w_i . En cada iteración del entrenamiento se presentan a la red los N conjuntos $\{\mathbf{x}^n\}_{n=1}^N$ de las variables de entrada y se calculan los correspondientes valores de salida \hat{y}^n . Estas aproximaciones se comparan contra los valores y^n esperados y las diferencias $y^n - \hat{y}^n$ se retropropagan como un error descrito por la ecuación (2.1) y que se usa para modificar los pesos de las conexiones entre capas de la red contenidos en la variable ficticia Θ , de acuerdo a la siguiente expresión

$$\Theta^{(p+1)} = \Theta^{(p)} - \eta \cdot \Delta\Theta^{(p)} \quad (2.2)$$

donde el superíndice p es el contador de las iteraciones, η es una tasa de aprendizaje definido por el usuario, $\Theta^{(p)}$ representa a las variables actuales, $\Theta^{(p+1)}$ representa los nuevos valores de las variables que se obtendrán después de cada iteración y $\Delta\Theta^{(p)}$ son los gradientes de error obtenidos de derivar la función de error E_r respecto a cada uno de los pesos w_{jk} y w_i de la red,

$$\Delta\Theta = E_r' = \begin{cases} \Delta w_{jk} = \frac{\partial E_r}{\partial w_{jk}} = -\frac{\partial E_r}{\partial \hat{y}^n} \frac{\partial \hat{y}^n}{\partial w_{jk}} \\ \Delta w_i = \frac{\partial E_r}{\partial w_i} = -\frac{\partial E_r}{\partial \hat{y}^n} \frac{\partial \hat{y}^n}{\partial w_i} \end{cases} \quad (2.3)$$

Una vez actualizados los valores de w_{jk} y w_i se vuelve a calcular la salida de la red y se repite el proceso de modificación de los pesos hasta que se obtiene el valor de error deseado. Expresado en una serie de pasos, el procedimiento de aprendizaje de una red entrenada usando la regla delta es

1. Asignar valores iniciales a los pesos de las conexiones.
2. Presentar los patrones de entrada a la red con los correspondientes valores de salida esperados.

3. Calcular las salidas de la red y compararlas contra los valores esperados para calcular el error.
4. Ajustar los pesos de la red de acuerdo a los cambios calculados usando la regla delta anteriormente descrita.
5. Repetir los pasos 2 a 5 hasta que se minimiza el error a un valor deseado.

El objetivo final del entrenamiento es reducir el error con cada iteración y hacer que los valores calculados de salida estén cada vez más cerca de los valores esperados. Grandes cambios en los pesos de las conexiones aceleran el entrenamiento y pueden hacer que la convergencia sea rápida y que la red se sobreentrene, lo que se traduce como una poca capacidad de generalización de la red; por otro lado, si los cambios en los pesos son pequeños podría suceder que el aprendizaje sea lento y se necesiten muchas iteraciones para converger al valor deseado. Estos problemas en el desempeño del método ponen de manifiesto las dificultades que tienen las técnicas de optimización en la búsqueda de espacios multi-dimensionales para hallar soluciones óptimas o cercanas a ellas [76, 77]. Una manera de ayudar al entrenamiento de la red es añadiendo un término con momento a la actualización de las variables de la ecuación (2.2). Usando el momento, los términos que se ven afectados por el entrenamiento de la red se modifican usando la siguiente regla de actualización

$$\Theta^{(p+1)} = \Theta^{(p)} - \eta \cdot E_r' + \mu (\Theta^{(p)} - \Theta^{(p-1)}) \quad (2.4)$$

donde el nuevo parámetro μ es el coeficiente del momento. Los parámetros η y μ son determinados empíricamente y se debe tener cuidado en elegir su valor, si adoptan valores muy grandes la convergencia al error puede divergir del valor deseado u oscilar, si adoptan valores muy pequeños la convergencia será muy lenta. En un segundo intento de evitar esta segunda dificultad en la actualización de las variables en nuestras implementaciones usaremos una alternativa de entrenamiento basada en el método del gradiente conjugado. Este método puede considerarse como una extensión del gradiente descendiente con la diferencia de que tanto la tasa de aprendizaje como el momento son calculados en cada iteración en lugar de ser elegidos de manera empírica. A diferencia del gradiente descendiente en donde se sigue una línea

de minimización en la dirección indicada por el negativo del gradiente $(-E_r')$, en el método del gradiente conjugado se genera una serie sucesiva de direcciones conjugadas llamadas $D^{(p)}$ que tienen la propiedad de que cada nueva dirección no se interfiere con la anterior y en donde se buscan mínimos locales que permitan alcanzar el mínimo global de la función de minimización del error.

La minimización de E_r mediante el método del gradiente conjugado comienza con una estimación inicial de los parámetros $\Theta^{(0)}$ y una línea inicial de búsqueda $D^{(0)} = -E_r'(\Theta^{(0)})$ que permitan generar una serie de aproximaciones de los parámetros de la red para conseguir el objetivo final. A partir de estos valores se busca una longitud de paso $\alpha^{(p)}$ de la línea de búsqueda $D^{(p)}$ mediante la minimización de la función $\psi(\alpha) = E(\Theta^{(p)} + \alpha D^{(p)})$. El valor para $\alpha^{(p)}$ se obtiene haciendo

$$\alpha^{(p)} = -\frac{E_r'(\Theta^{(p)})^T \cdot D^{(p)}}{(D^{(p)})^T \cdot \text{diag}(E_r''(\Theta^{(p)})) \cdot D^{(p)}} \quad (2.5)$$

Con este valor de $\alpha^{(p)}$ se calculan los primeros cambios en $\Theta^{(p)}$ mediante

$$\Theta^{(p+1)} = \Theta^{(p)} + \alpha^{(p)} D^{(p)} \quad (2.6)$$

Para determinar una nueva línea de búsqueda $D^{(p+1)}$ hacemos

$$D^{(p+1)} = -E_r'(\Theta^{(p+1)}) + \beta^{(p+1)} D^{(p)} \quad (2.7)$$

donde el parámetro $\beta^{(p+1)}$ se calcula de acuerdo al método de Polak y Ribiere por ser numéricamente más estable [76]

$$\beta^{(p+1)} = \frac{(E_r'(\Theta^{(p+1)}) - E_r'(\Theta^{(p)}))^T \cdot E_r'(\Theta^{(p+1)})}{E_r'(\Theta^{(p)})^T \cdot E_r'(\Theta^{(p)})} \quad (2.8)$$

Una vez definidas las ecuaciones que describen los gradientes de cambio se actualizan los valores de las variables en $\Theta^{(p)} | p \geq 1$ usando las expresiones (2.5)-(2.8)

$$\Theta^{(p+1)} = \Theta^{(p)} + \alpha^{(p)} D^{(p)} = \Theta^{(p)} - \alpha^{(p)} E_r'(\Theta^{(p+1)}) + \alpha^{(p)} \beta^{(p+1)} D^{(p)} \quad (2.9)$$

La tasa de aprendizaje y el momento no desaparecen en el algoritmo de gradiente conjugado sino que se redefinen como $\eta = \alpha$ y $\mu = \alpha \cdot \beta$. El entrenamiento de la red es un procedimiento iterativo que se realiza hasta que la diferencia entre los valores dependientes reales de la función $y = f(x)$ y los valores obtenidos por la red a través de la aproximación $\hat{y} = f(\mathbf{x}, \Theta)$ se reducen hasta alcanzar un valor mínimo predefinido por el usuario.

Otras metodologías usadas para resolver el problema durante el entrenamiento de la red son el aprendizaje con paro anticipado y la regularización Bayesiana [78]. En el primero se usa un subconjunto de datos de prueba para evaluar el comportamiento de la red después de cada iteración, si la salida esperada para los datos de prueba mejora entonces se continúa el entrenamiento, si por el contrario la salida esperada empeora entonces se da por concluido el entrenamiento de la red. En la segunda metodología se minimiza una combinación lineal de sumas de errores y pesos al cuadrado que al final del entrenamiento permite tener una mejor capacidad de generalización de la red entrenada. Usando la regularización Bayesiana la función de costo ahora queda expresada por la ecuación (2.10) para incluir el término E_w que representa la suma de pesos al cuadrado

$$E = \alpha E_r + \beta E_w \quad (2.10)$$

Los términos α y β son parámetros de la función objetivo que no pueden elegirse al azar y por tanto deberán optimizarse. Si $\alpha \gg \beta$ entonces el entrenamiento enfatiza la reducción de los pesos y tolera errores más grandes, por el contrario, si $\beta \gg \alpha$ entonces se presenta sobreentrenamiento [79]. La optimización Bayesiana de los parámetros α y β requiere de procedimientos computacionales costosos, los cuales se evitan usando el algoritmo de optimización de Levengerg-Marquardt [79, 80]

2.3 Análisis multimodal de datos

El análisis de datos multimodo es el análisis descriptivo de datos que forman una estructura con 3 modos o más. Estas metodologías de análisis permiten obtener modelos que describen las relaciones estructurales entre las entidades de forma los datos [81]. Los datos de carácter multimodal se adquieren con esta estructura bajo la suposición a priori de que cada uno de los modos que lo forman es necesario para encontrar la relación entre sus variables. Estas relaciones ocultas pueden ser de carácter trilineal, cuatrilineal o multilineal. El concepto de linealidad que se maneja en el análisis multimodal hace referencia a la linealidad del modelo en uno de sus conjuntos de parámetros dados los conjuntos de parámetros restantes. Para ejemplificar este concepto consideremos el modelo $x_{ijk} = a_{i1}b_{j1}c_{k1} + a_{i2}b_{j2}c_{k2}$ formado por dos términos, los elementos x_{ijk} del modelo anterior son trilineales en los términos a , b y c porque dados dos cualesquiera de ellos los valores de x_{ijk} pueden calcularse usando el tercero [81, 83].

Otro concepto importante en la descripción de las técnicas de análisis multimodo es el concepto de *rango*. El rango es una propiedad importante de las matrices que sirve para indicar el número de filas o columnas linealmente independientes. En el área química el rango de una matriz no es un concepto con mucha utilidad debido a los errores en las mediciones y a fuentes de ruido, es por ello que en su lugar se ocupa el concepto de *pseudorango* o *rango esencial* [82, 83]. Expresado de una manera concisa, el pseudorango de una matriz que originalmente tenía un conjunto de filas o columnas linealmente independientes pero que ha sido contaminada con ruido es igual al rango de la misma matriz una vez que se ha eliminado el ruido. En la práctica el pseudorango de una matriz de datos experimentales es considerablemente menor que el rango de la misma. Los términos de linealidad y rango están estrechamente relacionados. El rango de un arreglo de tres modos es el mínimo número de componentes que se necesitan en un modelo trilineal para reproducirlo [83]. Para obtener el pseudorango de un arreglo de tres modos primero se desdobra el arreglo en las tres direcciones posibles para formar tres matrices aumentadas (Fig. 2.4). El proceso de matrización da origen a tres matrices: una matriz aumentada por fila, otra aumentada por columna y una tercera aumentada por tubo. En cada una de las matrices aumentadas se hace

un análisis de rango, cuando las tres matrices tienen el mismo rango entonces se dice que el arreglo de tres modos es *trilineal*, cuando los rangos son diferentes entonces al arreglo *no es trilineal*.

Las técnicas de análisis multimodo se clasifican de acuerdo a su capacidad de manejo de la linealidad de los datos. La *descomposición trilineal directa (DTD)* y el *análisis de factores paralelos (PARAFAC)* son ejemplos de técnicas de análisis que asumen trilinealidad en la estructura de los datos; *TUCKER3*, *PARAFAC2* y la *resolución multivariable de curvas-mínimos cuadrados alternantes (MCR-ALS)* son, por el contrario, ejemplos de técnicas que no asumen la trilinealidad en el análisis de la información [82]. Los métodos que asumen trilinealidad tienen características matemáticas deseables tales como la unicidad de la solución, sin embargo, la solución obtenida con estos métodos no puede considerarse correcta si los datos analizados no son trilineales. Los métodos para el análisis de datos que no son trilineales son más flexibles en el manejo de la información aunque deban aplicarse restricciones que permitan asegurar que los resultados son correctos debido a que sin ellos se presentarían cierto grado de ambigüedad.

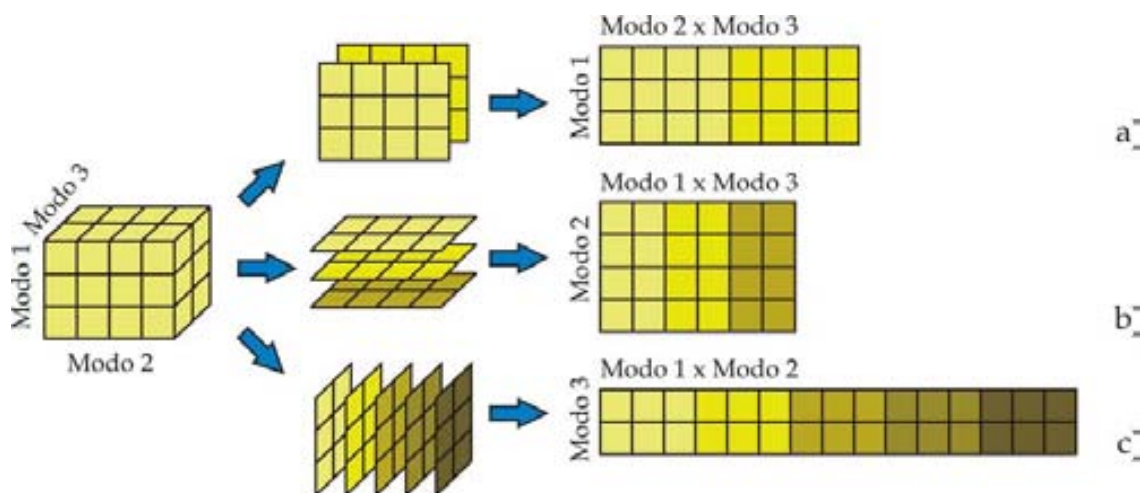


Fig. 2.4. Matricización de un arreglo de tres modos en a) una matriz aumentada por fila, b) una matriz aumentada por columna y c) una matriz aumentada por tubo.

2.3.1 PARAFAC y PARAFAC2

El análisis de *factores paralelos proporcionales (PARAFAC)* es un método de descomposición que tiene su origen en la psicometría y que puede

considerarse como una extensión directa de *análisis de componentes principales (PCA)* que se aplica a los datos de dos modos. El modelo fue propuesto de manera independiente por Harshman y por Carroll y Chang, quienes lo llamaron CANDECOMP (descomposición canónica) [84, 85]. El objetivo de la mayoría de los análisis de datos hechos con PARAFAC es revelar la existencia de componentes que tienen *perfiles proporcionales paralelos*, y si es posible, identificar estos componentes como los que tienen un significado sustantivo verdadero [86]. El principio de perfiles proporcionales paralelos establece que se puede hallar un conjunto de factores comunes que puedan ajustar al mismo tiempo (con diferentes pesos) a varias matrices de datos. Esto es equivalente a hallar un conjunto de factores para un grupo de matrices, es decir, para un arreglo de tres modos.

El modelo de PARAFAC que se obtiene de un conjunto de datos \mathbf{X} de tres modos con elementos x_{ijk} formado por I sujetos medidos en J variables y K condiciones tiene la forma

$$x_{ijk} = \sum_{f=1}^F a_{if} b_{jf} c_{kf} + e_{ijk} \quad (2.11)$$

donde los F elementos de sujetos a_{if} , variables b_{jf} y ocasiones c_{kf} son elementos que forman a las matrices \mathbf{A} , \mathbf{B} y \mathbf{C} , respectivamente (Fig. 2.5) [85, 87]. El modelo PARAFAC es trilineal, es decir, si fijamos dos de los conjuntos de parámetros (e.g. las a 's y b 's) entonces x_{ijk} puede ser expresado como una función lineal de las c 's. La idea básica de PARAFAC es buscar el mínimo número de parámetros para describir la máxima cantidad de correlación entre las variables. Este modo de operación es precisamente la idea principal del principio de perfiles proporcionales paralelos, es por esta razón que el modelo PARAFAC adopta este nombre.

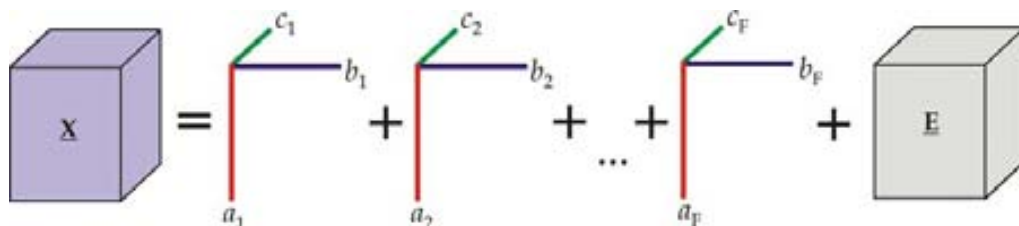


Fig. 2.5. Descomposición de un arreglo de datos de tres modos mediante el modelo PARAFAC.

Una característica relevante de PARAFAC es la unicidad de la solución, es decir, las matrices A , B y C que forman el modelo no pueden ser modificadas sin cambiar los residuales. El significado matemático de la unicidad es que el modelo PARAFAC no puede ser girado sin perder ajuste, es decir, carece de libertad de rotación.

2.3.2 Tucker3

De manera semejante al modelo PARAFAC, el modelo *Tucker3* también es usado en el análisis de datos de tres modos para hallar un conjunto limitado de componentes con los que se pueda describir la mayor parte de la variabilidad de la información analizada, por lo que resulta especialmente útil para reducir datos. El modelo Tucker3 permite descomponer un arreglo de tres modos \underline{X} en cuatro conjuntos de parámetros que corresponden a una matriz \mathbf{A} que resume en P componentes las I entidades del modo A (sujetos), una matriz \mathbf{B} que resume mediante Q componentes las J entidades del modo B (variables), una matriz \mathbf{C} que resume mediante R componentes las K entidades del modo C (ocasiones) y un arreglo central \underline{G} de tres modos y tamaño $P \times Q \times R$ formado por elementos que representan las contribuciones de cada una de las entidades obtenidas del análisis [81-83, 88]. La función del arreglo central es dar una breve descripción de la información original contenida en \underline{X} en términos de los componentes P , Q y R de los tres diferentes modos. Expresado de otra manera, la magnitud de cada pqr -ésimo elemento de \underline{G} es proporcional a la contribución del p -ésimo elemento de \mathbf{A} , el q -ésimo elemento de \mathbf{B} y el r -ésimo elemento de \mathbf{C} en la reproducción del arreglo original, por lo que \underline{G} puede considerarse como una versión reducida del arreglo \underline{X} con elementos que capturan las interacciones entre los tres modos pero en términos de los componentes obtenidos con la descomposición. El modelo obtenido con Tucker3 se describe en términos de la triple sumatoria expresada en la ecuación (1.2) entre los elementos contenidos en cada una de las matrices componentes y del arreglo central más un término de error correspondiente a cada estimación de los valores del arreglo original [87].

$$x_{ijk} = \sum_{p=1}^P \sum_{q=1}^Q \sum_{r=1}^R a_{ip} b_{jq} c_{kr} g_{pqr} + e_{ijk} \quad (2.12)$$

La representación esquemática de la descomposición de un arreglo de tres modos mediante Tucker3 es la mostrada en la Fig. 2.6. La principal razón de que Tucker3 no se use tan frecuentemente como otras técnicas de análisis multimodos en la búsqueda de patrones ocultos en datos de tres modos es su libertad de rotación [88]. El resultado básico que se obtiene al aplicar Tucker3 a un conjunto de datos de tres modos son componentes espaciales en las que cualquier orientación de los ejes es buena en términos del ajuste de la solución. Las tres matrices A , B y C pueden ser rotadas de manera independiente siempre que esas rotaciones sean compensadas en el arreglo central \underline{G} , esto significa que Tucker3 no entrega soluciones únicas sino que es posible estimar una infinidad de soluciones para A , B , C y \underline{G} . Esta característica no impacta la interpretación del modelo ya que el comportamiento sistemático atrapado por un modelo es el mismo en todos los modelos.

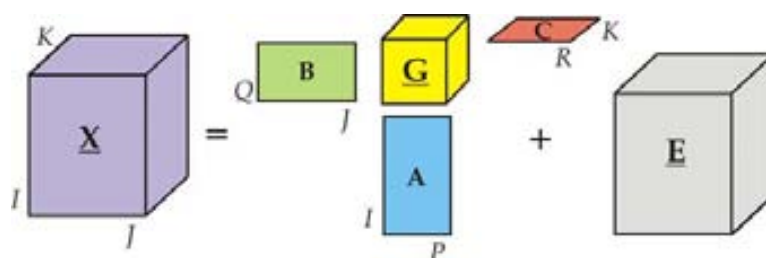


Fig. 2.6. Representación esquemática del modelo Tucker3.

2.3.3 Mínimos Cuadrados Parciales Multi-modo (N-PLS)

La regresión por Mínimos Cuadrados Parciales Multi-modo (*N-PLS*) tiene su origen en la regresión por Mínimos Cuadrados Parciales o (*PLS*). *PLS* es una herramienta comúnmente usada en la quimiometría para construir modelos de calibración usando conjuntos de datos formados por predictores y respuestas [89, 90, 91]. El objetivo de la metodología es predecir una sola (y) o varias respuestas (Y) a partir de una matriz de predictores X , así como describir la estructura común subyacente entre las dos variables [92].

En general, *PLS* descompone simultáneamente a X y y en un conjunto de variables latentes que maximizan la covarianza entre los datos independientes y dependientes. Para una regresión univariable, X y y se descomponen en un matriz T de *scores* comunes y vectores de *loadings*

individuales \mathbf{P} y \mathbf{q} . La matriz \mathbf{T} es el elemento de relación interna entre las ecuaciones lineales $\mathbf{X}=\mathbf{TP}^t+\mathbf{E}_x$ y $\mathbf{y}=\mathbf{Tq}^t+\mathbf{e}_y$, que modelan a los predictores y respuestas (\mathbf{E}_x y \mathbf{e}_y representan a la matrix y vector de residuos para \mathbf{X} y \mathbf{y} , respectivamente). Para predecir valores de la variable dependiente se usa el modelo de regresión lineal $\hat{\mathbf{y}}=\mathbf{Xb}+\mathbf{e}_y$, cuyos coeficientes de regresión \mathbf{b} son obtenidos a partir de los parámetros del modelo [93, 35].

La regresión *N-PLS* es una generalización del método de regresión *PLS* de dos modos para conjuntos de datos de orden superior [94,87]. La metodología fue presentada por Rasmus Bro en 1996 y mejorada en el 2001 [95] por el mismo autor. Existen varias publicaciones que describen los fundamentos matemáticos de esta metodología [83, 94, 96, 97, 98], por lo que aquí sólo daremos una breve descripción.

Al ser una extensión de la regresión *PLS* de dos modos, la regresión *N-PLS* también permite construir un modelo de calibración incorporando una relación entre los conjuntos de descriptores y respuestas basada en la estructura multimodo de los arreglos, lo que permite predecir respuestas con una estructura superior a dos modos. La principal diferencia entre *N-PLS* y la regresión *PLS* de dos modos es el uso de matrices de *loadings* de mayor dimensión que se usan para descomponer los arreglos multi-modo. Tanto en *PLS* como en *N-PLS* se agrega un número al final del acrónimo para identificar el número de modos de las respuestas, en el caso de *PLS* este número queda restringido a 2.

El objetivo de la metodología en la que se basa *N-PLS* es ajustar simultáneamente modelos multilineales formador por vectores de *score* y *loadings* obtenidos de los arreglos de predictores y respuestas, junto con un modelo de regresión que relaciona los dos modelos de descomposición. De acuerdo a la teoría subyacente de la regresión *PLS*, el modelo *N-PLS* es ajustado de tal manera que los vectores de *score* tienen la máxima covarianza con la parte sin explicar de la variable dependiente [94].

Para el caso de dos estructuras \mathbf{X} y \mathbf{Y} de tres modos con índices ($I \times J \times K$) y ($I \times L \times M$), respectivamente, la metodología modela el arreglo de predictores y respuestas en una descomposición multilineal del tipo PARAFAC. Los modelos

para los arreglos matricizados quedan descritos mediante $\mathbf{X}=\mathbf{T}\mathbf{G}_x(\mathbf{P}^J \otimes \mathbf{P}^K)^t + \mathbf{E}_x$ y $\mathbf{Y}=\mathbf{U}\mathbf{G}_y(\mathbf{Q}^L \otimes \mathbf{Q}^M)^t + \mathbf{E}_y$, respectivamente, donde \mathbf{T} y \mathbf{U} son vectores de *score*, \mathbf{P} y \mathbf{Q} son vectores de *loadings*, y \mathbf{E}_x y \mathbf{E}_y son matrices de residuales. Los superíndices J , K , L y M indican con cuál de los modos está relacionado el correspondiente vector de *loadings*. \mathbf{G}_x es un núcleo matricizado definido mediante $\mathbf{G}_x=\mathbf{T}^+\mathbf{X}\left(\left(\mathbf{P}^J\right)^+ \otimes \left(\mathbf{W}^K\right)^+\right)^t$, que en la versión mejorada de *N-PLS* resuelve varios problemas relacionados con el modelado de $\underline{\mathbf{X}}$ [95]. \mathbf{G}_y queda definida de manera similar para $\underline{\mathbf{Y}}$.

Usando los modelos trilineales previamente obtenidos, la relación entre $\underline{\mathbf{X}}$ y $\underline{\mathbf{Y}}$ se obtiene mediante un modelo de regresión del tipo $\mathbf{U}=\mathbf{T}\mathbf{B}+\mathbf{E}_u$. Los coeficientes de regresión \mathbf{B} para el modelo *N-PLS* pueden obtenerse siguiendo los procedimientos descritos en [97, 98].

Sección 2

3

Redes Neuronales Wavelet

En los últimos años las ANNs han sobresalido como herramientas universales de aproximación para el ajuste de modelos tanto lineales como no-lineales a partir de conjuntos apareados de datos de entrada y salida [99, 100]. Lo anterior se confirma con el contenido de las referencias [101, 102, 103, 104, 105, 106, 107, 108], las cuales son algunas publicaciones de los últimos 10 años donde se hace una revisión de los sistemas basados en sensores electroquímicos acoplados con ANNs (entre otras herramientas de calibración) para realizar tareas de clasificación y cuantificación.

En una red neuronal feed-forward, la salida de una neurona en cualquiera de sus capas se calcula aplicando una función de activación a la suma ponderada de valores de entrada. Una generalización de esta estructura que permita conservar de alguna manera la característica multi-dimensional de los datos de entrada consiste en implementar neuronas con funciones de activación multi-dimensional.

A pesar de que las redes neuronales feedforward con función de activación sigmoideal son capaces de aproximar cualquier función continua, se ha demostrado que no todas las funciones con cierto grado de complejidad pueden ser representadas de manera simple por medio de funciones con menor grado de complejidad [109]. El uso de funciones de transferencia más complejas en las ANNs, como las funciones wavelet ha permitido construir herramientas alternativas para la obtención de modelos de calibración con el grado de complejidad que se necesita en una lengua electrónica. La idea de

fusionar las teoría wavelet con las redes neuronales ha dado origen a la *Red Neuronal Wavelet (WNN)*, propuesta originalmente por Zhang y Benveniste en 1992 como una alternativa a la aproximación de funciones arbitrarias no-lineales [110]. La capacidad de aproximación de funciones de esta estructura de red neuronal relativamente nueva ha sido superior que la de las redes MLP y las redes con funciones base del tipo radial [101, 111, 112]. La efectividad de las WNNs ha sido demostrada en las tareas de modelización, predicción y procesamiento de señales en las áreas de micro- y nano-sistemas, ingeniería mecánica, electrónica de potencia, medicina, tratamiento de imágenes y química; en esta última área se han publicado trabajos donde la WNN ha sido usada para el modelado del potencial de media onda de aldehídos y cetonas [113], la predicción de componentes usando espectros cercanos al infrarrojo [114], la determinación simultánea de múltiples componentes [115, 116, 117, 118], el cálculo de constantes de enlace en derivados del benceno [119], la predicción de la polaridad del disolvente [120] y la solubilidad de hidrocarburos aromáticos policíclicos [121].

3.1 Transformación Wavelet

La *Transformación Wavelet (WT)* es una técnica de procesamiento de señales usada para convertir una señal (o función) de su dominio original a otro dominio donde ciertas características de la señal original son más fáciles de estudiar [122, 123].

La técnica ha mostrado un elevado desempeño en el análisis de señales no estacionarias, ruidosas, intermitentes o transitorias debido a sus buenas propiedades de localización tanto en el dominio del tiempo como en el dominio de la frecuencia [124].

La WT precisa del uso de pequeñas funciones ondulatorias localizadas conocidas como *wavelets*. Matemáticamente hablando, la transformación es la convolución de la señal a analizar con un conjunto de funciones obtenidas mediante la dilatación (o escalamiento) y traslación de una función base conocida como la *wavelet madre* [122, 125]. El resultado de la transformación es un grupo de coeficientes que cuantifican la correspondencia local de la wavelet dilatada y trasladada con la señal a analizar. Si la wavelet tiene una

correspondencia elevada en una escala y posición específica entonces el coeficiente obtenido de la transformación es elevado. Si por el contrario, la wavelet y la señal no tienen una buena correlación entonces se obtiene un bajo coeficiente de correlación. Los coeficientes de la WT se calculan para varias posiciones en la señal y diferentes valores de escalamiento de la función wavelet. La transformación puede realizarse en tiempo continuo con la *Transformada Wavelet Continua (CWT)*, o en tiempo discreto con la *Transformada Wavelet Discreta (DWT)*.

Los resultados obtenidos de la transformación se grafican en un plano de posición contra escala, la gráfica obtenida se conoce como escalograma y proporciona información acerca de la correlación entre la señal y una wavelet a diferentes posiciones y valores de escalamiento (Fig. 3.1). La teoría del procesamiento Wavelet descrita a continuación se puede encontrar con mayor detalle en la bibliografía [122-126, 127, 128, 129, 130, 131, 132, 133, 134].

3.1.1 Función wavelet

Para propósitos de explicación de la función wavelet y la transformación correspondiente, usaremos el tiempo (t) como la variable independiente, aunque puede ser cualquier otra variable en función del experimento realizado. La WT de una señal precisa del uso de una función wavelet (algunos ejemplos están representados en la Fig. 3.2 para propósitos de visualización). Una wavelet $\psi(t)$ es una función cuadrática integrable que satisface ciertas condiciones matemáticas descritas en las referencias [122-134]. Las condiciones más importantes pero no suficientes que definen a $\psi(t)$ como una wavelet son las condiciones de *admisibilidad* y *regularidad* [122, 126].

La condición de admisibilidad establece que la Transformada de Fourier de la función wavelet debe valer cero para una frecuencia cero de la señal, lo que implica que la función debe ser oscilatoria y tener un promedio igual a cero. La condición de admisibilidad también asegura que una señal pueda ser analizada y sintetizada sin perder información [122, 126, 127, 134].

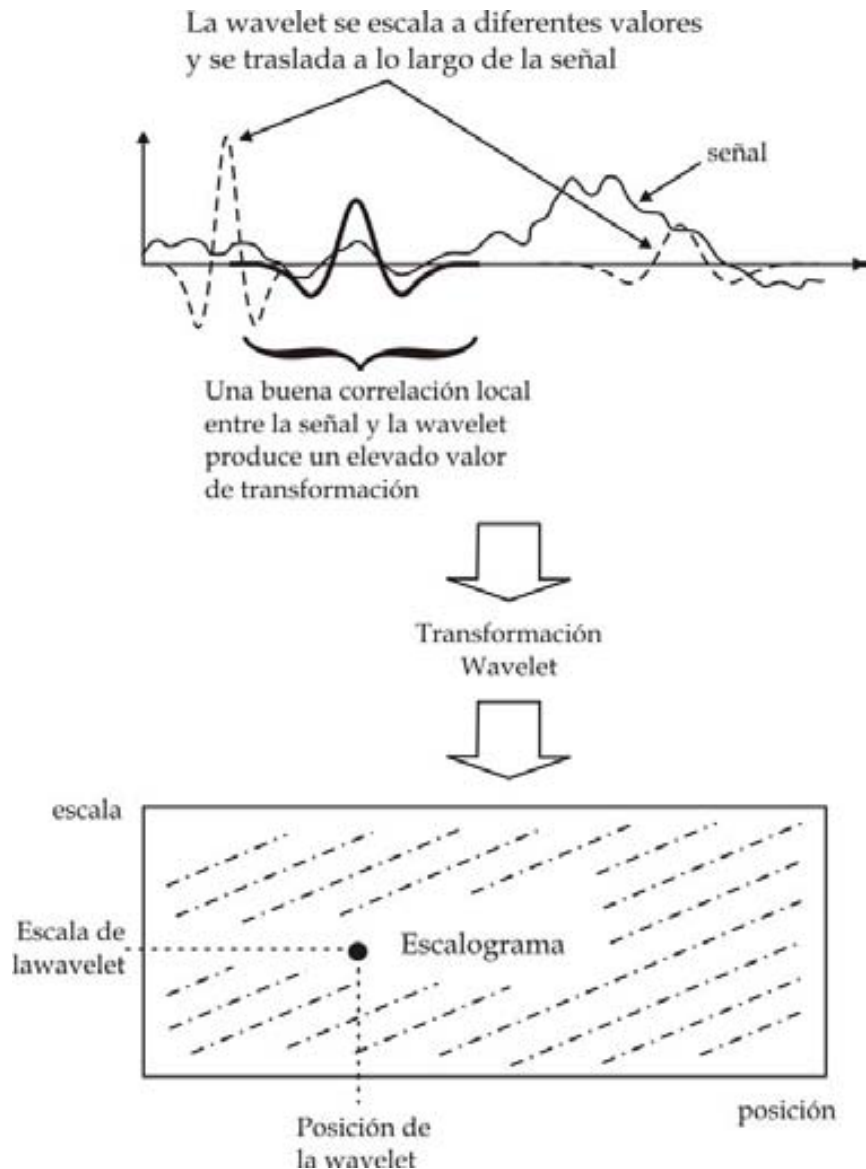


Fig. 3.1. La Transformación Wavelet es la convolución de la señal con una wavelet escalada y trasladada a diferentes posiciones. El resultado de la transformación es un conjunto de coeficientes que graficados en un plano posición-escala proporcionan información del contenido espectral de la señal y su localización.

La condición de regularidad establece que la wavelet debe ser una función suavizada y localizada tanto en tiempo como en frecuencia. El concepto de regularidad es más complejo que el de admisibilidad y está estrechamente relacionado con la cantidad de veces que una función wavelet puede diferenciarse y el número de momentos de desvanecimiento que posea [133]. La regularidad es una medida aproximada del número de derivadas continuas que posee una función wavelet: entre mayor sea la regularidad, la función

wavelet será más suave. Matemáticamente hablando, una wavelet que posee $n+1$ momentos de desvanecimiento es n veces diferenciable [135]. Los momentos de desvanecimiento se definen como $M_p = \int t^p \psi(t) dt = 0$ para $p = 0, 1, \dots, n$.

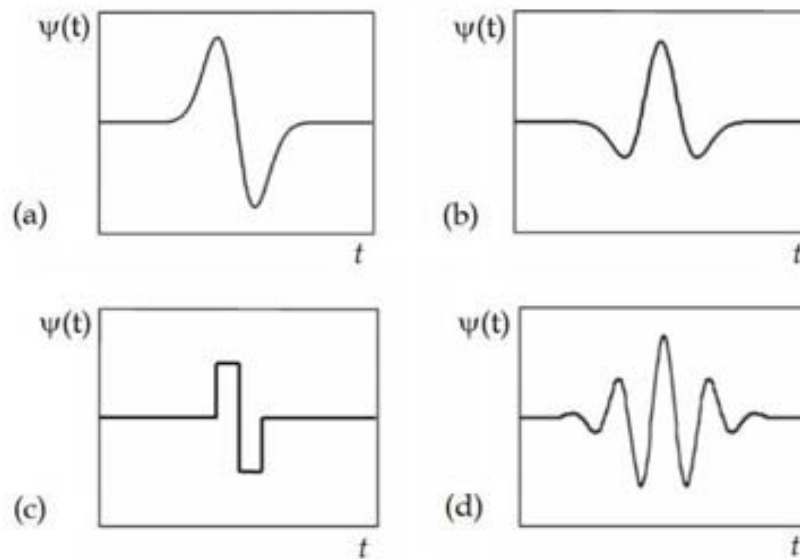


Fig. 3.2. Cuatro ejemplos de funciones wavelet. El número entre corchetes indica los momentos de desvanecimiento (a) Onda Gausiana (primera derivada de una Gausiana) [1]. (b) Función Marr (Sombrero Mexicano, segunda derivada de una Gausiana) [2]. (c) Haar [1]. (d) Morlet (parte real de la función) [1].

Una wavelet con $n+1$ momentos de desvanecimiento es ortogonal a los polinomios de grado n , por lo que si hacemos la transformación de una señal caracterizada por un polinomio definido por segmentos de grado n usando una wavelet con $n+1$ momentos de desvanecimiento, entonces los coeficientes wavelet serán igual a cero para los segmentos y diferente de cero en las discontinuidades. Los momentos de desvanecimiento de la wavelet es lo que permite caracterizar la regularidad de una función y ubicar las posibles discontinuidades.

Para resumir, juntando las dos condiciones anteriores tenemos que: a) de acuerdo a la condición de admisibilidad la wavelet debe valer cero a frecuencia cero, y b) de acuerdo a la condición de regularidad la wavelet debe estar

localizada tanto en tiempo como en frecuencia, por tanto, la transformada wavelet es un filtro pasa-banda en el plano de Fourier

3.1.2 Transformada Wavelet Continua

Para usar una wavelet madre en el análisis de una señal necesitamos que la función sea más flexible que sólo $\psi(t)$. Para conseguirlo realizamos dos manipulaciones sobre $\psi(t)$: incluimos un *parámetro de escalamiento* (s) para estirla o estrecharla y un *parámetro de traslación* (m) para desplazarla. Las versiones estrechadas (o dilatadas) y desplazadas de la *wavelet madre* $\psi(t)$, conocidas como *wavelets hijas*, se describen mediante la siguiente expresión

$$\psi_{s,m}(t) = \frac{1}{\sqrt{|s|}} \psi\left(\frac{t-m}{s}\right) \quad s, m \in \mathbb{R} \quad m \neq 0 \quad (3.1)$$

Usando el conjunto de funciones descritas por la expresión (3.1) definimos la Transformación Wavelet de una señal Continua (CWT) como la convolución de una señal $f(t)$ con una familia de funciones wavelet $\psi_{s,m}(t)$,

$$CWT_{s,m}\{f(t)\} = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{+\infty} f(t) \psi^*\left(\frac{t-m}{s}\right) dt \quad (3.2)$$

Los parámetros de escala y traslación anteriormente definidos para las wavelets hijas hacen que la integral sea una ventana flexible para el análisis de señales tanto en tiempo como en frecuencia. La ventana se estrecha para detectar los componentes de alta frecuencia y se estira para estudiar los componentes de baja frecuencia en cada segmento de la señal analizada (Fig. 3.3) [129].

De acuerdo a la condición de admisibilidad, una señal debe ser recuperable a partir de su transformación wavelet. La Transformada Wavelet Inversa permite recuperar la señal original a partir de los coeficientes wavelet obtenidos de la WT mediante la integración sobre todos los valores de escala (s) y traslación (m)

$$f(t) = \frac{1}{C_g} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} CWT_{s,m} \{f(t)\} \psi_{s,m}(t) \frac{ds dm}{s^2} \quad (3.3)$$

Si en la integración se acota el intervalo de valores que puede tomar el parámetro de escalamiento s , entonces se puede hacer un filtrado básico de la señal original ya que se estarán omitiendo contenido espectral en la reconstrucción de la señal.

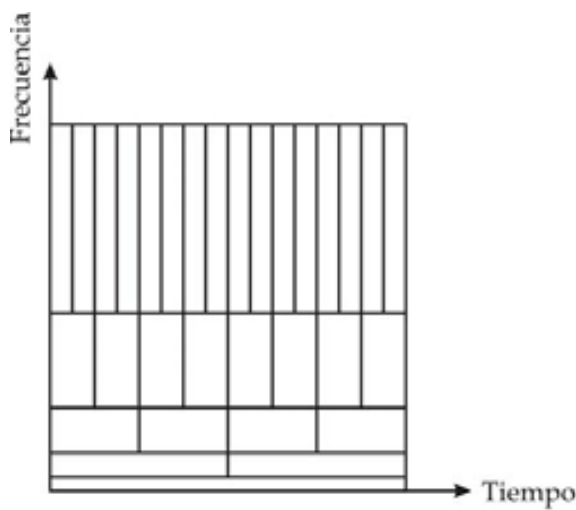


Fig. 3.3. División del plano tiempo-frecuencia por la transformada wavelet. Los largos intervalos de tiempo de la parte inferior corresponden a los eventos de baja frecuencia, los intervalos cortos de tiempo de la parte superior a los eventos de alta frecuencia.

3.2 Red Neuronal Wavelet

Las WNNs que se implementaron en esta tesis tienen una estructura similar a las redes MLP. Como se mencionó en el Capítulo 2, una red MLP básica está formada por tres capas de neuronas: una de entrada, una oculta y una de salida. Al igual que en las redes MLP las neuronas en la capa de entrada de la WNN no tienen ninguna función de activación, su tarea consiste en distribuir los datos de entrada hacia el interior de la red. La capa oculta de la WNN está formada por neuronas con funciones de activación wavelet y finalmente, las neuronas de la capa de salida contienen funciones de activación lineal. La característica que permite aproximar la transformación wavelet a una red neuronal MLP de una capa oculta es la similitud que existe entre la fórmula para sintetizar una señal a partir de los coeficientes wavelet obtenidos de su

descomposición, $f(t) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} D_{j,k} \psi_{j,k}(t)$ y la ecuación que permite calcular la

salida de una red MLP definida por $\hat{y} = \sum_j w_j f\left(\sum_i w_i x_i\right)$ [137, 138]. Resulta

relevante notar que la fórmula para la reconstrucción de $f(t)$ ya había sido descrita en 1927 por Strömberg en sus estudios de descomposición atómica de señales (los *átomos* son los elementos más simples de un espacio de funciones) antes de la descripción formal de la teoría Wavelet. La similitud entre ambas ecuaciones se vuelve más evidente cuando los índices de traslación y escalamiento de la doble sumatoria en la IDWT son reemplazados por un único índice j para realizar la suma sobre las neuronas wavelet. El reemplazo es posible debido a que en una WNN los parámetros de traslación y escalamiento son fijados en cada neurona al final del entrenamiento, como se verá más adelante.

El primer modelo propuesto para una red neuronal wavelet es el esquematizado en la Fig. 3.4, el cual corresponde a una estructura feed-forward con una sola capa oculta. En el esquema hemos limitado el número de salidas de la red a una sola para simplificar la explicación de su funcionamiento, la teoría que gobierna a la WNN con una sola salida es fácilmente extendible a una red con múltiples salidas.

La implementación de la WNN se facilita cuando se usa una familia de marcos redundante en lugar del análisis multiresolución [117, 136]. La redundancia en un marco no impone la condición de uso de funciones ortogonales, lo que otorga una gran flexibilidad y añade un efecto positivo que consiste en ayudar a la eliminación de ruido que muchas veces enmascara la señal [137, 138]. Un marco es, en general, un conjunto de vectores $\Phi = \{\varphi_i : i \in \mathbb{Z}\}$ que permiten una representación estable de una señal f por medio de una expansión del tipo $f = \sum_i c_i(f) \varphi_i$, donde los coeficientes $c_i(f)$ son el mapeo de la señal f sobre el espacio generado por el conjunto de vectores $\{\varphi_i\}$ los cuales no son necesariamente ortonormales. Para que un conjunto de vectores $\{\varphi_i\}$ sea considerado un marco se debe cumplir que la energía de los coeficientes $c_i(f)$ resultantes del mapeo yazcan dentro de dos intervalos positivos $A > 0$ y $B < \infty$ conocidos como límites del marco, y que

dichos coeficientes satisfagan la condición $A\|f\|^2 \leq \sum_i |\langle f, c_i \rangle|^2 \leq B\|f\|^2$ [36].

Existen tres posibilidades de clasificar el marco de acuerdo a los valores de A y B : i) cuando $A=B$ al marco resultante se conoce como ajustado, ii) cuando $A=B=1$ entonces la familia de vectores $\{\varphi_i\}$ forma una base ortonormal, iii) cuando $A=B>1$ entonces la familia $\{\varphi_i\}$ es redundante.

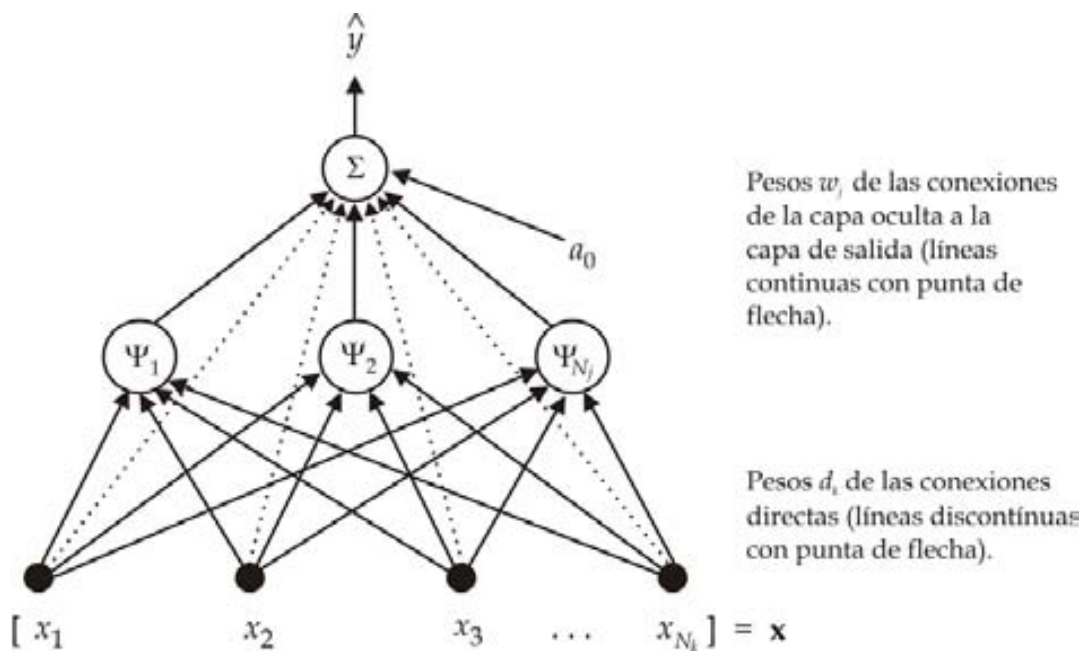


Fig. 3.4. Estructura básica para la implementación de la red neuronal wavelet multi-dimensional. Las funciones de activación multivariable Ψ_j en la capa oculta pueden definirse a partir de la norma Euclidiana o usando el producto tensorial de funciones mono-dimensionales. Los subíndices N_j y N_k indican el número total de neuronas wavelet y la dimensión del tensor \mathbf{x} de entrada, respectivamente.

La familia de marcos usada en la transformación wavelet es de la forma $\Psi = \{\psi_{j,k} : j, k \in \mathbb{Z}\}$, la cual se obtiene al discretizar los parámetros de traslación y escalamiento de la función Ψ . La representación de una señal a partir de estos elementos discretizados es

$$\mathbf{f} = \sum_k \sum_j c_{j,k}(\mathbf{f}) \psi_{j,k} \quad (3.4)$$

donde los factores $c_{j,k}(f)$ equivalen a los coeficientes $D_{j,k} = \langle f, \psi_{j,k} \rangle$ obtenidos de la Transformación Wavelet de una señal. Observando la ecuación (3.4) podemos ver que es equivalente a la transformación wavelet inversa, por lo que podemos considerar que la señal f se recupera a partir de los coeficientes $c_{j,k}(f)$ obtenidos previamente de la transformación wavelet usando la familia de elementos $\{\psi_{j,k}\}$. A partir de esta descripción se deduce que una característica importante de un marco es que debe permitir el análisis y síntesis de una señal, tal y como ocurre con el procesamiento Wavelet.

Además de las conexiones que unen a las neuronas entre capas adyacentes, en el modelo también se muestran conexiones directas que unen a las neuronas de entrada con la neurona de salida. Estas conexiones, cuando están presentes en redes con estructura MLP, permite añadir términos lineales a la función de salida y aumentar su capacidad de discriminación [139]. La salida \hat{y} calculada por la red de la Fig. 3.4 para un tensor multi-dimensional de entrada $\mathbf{x} = \{x_k\}_{k=1}^{N_k}$ es igual a

$$\hat{y} = \sum_{j=1}^{N_j} w_j \cdot \Psi_j(\mathbf{x}) + \sum_{k=1}^{N_k} d_k \cdot x_k + a_0 \quad (3.5)$$

la cual, además de ser función de la variable \mathbf{x} , también es función de los pesos $\mathbf{d} = \{d_k\}_{k=1}^{N_k}$ que forma las conexiones directas, los pesos $\mathbf{w} = \{w_j\}_{j=1}^{N_j}$ que representan a las conexiones de la capa oculta a la capa de salida, los parámetros de traslación y escala que afectan a cada función wavelet multi-dimensional de la capa oculta, y de un valor constante a_0 que es útil en el ajuste de funciones con promedio diferente a cero.

Las funciones multi-dimensionales de activación en las neuronas de la capa oculta de la WNN pueden implementarse tanto con funciones wavelet del tipo *isotrópico* como del tipo *anisotrópico*. La característica que diferencia ambas funciones es su dirección espacial. La primera es invariante a rotaciones, es decir, la función tiene el mismo comportamiento para cualquier dirección espacial, mientras que la segunda requiere de un parámetro de rotación por ser un tipo de función dependiente de la dirección (Fig. 3.5). La

diferencia en sus aplicaciones reside en la importancia que tenga para el análisis conocer la dirección en la que se detecta una singularidad. Nuestra aplicación no precisa de esta ventaja adicional, por lo que nos limitaremos al uso de wavelets isotrópicas que además facilita la implementación computacional y reduce el número de ecuaciones a evaluar.

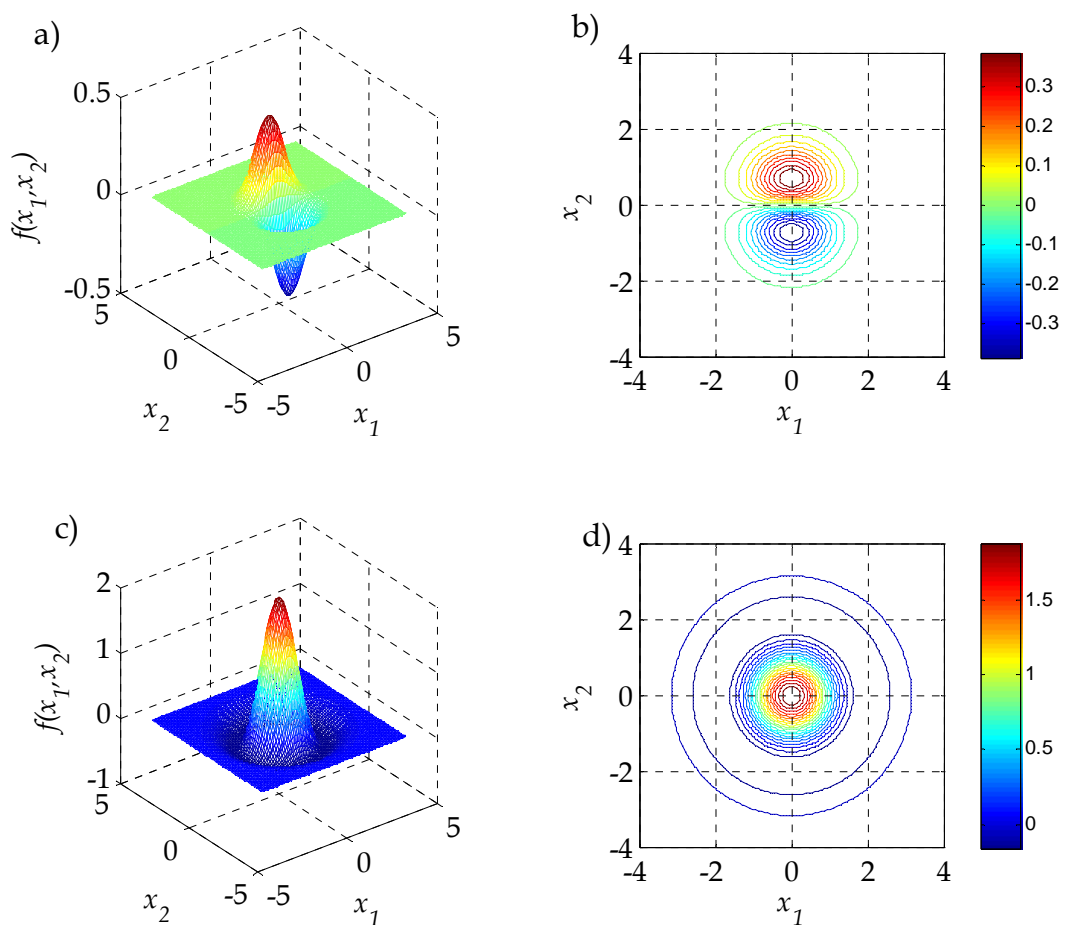


Fig. 3.5. Ejemplos de funciones wavelet bi-dimensionales que muestran cómo influye la dirección espacial. a) Función anisotrópica (wavelet Gaussiana). El valor de la función cambia para un mismo par coordinado (x_1, x_2) si se hace girar la función gira sobre el eje z . En la figura b) se muestra su correspondiente curva de nivel. c) Función isotrópica (wavelet Marr). La función tiene el mismo valor para un par coordinado (x_1, x_2) sin importar si la función gira sobre el eje z , lo que se corrobora en su curva de niveles en d).

Además de la característica de dirección espacial, otra característica adicional que nos es útil en la descripción de las funciones de activación $\Psi(\mathbf{x})$ usadas en la expresión (3.5) es la manera en como se evalúa la

multidimensionalidad de la función. En esta parte del trabajo se usaron marcos wavelet multi-dimensionales basados en la norma Euclidiana de las variables de entrada. Adicionalmente, se describe también otro tipo de marco multi-dimensional usado en las redes neuronales basado en el producto tensorial de funciones wavelet mono-dimensionales que fue desarrollado con anterioridad pero probado en este trabajo con la mejora adicional de contar con una técnica de entrenamiento más eficaz.

3.2.1 Red Neuronal Wavelet con función de activación basada en la norma Euclidiana

Cuando se discretiza una función wavelet cada uno de los miembros de la familia de marcos que se forma es función de un solo parámetro de traslación y otro de escalamiento. En 1995, Kugarajah y Zhang demostraron que una familia de marcos también puede ser formada por funciones wavelet multi-dimensionales construidas con varios coeficientes de traslación y solo uno de escalamiento [136]. Las funciones wavelets con esta característica se evalúan usando la *norma Euclidiana* del tensor construido con las variables de entrada trasladadas. La *norma Euclidiana* o *norma-2* de un vector \mathbf{x} con dimensión N_k se define como

$$\|\mathbf{x}\|_2 = \left(\sum_{k=1}^{N_k} x_k^2 \right)^{\frac{1}{2}} \quad (3.6)$$

Usando la anterior definición de norma vectorial definimos a una función wavelet multi-dimensional $\Psi(\mathbf{x})$ basada en la norma Euclidiana como

$$\Psi(\mathbf{x}) = \frac{1}{\sqrt{s^{N_k}}} \psi \left(\frac{\|\mathbf{x} - \mathbf{m}\|_2}{s} \right) \quad (3.7)$$

la cual nos permitirá construir un marco de funciones wavelet con un solo parámetro de escalamiento. Para el caso de estas funciones multi-dimensionales cada elemento del tensor \mathbf{x} debe verse afectado por su correspondiente parámetro de traslación, por lo que el tensor \mathbf{m} debe tener la misma longitud que el tensor de entrada \mathbf{x} .

Una vez definida la estructura de la red, el tipo de función de transferencia a usar en las neuronas de la capa oculta y la ecuación que describe la salida, se debe realizar el entrenamiento de la red para ajustar sus parámetros. La red se entrenó usando el algoritmo de gradiente conjugado, que será explicado más adelante y que está basado en la retro-propagación del error definida previamente en la sección *Redes Neuronales*. El entrenamiento se realizó haciendo un procesamiento por lote de la información, es decir, el cálculo del error y el ajuste de los parámetros de la red se realizaron una vez que se presentaron los N vectores $\{\mathbf{x}^{(n)}\}_{n=1}^N$ que forman el patrón de entrada para el entrenamiento y se calcularon las N salidas correspondientes a cada uno de ellos. Debido a la característica de dependencia multivariable del modelo expresado en la ecuación (3.5) y para ayudar a la visualización de las siguientes ecuaciones definiremos una variable ficticia Θ para englobar a los parámetros que afectan el comportamiento de la red y cuyos valores deberán ajustarse durante el entrenamiento.

$$\Theta = \{\mathbf{a}_0, \mathbf{d}, \mathbf{w}, \mathbf{s}_j, \mathbf{m}_j\} \quad (3.8)$$

Usando esta nueva variable la salida \hat{y} de la red wavelet puede ser expresada como una función de Θ . A partir de las expresiones en (3.5) y (3.8) y retomando de la Sección 2.2 que la función de costo a minimizar durante el entrenamiento es $E_r = \frac{1}{2} \sum_n (y - \hat{y})^2$ describiremos las ecuaciones que harán que el conjunto de parámetros contenidos en Θ se modifiquen de acuerdo al gradiente de la función de error hasta conseguir minimizar la diferencia entre los valores obtenidos \hat{y} y esperados y . El gradiente de E_r respecto a Θ está descrito por

$$\frac{\partial E_r}{\partial \Theta} = E'_r = - \sum_n (y - \hat{y}) \frac{\partial \hat{y}}{\partial \Theta} \quad (3.9)$$

Las derivadas parciales $\frac{\partial \hat{y}}{\partial \Theta}$ que determinarán los cambios de cada una de las variables contenidas en Θ son

$$\frac{\partial \hat{y}}{\partial \mathbf{a}_0} = 1 \quad (3.10)$$

$$\frac{\partial \hat{y}}{\partial d_k} = x_k \quad (3.11)$$

$$\frac{\partial \hat{y}}{\partial w_j} = \Psi_j(\mathbf{x}) \quad (3.12)$$

$$\frac{\partial \hat{y}}{\partial s_j} = w_j \frac{\partial \Psi_j(\mathbf{x})}{\partial s_j} \quad (3.13)$$

$$\frac{\partial \hat{y}}{\partial \mathbf{m}_j} = w_j \frac{\partial \Psi_j(\mathbf{x})}{\partial \mathbf{m}_j} \quad (3.14)$$

Para calcular el gradiente de los vectores de traslación y escalamiento primero definimos la función wavelet a usar en la red neuronal, la cual debe tener la forma de la ecuación (3.7). En la implementación de la WNN probamos dos funciones wavelet isotrópicas como funciones de transferencia en las neuronas de la capa oculta de la WNN. Las wavelets corresponden a las funciones *Marr* (ψ_1) también conocida como *sombrero mexicano*, y a la función *Halo* (ψ_2) que es el equivalente multi-dimensional de la wavelet *Morlet*. La wavelet Halo es una función real sin orientación espacial creada en 1993 por Dallard y Spedding [140]. Estas dos funciones se definen para el caso multi-dimensional como

$$\Psi_1(\mathbf{x}) = \frac{1}{\sqrt{s^{N_k}}} \left(N_k - \left(\frac{\|\mathbf{x} - \mathbf{m}\|_2}{s} \right)^2 \right) \exp \left(-\frac{\left(\frac{\|\mathbf{x} - \mathbf{m}\|_2}{s} \right)^2}{2} \right) \quad (3.15)$$

y

$$\Psi_2(\mathbf{x}) = \frac{1}{\sqrt{s^{N_k}}} \cos \left(k_0 \cdot \left(\frac{\|\mathbf{x} - \mathbf{m}\|_2}{s} \right) \right) \exp \left(-\frac{\left(\frac{\|\mathbf{x} - \mathbf{m}\|_2}{s} \right)^2}{2} \right) ; k_0 = \pi \sqrt{\frac{2}{\ln(2)}} \quad (3.16)$$

La función Marr fue presentada en los apartados c) y d) de la Fig. 3.5. La función Halo es la que se muestra en la Fig. 3.6 junto con su gráfica de nivel para poder visualizar la carencia de dirección espacial.

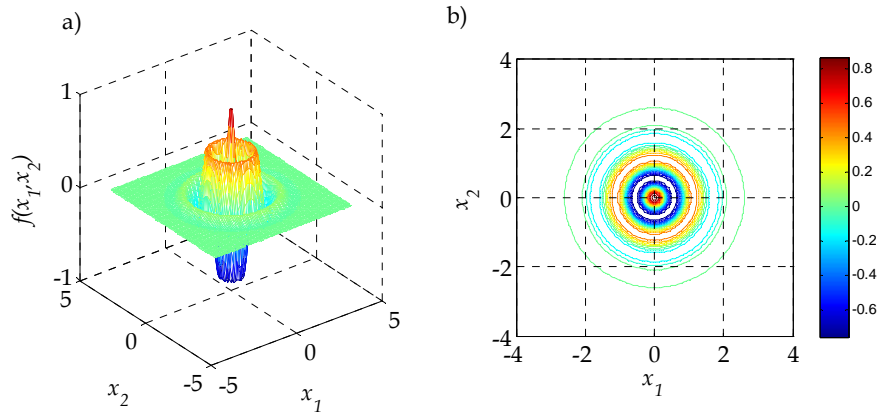


Fig. 3.6. a) Función wavelet isotrópica Halo, b) correspondiente curva de nivel.

Las derivadas de las funciones (3.15) y (3.16) respecto a los parámetros de traslación y escala que se necesitan en las ecuaciones (3.13) y (3.14) para el entrenamiento de la red son

$$\frac{\partial \Psi_j(\mathbf{x})}{\partial s_j} = \frac{\Psi_j(\mathbf{x})}{s_j} \cdot \left(z_j^2 - \frac{N_k}{2} \right) + \frac{2}{s_j \sqrt{s_j^{N_k}}} \cdot z_j^2 \cdot \exp^{-\frac{z_j^2}{2}} \quad (3.17)$$

y

$$\frac{\partial \Psi_j(\mathbf{x})}{\partial m_{k,j}} = \frac{2 \cdot (x_k - m_{k,j})}{s_j^2} \cdot \left(\frac{1}{2} \Psi_j(\mathbf{x}) + \frac{1}{\sqrt{s_j^{N_k}}} \cdot \exp^{-\frac{z_j^2}{2}} \right) \quad (3.18)$$

para la función Marr, y

$$\frac{\partial \Psi_j(\mathbf{x})}{\partial s_j} = \frac{\Psi_j(\mathbf{x})}{s_j} \cdot \left(z_j^2 - \frac{N_k}{2} \right) + k_0 \cdot \frac{z_j}{s_j \sqrt{s_j^{N_k}}} \cdot \sin(k_0 \cdot z_j) \cdot \exp^{-\frac{z_j^2}{2}} \quad (3.19)$$

junto con

$$\frac{\partial \Psi_j(\mathbf{x})}{\partial m_{k,j}} = \frac{(x_k - m_{k,j})}{s_j^2} \cdot \left(\Psi_j(\mathbf{x}) + \frac{k_0}{z_j \sqrt{s_j^{N_k}}} \cdot \sin(k_0 \cdot z_j) \cdot \exp^{-\frac{z_j^2}{2}} \right) \quad (3.20)$$

para la función Halo. En las ecuaciones (3.17)-(3.20) se ha hecho el cambio de variable $\frac{\|\mathbf{x} - \mathbf{m}_j\|_2}{s_j} \rightarrow z_j$ para facilitar la escritura de las mismas.

Para minimizar la función de error E_r , actualizaremos las variables que afectan el comportamiento de la red y que están contenidas en la variable Θ de la expresión (3.8). El método del gradiente descendiente se implementó como primera metodología para el entrenamiento de la WNN, sin embargo, demostró ser ineficiente por la lentitud de la convergencia y por ser propenso a quedar atrapado en un mínimo local, la segunda metodología implementada fue la regla delta basada en el gradiente conjugado. Para aplicar la técnica del gradiente conjugado se necesita conocer las derivadas primera y segunda de la función de error E_r . Las expresiones de la derivada primera para E_r y para cada una de las variables en Θ fueron descritas en las ecuaciones (3.10)-(3.12) y (3.17)-(3.20). En el caso de la derivada segunda de E_r , ésta queda definida como

$$E_r'' = \left(\frac{\partial \hat{y}}{\partial \Theta} \right)^2 - (y - \hat{y}) \frac{\partial^2 \hat{y}}{\partial \Theta^2} \quad (3.21)$$

donde la derivada segunda de \hat{y} respecto a las variables $\{\mathbf{a}_0, \mathbf{d}, \mathbf{w}\}$ en Θ son

$$\frac{\partial^2 \hat{y}}{\partial \mathbf{a}_0^2} = 0 \quad (3.22)$$

$$\frac{\partial^2 \hat{y}}{\partial \mathbf{d}_k^2} = 0 \quad (3.23)$$

$$\frac{\partial^2 \hat{y}}{\partial \mathbf{w}_j^2} = 0 \quad (3.24)$$

Para la función Marr, la derivada segunda de la función (3.15) respecto a las variables $\{\mathbf{s}_j, \mathbf{m}_j\}$ son

$$\frac{\partial^2 \Psi_j(\mathbf{x})}{\partial s_j^2} = \frac{\Psi_j(\mathbf{x})}{s_j^2} \left(\left(\frac{N_k}{2} - z_j^2 \right)^2 + \frac{N_k}{2} - 3 \cdot z_j^2 \right) + \frac{2 \cdot z_j^2}{s_j^2 \sqrt{s_j^{N_k}}} \exp^{-\frac{z_j^2}{2}} (2 \cdot z_j^2 - N_k - 3)$$
(3.25)

y

$$\begin{aligned} \frac{\partial^2 \Psi_j(\mathbf{x})}{\partial m_{k,j}^2} &= 4(x_k - m_{k,j})^2 \left(\frac{\Psi_j(\mathbf{x})}{4 \cdot s_j^4} + \frac{1}{s_j^4 \sqrt{s_j^{N_k}}} \cdot \exp^{-\frac{z_j^2}{2}} \right) - \frac{2 \cdot N_k}{s_j^2 \sqrt{s_j^{N_k}}} \cdot \exp^{-\frac{z_j^2}{2}} \\ &\quad - \Psi_j(\mathbf{x}) \cdot \frac{N_k}{s_j^2} \end{aligned}$$
(3.26)

respectivamente. Para el caso de la wavelet Halo descrita por la ecuación (3.16), las derivadas segundas respecto a cada una de las mismas variables son

$$\begin{aligned} \frac{\partial^2 \Psi_j(\mathbf{x}^{(n)})}{\partial s_j^2} &= \frac{\Psi_j(\mathbf{x}^{(n)})}{s_j^2} \left(\left(\frac{N_k}{2} - z_j^2 \right)^2 + \frac{N_k}{2} - z_j^2 \cdot (k_0 + 3) \right) \\ &\quad + \frac{k_0 \cdot z_j}{s_j^2 \sqrt{s_j^{N_k}}} \cdot \sin(k_0 \cdot z_j) \cdot \exp^{-\frac{z_j^2}{2}} (2 \cdot z_j^2 - N_k - 2) \end{aligned}$$
(3.27)

$$\begin{aligned} \frac{\partial^2 \Psi_j(\mathbf{x}^{(n)})}{\partial m_{k,j}^2} &= \frac{\Psi_j(\mathbf{x}^{(n)})}{s_j^2} \left[\frac{(x_k^{(n)} - m_{k,j})^2}{s_j^2} - \frac{k_0^2 \cdot (x_k^{(n)} - m_{k,j})^2}{z_j^2 \cdot s_j^2} - N_k \right] \\ &\quad + \frac{\sin(k_0 \cdot z_j) \cdot k_0 \cdot \exp^{-\frac{z_j^2}{2}}}{s_j^2 \sqrt{s_j^{N_k}} \cdot z_j} \left[\frac{(x_k^{(n)} - m_{k,j})^2}{z_j^2 \cdot s_j^2} + \frac{2 \cdot (x_k^{(n)} - m_{k,j})^2}{s_j^2} - N_k \right] \end{aligned}$$
(3.28)

Con el conjunto de ecuaciones descritos por las expresiones (3.22)-(3.28) se completan las ecuaciones para evaluar de la derivada segunda de la función de error E_r que es necesaria en el cálculo de los parámetros α y β usados en la técnica de entrenamiento basada en el gradiente conjugado.

La anterior descripción de la metodología de construcción y entrenamiento de la red neuronal con funciones de activación wavelet basadas en la norma Euclidiana se condensa en el algoritmo presentado en el Apéndice 1. A pesar de la complejidad de las ecuaciones relacionadas con las derivadas parciales usadas en el cálculo del gradiente conjugado para el entrenamiento de la WNN, su implementación se minimiza porque existen valores tales como z_j y las funciones $\Psi_j(\cdot)$, $\sin(\cdot)$, $\cos(\cdot)$ y $\exp(\cdot)$ que se calculan una vez por iteración.

3.2.2 Red Neuronal Wavelet con función de activación basada en el producto tensorial

Una alternativa a las funciones wavelet multi-dimensionales basadas en la norma Euclidiana son las wavelet multi-dimensionales construidas con el producto tensorial de funciones wavelet mono-dimensionales. Este tipo de funciones se construyen mediante

$$\Psi(\mathbf{x}) = \psi(x_1) \cdots \psi(x_{N_k}) \quad (3.29)$$

donde cada función $\psi(x_k)$ se evalúa con su correspondiente pareja de parámetros $\{m_k, s_k\}$ que son diferentes para cada valor x_k , con lo que cada función multi-dimensional $\Psi(\mathbf{x})$ implementada en cada neurona oculta de esta nueva WNN debe contar con un conjunto de parámetros de traslación y escala de la misma longitud que el tensor \mathbf{x} de entrada. Debido a este nuevo esquema de correspondencia entre las variables que se evalúan en la función $\psi(x)$, el cambio de variable para el argumento de las funciones wavelet se

define ahora como $\frac{x_k - m_{k,j}}{s_{k,j}} \rightarrow z_{k,j}$, por lo que

$$\Psi_j(\mathbf{x}) = \psi(z_{1,j}) \cdots \psi(z_{k,j}) \cdots \psi(z_{N_k,j}) \quad (3.30)$$

La estructura de la red wavelet no cambia para esta segunda aproximación en la que usamos funciones wavelet basadas en el producto tensorial, lo que se modifica son las ecuaciones para el ajuste de los

parámetros de la red durante el entrenamiento además de la función de transferencia. El entrenamiento de la red también está basado en la minimización de la función de error E_r anteriormente descrita. El conjunto de variables contenidas en Θ se redefine para esta segunda aproximación como

$$\Theta = \{a_0, \mathbf{d}, \mathbf{w}, \mathbf{s}_j, \mathbf{m}_j\} \quad (3.31)$$

Respecto al anterior conjunto de variables en Θ de la expresión (3.8), la nueva expresión (3.31) cambia sólo en el parámetro de escalamiento s_j que ahora es de carácter univariable; esta modificación nos permite seguir usando las ecuaciones (3.10) a (3.12) para calcular el gradiente en los pesos de las conexiones directas \mathbf{d} , los pesos de las conexiones a la capa de salida \mathbf{w} , y el valor de desplazamiento a_0 , sin embargo, las ecuaciones (3.17) a (3.20) usadas para modificar los parámetros de traslación y escalamiento durante el entrenamiento de la red deben redefinirse debido a que ha cambiado la manera en como se evalúa la función wavelet. Para determinar el gradiente de cambio en cada uno de los k -ésimos elementos en los tensores de escalamiento \mathbf{s}_j y traslación \mathbf{m}_j haremos uso de la regla de la cadena y el cambio de variable anteriormente descrito. La derivada de \hat{y} respecto a cada k -ésimo elemento en \mathbf{s}_j y \mathbf{m}_j se definen ahora como

$$\frac{\partial \hat{y}}{\partial \mathbf{s}_{k,j}} = \frac{\partial \hat{y}}{\partial \Psi_j} \frac{\partial \Psi_j}{\partial \mathbf{s}_{k,j}} = w_j \frac{\partial \Psi_j}{\partial \mathbf{s}_{k,j}} \quad (3.32)$$

$$\frac{\partial \hat{y}}{\partial \mathbf{m}_{k,j}} = \frac{\partial \hat{y}}{\partial \Psi_j} \frac{\partial \Psi_j}{\partial \mathbf{m}_{k,j}} = w_j \frac{\partial \Psi_j}{\partial \mathbf{m}_{k,j}} \quad (3.33)$$

La derivada $\partial \Psi_j / \partial \mathbf{s}_{k,j}$ queda descrita para estas funciones basadas en el producto tensorial de funciones wavelet mono-dimensionales como

$$\frac{\partial \Psi_j}{\partial \mathbf{s}_{k,j}} = \psi(\mathbf{z}_{1,j}) \psi(\mathbf{z}_{2,j}) \dots \frac{\partial \psi(\mathbf{z}_{k,j})}{\partial \mathbf{s}_{k,j}} \dots \psi(\mathbf{z}_{N_k,j}) \quad (3.34)$$

La derivada $\partial\Psi_j/\partial m_{k,j}$ se define de manera semejante. Las wavelet *Marr* (ψ_1) y la parte real de la wavelet *Morlet* (ψ_2) son usadas en esta segunda aproximación (Fig. 3.7). Estas funciones mono-dimensionales se describen mediante

$$\psi_1(x) = \frac{1}{\sqrt{s}} \left(1 - \left(\frac{x-m}{s} \right)^2 \right) \exp \frac{\left(\frac{x-m}{s} \right)^2}{2} \quad (3.35)$$

y

$$\psi_2(x) = \frac{1}{\sqrt{s}} \cos \left(k_0 \cdot \left(\frac{x-m}{s} \right) \right) \exp \frac{\left(\frac{x-m}{s} \right)^2}{2} ; k_0 = \pi \sqrt{\frac{2}{\ln(2)}} \quad (3.36)$$

para (ψ_1) y (ψ_2), respectivamente. Las derivadas primeras de Ψ_j respecto a $s_{k,j}$ y $m_{k,j}$ para la función *Marr* (ψ_1) son:

$$\frac{\partial \psi_1(x)}{\partial s_{k,j}} = \frac{\psi_1(x_k)}{s_{k,j}} \cdot \left((z_{k,j})^2 - 2 \right) + 2 \cdot \exp \frac{(z_{k,j})^2}{2} \cdot \frac{(z_{k,j})^2}{s_j \sqrt{s_j}} \quad (3.37)$$

y

$$\frac{\partial \psi_1(x)}{\partial m_{k,j}} = \psi_1(x_k) \frac{z_{k,j}}{s_{k,j}} + \frac{2 \cdot z_{k,j}}{s_{k,j} \sqrt{s_{k,j}}} \cdot \exp \frac{(z_{k,j})^2}{2} \quad (3.38)$$

Las correspondientes derivadas para la parte real de la función *Morlet* (ψ_2) son

$$\frac{\partial \psi_2(x)}{\partial s_{k,j}} = \frac{\psi_2(x_k)}{s_{k,j}} \left((z_{k,j})^2 - \frac{1}{2} \right) + k_0 \cdot \sin(k_0 \cdot z_{k,j}) \frac{z_{k,j}}{s_{k,j} \cdot \sqrt{s_{k,j}}} \exp \frac{(z_{k,j})^2}{2} \quad (3.39)$$

y

$$\frac{\partial \psi_2(x)}{\partial m_{k,j}} = \psi_2(x_k) \frac{z_{k,j}}{s_{k,j}} + \frac{k_0}{s_{k,j} \sqrt{s_{k,j}}} \cdot \sin(k_0 \cdot z_{k,j}) \cdot \exp \frac{(z_{k,j})^2}{2} \quad (3.40)$$

Al igual que en el anterior modelo de WNN, el entrenamiento se hace por lote y los cambios en cada una de las variables de la red wavelet se calculan en cada iteración usando el método del gradiente conjugado anteriormente descrito.

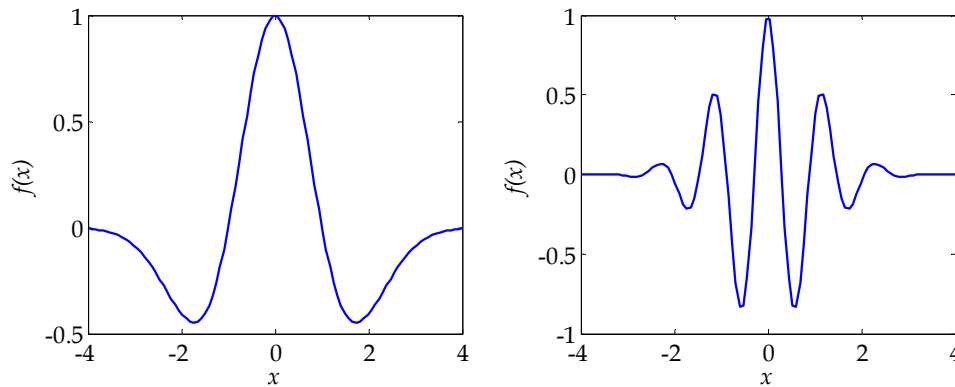


Fig. 3.7. Wavelets Marr (gráfica de la izquierda) y parte real de la wavelet Morlet (gráfica de la derecha) usadas como funciones de transferencia mono-dimensionales.

Las ecuaciones que necesitamos modificar para calcular los parámetros α y β que se necesitan en el entrenamiento de la red son las derivadas segunda de (ψ_1) y (ψ_2) respecto a $m_{k,j}$ y $s_{k,j}$. Para la función Marr estas derivadas quedan descritas por

$$\frac{\partial^2 \Psi_j(\mathbf{x})}{\partial s_{k,j}^2} = \frac{\Psi_j(\mathbf{x})}{s_{k,j}^2} \left[\frac{3}{4} - 4 \cdot (z_{k,j})^2 + (z_{k,j})^4 \right] + \frac{4 \cdot (z_{k,j})^2}{s_{k,j}^2 \sqrt{s_{k,j}}} \cdot \exp^{-\frac{(z_{k,j})^2}{2}} \cdot \left[(z_{k,j})^2 - 2 \right] \quad (3.41)$$

$$\frac{\partial^2 \Psi_j(\mathbf{x})}{\partial m_{k,j}^2} = \frac{\Psi_j(\mathbf{x})}{s_{k,j}^2} \left[(z_{k,j})^2 - 1 \right] + \frac{2}{s_{k,j}^2 \sqrt{s_{k,j}}} \cdot \exp^{-\frac{(z_{k,j})^2}{2}} \left[2 \cdot (z_{k,j})^2 - 1 \right] \quad (3.42)$$

y para la parte real de la función Morlet las derivadas segundas son

$$\begin{aligned} \frac{\partial^2 \Psi_j(\mathbf{x})}{\partial \mathbf{s}_{k,j}^2} &= \frac{\Psi_j(\mathbf{x})}{\mathbf{s}_{k,j}^2} \left[\frac{3}{4} - 4 \cdot (\mathbf{z}_{k,j})^2 - k_0^2 \cdot (\mathbf{z}_{k,j})^2 + (\mathbf{z}_{k,j})^4 \right] \\ &+ \frac{k_0 \cdot \mathbf{z}_{k,j}}{\mathbf{s}_{k,j}^2 \sqrt{\mathbf{s}_{k,j}}} \sin(k_0 \cdot \mathbf{z}_{k,j}) \cdot \exp\left(-\frac{(\mathbf{z}_{k,j})^2}{2}\right) \left[2 \cdot (\mathbf{z}_{k,j})^3 - 2 \cdot (\mathbf{z}_{k,j}) \right] \end{aligned} \quad (3.43)$$

$$\frac{\partial^2 \Psi_j(\mathbf{x})}{\partial m_{k,j}^2} = \frac{\Psi_j(\mathbf{x})}{\mathbf{s}_{k,j}^2} \left[(\mathbf{z}_{k,j})^2 - 1 - k_0^2 \right] + \frac{2 \cdot k_0 \cdot \mathbf{z}_{k,j}}{\mathbf{s}_{k,j}^2 \sqrt{\mathbf{s}_{k,j}}} \sin(k_0 \cdot \mathbf{z}_{k,j}) \cdot \exp\left(-\frac{(\mathbf{z}_{k,j})^2}{2}\right) \quad (3.44)$$

Las ecuaciones (3.30) y (3.32-3.44) que se usan en este nuevo esquema de red neuronal wavelet tienen la desventaja de ser computacionalmente más costosas que las ecuaciones de la red con funciones de transferencia basadas en la norma Euclidiana. Tomando como ejemplo la wavelet Marr descrita por las ecuaciones (3.15) y (3.35) y considerando que la multi-dimensionalidad de la segunda ecuación se rige por el producto tensorial descrito por la ecuación (3.29) se puede notar que se necesitan N_k evaluaciones de la función wavelet, además de $N_k - 1$ operaciones de multiplicación, mientras que en la primera sólo se añade el cálculo de la norma Euclidiana a una sola evaluación de la función Wavelet. La gran ventaja que tiene una red wavelet basada en el producto tensorial es la gran cantidad de grados de libertad que posee debido a que se crea una función wavelet por cada variable que existe en el tensor \mathbf{x} de entrada.

Cualquiera de las dos estructuras de WNN anteriormente descritas con una sola neurona en la capa de salida es fácilmente extendible a una estructura con múltiples neuronas de salida. Para conseguirlo se agregan las neuronas necesarias a la capa de salida (una para cada variable de salida que se desee calcular), conexiones entre las nuevas neuronas y las neuronas de la capa oculta, y conexiones directas de la entrada hacia las nuevas neuronas de salida. Esto obliga a aumentar en una unidad tanto el orden de los tensores de pesos \mathbf{w} y \mathbf{d} como a redefinir la función de error E_r con una doble sumatoria,

$$E_r = \frac{1}{2} \sum_n \sum_i (y_i^n - \hat{y}_i^n)^2, \text{ donde el nuevo subíndice } i \text{ sirve para señalar a las}$$

diferentes salidas de la red. Las ecuaciones para el entrenamiento de la WNN con múltiples salidas son fácilmente obtenibles a partir de las ecuaciones anteriormente planteadas.

El algoritmo para la construcción y entrenamiento de la red wavelet con funciones multi-dimensionales basadas en el producto tensorial está descrito en el Apéndice 2.

4

Redes Neuronales B-spline

En los últimos años ha surgido el interés por usar funciones de activación adaptativas en la construcción de ANNs tales como polinomios de Hermite [141] y funciones polinomiales en estructuras llamadas redes neuronales polinomiales adaptativas [142]. Además de estas dos funciones polinomiales también se han usado funciones de activación spline. Las redes neuronales basadas en funciones spline han demostrado tener mejor capacidad de aproximación y generalización [143, 144]. Dentro de las spline, las funciones B-spline, en particular, tienen una interesante propiedad que consiste en la capacidad de aproximar cualquier función arbitraria a cualquier nivel de precisión deseado [145]. Esta capacidad de aproximación junto con la capacidad de modelado no-lineal de las redes neuronales permite fusionar ambas teorías para dar origen a una estructura de red neuronal conocida como *Red Neuronal B-Spline (BSNN)*. Además de la propiedad de aproximación, los modelos basados en funciones B-spline han demostrado una mejor capacidad de generalización en funciones multi-dimensionales que la mayoría de los modelos basados en funciones base radial [146, 147].

La construcción de la BSNN no se basa únicamente en las dos características mencionadas en el párrafo anterior, sino también en estudios realizados en 1990 por Poggio y Girosi [148] que demostraron que existe una estrecha relación entre la aproximación de funciones usando splines y las redes neuronales, y en los avances en teoría de regularización que apoyan el uso de funciones kernel aditivas tipo spline en la construcción un modelo de aproximación (esto equivale a construir una red neuronal con una capa oculta)

[149]. Una interesante ventaja de la red neuronal B-spline sobre las redes MLP es que las BSNN almacenan información de manera local, lo cual significa que el aprendizaje sobre un intervalo del espacio de entrada afecta mínimamente al resto de la red.

La BSNN es también una clase de red neuronal feed-forward con tres capas de neuronas: una de entrada, otra oculta construida con funciones de activación B-spline y una de salida. Pueden construirse redes para variables de entrada de los tipos mono- y multi-dimensional. En el caso de las funciones B-spline multi-dimensional, ésta se construye usando el producto tensorial de funciones B-spline mono-dimensionales.

4.1 Técnicas de calibración univariable

Antes de explicar brevemente la teoría de las funciones *spline* y *B-spline* usadas en las herramientas de calibración propuestas este trabajo de tesis, haremos una breve mención de otras técnicas de calibración de carácter univariable tales como los *polinomios de Lagrange* y la *interpolación de Hermite*. El objetivo final es describir las ventajas que presenta construir modelos usando funciones spline y B-spline.

4.1.1 Interpolación polinomial

En el área de análisis numérico se conoce como interpolación polinomial a la obtención de una función f dentro de una cierta clase prefijada de funciones que relacione un conjunto de puntos $\{x_0, x_1, x_2, \dots, x_n\}$ con el valor de la función de ellos $\{y_0, y_1, y_2, \dots, y_n\}$ de tal manera que $y = f(x)$ [150].

La situación más común es el problema de interpolación polinómica, donde f se obtiene de un conjunto de polinomios de grado menor o igual a n . El uso de polinomios de interpolación de grado superior puede producir errores grandes debido al alto grado de oscilación que presentan. Para evitar este problema es posible aproximar la función desconocida en intervalos pequeños usando polinomios de grado bajo.

4.1.2 Polinomios de Lagrange

Un procedimiento para obtener un polinomio de interpolación de grado n es la fórmula de *Lagrange* [150, 151, 152]. Un polinomio de Lagrange de grado menor o igual a n usado para la interpolación de n pares de puntos $\{x_i, y_i\}$ tiene la forma $p_n(x) = \sum_{i=0}^n y_i L_i(x)$. Los términos $L_i(x)$ del polinomio de interpolación son independientes de los valores y_i , y una vez hallados sirven para escribir el polinomio en los puntos $\{x_i, y_i\}$ para $i=0, 1, \dots, n$. El polinomio de interpolación de Lagrange tiene el inconveniente de que si se añade un nuevo punto al conjunto $\{x_i, y_i\}$ entonces el nuevo polinomio $p_{n+1}(x)$ no se puede obtener del anterior y tiene que ser recalculado [151].

La falta de relación entre el primer polinomio $p_n(x)$ y el nuevo polinomio $p_{n+1}(x)$ que se obtiene agregando un punto de interpolación al conjunto $\{x_i, y_i\}$ se resuelve usando el algoritmo de Horner, también conocido como algoritmo de diferencias divididas [153]. El algoritmo de diferencias divididas sirve para crear polinomios de interpolación de manera sucesiva mediante un esquema recursivo. El algoritmo está definido por

$$p_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \dots + f[x_0, x_1, \dots, x_n](x - x_0)(x - x_1)\dots(x - x_{n-1}) \quad (4.1)$$

donde las diferencias divididas quedan determinadas por el siguiente método recursivo

$$f[x_0, x_1, \dots, x_n] = \frac{f[x_1, x_2, \dots, x_n] - f[x_0, x_1, \dots, x_{n-1}]}{x_n - x_0} \quad (4.2)$$

El algoritmo de diferencias divididas se usa también en el desarrollo de funciones B-spline.

4.1.3 Interpolación de Hermite

Los polinomios de interpolación de Lagrange de alto grado tienen un problema de oscilación en los extremos de interpolación $[x_0, x_n]$ conocido como fenómeno de Runge (Fig. 4.1) [150, 153, 154]. Para intentar mejorar o eliminar este fenómeno se hace que el polinomio interpolador no sólo coincida con la

función en los valores $f(x_i)$, sino también en la primera derivada en los nodos [150, 154]. Al polinomio que interpola de esta forma se le denomina *polinomio de Hermite*. Los polinomios de interpolación de Hermite no eliminan el fenómeno de Runge, la manera eficaz de hacerlo es con una interpolación polinomial a intervalos conocida como *interpolación por splines*.

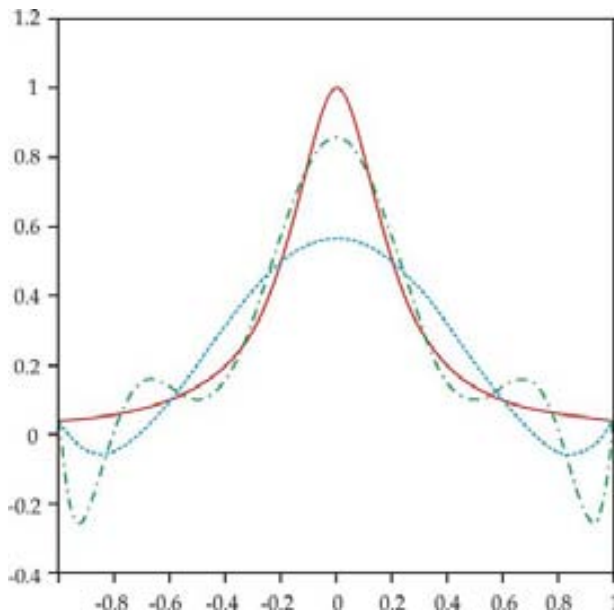


Fig. 4.1. Fenómeno de Runge que se presenta en las interpolaciones polinómicas. La línea roja continua es la función de Runge, la línea azul discontinua (línea-línea) es una aproximación con un polinomio de 5º grado y la línea verde discontinua (punto-línea) es una aproximación con un polinomio de 9º grado. La aproximación en los extremos de la función es peor para un polinomio de mayor grado.

4.2 Funciones spline

Las *funciones spline* deben su nombre a las reglas flexibles sujetas en dos puntos usadas en dibujo técnico para trazar curvas suaves pasando por puntos específicos [155]. Una *función spline* o simplemente *spline* es una función polinomial definida por segmentos, de grado definido, que se unen entre sí en puntos llamados nodos obedeciendo a ciertas condiciones de continuidad [150, 152, 153].

El número de parámetros que se necesitan para definir una función spline se puede obtener a partir de los parámetros de las funciones polinomiales que la definen por segmentos y de las restricciones de continuidad en los extremos de dichos segmentos. En general, se puede estimar que el número de parámetros que definen a una spline de grado k en un intervalo de n segmentos es $n+k$ [151].

Las funciones spline pueden ser creadas para cualquier grado k , pero las interpolaciones más frecuentes son las de grado 1 a 3; su desarrollo puede encontrarse en las referencias [150-155]. Los splines de grado 1 están contruidos con segmentos lineales a trozos y continuas. Gráficamente se representan por poligonales. Una spline de segundo grado o cuadrática es un polinomio a trozos cuyos segmentos son de segundo grado. Gráficamente son parábolas en cada uno de los segmentos conectados, con continuidad de la función y su primera derivada.

Las funciones spline cúbicas son las funciones polinomiales a trozos más usadas en la práctica, definidas como polinomios de tercer grado con continuidad tanto en la función misma como en sus derivadas primera y segunda. El spline cúbico está determinado por $n+3$ parámetros. Para determinar de forma única a un spline cúbico de interpolación se necesitan dos condiciones adicionales a los $n+1$ nodos. Las condiciones más usadas dan origen a *splines naturales* o *con frontera libre* (la segunda derivada de los nodos extremos es igual a cero, $s''(x_0) = s''(x_n) = 0$), *splines con frontera sujeta* (la primer derivada de los nodos extremos tiene un valor predeterminado, $s'(x_0) = y'_0$ y $s'(x_n) = y'_n$), y *splines periódicos* (la primer y segunda derivadas de la función en el nodo inicial es igual a la primer y segunda derivada de la función en el nodo final, respectivamente, $s'(x_0) = s'(x_n)$ y $s''(x_0) = s''(x_n)$).

4.2.1 Funciones B-splines

Las funciones spline anteriormente descritas tienen estructura de espacio vectorial con las operaciones usuales de suma de funciones y producto por escalares [156], esto permite construir funciones spline en un intervalo $[x_0, x_n]$ a partir de los miembros de una familia de funciones base [150, 156, 157]. Una spline construida a partir de funciones base es una ecuación paramétrica descrita mediante la siguiente expresión,

$$s(x) = \sum_{i=0}^n c_i B_i^k(x) \quad n \geq k-1 \quad (4.3)$$

Los términos $B_i^k(x)$ de la ecuación anterior son funciones polinómicas por trozos con soporte finito conocidas como *splines básicas* o *B-splines*, y los coeficientes c_i son puntos de control, los cuales forman un polígono envolvente y se encargan de modificar la forma de $s(x)$ formada por B-splines con nodos fijos. Una B-spline uniforme B^k de grado k (y orden $k+1$) queda definida por la siguiente relación recursiva

$$B^k(x) = \int_{x-1}^x B^{k-1} \quad (4.4)$$

la cual comienza con la función característica $B^0 = 1$ en el intervalo unitario $[0,1)$. De manera equivalente,

$$\frac{d}{dx} B^k(x) = B^{k-1}(x) - B^{k-1}(x-1) \quad (4.5)$$

Las propiedades básicas de una función B-spline son [157, 158]:

- Positividad y soporte local. Una función B-spline es positiva y tiene soporte mínimo en el intervalo $[0, n+1)$, de hecho las B-spline son las splines polinómicas más pequeñas posibles.
- Son una partición de la unidad. La suma de las B-spline dentro de un intervalo especificado es igual a 1, es decir, $\sum_{i=0}^k B_i^k(x) = 1$.
- Suavidad. Una B^k tiene $(k-1)$ derivadas continuas con discontinuidades de la k -ésima derivada en los nodos.
- Es una estructura polinomial definida por segmentos. B^k es un polinomio de grado k en cada intervalo $[i, i+1]$ para $i = 0, \dots, n$.
- Simetría. Una B-spline de grado k es simétrica, es decir, $B^k(x) = B^k(k+1-x)$ donde $k+1$ representa la posición del nodo final.

La ecuación recursiva (4.4) usada para definir las B-spline no es eficiente desde un punto de vista computacional [159]. Una manera más simple para evaluar una B-spline es a través del algoritmo desarrollado por Carl de Boor [158-160] basado en las diferencias divididas, la cual fue descrita en la ecuación (4.2),

$$B_i^1(x) = \begin{cases} 1, & x_i \leq x < x_{i+1} \\ 0 & \text{para los demás casos} \end{cases} \quad (4.6)$$

$$B_i^m(x) = \frac{x - x_{i-k}}{x_{i-1} - x_{i-k}} B_{i-1}^{m-1}(x) + \frac{x_i - x}{x_i - x_{i-k+1}} B_i^{m-1}(x)$$

El superíndice m en la ecuación anterior indica el orden de la función B-spline, $m=k+1$. La ecuación (4.6) muestra que las funciones B-spline de cualquier orden pueden describirse a partir de con una secuencia de nodos x_i de números reales no decreciente como una combinación lineal de funciones base de menor orden. Para definir una B-spline a partir de diferencias divididas se necesitan $m+1$ nodos. La forma de las funciones base depende del espaciamiento entre nodos ya que las B_i^m están basadas en las diferencias entre ellos. Para un espaciamiento unitario, las B-spline de órdenes 1 a 4 obtenidas con el algoritmo de diferencias divididas de de Boor se han graficado en la Fig. 4.2.

A partir de las ecuaciones descritas en la Fig. 4.2 para cada uno de los segmentos de las B-spline de órdenes 1 a 4 se obtiene su siguiente representación matricial definida por $B_0^m = \mathbf{x} \cdot \mathbf{M}$,

$$B_0^2(x) = [x \quad 1] \begin{bmatrix} -1 & 1 \\ 2 & 0 \end{bmatrix} \quad (4.7)$$

$$B_0^3(x) = [x^2 \quad x \quad 1] \begin{bmatrix} \frac{1}{2} & -1 & \frac{1}{2} \\ -3 & 3 & 0 \\ \frac{9}{2} & -\frac{3}{2} & 0 \end{bmatrix} \quad (4.8)$$

$$B_0^4(x) = [x^3 \quad x^2 \quad x \quad 1] \begin{bmatrix} 0 & \frac{2}{3} & \frac{22}{3} & \frac{32}{3} \\ 0 & -2 & 10 & -8 \\ 0 & 2 & -4 & 2 \\ \frac{1}{6} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{6} \end{bmatrix} \quad (4.9)$$

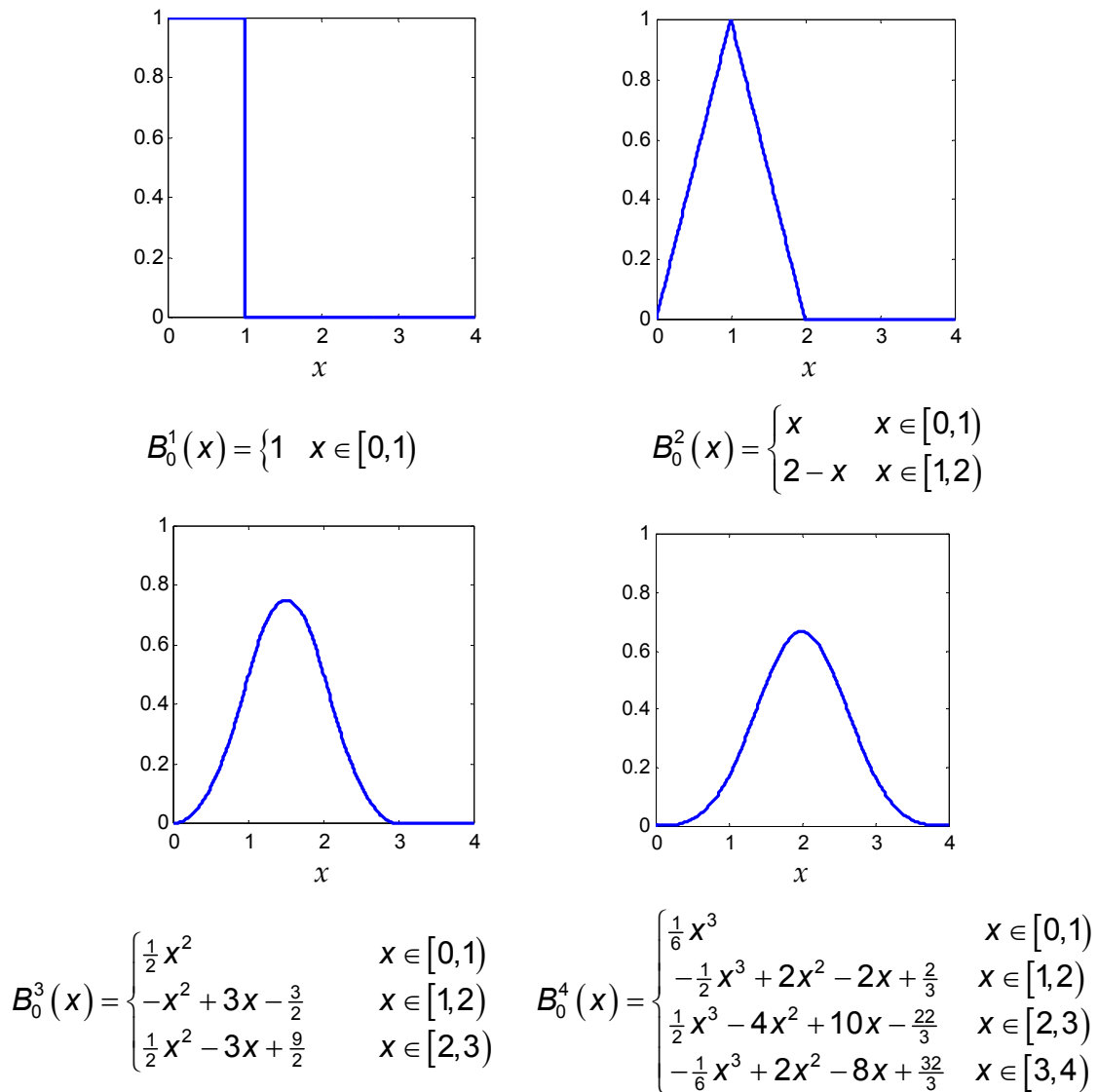


Fig. 4.2. Funciones B-spline de orden $m = 1, 2, 3, 4$ con nodos equidistantes.

Las cuales se usan para evaluar la función spline de la ecuación (4.4) de una manera más eficiente desde un punto de vista computacional mediante

$$s(x) = \mathbf{x} \cdot \mathbf{M} \cdot \mathbf{c} \quad (4.10)$$

donde \mathbf{x} es la base monomial de los valores en x que serán evaluados, \mathbf{M} es el conjunto de coeficientes que definen cada uno de los segmentos de una B-spline de orden m como se describen en las ecuaciones (4.7) a (4.9), y \mathbf{c} es un vector de puntos de control. En esta tesis se ha usado el algoritmo de de

Boor para encontrar las expresiones que definen los segmentos que forman las B-spline de órdenes 2, 3 y 4 con espaciamiento uniforme y longitud arbitraria entre nodos. Dichas expresiones se implementaron en el diseño de redes neuronales con funciones de transferencia B-spline.

4.2.2 B-splines multivariable

Las B-spline multivariable son una extensión de las B-spline de una variable. El tipo más común se obtiene a partir del producto tensorial de B-splines univariable [161]. El producto tensorial entre dos vectores columna \mathbf{x} y \mathbf{y} se define como [162]

$$\mathbf{x} \otimes \mathbf{y} = \mathbf{x} \cdot \mathbf{y}^T \quad (4.11)$$

que para el caso de dos vectores \mathbf{x} y \mathbf{y} con 3 elementos cada uno el resultado del producto tensorial es

$$\mathbf{x} \otimes \mathbf{y} = \begin{bmatrix} x_1 y_1 & x_1 y_2 & x_1 y_3 \\ x_2 y_1 & x_2 y_2 & x_2 y_3 \\ x_3 y_1 & x_3 y_2 & x_3 y_3 \end{bmatrix} \quad (4.12)$$

Usando como ejemplo de construcción de una B-spline multivariable el caso más sencillo que es el que corresponde a una B-spline de dos variables, el resultado del producto tensorial será una superficie descrita por

$$\mathbf{s}(x_1, x_2) = \sum_{i=0}^k \sum_{j=0}^l d_{ij} B_i^m(x_1) B_j^m(x_2) \quad (4.13)$$

donde las funciones base B_i^m y B_j^m quedan definidas como en la ec. (4.6) y los términos d_{ij} son elementos una matriz de puntos de control. De manera semejante a las funciones B-spline de una variable, la superficie B-spline definida por la ec. (4.13) está formada por una suma ponderada de superficie base spline.

Una B-spline bi-variable también puede evaluarse usando una forma matricial equivalente a la expresión (4.10) siempre que los nodos que forman las B-spline estén uniformemente espaciados,

$$\mathbf{s}_{ij}(x,y) = \mathbf{x} \cdot \mathbf{M} \cdot \mathbf{D} \cdot \mathbf{M}^T \cdot \mathbf{y}^T \quad (4.14)$$

La ecuación (4.14) es la forma matricial de una B-spline de dos dimensiones donde matriz \mathbf{M} es la matriz de coeficientes descrita por las ecuaciones (4.7) a (4.9), dependiendo del orden de la B-spline. Si los nodos usados para describir a \mathbf{s}_{ij} no son equidistantes entonces las funciones base spline deberán evaluarse usando el algoritmo de diferencias divididas de la ecuación (4.6).

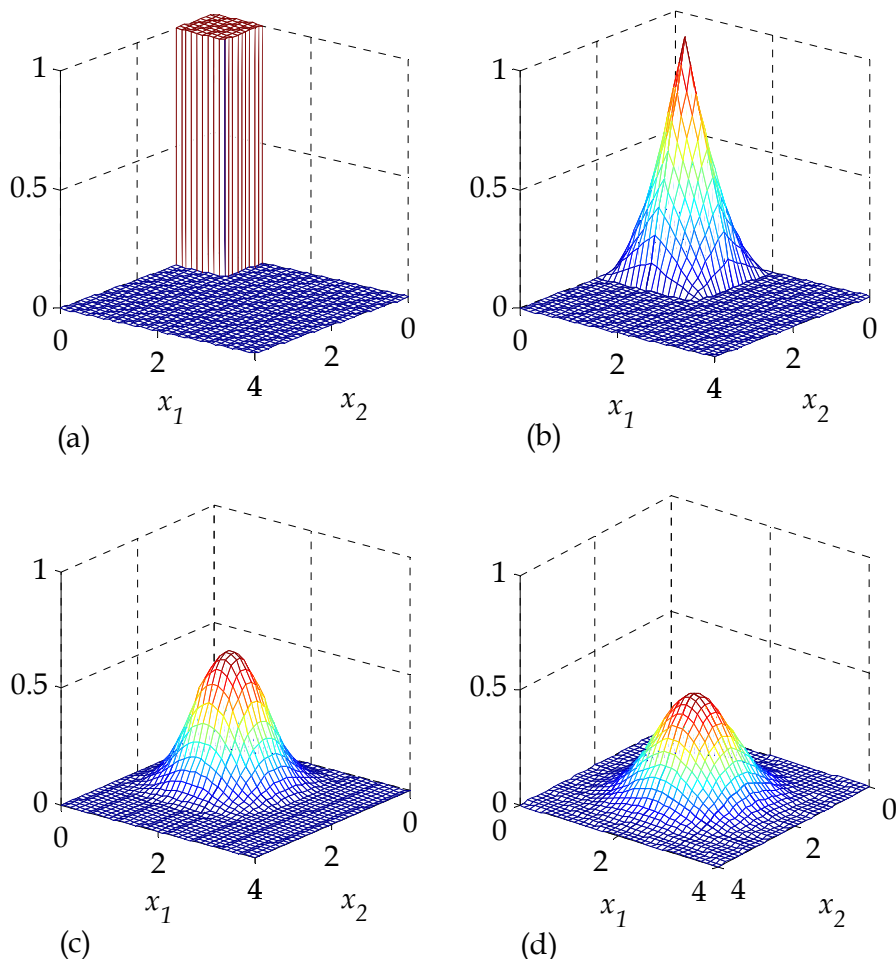


Fig. 4.3. Funciones B-spline bi-variable obtenidas a partir del producto de funciones B-spline mono-dimensionales. a) Primer orden, b) segundo orden, c) tercer orden, d) cuarto orden.

4.3 Red B-spline mono-dimensional

El esquema de una BSNN mono-dimensional se muestra en la Fig. 4.4. La aproximación producida por una red B-spline es el resultado de la superposición de un conjunto de funciones base locales que están afectadas por un coeficiente que representa la menor o mayor contribución de cada una de las funciones B-spline al ajuste de la relación entre los valores de entrada y salida.

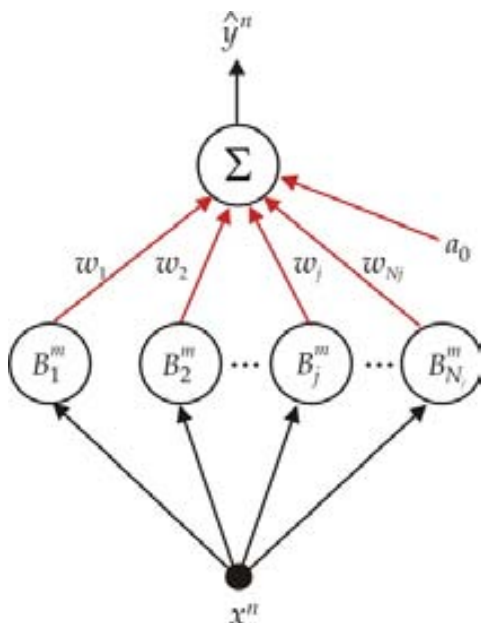


Fig. 4.4. Estructura básica de una BSNN mono-dimensional. Cada neurona B-spline cubre un subintervalo dentro del intervalo de la variable x de entrada.

Las funciones de transferencia en las neuronas de la capa oculta de la BSNN se construyeron con funciones de activación B-spline de órdenes 2 y 3, y nodos equidistantes. No se implementaron funciones de mayor orden debido a la complejidad y exigencias de cálculo que representan. Las funciones B-spline se evaluaron usando el algoritmo de de Boor descrito por la ecuación (4.2). Tomando como base a la función B-spline de orden 1 que se define como

$$B_i^1(x) = \begin{cases} = 1, & x \in [x_i, x_{i+1}) \\ = 0, & \text{fuera del intervalo} \end{cases} \quad (4.15)$$

describimos a las funciones B-spline de orden 2 y 3 implementadas en la BSNN como

$$\begin{aligned}
 B_i^2(x) &= \frac{x - x_i}{x_{i+1} - x_i} B_i^0(x) + \frac{x_{i+2} - x_i}{x_{i+2} - x_{i+1}} B_{i+1}^0(x) \\
 &= \begin{cases} \frac{x - x_i}{x_{i+1} - x_i}, & x \in [x_i, x_{i+1}) \\ \frac{x_{i+2} - x}{x_{i+2} - x_{i+1}}, & x \in [x_{i+1}, x_{i+2}) \\ 0, & \text{fuera del intervalo} \end{cases} \quad (4.16)
 \end{aligned}$$

y

$$\begin{aligned}
 B_i^3(x) &= \frac{x - x_i}{x_{i+2} - x_i} B_i^2(x) + \frac{x_{i+3} - x}{x_{i+3} - x_{i+1}} B_{i+1}^2(x) \\
 &= \begin{cases} \frac{(x - x_i)^2}{(x_{i+1} - x_i)(x_{i+2} - x_i)}, & x \in [x_i, x_{i+1}) \\ \frac{1}{x_{i+2} - x_{i+1}} \left[\frac{(x - x_i)(x_{i+2} - x)}{x_{i+2} - x_i} + \frac{(x - x_{i+1})(x_{i+3} - x)}{x_{i+3} - x_{i+1}} \right], & x \in [x_{i+1}, x_{i+2}) \\ \frac{(x_{i+3} - x)^2}{(x_{i+3} - x_{i+1})(x_{i+3} - x_{i+2})}, & x \in [x_{i+2}, x_{i+3}) \\ 0, & \text{fuera del intervalo} \end{cases} \quad (4.17)
 \end{aligned}$$

respectivamente. Las gráficas que corresponden a estas funciones son las que se presentaron en la Fig. 4.2 de la sección 4.2.1. A diferencia de las ecuaciones que aparecen en la Fig. 4.2, las nuevas expresiones (4.16) y (4.17) que ahora describen a las B-spline nos permiten construir funciones base con intervalos de soporte diferentes a la unidad.

Para entender el funcionamiento de una red neuronal B-spline multi-dimensional comenzaremos la explicación del caso que corresponde a una estructura mono-dimensional, la cual está esquematizada en la Fig. 4.4. Para construir una red neuronal B-spline mono-dimensional con J neuronas en la capa oculta y funciones de activación de orden m que nos permita ajustar la relación $y = f(x)$ para un intervalo de valores de la variable de entrada $x = [a, b]$, necesitamos definir primero una familia de nodos $\{p_i\}_{i=1}^{J+m}$ que nos permita:

- formar J subintervalos traslapados de la variable de entrada
- ubicar los nodos inicial y final por debajo y por arriba de las cotas a y b , es decir, que $p_1 < a$ y $p_{J+m} > b$
- colocar un par de nodos internos en la posición de las cotas a y b (Fig. 4.5).

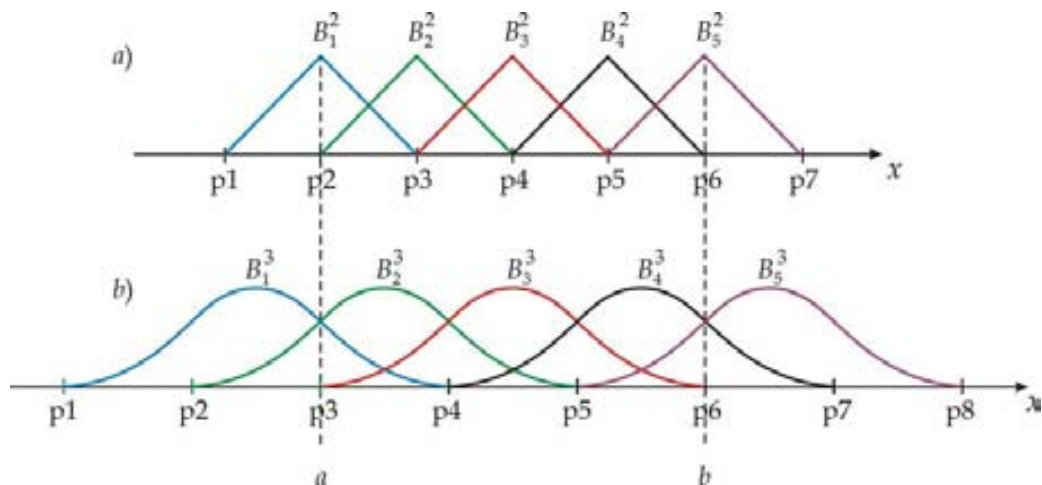


Fig. 4.5. Funciones B-spline de a) segundo y b) tercer orden cubriendo el intervalo $[a, b]$. Los $J+m$ nodos forman J subintervalos traslapados tanto en a) como en b). En la figura se han trazado $J=5$ funciones B-spline de órdenes $m=2$ y $m=3$, para los cuales se necesitan 7 y 8 nodos, respectivamente. Cada función B_j^m crea una neurona en la capa oculta.

En cada uno de los j -ésimos subintervalos de valores de la variable de entrada formado por la familia de nodos $\{p_i\}$ se construye una función B-spline de orden m que hereda todas las propiedades descritas en la sección 4.2.1. La construcción de una función B-spline cuyo intervalo de soporte queda

definido por un conjunto de $m+1$ nodos consecutivos equivale a crear una neurona oculta en la red neuronal. La característica principal de estas neuronas es que se activan solamente si el valor de la variable de entrada x yace dentro de su intervalo de soporte, si el valor de x está fuera de dicho intervalo entonces la salida de la neurona es igual a cero.

Una vez que se ha descrito la función de transferencia de las neuronas en la capa oculta y la condición que las activa, procederemos a ensamblar la red usando una capa de neuronas de entrada, otra de neuronas ocultas, una de neuronas de salida y conexiones entre ellas para construir una estructura jerárquica. La respuesta de la red a un estímulo de entrada se forma con la suma de las contribuciones individuales de cada neurona B-spline que responde a la excitación. Esta descripción del funcionamiento de una red neuronal B-spline mono-dimensional nos permite expresar su respuesta mediante

$$\hat{y} = \sum_j w_j B_j^m(x) + a_0 \quad (4.18)$$

donde cada j -ésima función $B_j^m(x)$ de orden m queda definida de acuerdo al algoritmo de de Boor. La salida \hat{y} de la red será igual la suma del valor de las funciones B-spline univariadas ponderadas por su correspondiente peso w_j más un desplazamiento a_0 que puede estar presente en funciones con un *off-set* diferente de cero. La ventaja de las funciones B-spline en la implementación de redes neuronales es el control local de la curva de ajuste, ya que ésta cambia solamente en el intervalo de soporte de la función si se modifica el peso de la neurona correspondiente (Fig. 4.6).

En esta red neuronal B-spline se asume que las funciones base son fijas y lo que se adapta durante el proceso de aprendizaje son los pesos de las conexiones entre las neuronas ocultas y la neurona de salida. La colocación de los nodos en este tipo de red donde la aproximación a una función $y = f(x)$ se consigue mediante la superposición de funciones locales no es un problema trivial. El número de nodos y por tanto el número de neuronas ocultas en la red puede ser estimado a priori si se tiene algún conocimiento de la función a aproximar, aunque si no se cuenta con esta información entonces la precisión del ajuste y la capacidad de generalización se rige por un proceso de prueba y

error en la elección del número de nodos y neuronas ocultas. En nuestra implementación no se optimizó la colocación de los nodos durante el ajuste de la aproximación, las funciones B-spline que se usaron en la red son fijas lo que tiene por consecuencia la restricción en el desempeño de la red [163]. Debe notarse el hecho de que debido a las características equidistantes de los nodos y la propiedad de partición de la unidad de las funciones B-spline, una BSNN cubre un intervalo de valores de la variable de entrada mayor al definido inicialmente por las cotas $[a, b]$.

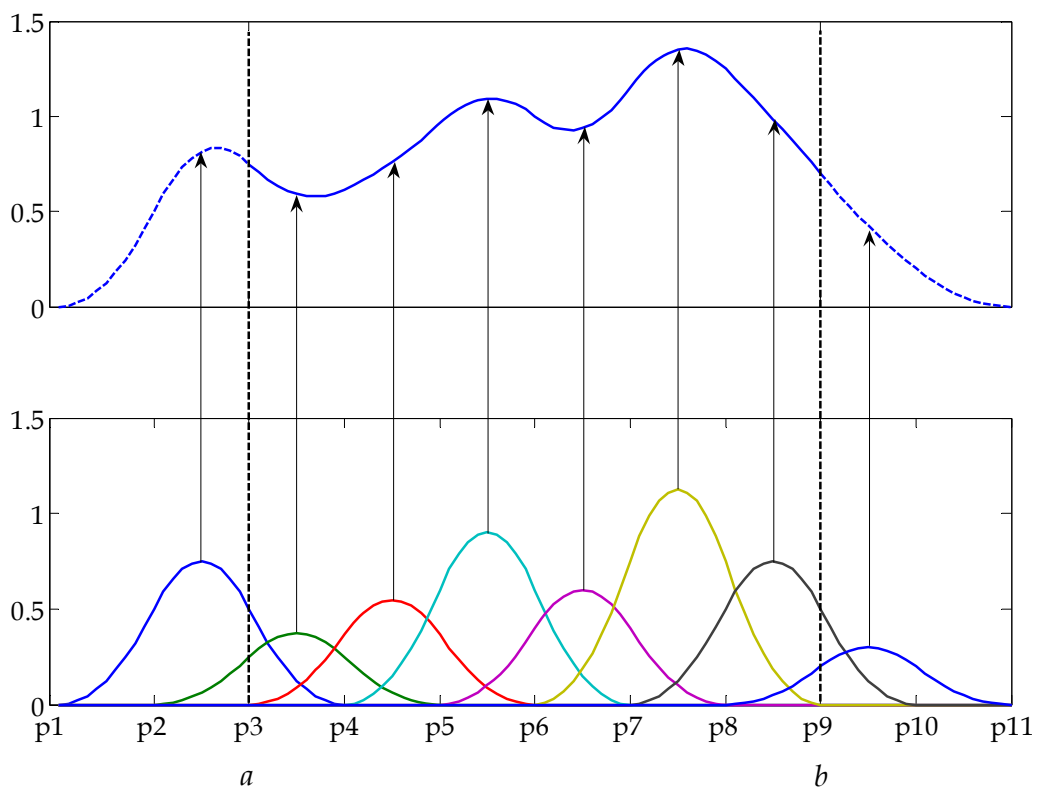


Fig. 4.6. Esquema del tipo de aproximación realizada por una BSNN mono-dimensional. En la gráfica inferior se han trazado 8 funciones B-spline de orden 3 que forman una partición de la unidad dentro del intervalo $[a,b]$ de la variable x de entrada. Cada una de estas funciones ha sido afectada por un peso que modifica la altura de cada B-spline. La gráfica superior es la aproximación que se obtiene al sumar las funciones trazadas en la gráfica inferior.

4.4 Red B-spline multi-dimensional

En una red B-spline multi-dimensional, las funciones de activación en la capa oculta se construyen con el producto tensorial de las funciones B-spline

mono-dimensionales anteriormente descritas. Las funciones B-spline por dimensión en una BSNN multi-dimensional se crean igual que en la red mono-dimensional y para obtener la característica de multidimensionalidad se realiza la multiplicación tensorial de las funciones mono-dimensionales.

Para ejemplificar lo anterior consideremos una red B-spline bi-dimensional cuya estructura es la representada en la Fig. 4.7. Las 16 neuronas que forman la red de la figura se construyen con funciones obtenidas a partir del producto tensorial de los conjuntos de 4 funciones B-spline mono-dimensionales encargados de cubrir los intervalos de valores de las variables de entrada x_1 y x_2 (Fig. 4.8). La extensión para el caso en que los tensores de entrada tienen una dimensión mayor a 2 es directa aunque la representación esquemática no es didáctica.

Extendiendo la explicación al caso multi-dimensional se puede establecer en términos generales que la respuesta de la j -ésima neurona B-spline de dimensión N_k se construye multiplicando N_k funciones base mono-dimensionales $B_{i_k}^m(x)$ de orden m .

$$\mathbf{B}_j^m(\mathbf{x}) = B_{i_1}^m(x_1) \cdots B_{i_k}^m(x_k) \cdots B_{i_{N_k}}^m(x_{N_k}) \quad (4.19)$$

El subíndice $i_k = 1..I_k$ de la expresión (4.19) sirve para señalar a la i -ésima B-spline mono-dimensional en la dimensión k . El número total de neuronas en la capa oculta que forman una red de este tipo es igual al resultado de multiplicar el número de funciones B-spline por cada dimensión, es decir, haciendo $\prod_{k=1}^{N_k} I_k$. En este tipo de red no se restringe que el número de funciones base por dimensión sea el mismo, cada una de las I_k puede ser diferente siempre que se conserve la condición de que las funciones formen una partición de la unidad dentro del intervalo de valores de la k -ésima variable de entrada que forma el vector \mathbf{x} . Computacionalmente hablando, cuando se construye una BSNN multi-dimensional no se crean las $\prod_{k=1}^{N_k} I_k$ neuronas ocultas, sino que sólo se construyen funciones B-spline unidimensionales y una red de

puntos de multiplicación a la salida de las funciones base para crear la característica de multi-dimensionalidad.

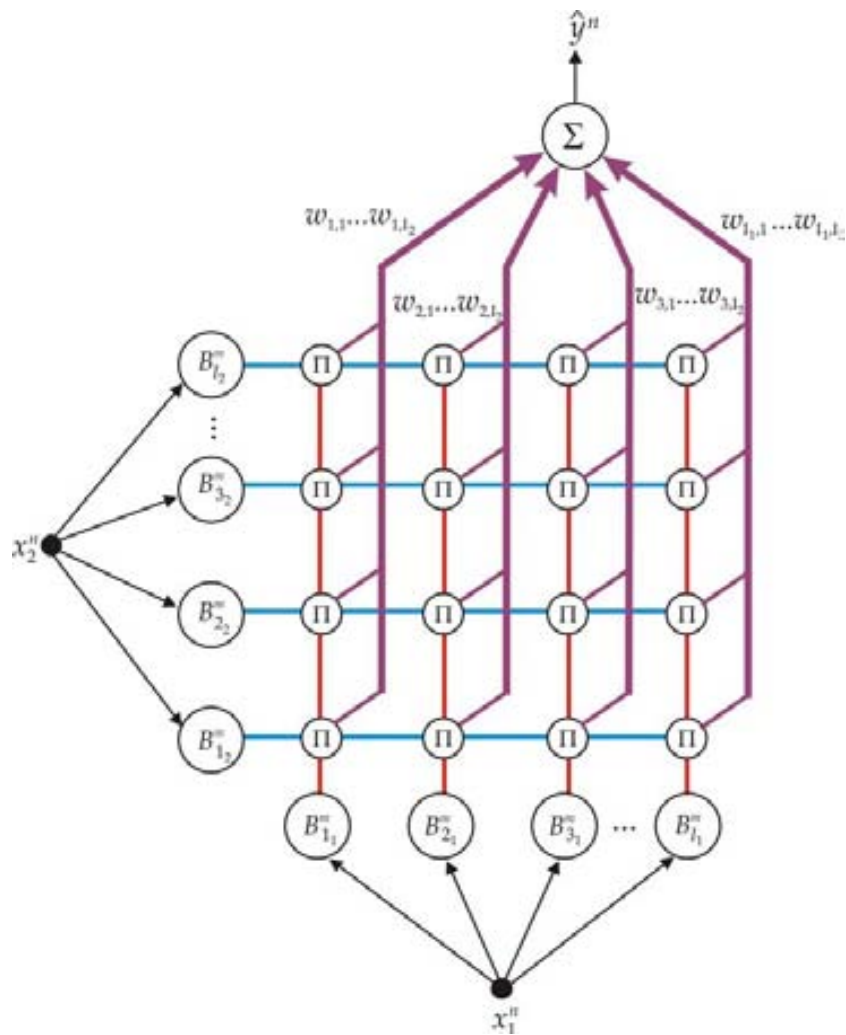


Fig. 4.7. Estructura de una BSNN bi-dimensional basada en el producto tensorial de funciones B-spline unidimensionales.

La estructura de una BSNN multi-dimensional con dimensión $N_k > 2$ es visualmente compleja. Desde un punto de vista conceptual, el espacio N -dimensional cubierto por una red B-spline puede visualizarse como un arreglo de N modos, donde cada modo es una dimensión y el número de elementos en cada modo es igual al número de subintervalos traslapados que dan origen a las J B-spline mono-dimensionales (Fig. 4.9).

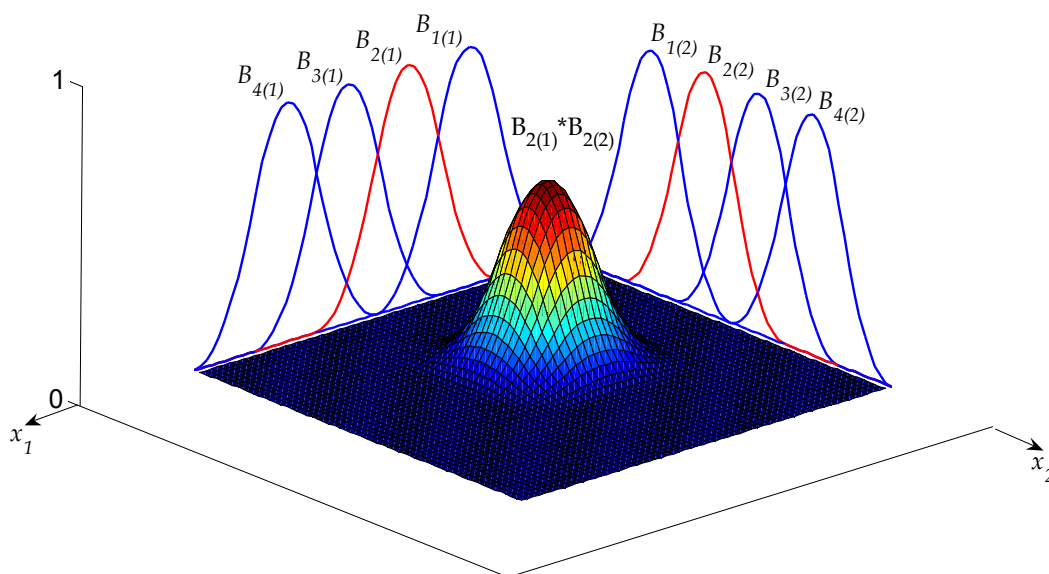


Fig. 4.8. El producto tensorial de las funciones mono-dimensionales que cubren los intervalos de las variables x_1 y x_2 genera funciones B-spline bidimensionales. En la figura se muestra solamente la superficie obtenida al multiplicar las funciones base resaltadas en color rojo.

La salida de la BSNN multi-dimensional será la combinación lineal de las funciones base multi-dimensionales definidas por la expresión (4.19). La aproximación obtenida con esta red es igual a

$$\hat{y} = \sum_j w_j \mathbf{B}_j^m(\mathbf{x}) + a_0 \quad (4.20)$$

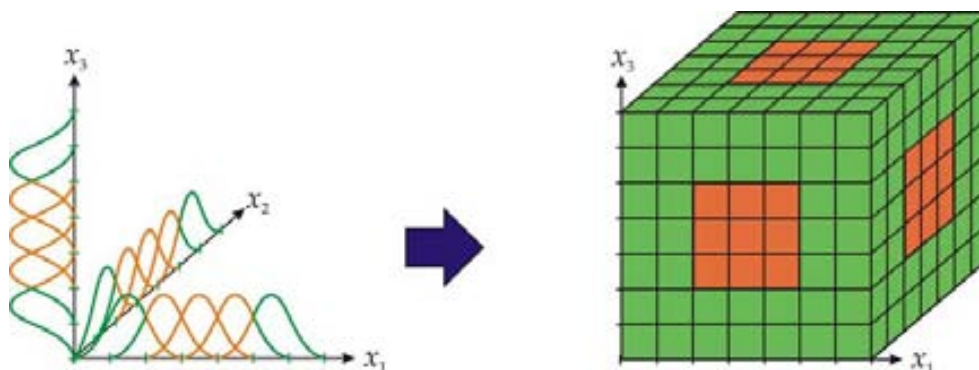


Fig. 4.9. A la derecha se representa el espacio tri-dimensional cubierto por funciones B-spline multi-dimensionales creadas a partir del producto tensorial de las funciones mono-dimensionales representadas a la izquierda.

De la anterior ecuación podemos ver que durante el entrenamiento de la red las únicas variables a modificar son el par $\Theta = \{w_j, a_0\}$ ya que las funciones $\mathbf{B}_j^m(\mathbf{x})$ son constantes una vez que se han fijado los nodos y el grado de la B-spline.

Para realizar el entrenamiento de la red mediante la técnica del gradiente descendiente necesitamos conocer las derivadas primera y segunda de \hat{y} respecto a las variables w_j y a_0 . En el caso de las derivadas primeras éstas son iguales a

$$\frac{\partial \hat{y}}{\partial w} = \mathbf{B}_j^m(\mathbf{x}) \quad (4.21)$$

$$\frac{\partial \hat{y}}{\partial a_0} = 1 \quad (4.22)$$

Las derivadas segundas de \hat{y} respecto a las mismas variables son iguales a cero. Usando las expresiones (4.21) y (4.22) se completa el conjunto de ecuaciones para construir y entrenar mediante la técnica del gradiente conjugado a una red neuronal B-spline multi-dimensional.

Las ecuaciones usadas para el entrenamiento de la red B-spline multi-dimensional son las mismas que para el modelo mono-dimensional, por lo que pueden usarse indistintamente; la diferencia entre ellas es la dimensionalidad de las funciones base. El algoritmo que condensa los pasos para construir y entrenar una red neuronal de este tipo se encuentra en el Apéndice 3.

5

Redes Neuronales Spline Catmull–Rom

Las *Redes Neuronales Spline Catmull–Rom* (CR-SNN) son redes neuronales basadas en funciones de activación adaptativas [164, 165]. En comparación con las Redes Neuronales B–Spine del Capítulo 4 donde las funciones de activación se expresaron mediante ecuaciones racionales recursivas, en las CR-SNN las neuronas de la capa oculta están formadas por coeficientes polinomiales que representan los puntos de control de una función spline cúbica Catmull–Rom.

5.1 Splines cardinales y Catmull–Rom

La *spline Catmull–Rom* es una spline cúbica de interpolación desarrollada para gráficos computacionales que presentan continuidad en la primera derivada, control local, interpolación y no yace dentro de la envolvente convexa de los puntos de control sino que pasa a través de ellos [166].

Al igual que con las funciones B–spline de orden n anteriormente definidas, la spline Catmull–Rom también tiene una forma matricial para su evaluación. La matriz \mathbf{M} de coeficientes en la spline cúbica Catmull–Rom está definida por

$$\mathbf{M} = \frac{1}{2} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (5.1)$$

Las spline Catmull–Rom son un caso especial de las *splines cardinales*, la cual incorpora dentro de la matriz \mathbf{M} anteriormente descrita, un parámetro de tensión que determina la suavidad con que se dobla una curva para ajustar una función. La matriz de coeficientes de una spline cardinal cúbica incorpora el parámetro de tensión en ciertos elementos de su matriz,

$$\mathbf{M} = \begin{bmatrix} -a & 2-a & a-2 & a \\ 2a & a-3 & 3-2a & -a \\ -a & 0 & a & 0 \\ 0 & a & 0 & 0 \end{bmatrix} \quad (5.2)$$

Los efectos del parámetro de tensión a en la generación de una curva de interpolación construida con splines cardinales se muestran en la Fig. 5.1.

El tipo de spline cardinal que se usó en las otra implementación de red neuronal corresponde a la spline Catmull–Rom, que es el caso particular de la spline cardinal para un valor del parámetro de tensión de $a=0.5$.

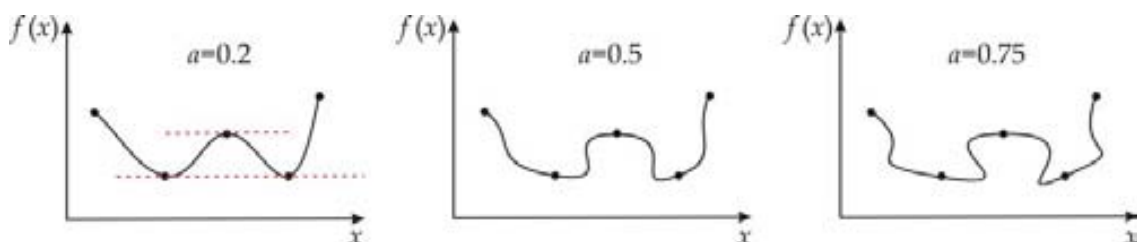


Fig. 5.1. Efectos del parámetro de tensión en la construcción de splines cardinales. En las tres figuras se han usado los mismos puntos de control para comparar los efectos de a .

5.2 Redes Neuronales Spline Catmull–Rom

Las *Redes Neuronales Spline Catmull–Rom (CR-SNN)* son también redes neuronales basadas en funciones de activación adaptativas [167, 168]. En

comparación con las BSNN, donde las funciones de activación se expresaron mediante ecuaciones racionales recursivas, en las CR-SNN las neuronas de la capa oculta están formadas por coeficientes polinomiales que representan los puntos de control de una función spline cúbica Catmull–Rom.

Durante la fase de aprendizaje de una CR-SNN, la forma de la función se modifica mediante la adaptación de los puntos de control además de los pesos en las conexiones, a diferencia de las BSNN donde solamente se ajustan los valores de los pesos para aproximar una función. Esta ventaja respecto a la BSNN se traduce en una mejora en la capacidad de generalización de la red y la buena convergencia de la misma debido al uso de funciones de activación con más grados de libertad.

Antes de comenzar la explicación del funcionamiento e implementación de una CR-SNN recordemos primero el principio de operación de una red neuronal artificial tipo perceptrón. La estimación \hat{y} en la neurona de salida de una ANN de este tipo se calcula aplicando una función de activación $f(\cdot)$ a la suma ponderada de las variables de entrada \mathbf{x} , es decir, haciendo $\hat{y} = f(u)$, donde $u = \sum_i w_i x_i = \mathbf{w} \cdot \mathbf{x}^T$. Si ahora sustituimos la función $f(u)$ por una función del tipo $\delta = f(u_1, \dots, u_M)$ donde el subíndice M es la dimensión de la función, entonces podemos crear neuronas con salidas definidas por funciones de activación multi-dimensionales del tipo

$$\delta = \Phi(u_1, \dots, u_M) \quad (5.3)$$

donde las entradas de activación $[u_1, \dots, u_M]$ forman un subconjunto M -dimensional creado por las combinaciones lineales de las N_k entradas $\{x_k\}_{k=1}^{N_k}$ ponderadas por un conjunto de pesos $w_{m,k}$ que forman la m -ésima variable multi-dimensional $u_m = \sum_k w_{m,k} x_k$. Usando estos términos lineales para u_m reescribimos la ecuación (5.3) como

$$\delta = \Phi\left(\sum_k w_{1,k} x_k, \dots, \sum_k w_{M,k} x_k\right) \quad (5.4)$$

donde la función $\Phi(\cdot)$ que define la activación de una neurona multi-dimensional se implementó usando la spline cúbica de interpolación Catmull-Rom, la cual tiene la capacidad de modificarse localmente para el ajuste de la función [18] y garantiza la continuidad tanto en la primera derivada como en la segunda, necesarias para implementar la técnica de entrenamiento basada en el gradiente descendiente.

La estructura final de la CR-SNN con neuronas en la capa oculta formadas por las funciones de transferencia anteriormente descritas es la representada en la Fig. 5.2. La salida de la red se obtiene haciendo

$$\hat{y} = \sum_j \Phi_j(\mathbf{u}^{(j)}) \quad (5.5)$$

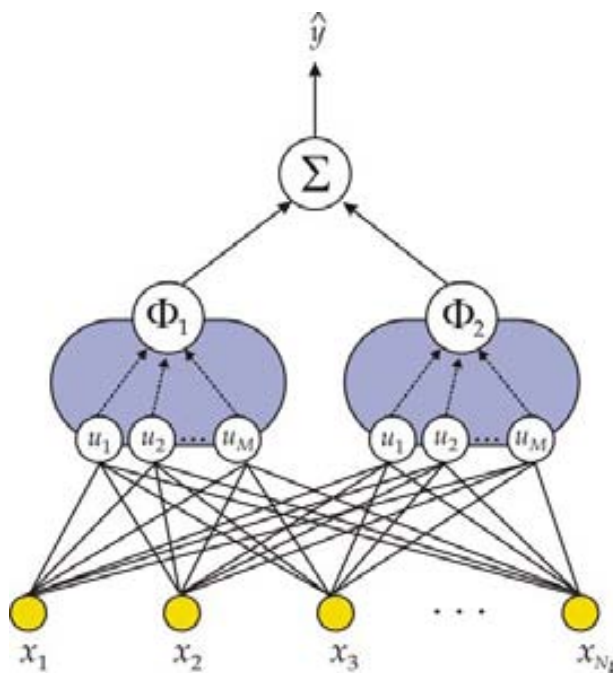


Fig. 5.2. Estructura de la red neuronal artificial con funciones de activación multi-dimensionales basadas en la spline cúbica Catmull-Rom. Las entradas a las neuronas de la capa oculta están formadas por las combinaciones lineales de pesos con la variable x de entrada.

donde cada vector $\mathbf{u}^{(j)}$ es un subconjunto de M variables formado por las combinaciones lineales anteriormente descritas de los pesos y las variables de entrada a la red. La salida \hat{y} queda representada de manera general como la concatenación de funciones locales $\Phi_j(u_1^{(j)}, \dots, u_M^{(j)})$, donde el elemento u de la

m -ésima dimensión está centrado entre 4 puntos de control y cada j -ésima función está controlada por un total de 4^M puntos de control.

Para ejemplificar la anterior descripción consideremos la función mono-dimensional $\phi(u)$ donde u es una variable ubicada en un intervalo $u_{(p1)} \leq u \leq u_{(p1+1)}$. Tomando la spline cúbica Catmull-Rom descrita en la sección 5.1 como función base $\phi(\cdot)$, la salida de $\phi(u)$ será el promedio ponderado de 4 puntos de control $[Q_{(p1-1)} \quad Q_{(p1)} \quad Q_{(p1+1)} \quad Q_{(p1+2)}]^T$ que están igualmente espaciados en el eje u . A partir de esta descripción y considerando que contamos con una función base $\phi(\cdot)$, puntos de control $\{Q_{(p1+n)}\}_{n=-1}^{n=2}$ y base monomial u entonces podremos calcular la salida de $\phi(u)$ usando la expresión (4.10) de la sección 4.2.1, por lo que

$$\phi(u) = \mathbf{u} \cdot \mathbf{M} \cdot \mathbf{q} \quad (5.6)$$

donde $\mathbf{u} = [u^3 \quad u^2 \quad u \quad 1]$. Sin perder generalidad de la anterior descripción podemos restringir el intervalo de valores de u a $[0,1]$ para facilitar el posterior entrenamiento de la red. Extendiendo la anterior explicación de la función mono-dimensional $\phi(u)$ al caso bi-dimensional tenemos,

$$\Phi(\mathbf{u}) = \Phi(u_1, u_2) = \mathbf{u}_2 \cdot \mathbf{M} \cdot (\mathbf{u}_1 \cdot \mathbf{M} \cdot \mathbf{Q}_2)^T \quad (5.7)$$

Dado que las funciones $\Phi(\mathbf{u})$ de una CR-SNN están controladas por 4^M puntos de control, la variable \mathbf{Q}_2 de la ecuación (5.7) se define ahora como una estructura bi-dimensional que contiene 16 puntos de control que yacen sobre una rejilla regular de dos dimensiones,

$$\mathbf{Q}_2 = \begin{bmatrix} Q_{(p1-1,p2-1)} & Q_{(p1-1,p2)} & Q_{(p1-1,p2+1)} & Q_{(p1-1,p2+2)} \\ Q_{(p1,p2-1)} & Q_{(p1,p2)} & Q_{(p1,p2+1)} & Q_{(p1,p2+2)} \\ Q_{(p1+1,p2-1)} & Q_{(p1+1,p2)} & Q_{(p1+1,p2+1)} & Q_{(p1+1,p2+2)} \\ Q_{(p1+2,p2-1)} & Q_{(p1+2,p2)} & Q_{(p1+2,p2+1)} & Q_{(p1+2,p2+2)} \end{bmatrix} \quad (5.8)$$

En la Fig. 5.3 se muestra la interpolación de los puntos de control para los casos de dimensión 1 y 2 anteriormente mencionados. Para el caso en que $M=3$, los puntos de control \mathbf{Q}_3 aumentan en una unidad el orden de su estructura y forman un arreglo tri-dimensional. La función $\Phi(\mathbf{u}) = \Phi(u_1, u_2, u_3)$ para este caso se evalúa mediante

$$\Phi(\mathbf{u}) = \Phi(u_1, u_2, u_3) = \mathbf{u}_3 \cdot \mathbf{M} \cdot \begin{bmatrix} \mathbf{u}_2 \cdot \mathbf{M} \cdot (\mathbf{u}_1 \cdot \mathbf{M} \cdot \mathbf{Q}_{3(\text{capa } 1)})^T \\ \mathbf{u}_2 \cdot \mathbf{M} \cdot (\mathbf{u}_1 \cdot \mathbf{M} \cdot \mathbf{Q}_{3(\text{capa } 2)})^T \\ \mathbf{u}_2 \cdot \mathbf{M} \cdot (\mathbf{u}_1 \cdot \mathbf{M} \cdot \mathbf{Q}_{3(\text{capa } 3)})^T \\ \mathbf{u}_2 \cdot \mathbf{M} \cdot (\mathbf{u}_1 \cdot \mathbf{M} \cdot \mathbf{Q}_{3(\text{capa } 4)})^T \end{bmatrix} \quad (5.9)$$

donde la variable \mathbf{Q}_3 ha sido descompuesta en arreglos bi-dimensionales para la evaluación de $\Phi(\mathbf{u})$. La extensión a funciones con un mayor número de variables sigue una metodología similar. Debido al tratamiento multi-modo de la información no es posible obtener una expresión general de $\Phi(\mathbf{u})$ para el caso multi-dimensional, aunque a partir de las ecuaciones (5.6), (5.7) y (5.9) se puede observar un patrón en la evaluación matricial de la información.

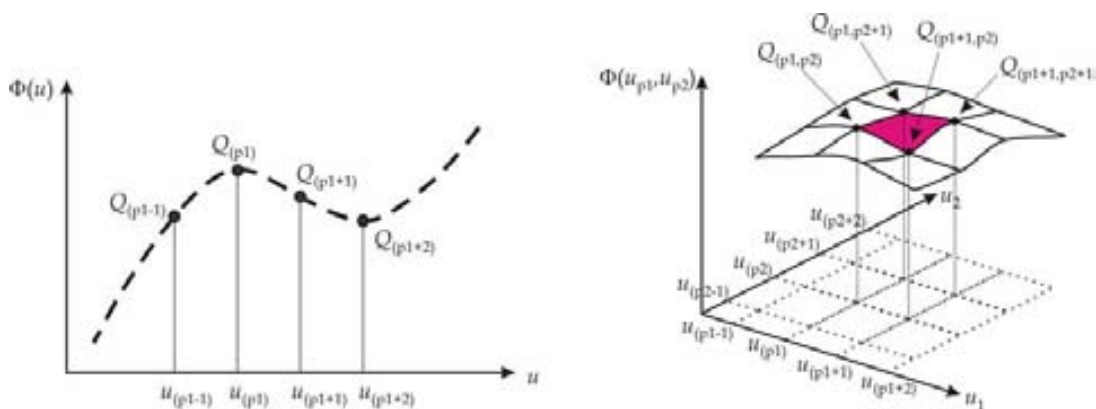


Fig. 5.3. Ejemplos de interpolación usando la spline cúbica Catmull-Rom. a) ajuste mono-dimensional, b) ajuste bi-dimensional.

La ecuación (5.5) define el comportamiento de la red, y las ecuaciones (5.6), (5.7) y (5.9) sirven como guía para evaluar las funciones spline

Catmull–Rom de 1 a 3 variables. A partir de estas ecuaciones podemos desarrollar el conjunto de expresiones que nos permitirán implementar la técnica de gradiente descendiente para el entrenamiento de una CR-SNN. Las variables que se modificarán para conseguir la minimización de la función de error en este tipo de red serán los pesos de las conexiones de entrada y los M puntos de control para cada j -ésima neurona oculta, $\Theta = \{\mathbf{w}_m^{(j)}, \mathbf{Q}_M^{(j)}\}$.

Partiendo de la función de error E_r anteriormente descrita, lo que resta por definir son las derivadas de \hat{y} para cada una de las variables contenidas en Θ . La derivada del tensor de pesos $\{\mathbf{w}_m^{(j)}\}_{m=1}^M$ que conecta a las entradas con la j -ésima neurona Φ_j es igual a

$$\frac{\partial \hat{y}}{\partial \mathbf{w}_1^{(j)}} = \dot{\mathbf{u}}_1^{(j)} \cdot \mathbf{M} \cdot \mathbf{q}_1^{(j)T} \cdot \mathbf{x} \quad (5.10)$$

Para el caso de 2 dimensiones la derivada respecto a cada tensor $\mathbf{w}_m^{(j)}$ que sirve para formar cada una de las dimensiones de entrada a Φ_j se define como

$$\begin{aligned} \frac{\partial \hat{y}}{\partial \mathbf{w}_1^{(j)}} &= \mathbf{u}_2^{(j)} \cdot \mathbf{M} \cdot [\dot{\mathbf{u}}_1^{(j)} \cdot \mathbf{M} \cdot \mathbf{Q}_2^{(j)}]^T \cdot \mathbf{x} \\ \frac{\partial \hat{y}}{\partial \mathbf{w}_2^{(j)}} &= \dot{\mathbf{u}}_2^{(j)} \cdot \mathbf{M} \cdot [\mathbf{u}_1^{(j)} \cdot \mathbf{M} \cdot \mathbf{Q}_2^{(j)}]^T \cdot \mathbf{x} \end{aligned} \quad (5.11)$$

donde $\dot{\mathbf{u}}_m = [3u_m^2 \quad 2u_m \quad 1 \quad 0]$.

Los puntos de control \mathbf{Q}_m también se actualizan con cada iteración usando al gradiente $\frac{\partial \hat{y}^n}{\partial \mathbf{Q}_m}$, que para los casos de 1 y 2 dimensiones que se han estado describiendo las respectivas derivadas son iguales a

$$\frac{\partial \hat{y}^n}{\partial \mathbf{Q}_1} = [\mathbf{u}_1 \cdot \mathbf{M}]^T \quad (5.12)$$

$$\frac{\partial \hat{y}^n}{\partial \mathbf{Q}_2} = \mathbf{u}_2 \cdot \mathbf{M} \cdot [\mathbf{u}_1 \cdot \mathbf{M}]^T \quad (5.13)$$

El número de puntos de control a ajustar depende de la dimensionalidad de la estructura y es igual a 4^M . Usando las expresiones anteriores es posible implementar una red neuronal spline Catmull–Rom entrenada con el algoritmo de gradiente descendiente. Para conseguir el entrenamiento es necesario determinar la derivada segunda de \hat{y} respecto a las variables de ajuste de la red neuronal. Para el caso de funciones mono-dimensionales, la segunda derivada respecto a los pesos de entrada es

$$\frac{\partial^2 \hat{y}}{\partial \{\mathbf{w}_1^{(j)}\}^2} = \ddot{\mathbf{u}}_1^{(j)} \cdot \mathbf{M} \cdot \mathbf{q}_1^{(j)T} \cdot (\text{diag}(\mathbf{x}) \cdot \mathbf{x}^T)^T \quad (5.14)$$

En el caso de funciones Φ_j bi-dimensiones la segunda derivada de \hat{y} respecto a los pesos para cada una de las dimensiones es igual a

$$\begin{aligned} \frac{\partial^2 \hat{y}}{\partial \{\mathbf{w}_1^{(j)}\}^2} &= \mathbf{u}_2^{(j)} \cdot \mathbf{M} \cdot [\ddot{\mathbf{u}}_1^{(j)} \cdot \mathbf{M} \cdot \mathbf{Q}_2^{(j)}]^T \cdot (\text{diag}(\mathbf{x}) \cdot \mathbf{x}^T)^T \\ \frac{\partial^2 \hat{y}}{\partial \{\mathbf{w}_2^{(j)}\}^2} &= \ddot{\mathbf{u}}_2^{(j)} \cdot \mathbf{M} \cdot [\mathbf{u}_1^{(j)} \cdot \mathbf{M} \cdot \mathbf{Q}_2^{(j)}]^T \cdot (\text{diag}(\mathbf{x}) \cdot \mathbf{x}^T)^T \end{aligned} \quad (5.15)$$

donde $\ddot{\mathbf{u}}_m$ se define igual en ambos casos y es igual a $\ddot{\mathbf{u}}_m = [6u \ 2 \ 0 \ 0]$. Tanto en el caso mono-dimensional como en el bi-dimensional, la derivada segunda de \hat{y} respecto a los puntos de control es igual a cero. La metodología para la construcción y entrenamiento de una Red Neuronal Spline Catmull–Rom se encuentra condensada en el Algoritmo 4 al final de la memoria de tesis.

6

Splines y B-splines Adaptativas de Regresión Multivariable (MARS y B-MARS)

Las *Splines y B-splines Adaptativas de Regresión Multivariable (MARS y B-MARS)* forman métodos de regresión basados en una serie de funciones base locales que sirven para modelar relaciones no-lineales entre múltiples variables de entrada y una (o varias) salida(s) definidas por la expresión $\mathbf{y} = f(\mathbf{x})$ [169], donde $f(\cdot)$ refleja la relación desconocida entre \mathbf{y} y \mathbf{x} . B-MARS se considera una extensión de MARS, por lo que primero revisaremos los fundamentos en los que se cimentan estos métodos de regresión.

6.1 Particionamiento recursivo

MARS tiene sus orígenes en la metodología de *particionamiento recursivo* (RP) [170, 171]. El RP es una técnica estadística de análisis multivariable que permite obtener una aproximación a una función desconocida $f(\mathbf{x})$ a partir de una expansión en funciones base construidas con funciones escalón unitario de carácter univariable. Usando el RP se puede obtener un modelo de regresión de la forma $\hat{y} = \sum_{j=1}^M a_j P_j(\mathbf{x})$ donde los términos $\{a_j\}_{j=1}^M$ son coeficientes de expansión que se calculan mediante un ajuste por mínimos cuadrados y que sirven para mejorar el modelo \hat{y} , y las funciones $\{P_j(\mathbf{x})\}_{j=1}^M$ son funciones

indicadoras que toman un valor unitario cuando su argumento es verdadero y cero en caso contrario. El argumento de una función $P_j(\mathbf{x})$ es verdadero si \mathbf{x} pertenece a alguna de las sub-regiones obtenidas de la partición del dominio de entrada $\mathbf{x} \in \mathfrak{R}^n$. Las funciones indicadoras en $P_j(\mathbf{x})$ están formadas a su vez por el producto de funciones base escalón unitario univariable $H[\cdot]$ de argumento positivo que describen a las sub-regiones obtenidas de la anterior partición de \mathbf{x} (una $H[\eta]$ vale 1 sólo si su argumento $\eta \geq 0$). Para ejemplificar la anterior descripción consideremos un conjunto de 4 funciones $\{P_j\}_{j=1}^4$ obtenidas a partir de la partición en 6 subregiones de un conjunto de 3 variables $\{\mathbf{x}_k\}_{k=1}^3$ acotadas en un intervalo $[a, b]$ (Fig. 6.1). Las subregiones se forman a partir de los puntos $\mathbf{t} \in \mathfrak{R}^3$ que optimizan la partición de las variables para el ajuste de una función. Cada punto t_k divide a la correspondiente variable \mathbf{x}_k en subregiones izquierda y derecha que dan origen a dos funciones escalón $H[\cdot]$ por variable \mathbf{x}_k . Estas funciones escalón son las funciones base que finalmente se usan para crear cada una de las $P_j(\mathbf{x})$, las cuales adoptan la forma

$$P_j(\mathbf{x}) = \prod_i H\left[\text{sgn}\left(x_{k(i,j)} - t_{i,j}\right)\right] \quad (6.1)$$

donde $\text{sgn}(\cdot)$ asigna el signo +/- al resultado del argumento $(x_{k(i,j)} - t_{i,j})$ para indicar la dirección izquierda o derecha de la función escalón, si el argumento es positivo entonces $H[\cdot]$ es un escalón dirigido hacia la derecha, si el argumento es negativo entonces $H[\cdot]$ está dirigido hacia la izquierda. Los subíndices k , i y j en la expresión (6.1) señalizan a la k -ésima dimensión de la variable \mathbf{x} , a la i -ésima función escalón unitario $H_i[\cdot]$ y a la j -ésima función $B_j(\cdot)$, respectivamente. Los coeficientes de expansión a_j se calculan minimizando una función que evalúa el ajuste de la aproximación \hat{y} a los valores reales y .

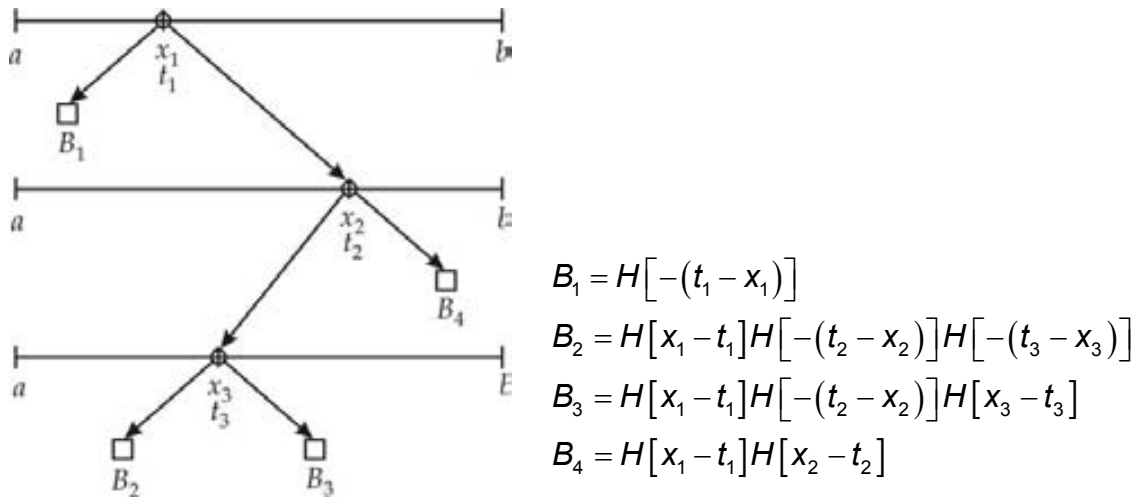


Fig. 6.1. Representación de la partición recursiva y sus funciones base asociadas.

La metodología para crear un modelo usando el RP es una tarea que se realiza en dos etapas y que empieza con la asignación de una sola subregión R_1 que abarca todo el intervalo $[a, b]$ de la variable de entrada. La primera etapa consiste en la división recursiva de subregiones para crear un número grande de S subregiones $\{R_j\}_{j=1}^S$ sin traslape por dimensión. La segunda etapa consiste en la eliminación de $S - M$ subregiones excedentes que se eligen de acuerdo a un criterio de evaluación que toma en cuenta tanto el ajuste del modelo como el número de subregiones en el mismo. El objetivo final de la metodología es obtener un conjunto de subregiones $\{R_j\}_{j=1}^M$ a partir de los datos \mathbf{x} , y un conjunto de coeficientes $\{a_j\}_{j=1}^M$ que permitan crear una buena aproximación \hat{y} de la función $f(\mathbf{x})$.

6.2 MARS

MARS es un procedimiento de regresión similar al RP por tener su origen en ella. El proceso de creación de un modelo MARS es adaptativo debido a que la elección de las funciones base es específico al problema que se trata. En MARS el espacio de variables de entrada se divide en espacios traslapados a los que se ajusta funciones base spline. Al igual que en el RP, el modelo de

aproximación obtenido con MARS está formado por la suma ponderada de términos $a_j P_j(\mathbf{x})$,

$$\hat{y} = \sum_{j=0}^J a_j P_j(\mathbf{x}) \quad (6.2)$$

De la anterior ecuación se puede apreciar que, al igual que en la RP, la aproximación \hat{y} está formada por una sumatoria de términos formados por funciones $P_j(\mathbf{x})$ multiplicadas por su correspondiente coeficiente de expansión a_j . El primer término de la aproximación (6.2) consiste de un término constante a_0 debido a que $P_0(\mathbf{x}) = 1$. En lugar de usar funciones escalón unitario para construir las funciones $P_j(\cdot)$, en MARS cada una de estas funciones está formada por una sola spline o el producto de dos más splines. La interacción entre funciones base permite que el modelo tenga términos de orden superior.

El algoritmo para la construcción de un modelo de regresión del tipo MARS fue desarrollado por Jerome Friedman en 1991 [ref]. En MARS se construye un modelo a partir de la adición paso a paso de funciones base de orden 1 o superior. Las funciones base usadas para crear las funciones $P_j(\cdot)$ son funciones spline truncadas del tipo

$$\begin{aligned} b_+^{(q)}(x-t) &= [+(x-t)]_+^{(q)} = \begin{cases} (x-t)^{(q)} & \text{si } x > t \\ 0 & \text{caso contrario} \end{cases} \\ b_-^{(q)}(x-t) &= [-(x-t)]_+^{(q)} = \begin{cases} (t-x)^{(q)} & \text{si } x < t \\ 0 & \text{caso contrario} \end{cases} \end{aligned} \quad (6.3)$$

donde q es el grado de la spline, la variable t es el nodo que divide la región de x en porciones izquierda y derecha, $[\cdot]_+$ es un operador que indica que la función es positiva, y el signo +/- en el subíndice señala las porciones derecha e izquierda. Estas funciones truncadas $b_{\pm}^{(q)}(x-t)$ son splines de orden 1, las cuales están representadas en la Fig. 6.2. Para obtener funciones de mayor orden se realiza el producto tensorial entre estas funciones base hasta un máximo orden q predeterminado por el usuario. Usando las funciones

truncadas de la expresión (6.3) se construyen las funciones base $P_j(\cdot)$ del modelo MARS mediante

$$P_j(\mathbf{x}) = \prod_i \left[\text{sgn}(\mathbf{x}_k - t_k^{(n)}) \right]_{i,j}^+ \quad (6.4)$$

La diferencia entre estas funciones $P_j(\mathbf{x})$ construidas con MARS y las funciones en la ecuación (6.1) construidas con el RP es el uso de una función truncada en lugar de una función escalón unitario.

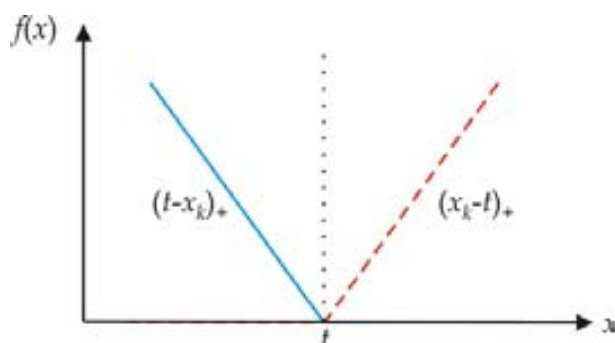


Fig. 6.2. Representación de una función base spline usada en los modelos MARS. La spline para $x > t$ está representada por la línea roja punteada, la spline para $x < t$ está representada por la línea azul continua.

La metodología para crear un modelo MARS es similar a la metodología para crear un modelo basado en el RP. La tarea se realiza en dos etapas, en la primera de ellas se eligen los mejores pares de funciones base $b_{\pm}^{(q)}(x-t)$ para crear los términos $P_j(\mathbf{x})$ que mejoran el ajuste del modelo \hat{y} . Para incluir un par de funciones base se evalúan las N_k variables que existen en \mathbf{x} así como todos los posibles nodos en cada variable, es decir, se evalúa cada uno de los n -ésimos elementos presentes en el k -ésimo tensor $\{x_k^{(n)}\}_{n=1}^N$. La búsqueda de la mejor variable y nodo se realiza de una manera iterativa. De manera adicional, también se evalúa al final de cada iteración si la posible interacción entre variables puede mejorar el modelo. Las interacciones se expresan como el producto entre dos o más funciones $b_{\pm}^{(q)}(x-t)$ y sólo son posibles entre funciones base con diferente variable \mathbf{x}_k . El orden del modelo MARS indica el máximo número de funciones base que pueden interactuar

para dar origen a un término de orden superior. Durante la construcción del modelo de la ecuación (6.2) se van agregando funciones $P_j(\cdot)$ hasta que se ha alcanzado el número máximo S de funciones que se ha especificado, el cual debe ser considerablemente grande.

La segunda etapa de la metodología consiste en la eliminación de $S - J$ términos del modelo sobre-ajustado anteriormente obtenido. El proceso se realiza probando la eliminación uno a uno de los términos presentes en el modelo y descartando al final de la iteración a aquél que contribuye poco al ajuste de la aproximación \hat{y} . El criterio para evaluar la contribución de cada término (ya sea para incluirlo durante la primera etapa del modelo o eliminarlo en la segunda) está basado en la *Validación Cruzada Generalizada* (GCV) definida como

$$GCV(J) = \frac{1}{N} \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\left(1 - \frac{C(J)}{N}\right)^2} \quad (6.5)$$

La GCV es una suma ponderada de residuales al cuadrado (el numerador) a la que se incluye un término de penalización debida a la complejidad del modelo (el denominador). De la ecuación (6.5), $C(J)$ es el término de costo de la complejidad del modelo y se define como $C(J) = J + d \cdot J$, donde J es el número de funciones $P_j(\mathbf{x})$ y d funciona tanto como un costo adicional por cada término que se optimiza como un parámetro de suavizado. Entre mayor sea el valor de d más suave será el ajuste. Cada vez que se agrega un par de funciones base al modelo sobre-ajustado de la primera etapa de la metodología, o se quita una de las funciones $P_j(\mathbf{x})$ de la segunda etapa se debe recalculer el valor de los coeficientes de expansión a_j mediante un ajuste por mínimos cuadrados.

Al finalizar la eliminación de los términos que sobreajustan el modelo se da por concluida la metodología para la construcción del modelo MARS. La aproximación \hat{y} que se obtiene tiene la forma

$$\hat{y} = a_0 + \sum_{j=1}^J a_j \prod_{i=1}^q \left[\text{sgn}(\mathbf{x}_k - \mathbf{t}_k^{(n)})_{i,j} \right]_{+} \quad (6.6)$$

la cual estará formada por los términos que más contribuyen a la disminución de la diferencia $y - \hat{y}$.

La anterior descripción de pasos para la creación de un modelo de regresión del tipo MARS se condensa en el algoritmo mostrado en el Apéndice 5.

6.3 B-MARS

La construcción de un modelo \hat{y} con funciones spline truncadas puede mejorarse si se usan otros tipos de funciones base tales como las funciones B-spline. Esta mejora se refleja en el momento de calcular los coeficientes a_j que minimizan la diferencia entre los valores obtenidos y los esperados, ya que el sistema de ecuaciones lineales formado por las funciones spline truncadas que se usa para calcular los coeficientes a_j puede formar un sistema mal condicionado, lo que no ocurre con las funciones B-spline por tener propiedades numéricas superiores que evitan esta dificultad [172]. La metodología MARS modificada para usar funciones base B-spline en lugar de funciones base truncadas recibe el nombre de *B-MARS*. De manera semejante a MARS, con B-MARS se construye un modelo de regresión en dos etapas; en la primera de ellas se obtiene un modelo con una gran cantidad de términos formados por las funciones base y sus interacciones mientras que en la segunda se quitan los términos sub-óptimos obtenidos en el paso anterior y que contribuyen poco en la reducción del error del modelo. Al final de la metodología B-MARS también se obtiene un modelo de la forma $\hat{y} = \sum_{j=0} a_j P_j(\mathbf{x})$ donde los términos $P_j(\mathbf{x})$ son construidos en esta ocasión a partir del producto tensorial de funciones B-spline univariable del tipo,

$$P_j(\mathbf{x}) = \prod_{i=1}^q B_{\{S(n,i)\}}(\mathbf{x}_k) \quad (6.7)$$

La familia de funciones $B_{\{S(n,l)\}}(\mathbf{x})$ usadas en la ecuación (6.7) está formada por B-splines univariable que tienen el mismo orden pero diferentes intervalos de soporte. Dos funciones $B(x)$ con diferente intervalo de soporte cubren diferentes intervalos de valores de la variable x de entrada, la B-spline con el intervalo de soporte más grande cubrirá un intervalo de valores más amplio de la variable de entrada, en comparación con la B-spline con un intervalo de soporte menor.

Para explicar la manera en como se construyen los términos $P_j(\mathbf{x})$ introducimos el concepto de *escala*. En el resto del desarrollo de este apartado llamaremos escala a una familia de funciones B-spline que tienen el mismo intervalo de soporte sin importar su posición. Las funciones B-spline con el mismo intervalo de soporte las agruparemos en familias a las que llamaremos *escalas*, las cuales numeraremos para identificarlas como baja o alta. Los miembros de una escala baja serán B-splines con un intervalo de soporte grande en comparación con los miembros de una escala alta que tendrán un intervalo de soporte más pequeño. Para construir la familia de funciones $B_{\{S(n,l)\}}(\cdot)$ se empieza definiendo un conjunto de secuencias de nodos $S_{(n,l)}$, donde cada conjunto S permite construir n funciones B-spline a una escala l . El número máximo de escalas es un parámetro elegido por el usuario. Al igual que en MARS, la construcción de los términos descritos por la expresión (6.7) empieza con la definición de la función constante $P_0(\mathbf{x}) = 1$ a partir de la cual se busca el siguiente término haciendo

$$P_1(\mathbf{x}) = P_0(\mathbf{x})B_{\{S(n,l)\}}^m(\mathbf{x}_k) \quad (6.8)$$

donde la nueva función $B_{\{S(n,l)\}}^m(\mathbf{x}_k)$ es un miembro de la familia de B-splines de la escala más baja posible que hace que disminuya la diferencia entre los valores obtenidos con la aproximación \hat{y} y los datos reales y . Cada nueva B-spline que contribuye a formar el j -ésimo término $P_j(\mathbf{x})$ se agrega después de que se han evaluado cada una de las funciones base presentes en todas las escalas disponibles desde la más baja hasta la más alta, y para todas las variables que todavía no están involucradas en el término $P_{j-1}(\mathbf{x})$. Este

procedimiento para ir agregando términos en la expresión (6.7) permite construir un modelo de regresión comenzando con una aproximación suavizada de \hat{y} (contribución de baja frecuencia debidas a las B-spline de escala baja) e ir añadiendo términos que aumentan los detalles de la aproximación (contribuciones de alta frecuencia debidas a B-splines de escala alta) hasta un total de M términos definido por el usuario.

El modelo B-MARS obtenido al final de esta primera parte es una aproximación que sobre-ajusta a los datos de la función objetivo y . El proceso de eliminación de términos $P_j(\mathbf{x})$ que tienen la menor contribución al modelo \hat{y} y que permitirán aumentar su capacidad de generalización sigue el mismo procedimiento que el usado en el proceso de eliminación de términos de un modelo MARS: del modelo sobre-ajustado se quita uno de los M términos, se recalculan los coeficientes a_j del modelo formado por los $M-1$ términos restantes y se evalúa el ajuste, si la diferencia $y - \hat{y}$ debida al nuevo modelo no disminuye entonces se devuelve el término eliminado, se quita uno diferente y se recalculan los coeficientes a_j y la diferencia $y - \hat{y}$. El proceso se realiza hasta que se han probado los efectos de la eliminación de cada término para descartar el menos eficiente. Al final de la metodología se tiene un modelo formado por las funciones $P_j(\mathbf{x})$ que más contribuyen a mejorar el modelo \hat{y} .

En las dos etapas de la metodología para la creación de un modelo B-MARS se usa la Validación Cruzada Generalizada (GCV) descrita en la ecuación (6.5) como criterio de elección de los términos $P_j(\mathbf{x})$ que contribuye mucho o poco (dependiendo si el criterio es aplicado a la creación del modelo inicial o la eliminación de términos) al ajuste de la función \hat{y} . El modelo B-MARS que se obtiene al final es de la forma

$$\hat{y} = a_0 + \sum_{j=1}^J a_j \prod_{i=1}^q B_{\{S(n,i)\}}^m(\mathbf{x}_k) \quad (6.9)$$

donde los coeficientes de regresión $\{a_j\}_{j=0}^J$ se determinan también mediante un ajuste de mínimos cuadrados. La anterior descripción de la metodología para la

creación de un modelo de regresión del tipo B-MARS se condensa en el algoritmo mostrado en el Apéndice 6.

Además de la ventaja numérica de BMARS sobre MARS en lo que respecta al cálculo de los coeficientes a_j , otras dos ventajas que se desprenden del desarrollo de su metodología son:

- La reducción del número de funciones base que se evalúan para formar cada uno de los términos $P_j(\mathbf{x})$. Mientras que MARS construye un par de funciones spline truncadas por cada uno de los puntos $\{t_k^{(n)}\}_{n=1}^N$ contenidos en la k -ésima variable $\{\mathbf{x}_k\}_{k=1}^{N_k}$, en B-MARS sólo se permite formar parte de los términos $P_j(\mathbf{x})$ a una pequeña sub-familia de todas las posibles funciones B-spline univariable que se pueden crear en un intervalo $[a,b]$ de la variable de entrada.
- Es posible obtener aproximaciones más suaves usando solamente funciones base de segundo orden o mayor, sin interacción entre variables.

7

Preprocesamiento y Modelado con Redes Neuronales

Otra alternativa en la construcción de modelos de calibración para lenguas electrónicas es la formada por la combinación del preprocesamiento de las señales electroquímicas y el posterior modelado de los coeficientes obtenidos en el preprocesamiento mediante redes neuronales artificiales. Esta combinación la implementamos usando los *Toolboxes de Matlab* de *Procesamiento Wavelet* y *Redes Neuronales Artificiales*, además del *N-way Toolbox* desarrollado por Rasmus Bro de la Universidad de Copenhague. El primero de estos dos Toolboxes está basado en el algoritmo de la Pirámide de Mallat, el segundo usa funciones sigmoidales y lineales como funciones de transferencia para las capas oculta y de salida, y el tercero en el análisis multimodo con N-PLS y Tucker3.

7.1 Transformada Wavelet Discreta

La ecuación (2.2) descrita en el Capítulo 2 para la Transformación Wavelet Continua de una señal $f(t)$ no es práctica para implementarla en un eficiente algoritmo computacional que permita el análisis y síntesis de una señal porque i) las funciones obtenidas de la wavelet madre no forman una base orthonormal, ii) los parámetros a y m son variables continuas, lo que significa que una función bajo análisis puede descomponerse en un número infinito de funciones wavelet y iii) no hay soluciones analíticas para la mayoría de las WT y su cálculo numérico es computacionalmente ineficiente [173].

Las soluciones a estos inconvenientes se consiguen con el uso de la Transformada Wavelet Discreta (*DWT*) [135, 173, 174]. La principal diferencia entre la *CWT* y la *DWT* es que los parámetros de escala y traslación toman valores discretos para el segundo. Si cambiamos los valores de m por m_0^j y de s por $ks_0m_0^j$ entonces la función wavelet discreta toma la forma

$$\psi_{j,k}(t) = \frac{1}{\sqrt{a_0^j}} \psi\left(\frac{t - km_0a_0^j}{a_0^j}\right), \quad j, k \in \mathbb{Z} \quad (7.1)$$

Las nuevas variables j y k son ahora las encargadas de controlar el escalamiento y la traslación de la wavelet, respectivamente. Si se asignan valores a m_0 y s_0 iguales a 2 y 1, respectivamente, a la representación de la wavelet discreta de la expresión (1.4) entonces se obtiene una discretización del plano tiempo-escala conocida como rejilla diádica (Fig 7.1), la cual conduce a la construcción de una base ortonormal de funciones wavelet [44]. La elección de estos valores está dictada por el teorema de muestreo de Shanon [173]. A partir de los nuevos valores para m_0 y s_0 definimos las wavelets diádicas $\psi_{j,k}(t)$ como

$$\psi_{j,k}(t) = \frac{1}{\sqrt{2^j}} \psi\left(\frac{t - k2^j}{2^j}\right) = 2^{-j/2} \psi(2^{-j}t - k), \quad (7.2)$$

a la transformada wavelet discreta como

$$D_{j,k} = \int_{-\infty}^{\infty} f(t) 2^{-j/2} \psi(2^{-j}t - k) dt, \quad (7.3)$$

y a la transformada inversa como

$$f(t) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} D_{j,k} 2^{-j/2} \psi(2^{-j}t - k) \quad (7.4)$$

La ortonormalidad de las wavelets diádicas asegura que la información almacenada en un coeficiente wavelet es única y no se repite en otra parte, con lo que la síntesis de una señal a partir de los estos coeficientes wavelet es completa y sin redundancia.

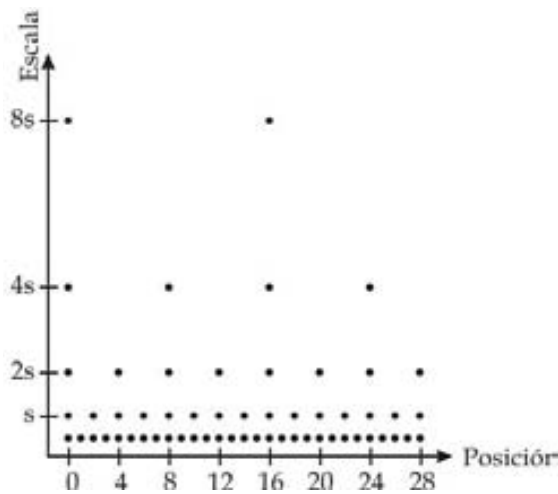


Fig. 7.1. Localización de las wavelets discretas en el muestreo diádico del plano tiempo-escala.

7.1.1 Bancos de filtros y análisis multiresolución

La Transformación Wavelet de una señal puede verse como si la señal se procesase a través de un banco de filtros. Para explicar esta equivalencia retomemos el concepto final de la sección 3.1.1 donde se describe a la función wavelet como un filtro pasa-banda en el plano de Fourier. Adicionalmente a esta descripción tenemos que cuando una señal se comprime en el dominio del tiempo, como es el caso de las wavelets hijas, entonces su correspondiente espectro en el plano de Fourier se estira y desplaza hacia arriba en la escala de frecuencias [123],

$$F\{f(at)\} = \frac{1}{|a|} F\left(\frac{\omega}{a}\right) \quad (7.5)$$

Juntando los dos conceptos anteriores podemos concluir que si comprimimos una función wavelet en un factor de 2 en el dominio del tiempo entonces doblaremos su espectro en frecuencia y desplazaremos sus componentes por el mismo factor. A partir de esta idea se concluye que podemos cubrir el espectro de una señal a analizar usando el espectro de wavelets escaladas, de la misma manera en como abarcamos la señal en el dominio del tiempo con waveletes trasladadas (Fig. 7.2).

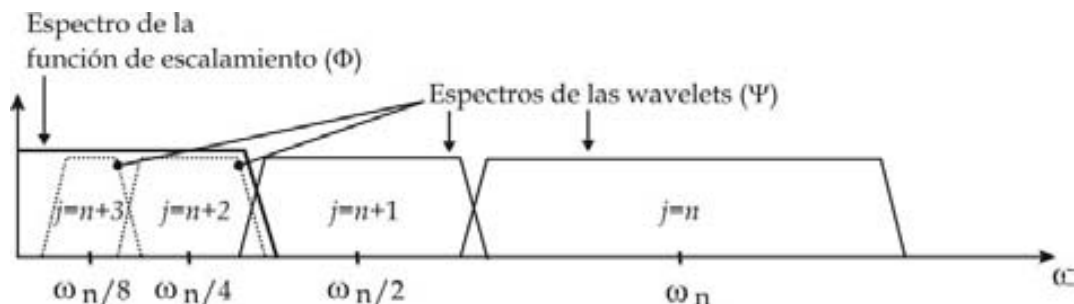


Fig. 7.2. La función de escalamiento se encarga de cubrir el espectro de baja frecuencia hasta 0. Su uso evita la creación de un número infinito de funciones wavelet.

Para que la DWT logre cubrir todo el espectro frecuencial de una señal hasta la frecuencia cero, Stéphane Mallat introdujo una función con un espectro de filtro paso-bajas llamada *función de escalamiento*, que se encarga de cubrir el espectro vacío de baja frecuencia dejado por las wavelets [174]. La función de escalamiento $\phi_{j,k}(t)$ propuesta por Mallat tiene la misma forma que la función wavelet, con la diferencia de que es ortogonal a traslaciones sobre sí misma pero no a escalamientos sobre sí misma. Dicha función está descrita por la expresión (7.6).

$$\phi_{j,k}(t) = 2^{-j/2} \phi(2^{-j}t - k) \quad (7.6)$$

La convolución de una señal $f(t)$ con la función de escalamiento $\phi(t)$ produce un conjunto de coeficientes de aproximación etiquetados como $A_{j,k}$

$$A_{j,k} = \int_{-\infty}^{\infty} f(t) 2^{-j/2} \phi(2^{-j}t - k) \quad (7.7)$$

A partir de los coeficientes $A_{j,k}$ y $D_{j,k}$ de las expresiones (1.10) y (1.6), respectivamente, se puede conseguir la síntesis de $f(t)$ mediante la combinación de la aproximación continua de la señal al primer nivel de descomposición j_0 y la suma de las contribuciones de los coeficientes y funciones wavelet desde j_0 hasta menos infinito.

$$f(t) = \sum_{k=-\infty}^{\infty} A_{j_0,k} \phi_{j_0,k}(t) + \sum_{j=-\infty}^{j_0} \sum_{k=-\infty}^{\infty} D_{j,k} \psi_{j,k}(t) \quad (7.8)$$

Las características de filtros paso-banda y paso-bajas que poseen las funciones $\psi(t)$ y $\phi(t)$ permiten que una función de escalamiento junto con un grupo de wavelets escaladas se interpreten como un banco de filtros en un esquema de codificación en sub-bandas, como el representado en la Fig. 7.3 [135, 174, 175, 176,].

La codificación en sub-bandas permite obtener la aproximación de una señal a un nivel j_{n-1} a partir de la suma de la aproximación y detalles a un nivel inferior j_n . Este tipo de representación se conoce como *análisis multiresolución* [174],

$$\hat{f}_{j_{n-1}}(t) = \sum_{k=-\infty}^{\infty} A_{j_n,k} \phi_{j_n,k} + \sum_{k=-\infty}^{\infty} D_{j_n,k} \psi_{j_n,k} \tag{7.9}$$

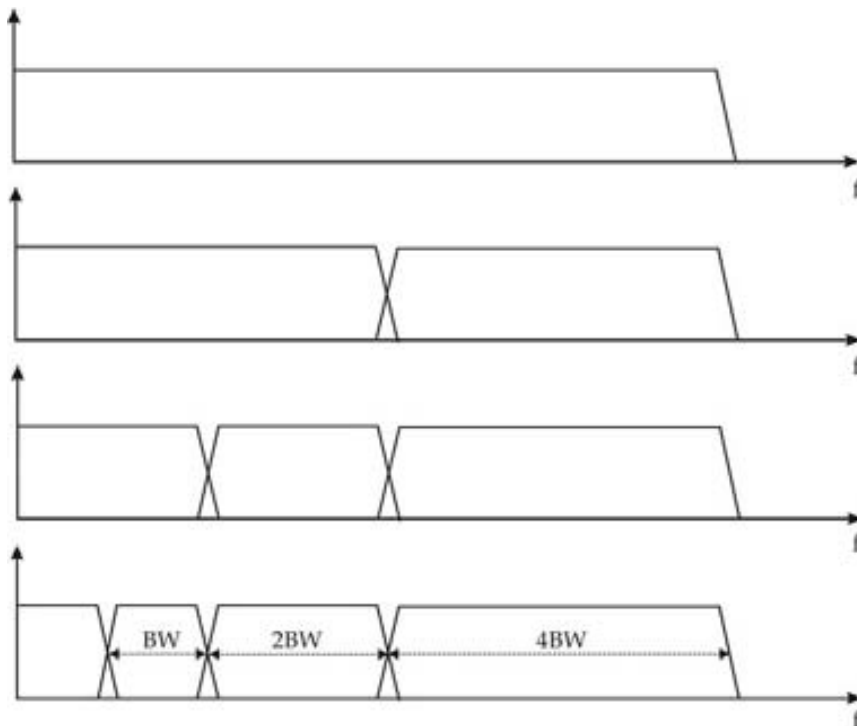


Fig. 7.3. Esquema de codificación en sub bandas de la DWT. La salida de cada filtro pasa-bajas se descompone de nuevo en términos de alta y baja frecuencia, creando un banco de filtros iterativos.

Las características de filtro de las funciones de escalamiento y wavelet junto con el análisis multiresolución se usan en el *Toolbox de Procesamiento*

Wavelet de Matlab para implementar la DWT en un algoritmo eficiente conocido como la pirámide de Mallat. En este algoritmo la DWT se obtiene haciendo pasar la señal discreta $f(k)$ a través de filtros consecutivos paso-altas y paso-bajas, como se muestra en la Fig. 1.10. En cada nivel el filtro pasa-altas produce coeficientes de aproximación ($A_{j,k}$) y detalle ($D_{j,k}$). Las expresiones para obtener $A_{j,k}$ y $D_{j,k}$ a partir de la convolución con los filtros pasa-baja y pasa-alta son

$$\begin{aligned} A_{j+1,k} &= \sum_n h_0(n-2k) \cdot A_j(n) \\ D_{j+1,k} &= \sum_n g_0(n-2k) \cdot A_j(n) \end{aligned} \quad (7.10)$$

En cada nivel de descomposición los filtros producen señales que tienen la mitad del contenido espectral y el mismo periodo de muestreo que el nivel anterior, por lo que las señales filtradas pueden decimarse en un factor de 2 sin perder información. El proceso de filtrado y decimación se continúa hasta que se alcanza el nivel deseado, siendo el máximo número de niveles un parámetro dependiente de la longitud de la señal. La DWT final se obtiene concatenando los coeficientes $A_{j,k}$ y $D_{j,k}$ empezando por el último nivel de descomposición.

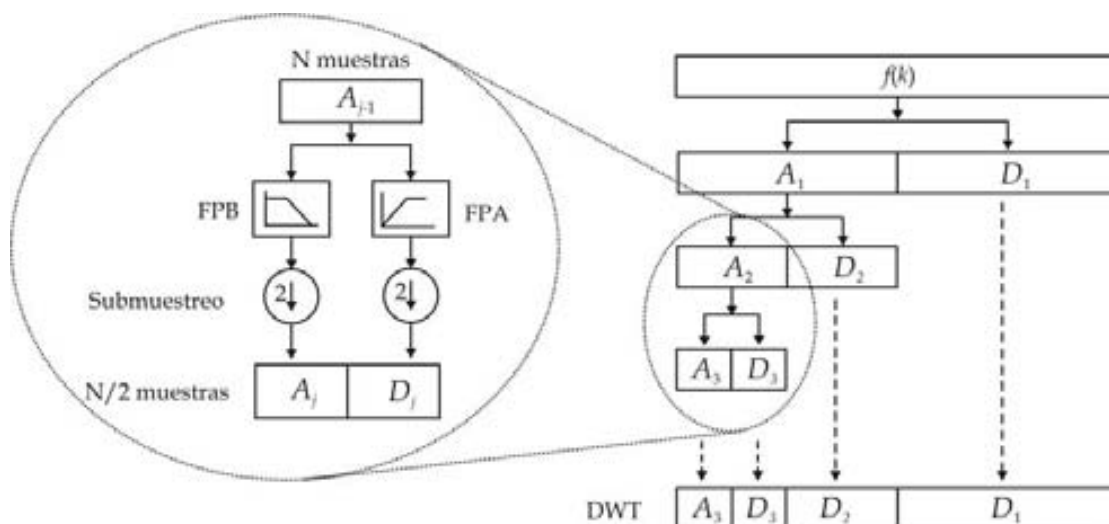


Fig. 1.10. Esquema del cálculo de la DWT usando la pirámide de Mallat. La señal muestreada se descompone en términos de alta y baja frecuencia en cada nivel de análisis y luego se sub-muestran. El proceso se repite sobre cada nuevo grupo de coeficientes de aproximación hasta el nivel de descomposición deseado, para formar al final la DWT con la concatenación de los coeficientes $A_{j,k}$ y $D_{j,k}$ desde el último nivel de descomposición hasta el primero.

La señal original se puede reconstruir a partir de los coeficientes de aproximación y detalle obtenidos de la DWT. El proceso de reconstrucción, conocido como Transformada Wavelet Discreta Inversa (IDWT) es básicamente lo opuesto al proceso de descomposición. Los coeficientes $A_{j,k}$ y $D_{j,k}$ se sobremuestran por dos, se pasan a través de filtros complementarios de síntesis pasa-bajas y pasa-altas y luego se suman. El proceso se realiza un número de veces igual al número de niveles usados en la descomposición hasta obtener finalmente la señal original.

$$A_{j,k} = \sum_n h_0(k-2n) \cdot A_{j+1}(n) + \sum_n g_0(k-2n) \cdot D_{j+1}(n) \quad (7.11)$$

La señal reconstruida en cada nivel es una aproximación obtenida a partir del análisis multiresolución. La señal $f(k)$ se reconstruye aplicando iterativamente la ecuación (1.14) hasta el primer nivel de descomposición.

7.2 Preprocesamiento con DWT + ANNs

El objetivo que se persiguió con el procesamiento wavelet fue reducir la longitud original de cada registro electroquímico a la vez de intentar retener la mayor cantidad de información de la señal original. La utilidad de la transformación wavelet en la extracción de características significativas de señales electroquímicas y el subsecuente uso de estas características en la construcción de modelos cuantitativos usando técnicas convencionales de calibración multivariable ha sido demostrada en publicaciones anteriores [177,178, 179]. Para realizar la compresión de cada matriz de datos se usaron una de las dos metodologías que se describen a continuación y que se esquematizan en la Fig. 2.11.

1. *Retención de los coeficientes de aproximación.* La transformación wavelet de los registros se probó desde el primer nivel de descomposición hasta el máximo permitido por la entropía de Shannon de la señal, definida como el límite absoluto de la mejor compresión posible sin pérdida de información, bajo ciertas condiciones [65]. La compresión se hizo reteniendo únicamente los coeficientes de aproximación obtenidos en cada uno de los niveles de descomposición y desechando los coeficientes de detalle. La efectividad en la compresión conseguida en cada uno de los niveles de descomposición se

evaluó comparando la similitud entre la señal original y la señal sintetizada a partir de los coeficientes retenidos. La técnica de comparación propuesta entrega una figura de mérito con valor acotado en el intervalo $[0, 1]$ que indica el nivel de similitud: 0 para señales totalmente diferentes y 1 para señales idénticas [179].

2. *Combinando coeficientes de aproximación y detalle.* La transformación de los registros se realizó hasta el máximo nivel de descomposición permitido por la entropía de la señal. La compresión se realizó tomando los coeficientes de aproximación del máximo nivel de descomposición además de varios coeficientes de detalle obtenidos en los niveles de descomposición anteriores. Al igual que en caso anterior, la efectividad en la compresión se evaluó comparando la similitud entre la señal original y la señal sintetizada a partir de los coeficientes retenidos. Ambas aproximaciones se han esquematizado en la Fig. 2.17.

En cualquiera de los dos casos, la compresión resulta en un compromiso entre el número de coeficientes retenidos y el grado de similitud entre las señales original y sintetizada. Los conjuntos de señales electroquímicas obtenidas de cada sensor se transformaron usando diferentes familias de wavelets discretas disponibles en Matlab. El porcentaje de compresión de los registros electroquímicos procesados con la TWD nos permite tener un indicador de la cantidad de información preservada, aunque no es un parámetro definitivo. La función wavelet y el nivel de descomposición elegido para la compresión de los registros se escogieron tomando en cuenta el grado de similitud entre el registro original y el sintetizado a partir de los coeficientes escogidos de la compresión. Para cuantificar la similitud hemos usado dos parámetros: el primero es el clásico coeficiente de correlación lineal r , y el segundo es un coeficiente que resulta de un proceso de comparación del área bajo la curva que hay en cada una de las dos señales que se comparan. El coeficiente para la evaluación de la similitud se define como la relación entre el área intersectada por ambas curvas y el área total que hay bajo ambas. Etiquetando como A y B a las áreas anteriormente mencionadas y usando la teoría de conjuntos para describir la relación, ésta puede expresarse como $(A \cap B) / (A \cup B)$. El resultado de esta relación varía entre 0 y 1 dependiendo del grado de similaridad, es cero cuando las señales no tienen nada en común y aumenta su valor conforme aumenta la similitud entre ambas señales. La Fig.

2.19 muestra gráficamente los términos usados para calcular este coeficiente. A partir de señales discretizadas de longitud N_k , como es nuestro caso con los registros electroquímicos, el coeficientes de similitud se evalúa mediante

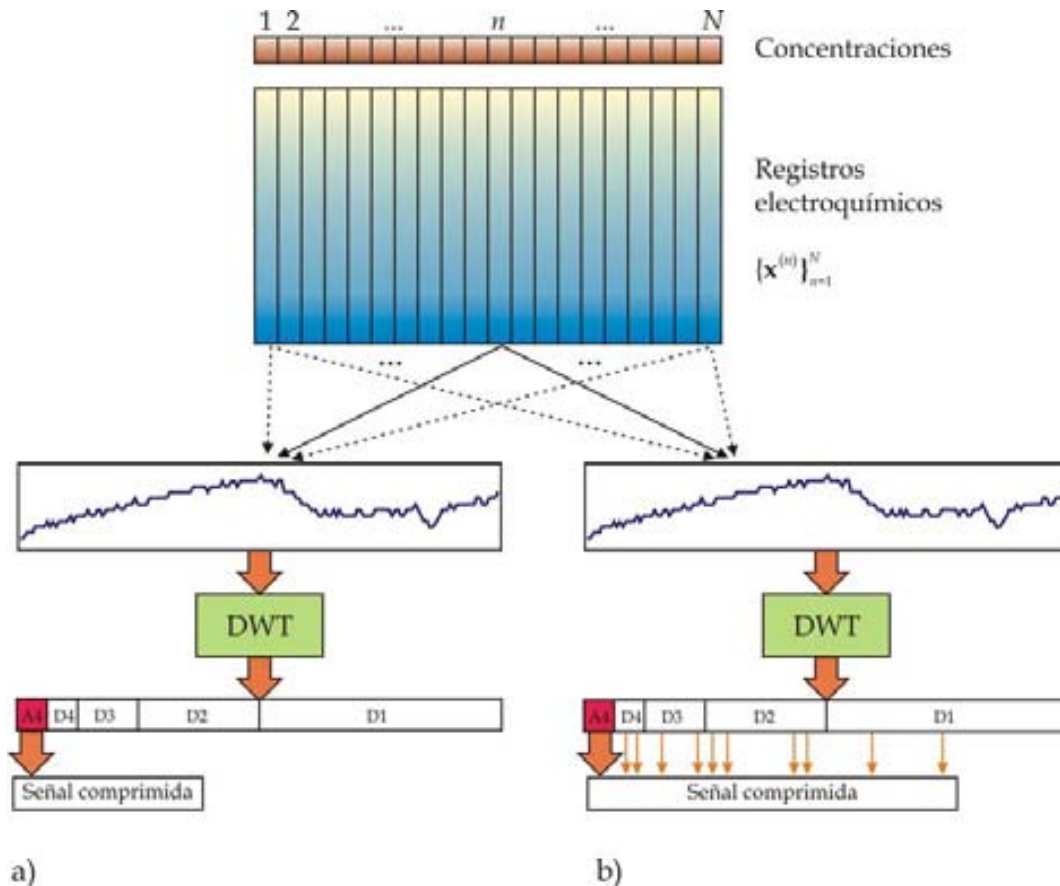


Fig. 2.17. Esquemas de las metodologías usadas en la compresión de señales. a) Retención de los coeficientes de aproximación únicamente. b) Combinación de los coeficientes de aproximación al máximo nivel de compresión más coeficientes de detalle de diferentes niveles de descomposición.

$$\frac{A \cap B}{A \cup B} = \frac{\sum_{k=1}^{N_k} (\max\langle a_k, b_k \rangle - |a_k - b_k|)}{\sum_{k=1}^{N_k} (\min\langle a_k, b_k \rangle + |a_k - b_k|)} \quad (7.12)$$

donde a_k y b_k son los k -ésimos elemento del registro electroquímico original y del sintetizado a partir de los coeficientes wavelet. Los coeficientes obtenidos de la compresión se arreglaron de dos maneras diferentes antes de ser usados en el entrenamiento de las redes neuronales:

- i. Cada matriz de coeficientes wavelet se usó de manera independiente para intentar modelar con una sola ANN todos los analitos de interés presentes en la solución.
- ii. Las matrices de coeficientes wavelet se ordenaron para formar una sola matriz de datos antes de ser usadas en el modelado con ANNs.

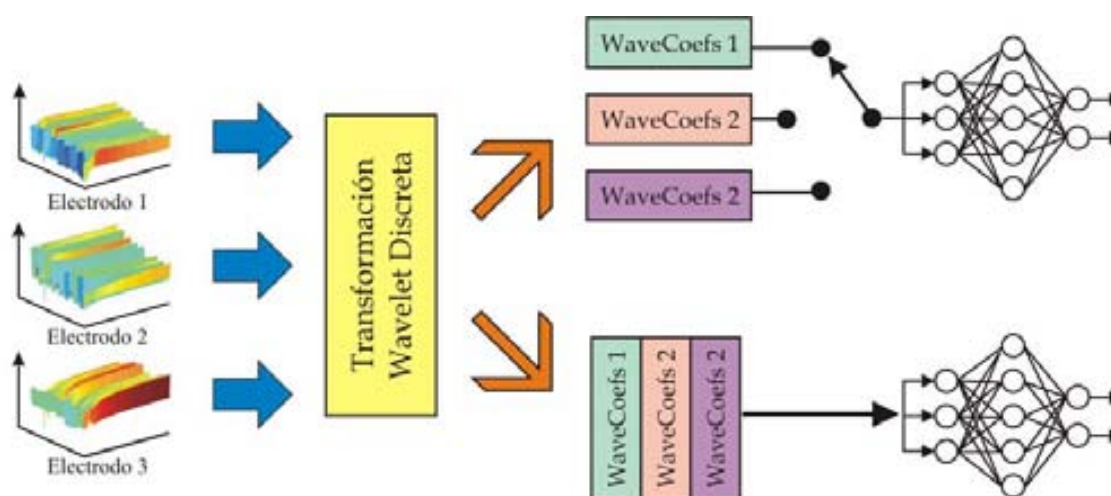


Fig. 2.18. Esquema de las modelizaciones realizadas con los coeficientes wavelet obtenidos del pre-procesamiento. Cada registro electroquímico es transformado usando una wavelet discreta, y los coeficientes resultantes de la transformación pueden usarse de manera independiente o unirlos en una sola matriz antes de ser introducidos en una ANN para su modelización.

En cuanto al tipo de redes usadas en esta aproximación, éstas fueron del tipo feedforward con retropropagación del error y entrenamiento basado en el algoritmo de regulación bayesiana, 1 o 2 capas ocultas con variable número de neuronas ocultas en ellas y funciones sigmoideas no-lineales de transferencia. El número de capas y neuronas ocultas es determinado por prueba y error. La elección de la estructura óptima comenzó con una capa y un número de neuronas ocultas igual al promedio geométrico obtenido de los coeficientes de entrada y las neuronas de salida. El número de neuronas ocultas se incrementó gradualmente en una unidad para mejorar el desempeño de la red hasta un máximo de neuronas igual al número de coeficientes wavelet de entrada. Una vez obtenida la estructura óptima de una capa oculta se evaluó el desempeño de la red con dos capas ocultas; en este caso, la suma de neuronas de las dos capas se igualó al de la estructura con una capa oculta, desde el mínimo hasta

el máximo. En el caso de las neuronas en la capa de salida ésta depende de cuántos analitos se modelarán.

El entrenamiento de las redes se hizo con el algoritmo de regularización Bayesiana descrito en la sección 1.4.1 y el ajuste durante el entrenamiento se monitorizó evaluando el error E_r obtenido en cada iteración y que se ha descrito a lo largo de esta memoria.

Sección 3

8

Pruebas y resultados

Las soluciones que se usaron en los experimentos que se describirán a continuación se prepararon usando el sistema SIA descrito en la Sección 1.3, con excepción de los datos que se describen en el apartado 8.3 y que están relacionados con el control de calidad en la producción del azúcar a partir de la remolacha. Estos datos fueron descargados de una base de datos pública a través de Internet, donde se ofrecen como datos de prueba para comprobar las prestaciones de los algoritmos de calibración multivariante.

El sistema SIA comentado está controlado por ordenador a través de un programa escrito en LabVIEW. Por cada experimento realizado, se programó al sistema SIA para preparar varios estándares multi-analito mezclando apropiadamente diferentes volúmenes de soluciones tomados de manera secuencial usando la microbureta y la válvula de selección. Los volúmenes de analitos y soluciones tampón se bombearon al interior de una celda de mezcla de 10ml de cavidad construida con un bloque de Perspex, donde se homogenizaron usando un núcleo de agitación y un agitador magnético. Las diferentes concentraciones de analitos que se usaron en cada experimento se obtuvieron usando otro programa de ordenador creado en el GSB y que elige, de acuerdo a un conjunto de valores aleatorios acotados entre un máximo y un mínimo, aquellos que representan una completa independencia entre las muestras para asegurar que no existe correlación entre las variables, permitiendo minimizar posibles tendencias en los valores, derivas y efectos de memoria.

Por cada experimento que se llevó a cabo se obtuvieron dos conjuntos de datos que se usaron en la construcción de los modelos de calibración: uno es el formado por las concentraciones de analitos usados en cada experimento y que representan las variables dependientes a predecir, y otro es el formado por los registros voltamperométricos o potenciométricos que representan a las variables independientes de entrada a los modelos de calibración y que permitirán predecir las concentraciones de analitos anteriormente mencionadas. Cada conjunto de datos experimentales se dividió en dos subconjuntos con el propósito de usarlos en los procesos de construcción del modelo y prueba del mismo. Por conveniencia en el proceso de modelización, el subconjunto correspondiente a esta etapa se normalizó al intervalo $[-1, 1]$ usando sus propios valores mínimos y máximos, y posteriormente, estos valores se usaron en la normalización del subconjunto de prueba para hacer compatibles sus valores con el modelo de calibración obtenido. Las salidas que se obtuvieron de los modelos de calibración tanto al finalizar el entrenamiento como en la etapa de prueba se desnormalizaron a sus valores reales con la finalidad de compararlos contra los valores esperados y así evaluar el desempeño de la red. A menos que se indique lo contrario, las herramientas se programaron para que el error máximo promedio por cada una de las salidas en entrenamiento no fuese mayor al 10%. En el caso de los modelos construidos con redes neuronales artificiales, la inicialización de sus parámetros fue aleatoria en un intervalo $[-1, 1]$ con una distribución normal y media cero.

Todas las herramientas que se describieron en los Capítulos 6-10 y que se prueban en este apartado fueron programadas en MATLAB 7.0 usando un ordenador portátil con procesador Intel T7300 de doble núcleo a 2.00GHz y memoria RAM de 2GBytes.

8.1 Determinación de compuestos oxidables usando una WNN con función de transferencia basada en la norma Euclidiana

Con el objetivo de cuantificar un conjunto de compuestos oxidables normalmente hallados en las formulaciones de fármacos, que tienen potenciales de oxidación similares y por lo tanto presentan un alto grado de

traslape en las pruebas voltamperométricas, se plantea en este apartado experimental la determinación simultánea de ácido ascórbico (AA), 4-aminofenol (4-Aph) y paracetamol (Pct) en soluciones tri-componentes sintetizadas en laboratorio [4]. El sensor electroquímico que se usó fue construido con una resina epóxica Epotek H77 (Epoxy Technology, USA) mezclada con grafito [180, 181], y como electrodo de referencia se usó un electrodo de Ag/AgCl (Modelo 52-40 de Crison Instruments). La técnica electroquímica empleada fue la voltamperometría lineal de barrido [182]. El potencial se varió desde 0 hasta 1.0V con una tasa de barrido de 100mV/s y escalones de voltaje 10mV, lo que permitió obtener 101 puntos de medición. Las intensidades de respuesta del sistema se adquirieron en cada potencial aplicado usando un sistema electroquímico Autolab/PGSTAT 20 (Ecochemie).

Los datos para la construcción de los modelos de calibración se obtuvieron a partir de 62 soluciones experimentales que se prepararon usando concentraciones de componentes puros disueltos en una solución tampón de KCl con concentración 0.1M. Las concentraciones de los analitos usados en las soluciones sintetizadas estuvieron en los intervalos [12–410] μM para el AA, [17–530] μM para el 4-Aph y [10–420] μM para el Pct; estos tres intervalos se eligieron en base a los resultados obtenidos con pruebas preliminares relacionadas con la linealidad y límite de detección de los electrodos.

Al finalizar el experimento se obtuvo un conjunto de datos formado por una matriz \mathbf{Y} de concentraciones de tamaño 3×62 y una matriz \mathbf{X} de predictores construida con las intensidades de corrientes de tamaño 101×62 . Las concentraciones de analitos son las salidas a modelar por la red mientras que los voltamperogramas son los datos de entrada del modelo. La distribución de las concentraciones usadas en este experimento está graficada en la Fig. 1, donde se puede observar la ausencia de tendencias y agrupaciones. Cada punto azul de la figura izquierda representa una de las triadas de concentraciones preparadas por el sistema SIA. El conjunto de voltamperogramas obtenidos con el electrodo de grafito está graficado en la Fig. 2. Cada uno de los registros voltamperométricos que forman la superficie se corresponde con cada uno de los puntos en el espacio tridimensional de concentraciones. Las intensidades de corriente adquiridas con el electrodo estuvieron en el intervalo $[-1.4, 52.4]$ mA. La separación de los datos en dos subconjuntos se hizo tomando las parejas de registros/concentraciones con las

dos primeras posiciones de cada grupo de tres para entrenamiento (registros 1, 2, 4, 5, ... hasta 62), dejando la tercera posición para prueba (registros 3, 6, 9, ... hasta 62).

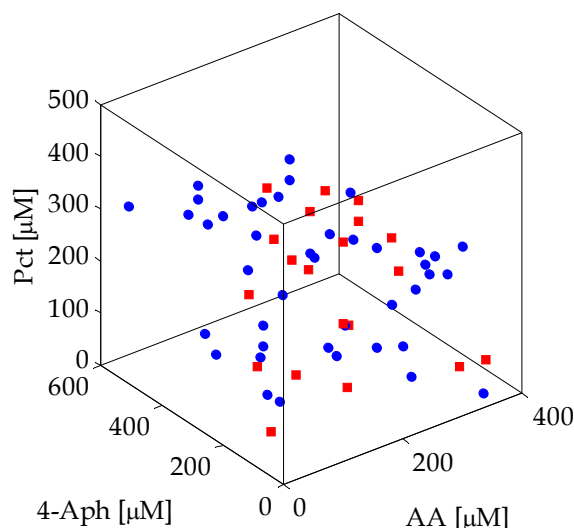


Fig. 1. Gráfica tridimensional de las concentraciones de analitos usados en el experimento. Los puntos azules son los elementos usados en la construcción del modelo y los cuadros rojos son los elementos usados en la prueba del mismo.

En este primer experimental se probó la WNN con funciones de transferencia basadas en la norma euclidiana. Se entrenaron redes neuronales wavelet con 3 neuronas en la capa de salida y función de transferencia lineal para modelar simultáneamente los 3 analitos, 101 neuronas de entrada y un número variable de neuronas en la capa oculta (desde 10 hasta 30 en intervalos de 2 neuronas) con funciones de transferencia *Wavelet Marr* y *Halo* del tipo norma-2. Las redes con 10 a 14 neuronas ocultas fueron ineficaces y no alcanzaron el error programado en entrenamiento, por lo que no se presentarán los resultados obtenidos con ellos. De las redes restantes, una vez que alcanzaron el error en entrenamiento, cada una se re-entrenó cinco veces para verificar que la estructura era capaz de alcanzar el ajuste debido a que la aleatoriedad en la inicialización de los parámetros de la red hace que sus valores finales no sean siempre los mismos, lo que tiene como consecuencia que no se pueda alcanzar el error esperado en entrenamiento porque la dispersión de las diferencias entre los valores obtenidos y esperados sea grande.

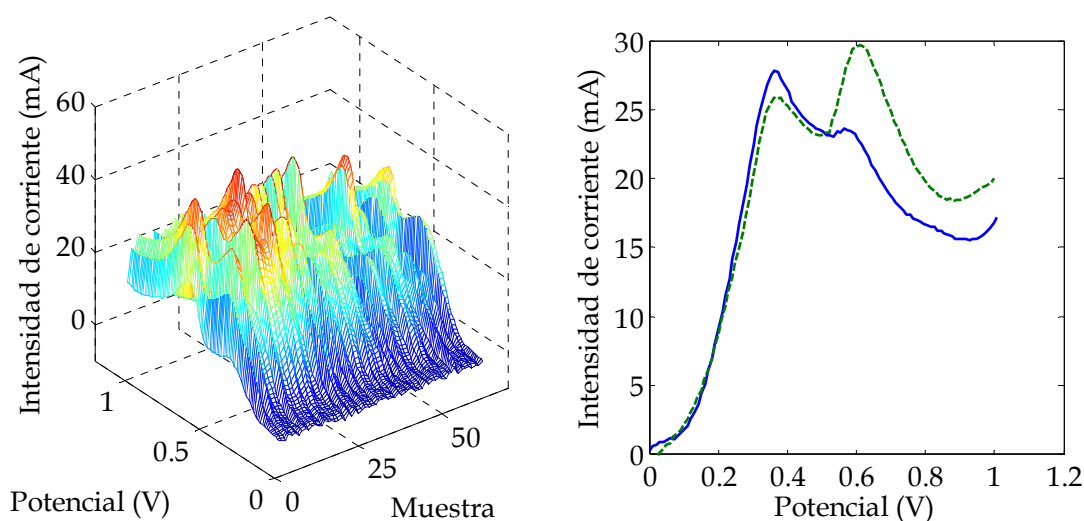


Fig. 2. La gráfica de la izquierda muestra los 62 registros voltamperométricos obtenidos con las triadas de concentraciones, la gráfica de la derecha muestra los registros que corresponden a las mezclas de concentraciones [190,17; 165,00; 40,74] μM y [212,39; 113,43; 127,33] μM . En ella puede observarse el elevado grado de traslape en los voltajes de oxidación de los componentes.

Se evaluó el desempeño de las redes entrenadas comparando los valores obtenidos en sus salidas contra los valores reales esperados mediante un análisis de regresión lineal. Los parámetros de correlación (R), pendiente (m) e intersección de la recta con el eje de ordenadas (b) obtenidos del análisis permiten conocer la eficiencia de la red como modelo de calibración, en el que se esperan valores de R próximos a 1, m prácticamente 1 y b prácticamente 0. Adicional al análisis de regresión se calculó un cuarto parámetro conocido como *Error Porcentual Absoluto Promedio (EPAP)*, este parámetro se define mediante

$$EPAP = \frac{\sum_{i=1}^N \left| \frac{y_{real(i)} - y_{obt(i)}}{y_{real(i)}} \right|}{N} \times 100 \quad \text{Equation Section 8(8.1)}$$

donde $y_{real(i)}$ es el i -ésimo valor esperado, y $y_{obt(i)}$ es el i -ésimo valor conseguido con la red. Los resultados obtenidos con una red neuronal de 20 neuronas ocultas y función de transferencia *Wavelet Marr* se muestran en las Figs. 3 a 7. Las Figs. 8 a 12 muestran los correspondientes resultados obtenidos con una red de la misma dimensión y función de transferencia

Wavelet Halo. Se han mostrado estos resultados solo como una muestra del desempeño de las Redes Neuronales Wavelet basadas en la norma Euclidiana.

Las gráficas de la Fig. 3 muestran los valores de concentraciones reales por cada solución preparada y los obtenidos con la WNN para ambos subconjuntos de entrenamiento y prueba, la Fig. 4 muestra las gráficas de regresión lineal obtenidas con los mismos subconjuntos de datos. Los correspondientes resultados obtenidos con la WNN con función de transferencia *Wavelet Halo* están en las Figs. 8 y 9.

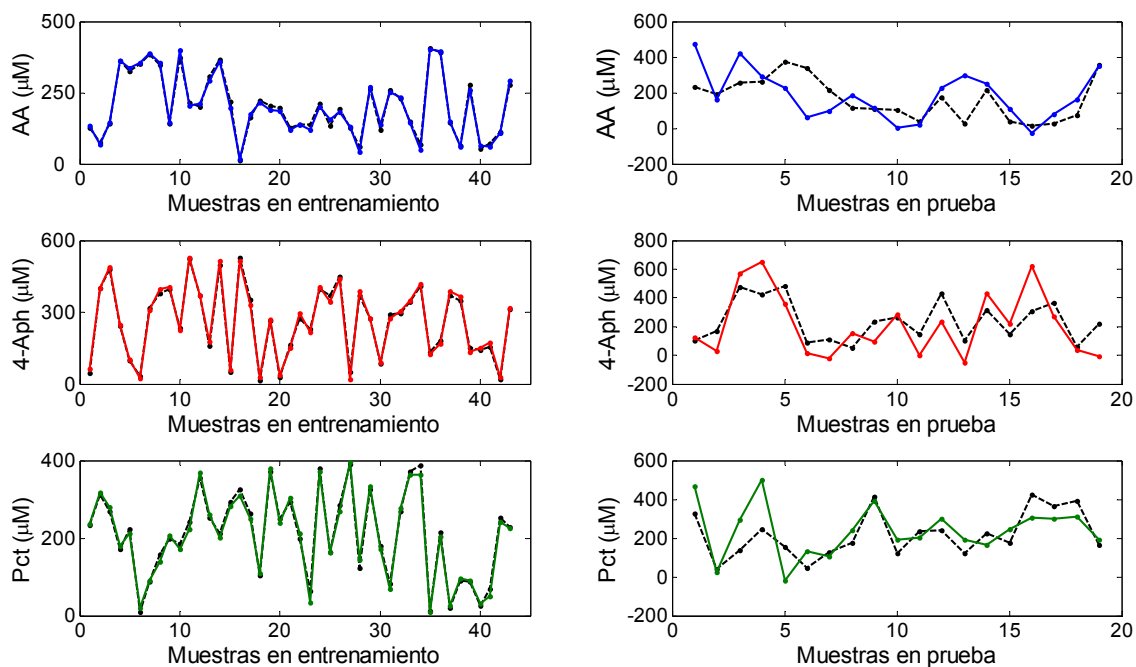


Fig. 3. Valores reales de concentraciones y valores obtenidos con una WNN con función de transferencia *Wavelet Marr* y 20 neuronas en la capa oculta. La columna izquierda corresponde al proceso de entrenamiento y la columna derecha a prueba. Los puntos negros unidos por una línea segmentada son los valores reales, los puntos de colores unidos por una línea continua son los valores obtenidos.

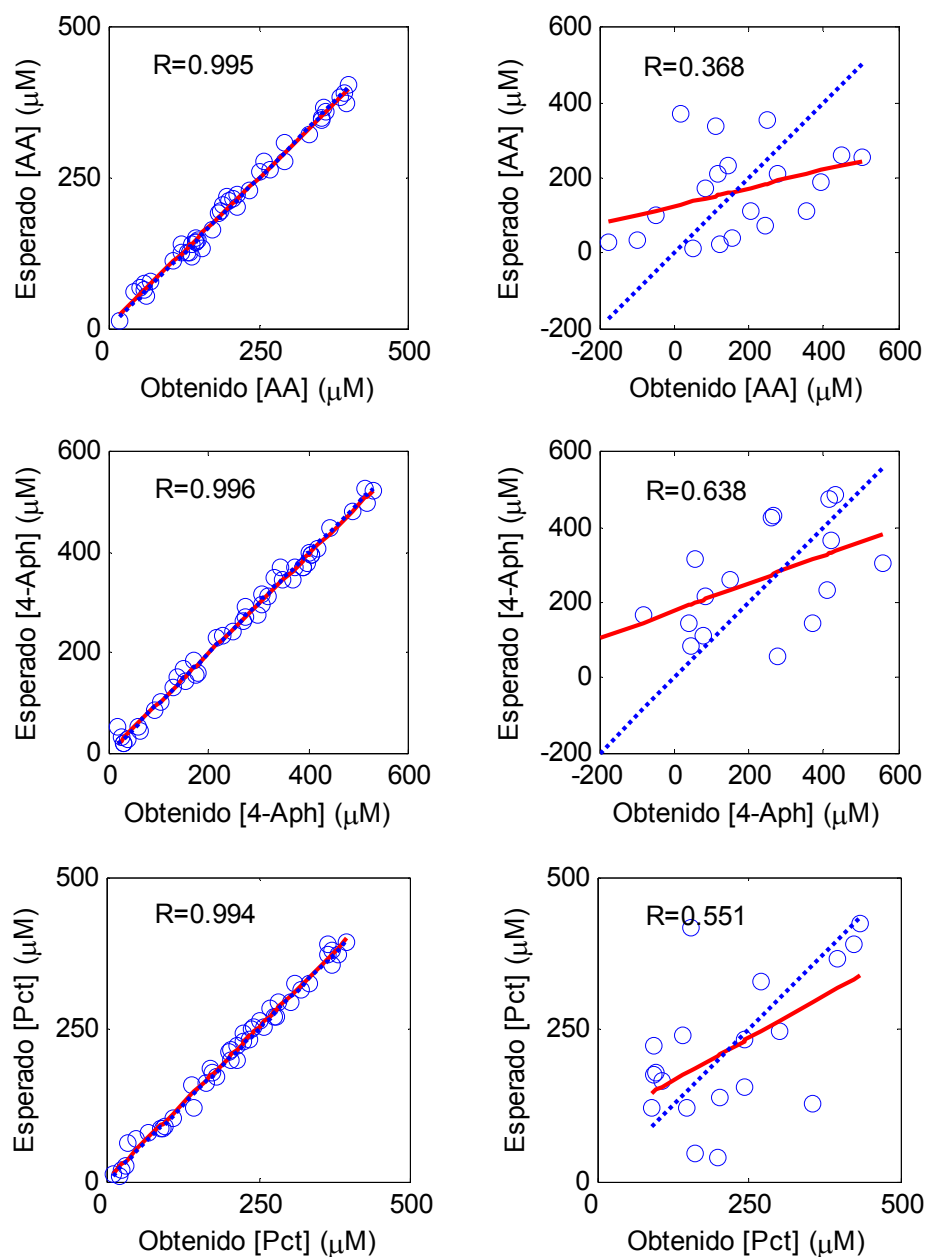


Fig. 4. Gráficas del análisis de regresión lineal entre los resultados reales y los obtenidos con una red neuronal de 20 neuronas con funciones de transferencia Wavelet Marr basadas en la norma Euclidiana. Las gráficas en la columna de la izquierda corresponden al entrenamiento y las de la derecha a la prueba. La línea continua es la regresión que mejor se ajusta y la discontinua la ideal.

De las gráficas correspondientes a la WNN con función Wavelet Marr puede observarse que el entrenamiento fue un procedimiento exitoso que permitió obtener valores de correlación de 0.995, 0.996 y 0.994 para el AA, 4-Aph y PCT, respectivamente. Sin embargo, la generalización de la red a los

datos de prueba se aleja mucho de lo esperado en un modelo de calibración, teniendo valores de correlación de 0.368 para el AA, 0.638 para el 4-Aph y 0.551 para el PCT. Estos valores de correlación, junto con la pendiente, intercepción y error porcentual obtenidos con una red Wavelet con 20 neuronas ocultas se condensan en la Tabla 1.

Tabla 1. Coeficientes de regresión lineal y error porcentual obtenidos con el primer entrenamiento hecho con una red neuronal wavelet de 20 neuronas con función de transferencia Wavelet Marr. La tabla contiene los valores obtenidos tanto en entrenamiento como en prueba.

Analito	m		b		R		Error (%)	
	tr	ts	tr	ts	tr	ts	tr	ts
AA	0.976	0.238	6.096	126.23	0.995	0.368	7.528	178.03
4-Aph	0.985	0.368	1.907	177.66	0.996	0.637	10.059	132.93
Pct	0.997	0.561	3.234	95.09	0.994	0.551	10.270	69.072

Debido a los resultados obtenidos en la etapa de prueba, se hicieron 5 entrenamientos de validación cruzada por cada estructura de red neuronal para evaluar si la elección de los registros para el modelado tiene influencia en la capacidad de generalización de la red. Para cada uno de los entrenamientos realizados se hizo un análisis de regresión lineal entre los valores obtenidos y los esperados. Los subconjuntos del primer entrenamiento de validación cruzada se construyeron tomando las posiciones $[3n - 2, 3n, \dots, 62]_{n=1 \dots 21}$ para entrenamiento y $[3n - 1, \dots, 62]_{n=1 \dots 21}$ para prueba, los del segundo entrenamiento se construyeron usando las posiciones $[3n - 1, 3n, \dots, 62]_{n=1 \dots 21}$ para entrenamiento y $[3n - 2, \dots, 62]_{n=1 \dots 21}$ para prueba, y los 3 entrenamientos restantes se hicieron separando aleatoriamente las parejas de registros/concentraciones. Los promedios de los parámetros de regresión lineal junto con los errores porcentuales obtenidos en las etapas de entrenamiento y prueba para los tres compuestos estudiados se muestran gráficamente en las Figs. 5 a 7.

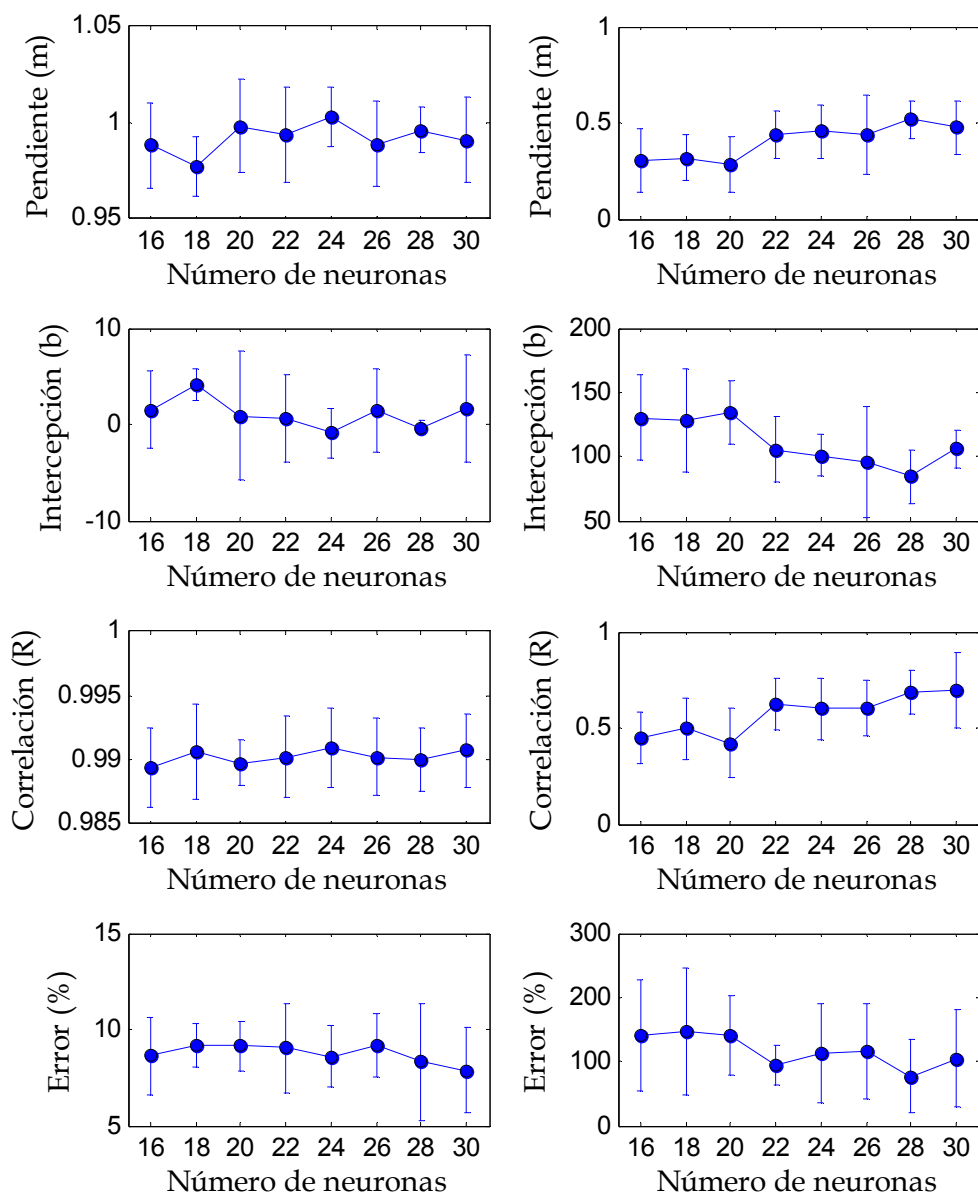


Fig. 5. Promedios de los resultados del análisis de regresión lineal y porcentajes de recuperación obtenidos con la validación cruzada usando una WNN con función de transferencia Marr. Las gráficas en la columna izquierda corresponden al entrenamiento de la red y las gráficas la columna derecha a la prueba. Las marcas de error indican el 95% del intervalo de confianza. Los resultados corresponden al ácido ascórbico.

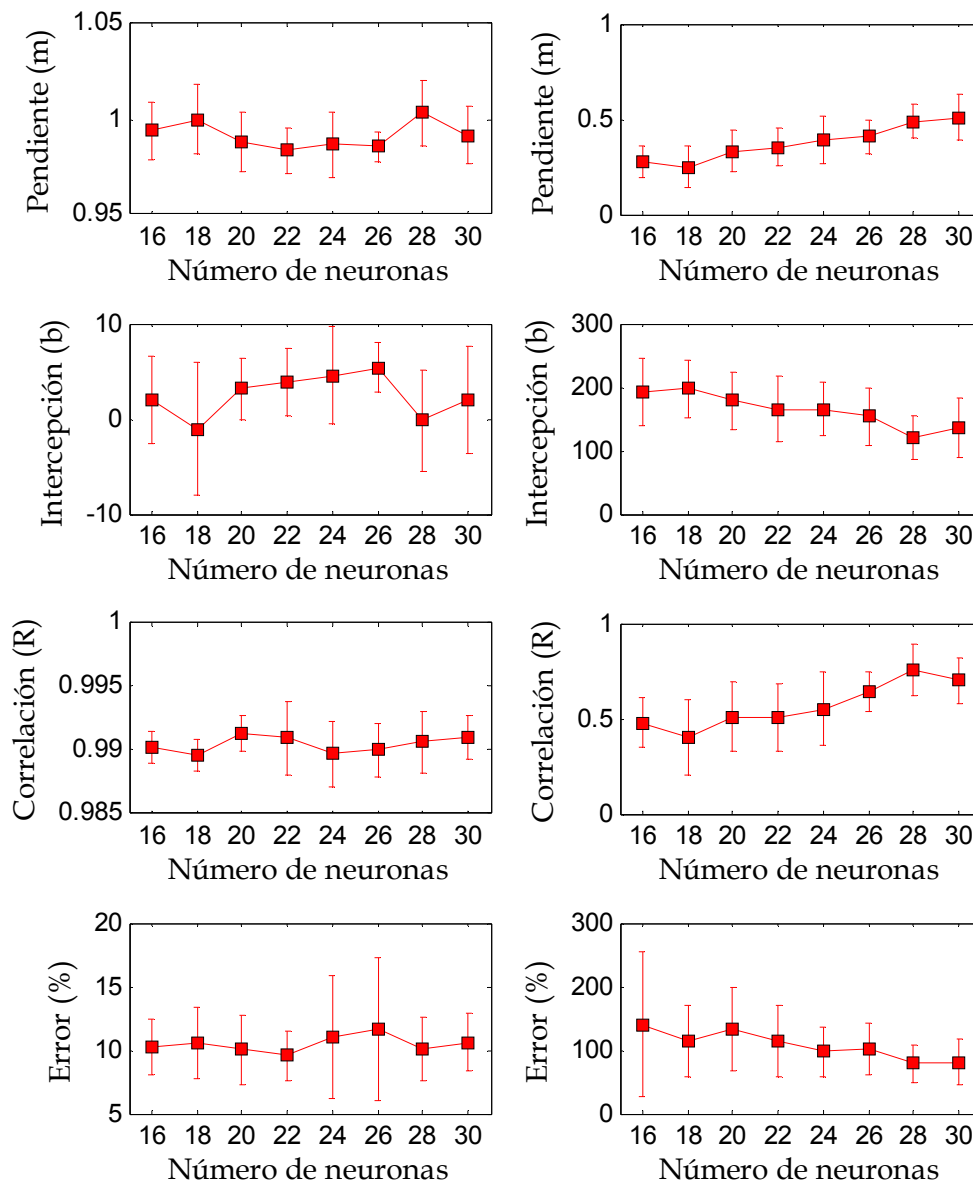


Fig. 6. Promedios de los resultados del análisis de regresión lineal y porcentajes de recuperación obtenidos con la validación cruzada usando una WNN con función de transferencia Marr. Las gráficas en la columna izquierda corresponden al entrenamiento de la red y las gráficas la columna derecha a la prueba. Las marcas de error indican el 95% del intervalo de confianza. Los resultados corresponden al 4-aminofenol.

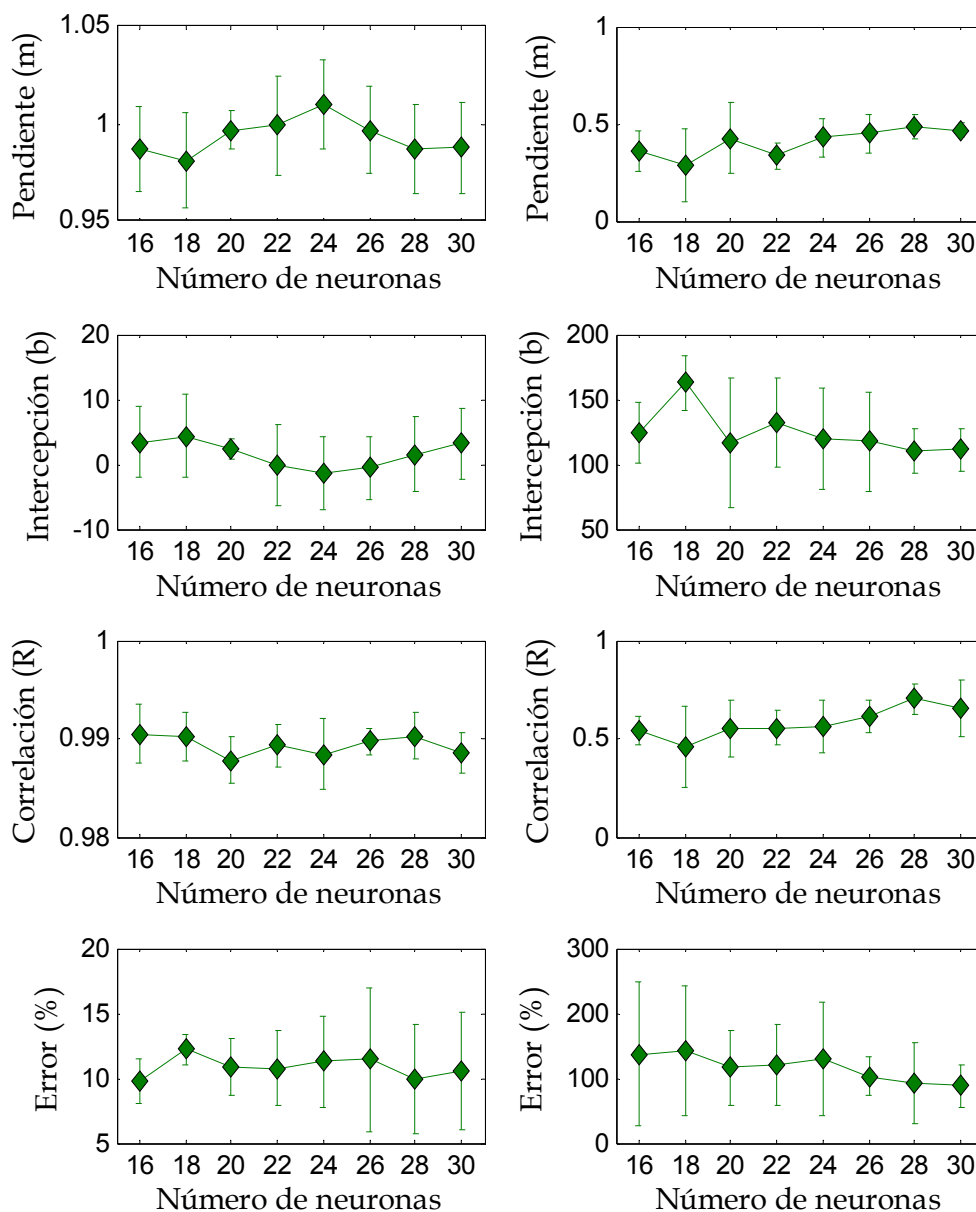


Fig. 7. Promedios de los resultados del análisis de regresión lineal y porcentajes de recuperación obtenidos con la validación cruzada usando una WNN con función de transferencia Marr. Las gráficas en la columna izquierda corresponden al entrenamiento de la red y las gráficas la columna derecha a la prueba. Las marcas de error indican el 95% del intervalo de confianza. Los resultados corresponden al paracetamol.

Las redes neuronales con función de transferencia *Wavelet Halo* tuvieron un comportamiento similar a las redes con función *Wavelet Marr*. La Fig. 8 muestra los valores reales y los obtenidos con una estructura con 20 neuronas, y la Fig. 9 muestra las correspondientes gráficas comparativas obtenidas

mediante regresión lineal. Los entrenamientos con estas redes también alcanzaron el error programado, lo que permitió conseguir valores de correlación de 0.95, 0.98 y 0.995 entre los valores obtenidos con la red y los reales para el AA, 4-Aph y PCT, respectivamente; sin embargo, la capacidad de generalización a los datos de prueba tampoco es buena para este conjunto de redes neuronales. Los valores de correlación que se obtuvieron para las tres salidas de la red fueron 0.376 para el AA, 0.606 para el 4-Aph y 0.436 para el PCT. Estos resultados corresponden al primer entrenamiento realizado con una estructura con 20 neuronas y se presentan en la Tabla 2 junto con el error porcentual y el resto de los parámetros obtenidos del análisis de regresión lineal.

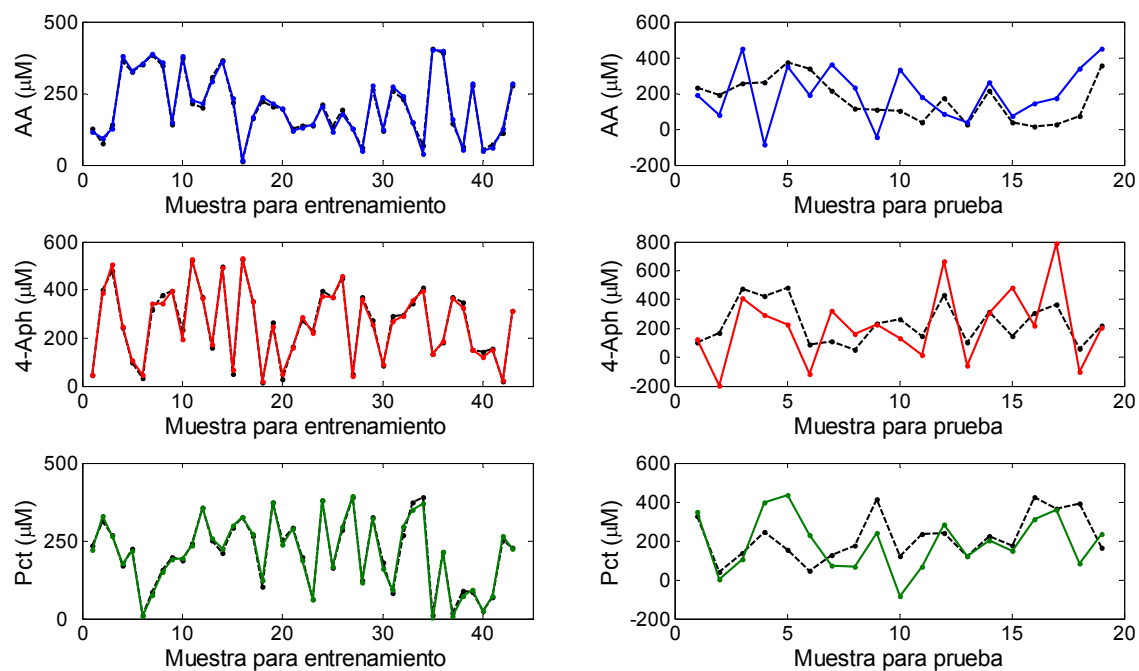


Fig. 8. Gráfica comparativa de resultados obtenidos y reales para los procesos de entrenamiento (columna izquierda) y prueba (columna derecha) obtenidos con una WNN con función Wavelet Halo. Los puntos negros unidos por una línea segmentada son los valores reales, los puntos de colores unidos por una línea continua son los valores obtenidos.

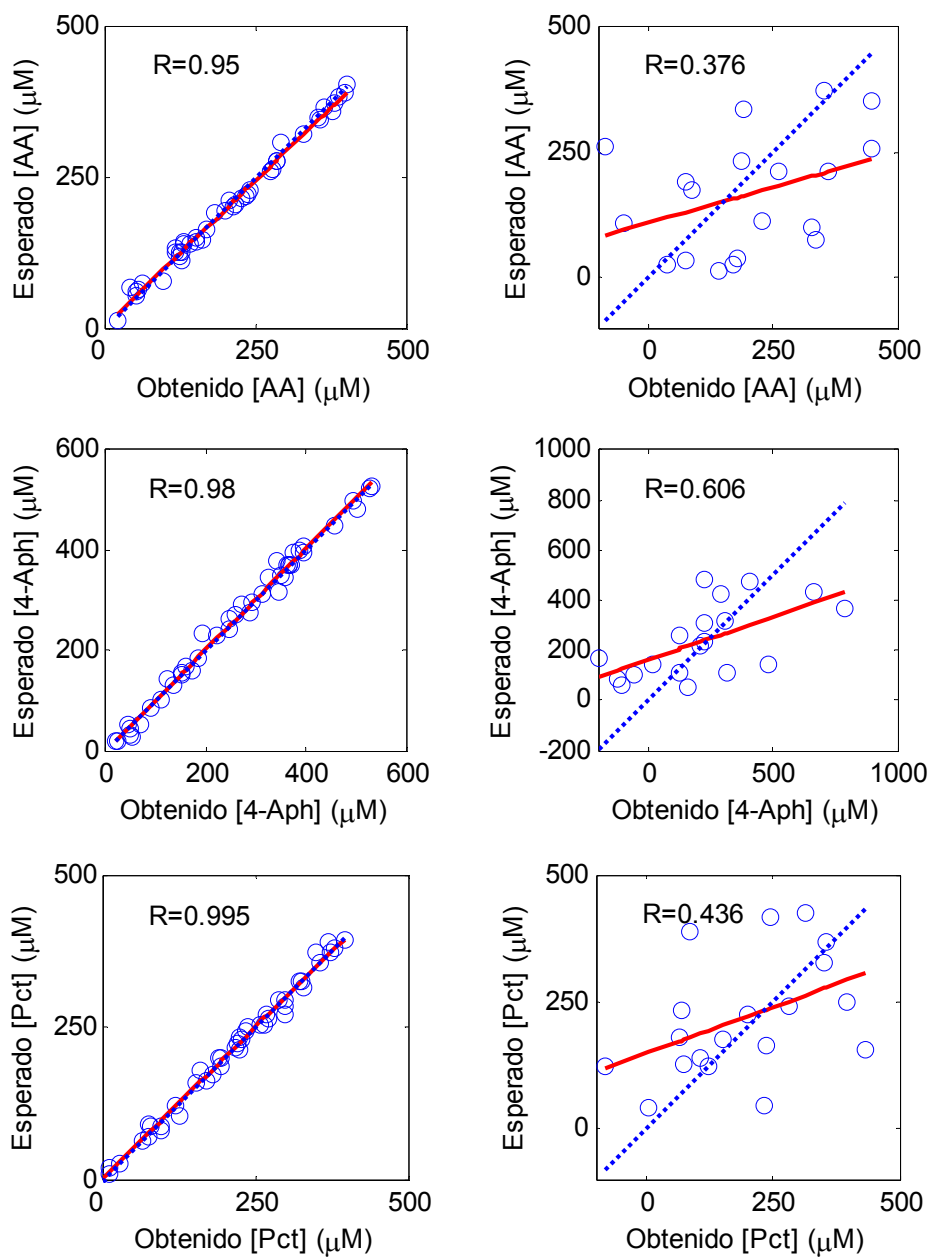


Fig. 9. Gráficas del análisis de regresión lineal entre los resultados reales y los obtenidos con una red neuronal de 20 neuronas con funciones de transferencia Wavelet Halo basadas en la norma Euclidiana. Las gráficas en la columna de la izquierda corresponden al entrenamiento y las de la derecha a la prueba. La línea continua es la regresión que mejor se ajusta y la discontinua la ideal.

Tabla 2. *Coefficientes de regresión lineal y error porcentual obtenidos con el primer entrenamiento hecho con una red neuronal wavelet de 20 neuronas con función de transferencia Wavelet Halo. La tabla contiene los valores obtenidos tanto en entrenamiento como en prueba.*

<i>Analito</i>	<i>m</i>		<i>b</i>		<i>R</i>		<i>Error (%)</i>	
	<i>tr</i>	<i>ts</i>	<i>tr</i>	<i>ts</i>	<i>tr</i>	<i>ts</i>	<i>tr</i>	<i>ts</i>
AA	0.961	0.286	6.005	108.75	0.950	0.376	7.662	177.01
4-Aph	1.008	0.345	-0.490	161.82	0.980	0.606	8.404	104.00
Pct	0.987	0.362	3.790	148.33	0.995	0.436	9.559	70.89

Se volvió a evaluar la influencia que tiene la elección de los registros en el entrenamiento de la red haciendo 5 entrenamientos adicionales de validación cruzada. La elección de las parejas de registro/concentraciones fue igual que en caso anterior. Los promedios obtenidos con los factores de correlación, pendientes, intercepciones y porcentajes de recuperación de los seis entrenamientos realizados se condensan gráficamente en las Figs. 10 a 12, junto con los promedios de las sumas de errores porcentuales.

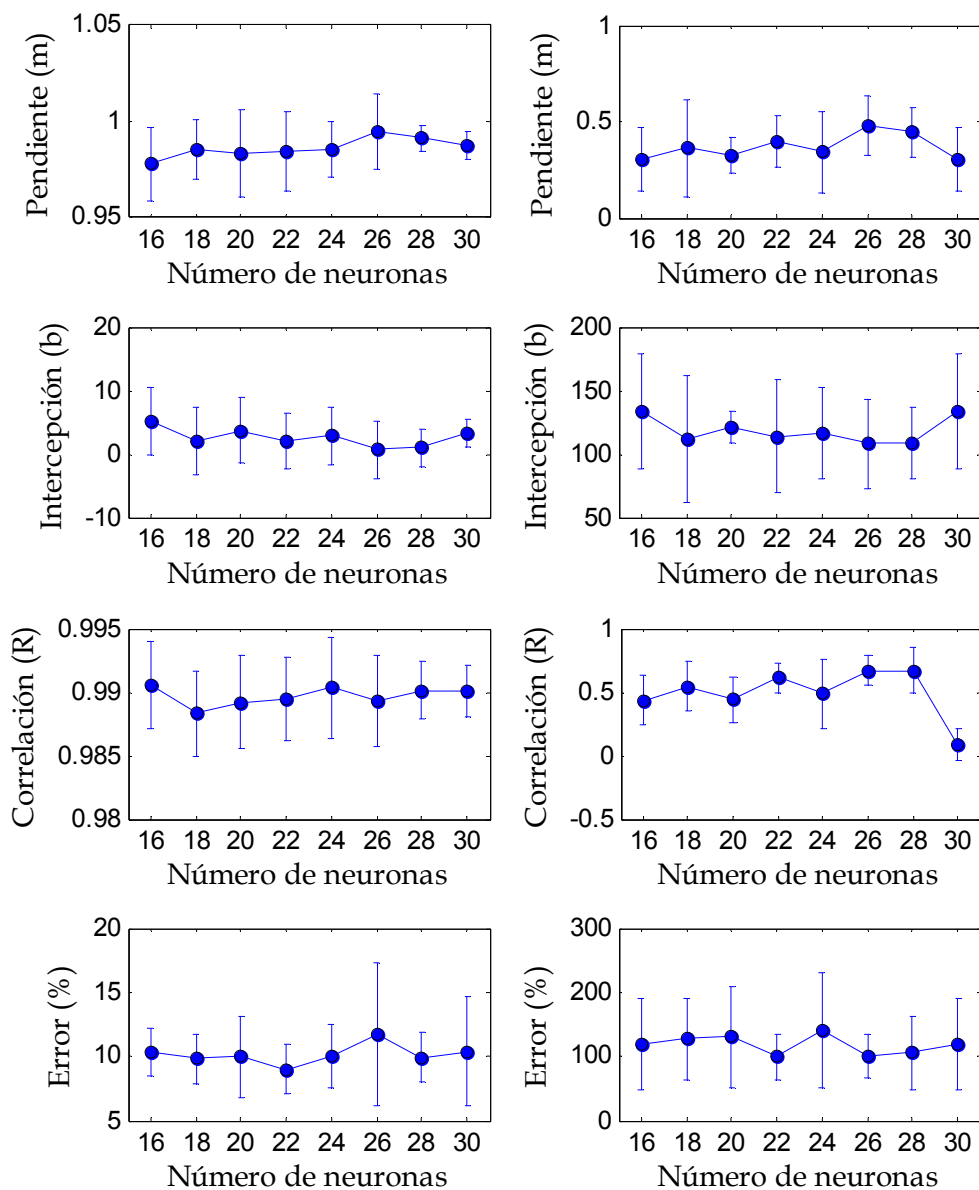


Fig. 10. Promedios de los resultados del análisis de regresión lineal, suma de errores al cuadrado y porcentajes de recuperación obtenidos en la etapa de prueba con la validación cruzada usando una WNN con función de transferencia Halo. Las marcas de error indican el 95% del intervalo de confianza. Los resultados corresponden al ácido ascórbico.

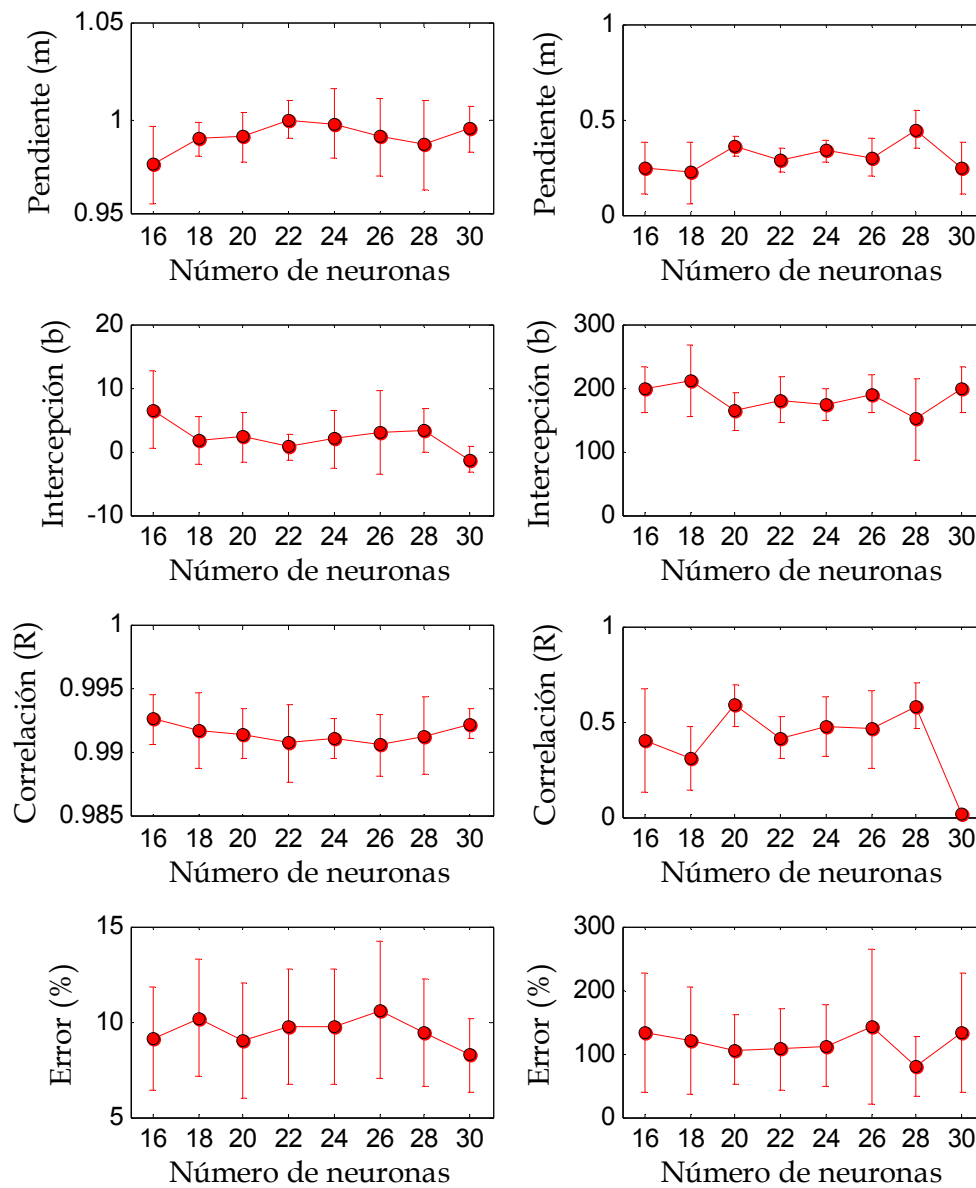


Fig. 11. Promedios de los resultados del análisis de regresión lineal, suma de errores al cuadrado y porcentajes de recuperación obtenidos en la etapa de prueba con la validación cruzada usando una WNN con función de transferencia Halo. Las marcas de error indican el 95% del intervalo de confianza. Los resultados corresponden al 4-aminofenol.

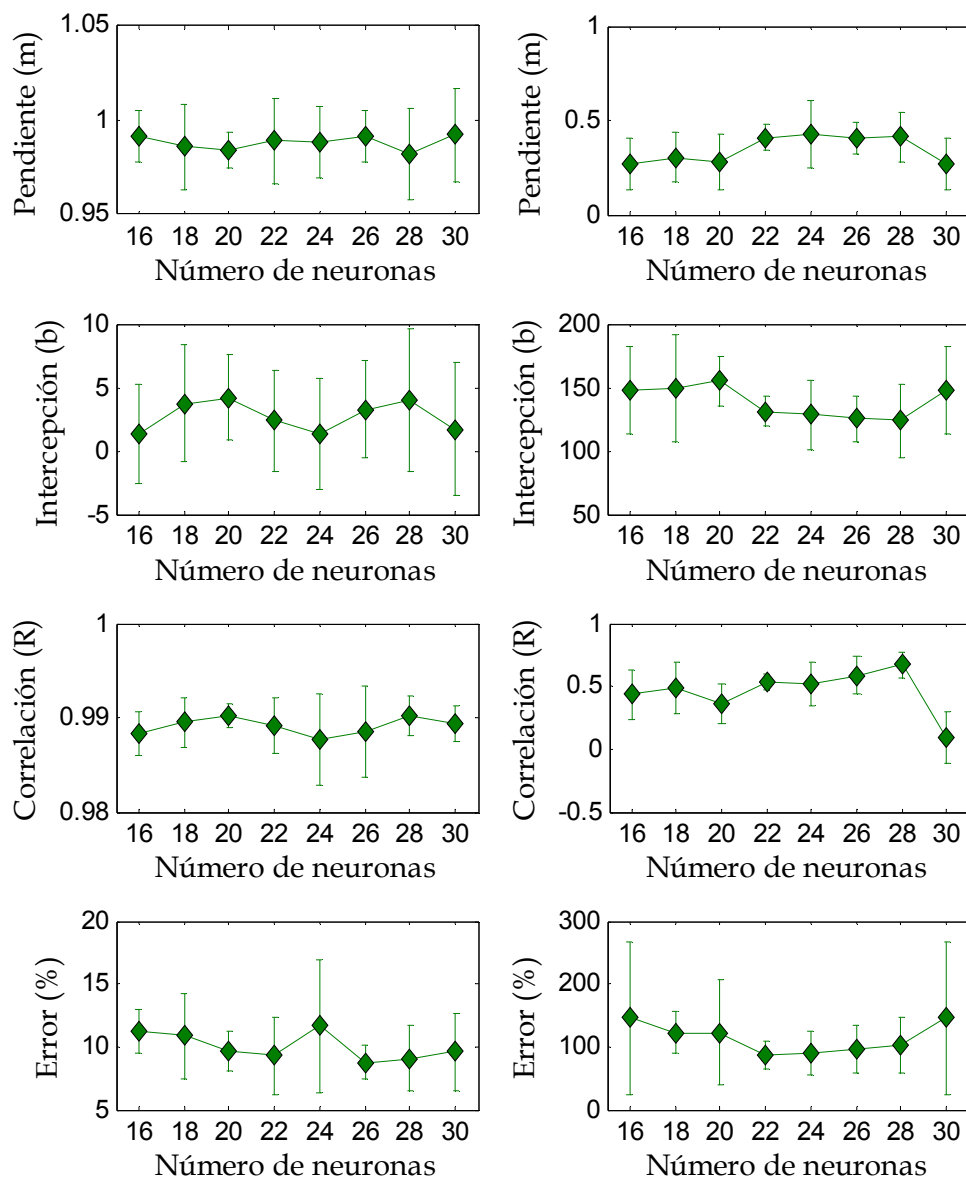


Fig. 12. Promedios de los resultados del análisis de regresión lineal, suma de errores al cuadrado y porcentajes de recuperación obtenidos en la etapa de prueba con la validación cruzada usando una WNN con función de transferencia Halo. Las marcas de error indican el 95% del intervalo de confianza. Los resultados corresponden al paracetamol.

La discrepancia que se observa entre en el buen ajuste de estas redes a datos de entrenamiento y su poca capacidad de generalización a datos de prueba puede deberse a que las funciones de transferencia wavelet basadas en la norma-2 que se usaron en la capa oculta de esta red forman una hipersuperficie que se vuelve muy específica a los valores de entrenamiento,

dejando poca flexibilidad al ajuste de nuevos datos que se presentan a la red entrenada. A pesar de que la función wavelet descrita por la ecuación (3.7) de la Sección 3.2.1 tiene muchos grados de libertad, el resultado de su evaluación es un punto en el hiperespacio al que convergen todos los valores de entrada, lo que reduce las posibilidades de ajuste de la red a menos que se incluyan un número elevado de funciones de este tipo.

8.2 Estructuras de WNNs paralelas en la determinación de compuestos fenólicos para la monitorización de contaminantes en agua

La monitorización de la calidad del agua requiere un estricto control debido a razones medioambientales y a que se usa para el consumo humano. Entre los contaminantes del agua que se precisan determinar en plantas de purificación están los metales pesados, fosfatos, sulfatos y contaminantes orgánicos. Los compuestos fenólicos son un grupo particular de sustancias que requieren especial atención por estar clasificadas como contaminantes orgánicos de alta prioridad. Estos compuestos son producto de varias actividades industriales, deshechos humanos y también fruto de la degradación biológica, por lo que la detección de fenoles en alimentos también puede usarse como un marcador que indica la frescura de los alimentos y bebidas. La extrema toxicidad de algunos de estos compuestos hace que su determinación y cuantificación sea de gran relevancia en el análisis ambiental.

Por las razones anteriormente expuestas, se ha planteado la cuantificación simultánea de mezclas de los compuestos fenólicos o-cresol, p-clorofenol y 4-cloro-3-metilfenol a partir de sus señales de oxidación directa obtenidas con voltamperometría lineal de barrido [49]. Los analitos que se usaron fueron de grado analítico con concentraciones que variaron en los intervalos [0, 40] μM para el o-cresol, [0.5, 80] μM para el p-clorofenol y [1, 50] μM para el 4-cloro-3-metilfenol. Para preparar las soluciones estándares se disolvieron los analitos en un tampón fosfato al 0.1M con un pH de 7.0 mezclado con cloruro potásico al 0.1M, para asegurar una alta conductividad eléctrica. Se usaron dos electrodos de trabajo construidos con una mezcla base de resina epóxica Epotek H77 y grafito. Uno de los electrodos fue

adicionado con una mezcla de AuPd y el otro con Pd. Para completar la celda de medida se usó acero inoxidable como contra-electrodo y se usó un electrodo de Ag/AgCl como electrodo de referencia. El potencial de la técnica voltamperométrica usada se varió desde 0.2V hasta 0.82V con escalones de voltaje de 15mV y una tasa de barrido de 100mV/s, lo que permitió obtener 42 puntos de medición. Estas mediciones se realizaron usando un sistema electroquímico Autolab/PGSTAT 20 (Ecochemie).

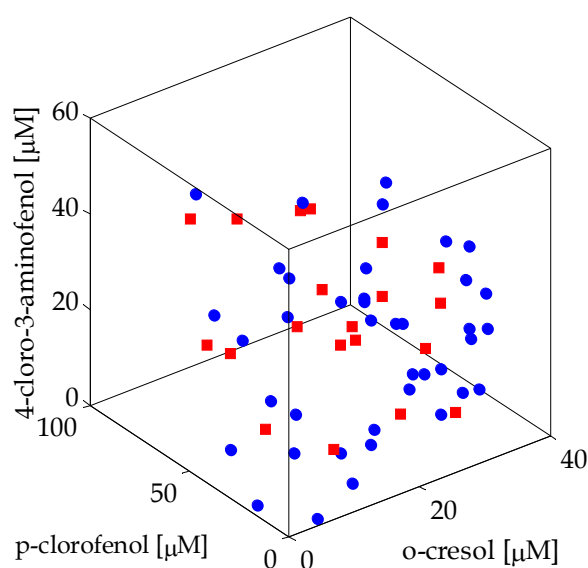


Fig. 13. Distribución espacial de las triadas de analitos usadas en el experimento. La figura de la izquierda muestra el total de triadas, y la de la derecha las separa en elementos usados en la construcción del modelo (puntos azules) y elementos usados en la prueba (puntos rojos).

El número total de estándares que se programaron para ser preparados automáticamente por el sistema SIA fue 60. En la Fig. 13 está graficada la distribución espacial de las 60 triadas de concentraciones para mostrar la falta de tendencias entre las mezclas de analitos. Se usó voltamperometría de barrido lineal y se midió la corriente producida por los 42 escalones de voltaje aplicados a cada una de las 60 soluciones. Al finalizar el experimental se tuvo un conjunto de datos formado por una matriz de concentraciones \mathbf{Y} de tamaño [3, 60] y un tensor \mathbf{X} construido a partir de las dos matrices de intensidades de corriente de tamaño [42, 60], una de ellas obtenida con el electrodo de Pd y la otra con el electrodo de AuPd. Las gráficas con los 60 registros voltamperométricos obtenidos con cada uno de los electrodos se muestran en

la Fig. 14. Las intensidades de corriente mínima y máxima que se obtuvieron en los intervalos de potencial anteriormente descritos fueron $[-0.29, 6.91]$ mA para el electrodo de AuPd y $[-0.098, 7.23]$ mA para el electrodo de Pd.

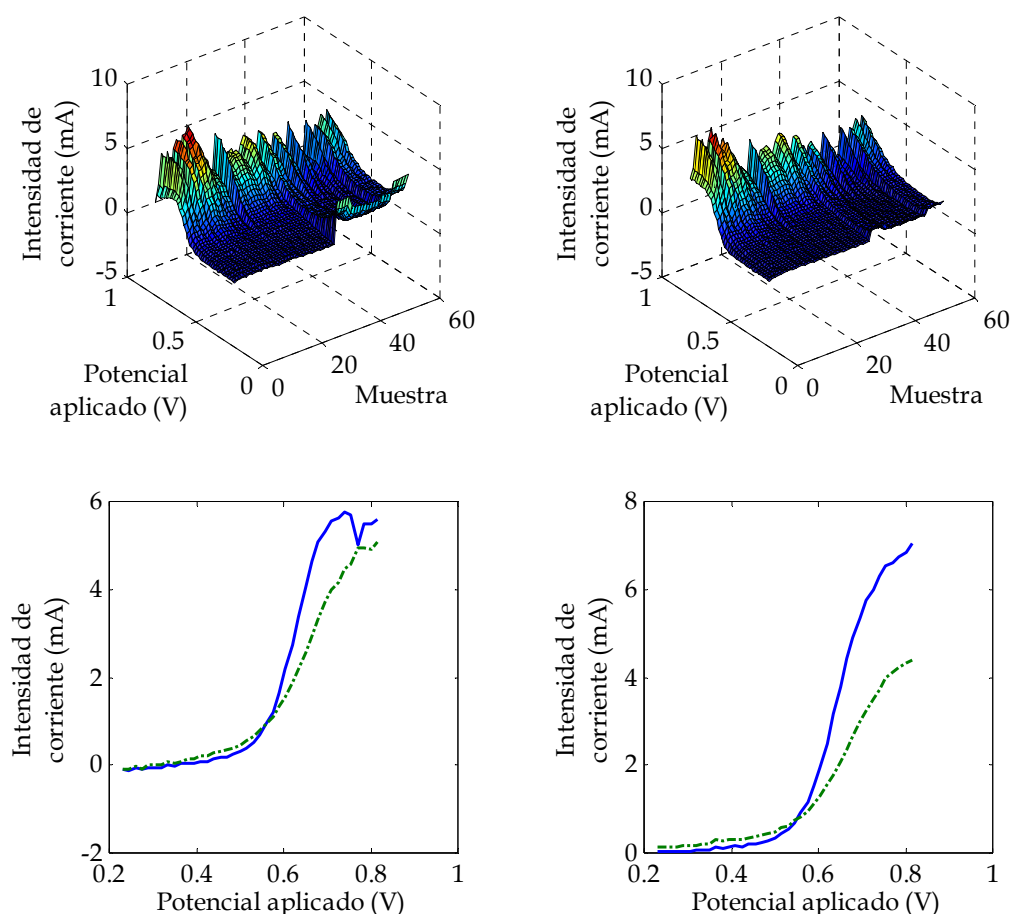


Fig. 14. Las gráficas superiores están formadas por los 60 voltamperogramas adquiridos con los electrodos de Au-Pd (izquierda) y Pd (derecha). Las gráficas inferiores muestran el detalle de dos registros adquiridos con las concentraciones $[31,39; 26,89; 25,00]$ μM .

El modelo de calibración para este trabajo experimental se construyó con la segunda herramienta descrita en el Capítulo 1. Se implementaron estructuras con 2 redes neuronales paralelas usando las funciones de transferencia *Wavelet Marr* y *Halo* basadas en la norma Euclidiana. En cada estructura una de las redes recibió como entrada los voltamperogramas obtenidos con el electrodo de AuPd y la otra los voltamperogramas obtenidos con el electrodo de Pd. Las neuronas ocultas en cada una de las redes

neuronales que formaron una estructura se variaron igualmente en número desde 10 neuronas hasta 20 en intervalos discretos de 2 neuronas; las cotas mínima y máxima en el número de neuronas se propusieron en base a la experiencia del anterior experimento. Las estructuras que finalmente se evaluaron tuvieron 10, 12, 14, 16, 18 y 20 neuronas ocultas por red.

Las salidas de las neuronas ocultas de las dos redes paralelas se hicieron converger en tres neuronas de salida con el objetivo de combinar la información electroquímica de los sensores de AuPd y Pd y poder cuantificar simultáneamente la triada de fenoles bajo estudio.

La repetibilidad en el ajuste de las estructuras usadas en esta sección experimental se verificó de la misma manera que en el caso anterior. Se realizaron 5 entrenamientos adicionales por cada estructura con dos redes paralelas y para cada una de las dos funciones de transferencia probadas. Los parámetros de pendiente (m), intersección de la recta con el eje de ordenadas (b) y correlación (R) obtenidos mediante análisis de regresión lineal entre las salidas de las redes y los valores reales de concentraciones esperados, además del EPAP, permiten conocer la eficiencia de esta estructura de red neuronal propuesta como modelo de calibración.

Para ejemplificar el desempeño conseguido con esta propuesta de solución, en la Fig. 15 se muestran las gráficas de correlación entre los valores reales de concentraciones y las salidas obtenidas para uno de los entrenamientos hechos con la estructura de red neuronal formada con dos redes paralelas con 16 neuronas ocultas en cada una de ellas y función de transferencia *Wavelet Marr*; el resto de los parámetros resultantes del análisis de regresión lineal están en la Tabla 3 junto con el error porcentual de las predicciones. De las gráficas de regresión de la Fig. 15 se observa que el entrenamiento de las redes permitió obtener valores de correlación de 0.997, 0.999 y 0.997 para el o-cresol, p-clorofenol y 4-cloro-3-metilfenol, respectivamente; sin embargo, la generalización de la red a los datos de prueba disminuyó hasta alcanzar valores de correlación de 0.747 para el o-cresol, 0.681 para el p-clorofenol y 0.633 para el 4-cloro-3-metilfenol. Estos resultados no son exclusivos de la estructura con 16 neuronas ocultas por red, las demás estructuras mostraron un comportamiento similar.

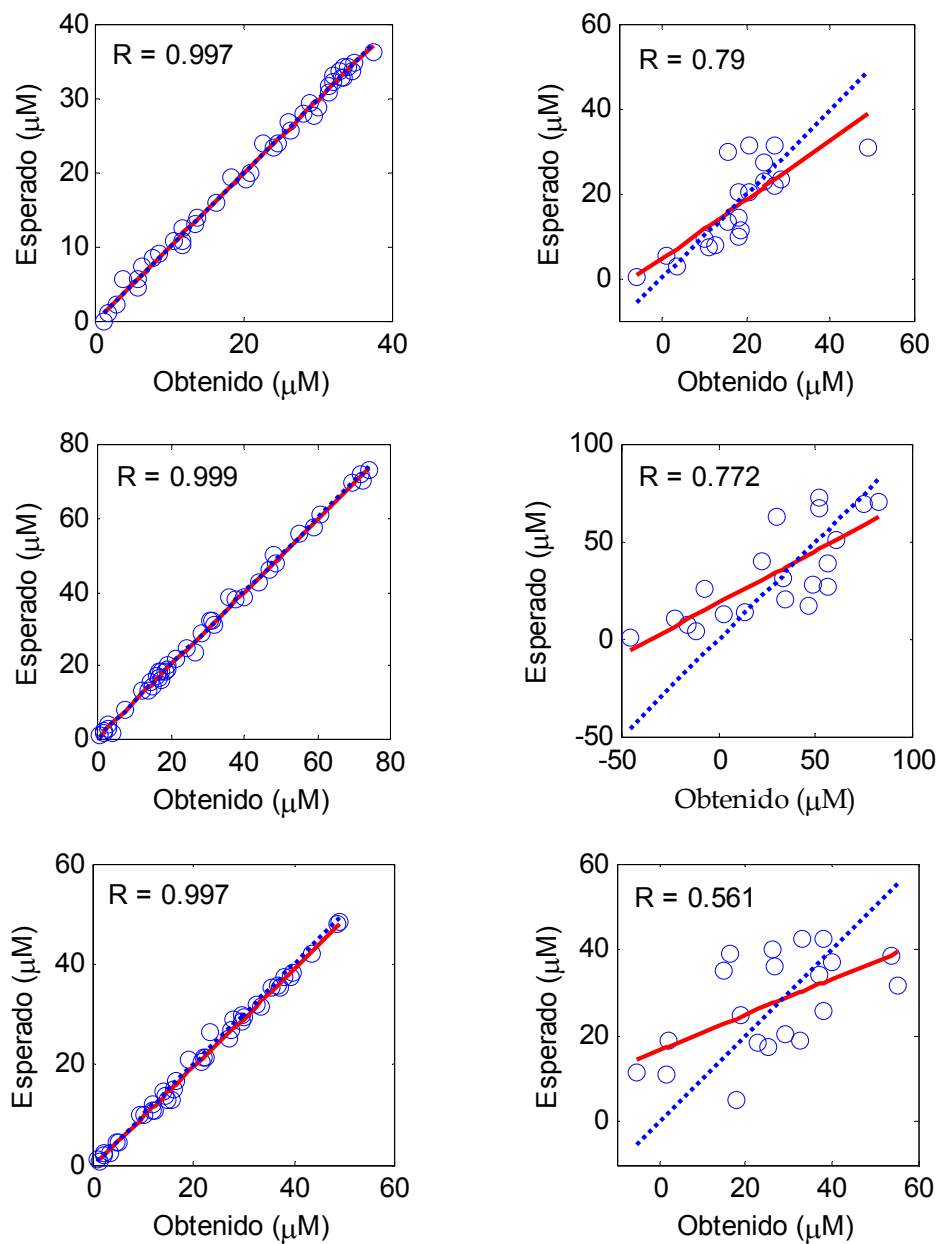


Fig. 15. Gráficas del análisis de regresión lineal entre los resultados reales y los obtenidos con una estructura de dos red neuronales paralelas de 16 neuronas cada una y función de transferencia Wavelet Marr basadas en la norma Euclidiana. Las gráficas en la columna de la izquierda corresponden al entrenamiento y las de la derecha a la prueba. La línea continua es la regresión que mejor se ajusta y la discontinua la ideal.

Tabla 3. Coeficientes de regresión lineal y error porcentual obtenidos con el primer entrenamiento hecho con una red neuronal wavelet de 16 neuronas con función de transferencia Wavelet Halo. La tabla contiene los valores obtenidos tanto en entrenamiento como en prueba.

Analito	<i>m</i>		<i>b</i>		<i>R</i>		Error (%)	
	<i>tr</i>	<i>ts</i>	<i>tr</i>	<i>ts</i>	<i>tr</i>	<i>ts</i>	<i>tr</i>	<i>ts</i>
<i>o-cresol</i>	0.994	0.695	0.082	4.731	0.997	0.790	7.577	55.556
<i>p-clorofenol</i>	0.994	0.535	1.289	18.764	0.998	0.772	10.247	240.44
<i>4-cloro-3-metilfenol</i>	0.979	0.409	-2.1608	16.774	0.996	0.561	8.310	53.380

Debido a la ineficiencia de los primeros resultados obtenidos, se realizaron 5 entrenamientos adicionales de validación cruzada por cada estructura de red neuronal para evaluar si la elección de las parejas registros/concentraciones para formar los subconjuntos de entrenamiento y prueba de la red tiene influencia en la capacidad de generalización de las mismas. La separación de los datos en subconjuntos de entrenamiento y prueba se hizo igual que en experimental anterior. Para el primer modelizado de validación cruzada se dividieron las parejas de registros/concentraciones tomando los registros $[3n-2, 3n, \dots, 60]_{n=1\dots 15}$ para el entrenamiento y los registros $[3n-1, \dots, 60]_{n=1\dots 15}$ para la etapa de prueba. Para el segundo modelizado los registros se dividieron tomando las posiciones $[3n-1, 3n, \dots, 60]_{n=1\dots 15}$ para el entrenamiento y las posiciones $[3n-2, \dots, 60]_{n=1\dots 15}$ para la prueba. Los 3 entrenamientos restantes se hicieron separando aleatoriamente las parejas de registros/concentraciones. Los promedios de los parámetros calculados con el análisis de regresión lineal, los errores porcentuales y las incertidumbres calculadas al 95% del intervalo de confianza obtenidos con los 5 entrenamientos realizados con cada una de las estructuras que se probaron están contenidos gráficamente en las Figs. 16 a 18 para los 3 compuestos fenólicos estudiados.

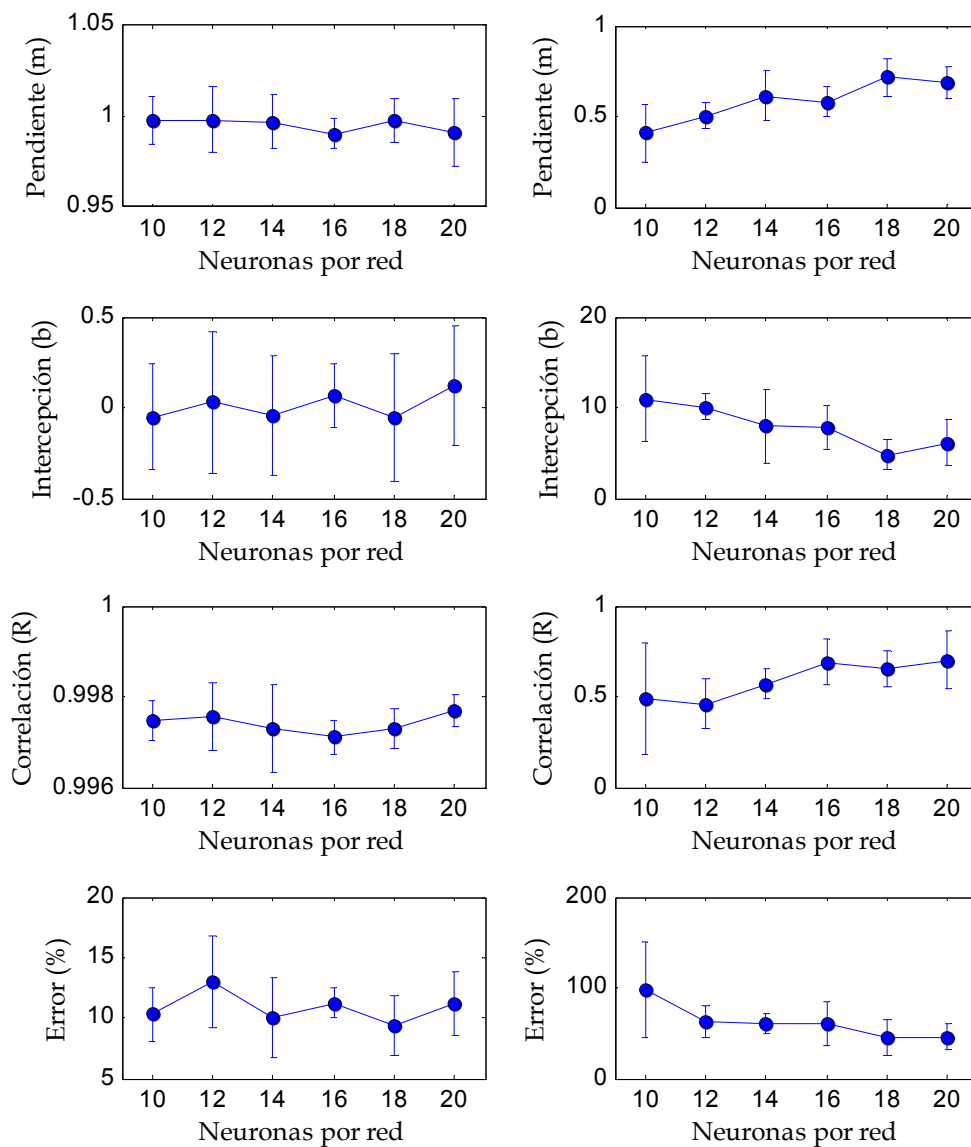


Fig. 16. Valores promedio e incertidumbres calculados al 95% del intervalo de confianza de los parámetros de regresión lineal y error porcentual obtenidos en la etapa de prueba con las estructuras de redes paralelas. Los resultados corresponden al o-cresol. La columna izquierda son los resultados de entrenamiento y la columna derecha son los de prueba.

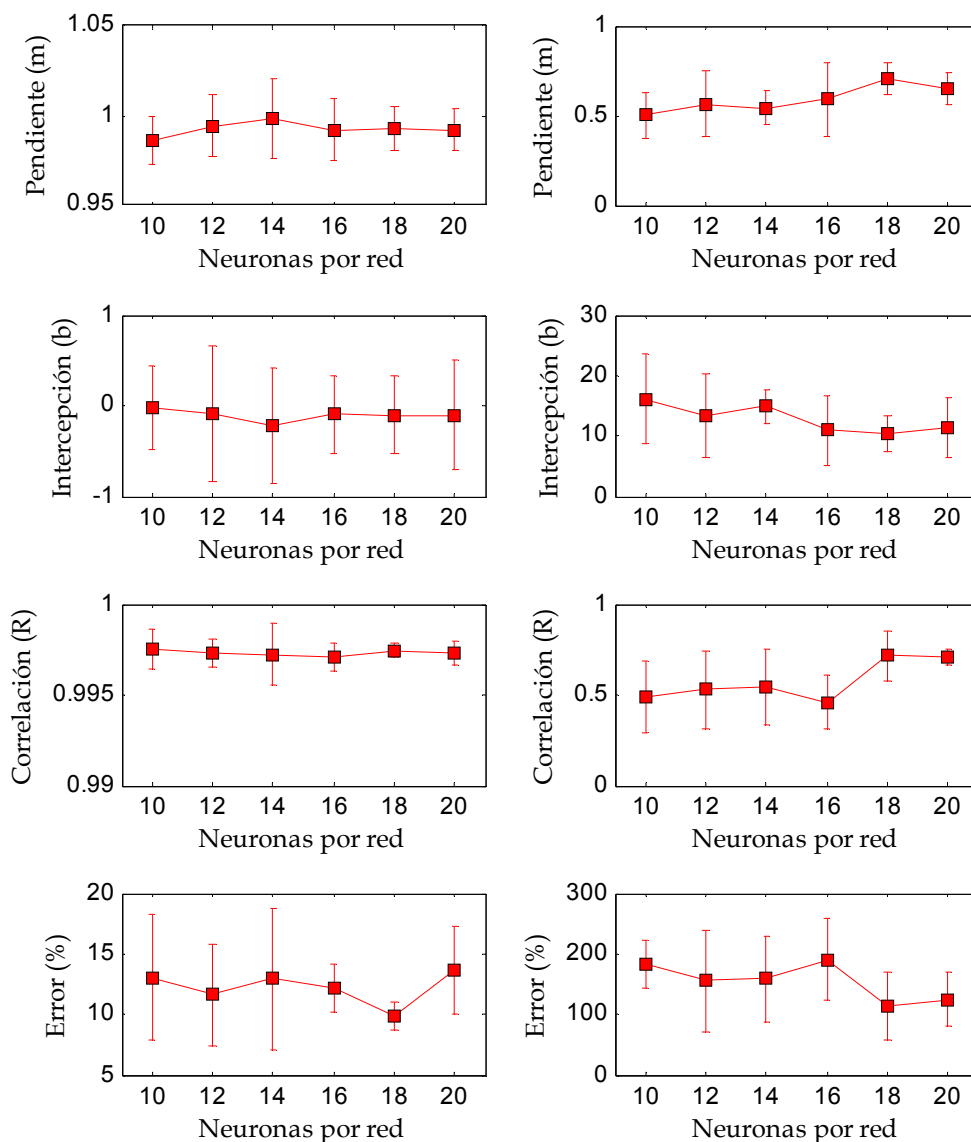


Fig. 17. Valores promedio e incertidumbres calculados al 95% del intervalo de confianza de los parámetros de regresión lineal y error porcentual obtenidos en la etapa de prueba con las estructuras de redes paralelas. Los resultados corresponden al p-clorofenol. La columna izquierda son los resultados de entrenamiento y la columna derecha son los de prueba.

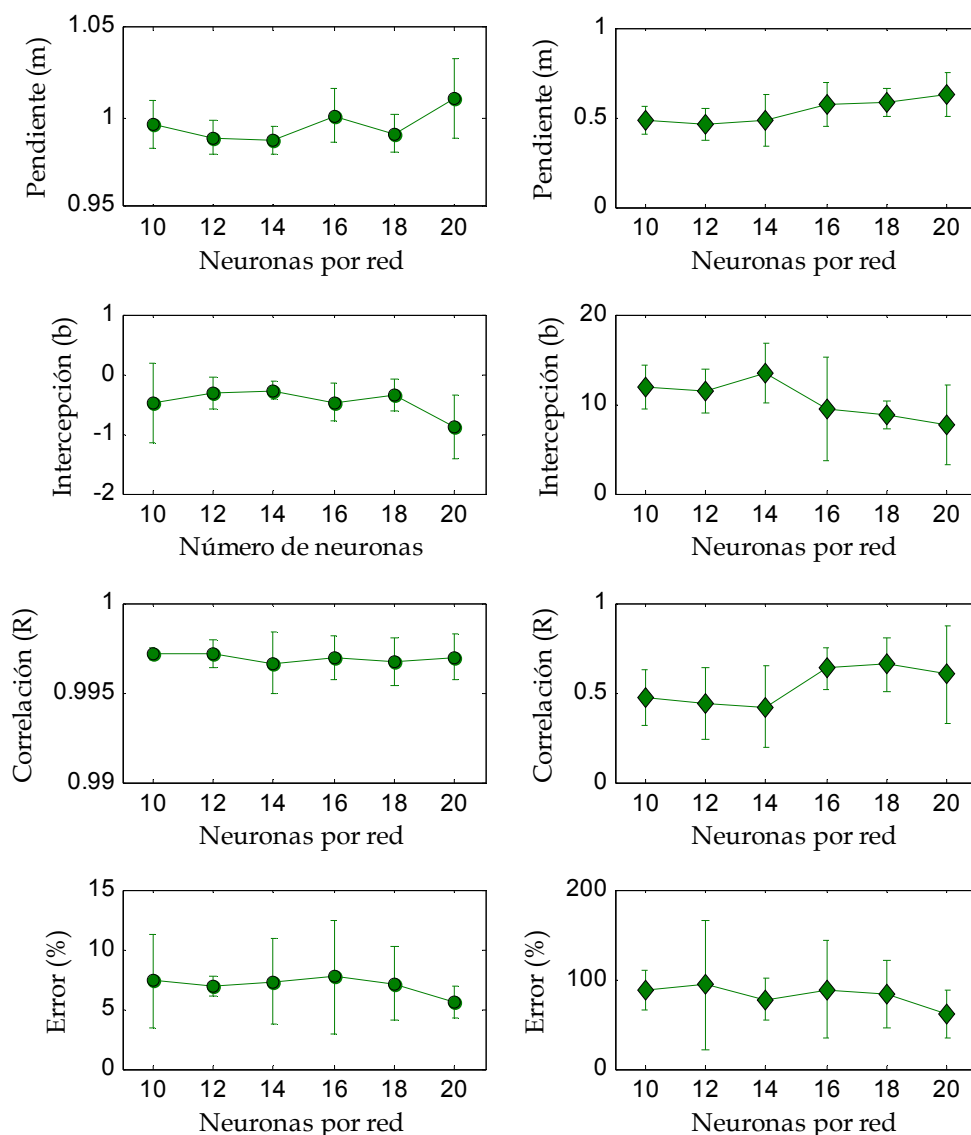


Fig. 18. Valores promedios e incertidumbres calculados al 95% del intervalo de confianza de los parámetros de regresión lineal y error porcentual obtenidos en la etapa de prueba con las estructuras de redes paralelas. Los resultados corresponden al 4-cloro-3-metilfenol. La columna izquierda son los resultados de entrenamiento y la columna derecha son los de prueba.

El comportamiento de la estructura con redes paralelas es similar tanto para las redes con función de transferencia *Wavelet Halo* como para las redes con función de transferencia *Wavelet Marr*. Las correlaciones que se obtuvieron al final del primer entrenamiento hecho con las redes paralelas con función de activación *Wavelet Halo* fueron de 0.95, 0.98 y 0.995 para el o-cresol, p-clorofenol y 4-cloro-3-metilfenol, respectivamente. Estos valores indican

que los entrenamientos se completaron con éxito. Respecto a las correlaciones obtenidas en la etapa de prueba para las tres salidas de la red éstas fueron 0.376 para el AA, 0.606 para el 4-Aph y 0.551 para el PCT. Estos valores, junto con el resto de parámetros del análisis de regresión lineal y el error porcentual obtenidos con una estructura con dos redes paralelas y 16 neuronas ocultas cada una de ellas, están contenidos en la Tabla 4. La influencia que tiene en el desempeño de la red la elección de los subconjuntos de entrenamiento y prueba se volvió a evaluar haciendo 5 entrenamientos adicionales de validación cruzada por cada una de las estructuras propuestas. La selección de las parejas de registro/concentraciones para formar los subconjuntos de entrenamiento y prueba fue igual que en caso anterior. Los promedios de los parámetros del análisis de regresión lineal junto con los valores de error porcentual que se obtuvieron entre los valores reales de las tres concentraciones y los obtenidos con las redes se muestran gráficamente en las Figs. 10 a 12.

Tabla 4. Coeficientes de regresión lineal y error porcentual obtenidos con el primer entrenamiento hecho con una red neuronal wavelet de 16 neuronas con función de transferencia Wavelet Halo. La tabla contiene los valores obtenidos tanto en entrenamiento como en prueba.

Analito	m		b		R		Error (%)	
	tr	ts	tr	ts	tr	ts	tr	ts
<i>o-cresol</i>	0.993	0.538	-0.137	7.577	0.996	0.629	7.883	99.381
<i>p-clorofenol</i>	0.975	0.574	1.496	15.141	0.996	0.660	12.623	207.76
<i>4-cloro-3-metilfenol</i>	1.002	0.411	-1.725	16.935	0.998	0.387	8.204	49.69

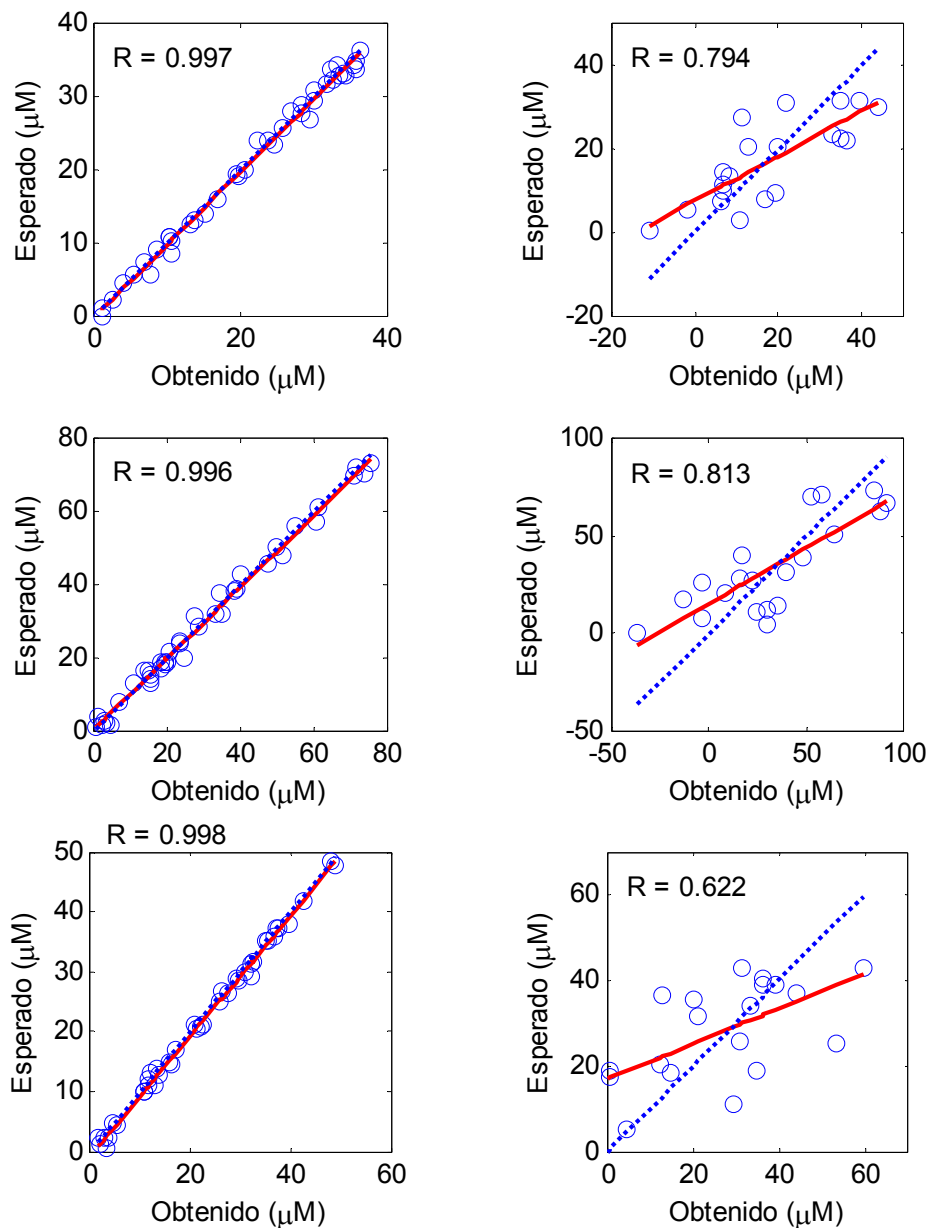


Fig. 19. Gráficas del análisis de regresión lineal entre los resultados reales y los obtenidos con una estructura de dos red neuronales paralelas de 16 neuronas cada una y función de transferencia Wavelet Halo basada en la norma Euclidiana. Las gráficas en la columna de la izquierda corresponden al entrenamiento y las de la derecha a la prueba. La línea continua es la regresión que mejor se ajusta y la discontinua la ideal. Las gráficas de la primera fila corresponden al o-cresol, las de la línea del medio al p-clorofenol y las de la línea inferior al 4-cloro-3-metilfenol.

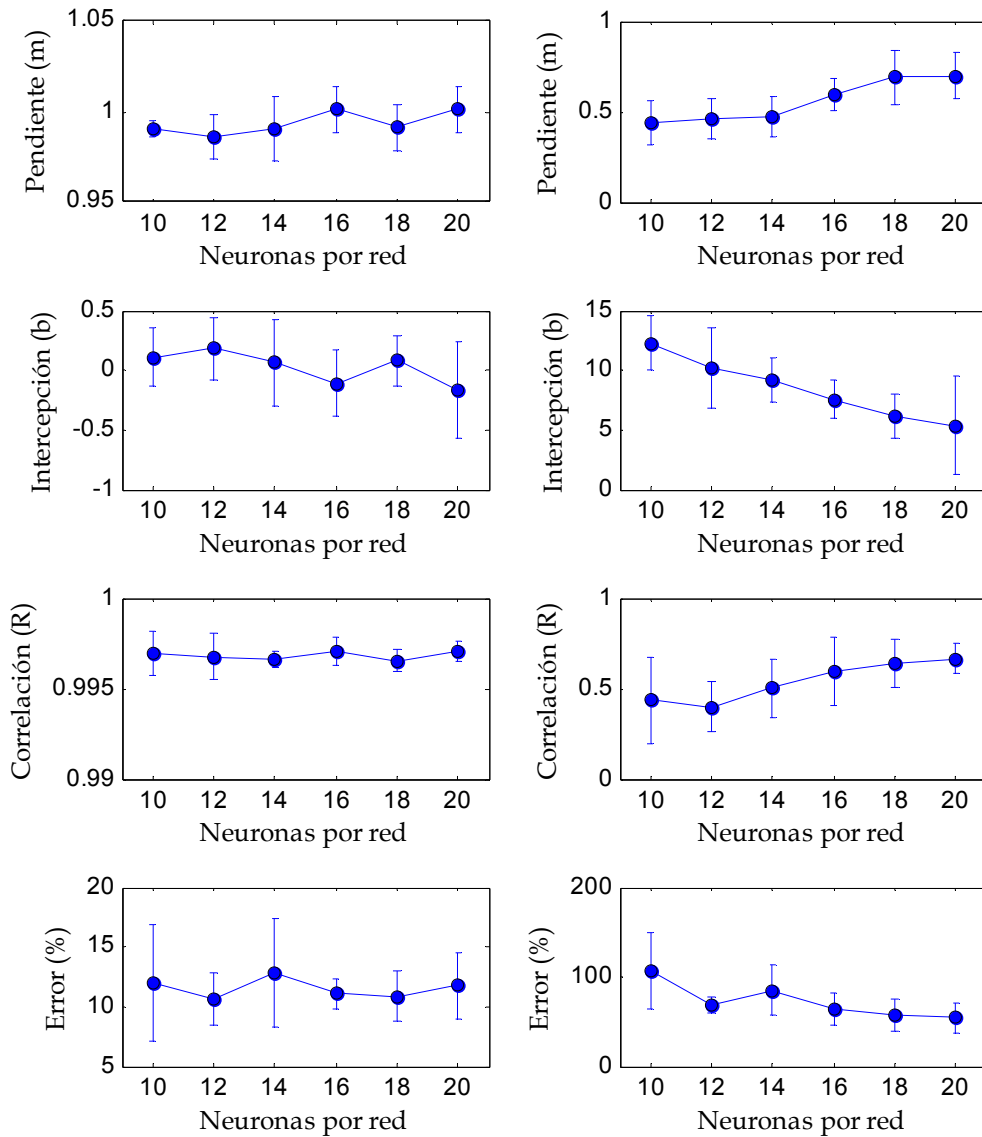


Fig. 20. Valores promedios e incertidumbres calculados al 95% del intervalo de confianza de los parámetros de regresión lineal y error porcentual obtenidos en la etapa de prueba con las estructuras de redes paralelas. Los resultados corresponden al o-cresol. La columna izquierda son los resultados de entrenamiento y la columna derecha son los de prueba.

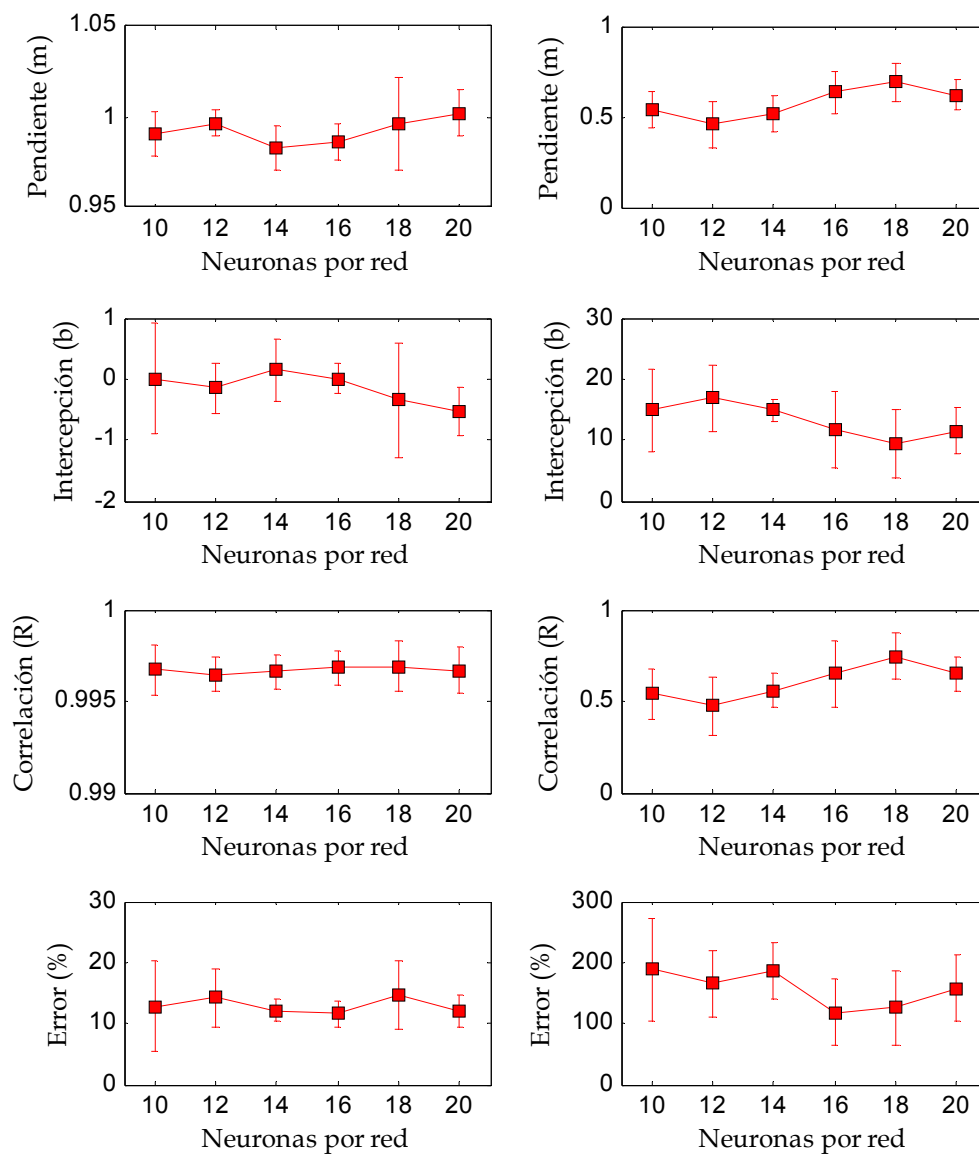


Fig. 21. Valores promedios e incertidumbres calculados al 95% del intervalo de confianza de los parámetros de regresión lineal y error porcentual obtenidos en la etapa de prueba con las estructuras de redes paralelas. Los resultados corresponden al *p*-clorofenol. La columna izquierda son los resultados de entrenamiento y la columna derecha son los de prueba.

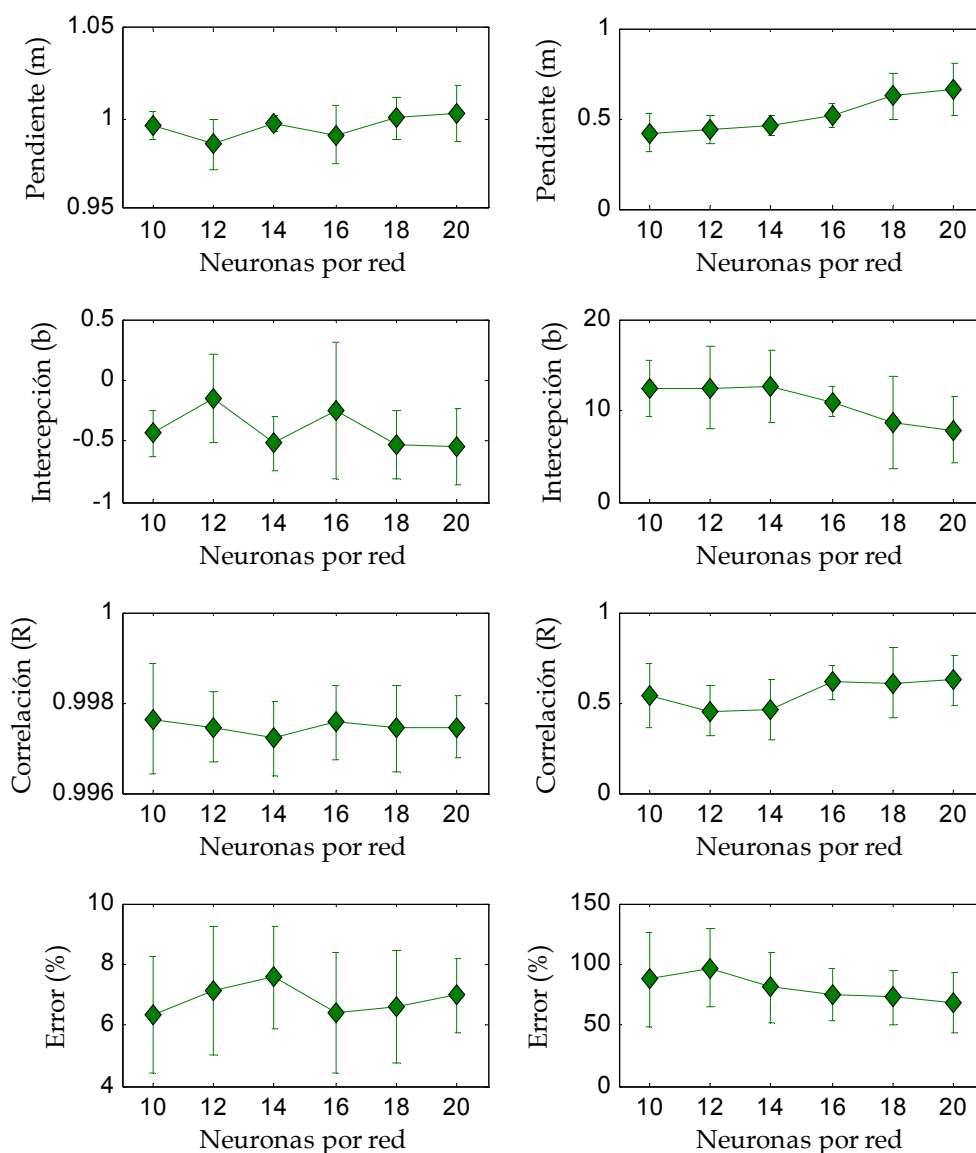


Fig. 22. Valores promedio e incertidumbres calculados al 95% del intervalo de confianza de los parámetros de regresión lineal y error porcentual obtenidos en la etapa de prueba con las estructuras de redes paralelas. Los resultados corresponden al 4-cloro-3-metilfenol. La columna izquierda son los resultados de entrenamiento y la columna derecha son los de prueba.

La razón por la que esta estructura propuesta tiene un comportamiento similar a la anterior también se justifica por el tipo de función propuesta. La hipersuperficie descrita por la función de transferencia wavelet basada en la norma-2 se vuelve muy específica a los valores de entrenamiento, dejando poca flexibilidad al ajuste de nuevos datos que se presentan a la red entrenada. El uso de redes paralelas no permitió mejorar la capacidad de generalización

porque el efecto final de la estructura es la superposición de hipersuperficies que han ajustado a pocos puntos del espacio descrito por la triada de concentraciones.

8.3 Redes neuronales Wavelet basadas en el producto tensorial

Para probar las WNNs con función de activación multi-dimensionales basadas en el producto tensorial de funciones wavelet mono-dimensionales, se usó la base de datos obtenida con el conjunto de compuestos oxidables de la Sección 8.1, donde ya se ha descrito el desarrollo experimental, por lo que a continuación sólo describiremos la estructura de la red, el proceso de evaluación y los resultados obtenidos.

Se preservó la división del conjunto formado por predictores \mathbf{X} y respuestas \mathbf{Y} en dos subconjuntos para los propósitos de entrenamiento y prueba de la red. Basándonos en anteriores experiencias con redes neuronales construidas con funciones Wavelet implementadas con el producto tensorial, se construyeron estructuras con 3 y 5 neuronas ocultas y tres neuronas de salida para la predicción simultánea de los 3 compuestos fenólicos. No se probaron estructuras de mayor dimensión en la capa oculta porque la dimensión de los datos de entrada (101 puntos por vector) y el proceso de entrenamiento ralentizaban la minimización de error. La inicialización de los parámetros de la red fue aleatoria, los subconjuntos de entrenamiento y prueba se normalizaron al intervalo $[-1, 1]$ y el error a alcanzar en entrenamiento se relajó y estableció en un máximo del 15% por cada analito.

Sólo se realizaron 5 entrenamientos por cada estructura debido al elevado tiempo de procesamiento requerido y a que los resultados obtenidos con el primer entrenamiento fueron satisfactorios. Para el primer modelizado se dividieron las parejas de registros/concentraciones tomando las parejas $[3n-2, 3n-1, \dots, 60]_{n=1\dots 15}$ para el entrenamiento y las parejas $[3n, \dots, 60]_{n=1\dots 15}$ para la prueba. Para el segundo modelizado los registros se dividieron tomando las posiciones $[3n-2, 3n, \dots, 60]_{n=1\dots 15}$ para el

entrenamiento y las posiciones $[3n-1, \dots, 60]_{n=1\dots 15}$ para la prueba. El tercer modelizado se realizó tomando las parejas con posiciones $[3n-1, 3n, \dots, 60]_{n=1\dots 15}$ para el entrenamiento y las parejas con posiciones $[3n-2, \dots, 60]_{n=1\dots 15}$ para la prueba. Los dos entrenamientos restantes se hicieron tomando dos terceras partes aleatorias de las matrices **X** y **Y** para entrenamiento y dejando la tercera parte restante para prueba.

Las tablas 5 y 6 junto con las figuras 23 a 25 corresponden a las redes con función de transferencia Wavelet Marr. La Tabla 5 condensa los resultados de los análisis de regresión y la evaluación del error porcentual para los 5 entrenamientos del modelo con 3 neuronas ocultas, y la Tabla 6 condensa los correspondientes resultados para el modelo con 5 neuronas ocultas. La Fig. 23 muestra visualmente la información contenida en estas dos tablas. En las Figs. 24 y 25 se aprecian las gráficas comparativas entre los valores esperados y los obtenidos con las redes con 3 y 5 neuronas ocultas, respectivamente.

Los parámetros de regresión lineal junto con los porcentajes de error obtenidos con los valores esperados de concentraciones y las salidas de las redes construidas con la función de transferencia Wavelet Morlet se condensan en las tablas 7 y 8. La Fig. 26 es la representación visual de estos parámetros y las Figs. 27 y 28 son las correspondientes gráficas comparativas.

Tabla 5. Coeficientes de regresión lineal y error porcentual obtenidos con los entrenamientos hechos con una red neuronal wavelet de 3 neuronas con función de transferencia tensorial Wavelet Marr. La tabla contiene los valores obtenidos tanto en entrenamiento como en prueba.

Analito	m		b		R		Error (%)	
	tr	ts	tr	ts	tr	ts	tr	ts
AA	1.010 ± 0.026	0.839 ± 0.057	2.324 ± 14.396	39.122 ± 9.414	0.988 ± 0.005	0.932 ± 0.017	19.644 ± 3.411	15.572 ± 0.883
4-Aph	0.969 ± 0.027	0.914 ± 0.039	2.467 ± 13.598	23.478 ± 14.951	0.986 ± 0.003	0.969 ± 0.014	18.659 ± 2.002	23.444 ± 2.478
Pct	0.930 ± 0.040	0.970 ± 0.028	21.586 ± 6.641	12.755 ± 7.337	0.972 ± 0.011	0.985 ± 0.010	15.088 ± 0.860	27.284 ± 4.327

Tabla 6. Coeficientes de regresión lineal y error porcentual obtenidos con los entrenamientos hechos con una red neuronal wavelet de 5 neuronas con función de transferencia tensorial Wavelet Marr. La tabla contiene los valores obtenidos tanto en entrenamiento como en prueba.

Analito	m		b		R		Error (%)	
	tr	ts	tr	ts	tr	ts	tr	ts
AA	1.001 ± 0.020	0.929 ± 0.048	-5.204 ± 7.201	17.902 ± 12.423	0.989 ± 0.006	0.959 ± 0.021	16.641 ± 2.454	10.713 ± 2.238
4-Aph	0.973 ± 0.029	0.956 ± 0.019	1.822 ± 15.541	13.101 ± 7.643	0.988 ± 0.003	0.981 ± 0.010	15.623 ± 1.448	20.522 ± 5.909
Pct	0.955 ± 0.013	0.993 ± 0.060	10.563 ± 13.866	5.992 ± 18.237	0.977 ± 0.009	0.986 ± 0.010	13.687 ± 0.836	29.636 ± 3.061

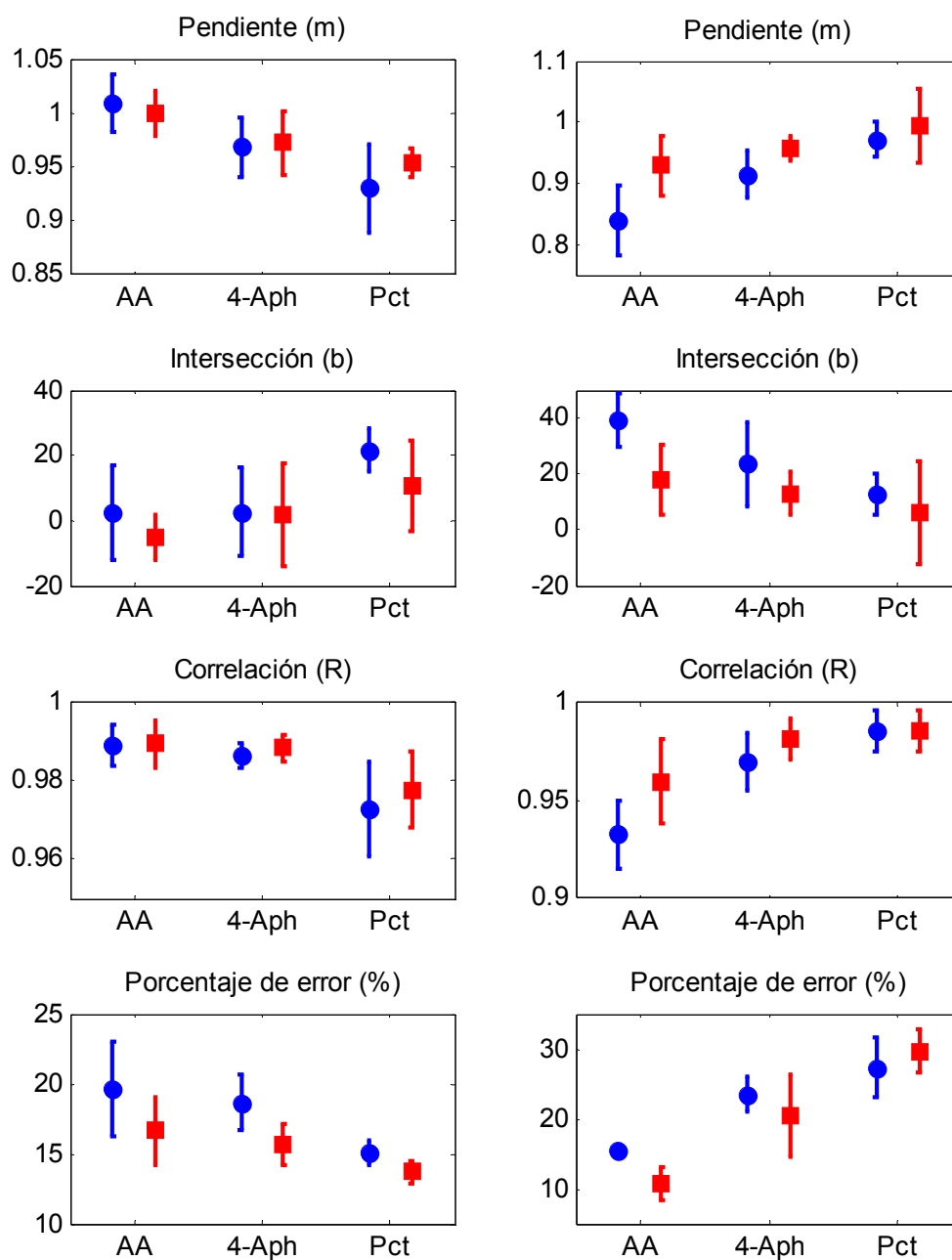


Fig. 23. Representación visual de la información contenida en las tablas 5 y 6. Los marcadores circulares corresponden a la red con 3 neuronas, los marcadores cuadrados a la red con 5 neuronas. Las gráficas de la columna izquierda son los resultados obtenidos en entrenamiento y las de la columna derecha son los resultados obtenidos en prueba.

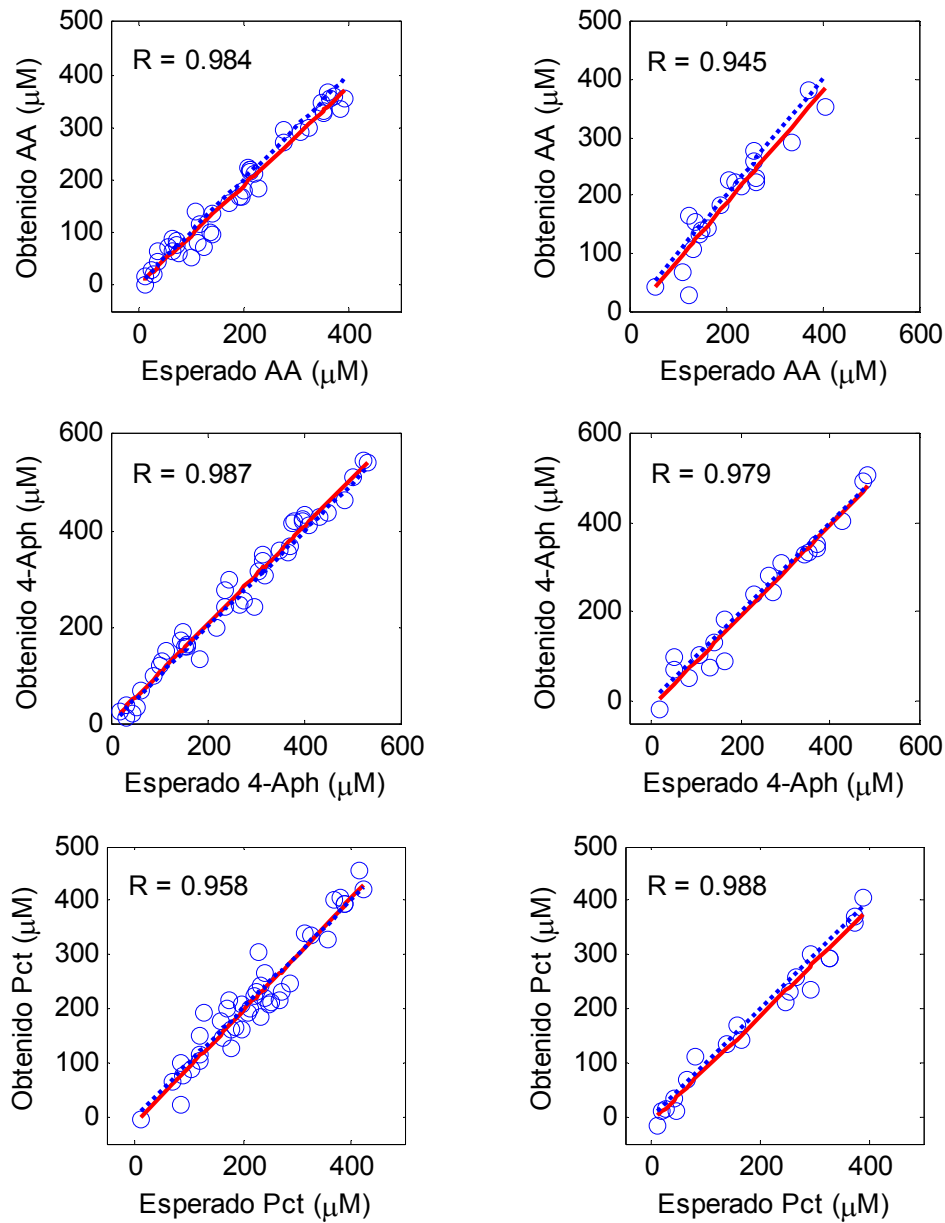


Fig. 24. Gráficas comparativas de regresión lineal entre los valores esperados y los obtenidos con una WNN con función Marr basada en el producto tensorial. Los resultados corresponden a una red con 3 neuronas ocultas. Las gráficas de la columna derecha son los resultados obtenidos en entrenamiento y las gráficas de la columna izquierda son los resultados obtenidos en prueba.

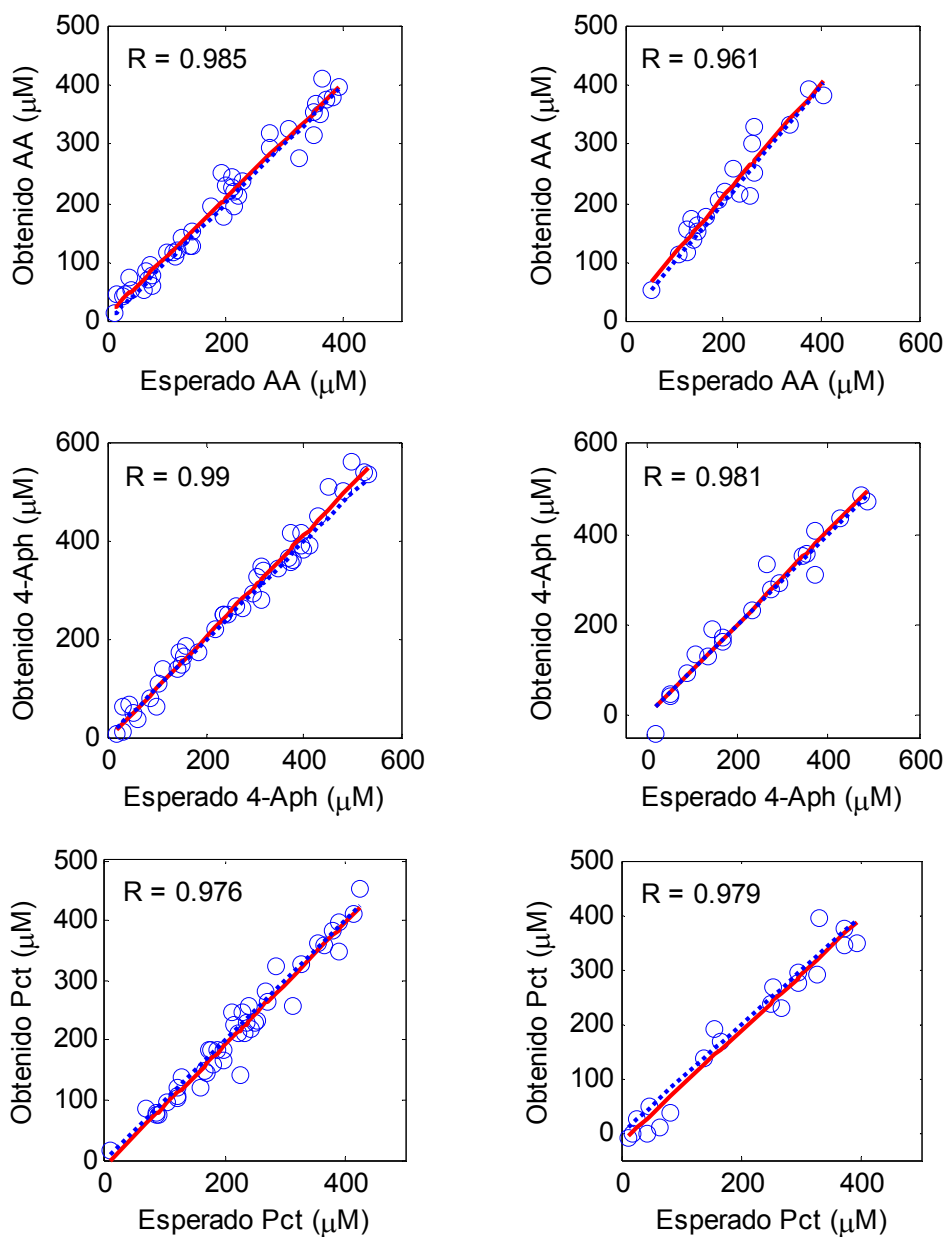


Fig. 25. Gráficas comparativas de regresión lineal entre los valores esperados y los obtenidos con una WNN con función Marr basada en el producto tensorial. Los resultados corresponden a una red con 5 neuronas ocultas. Las gráficas de la columna derecha son los resultados obtenidos en entrenamiento y las gráficas de la columna izquierda son los resultados obtenidos en prueba.

Tabla 7. Coeficientes de regresión lineal y error porcentual obtenidos con los entrenamientos hechos con una red neuronal wavelet de 3 neuronas con función de transferencia tensorial Wavelet Morlet. La tabla contiene los valores obtenidos tanto en entrenamiento como en prueba.

Analito	m		b		R		Error (%)	
	tr	ts	tr	ts	tr	ts	tr	ts
AA	1.001 ± 0.021	0.873 ± 0.087	4.413 ± 12.809	31.872 ± 14.129	0.988 ± 0.005	0.933 ± 0.011	20.363 ± 3.651	14.755 ± 1.467
4-Aph	0.966 ± 0.031	0.922 ± 0.051	3.219 ± 13.379	20.131 ± 15.564	0.983 ± 0.007	0.967 ± 0.013	18.984 ± 4.302	29.097 ± 15.301
Pct	0.941 ± 0.047	0.971 ± 0.080	19.518 ± 11.584	12.816 ± 19.158	0.969 ± 0.016	0.980 ± 0.015	16.855 ± 6.298	37.09 ± 19.39

Tabla 8. Coeficientes de regresión lineal y error porcentual obtenidos con los entrenamientos hechos con una red neuronal wavelet de 5 neuronas con función de transferencia tensorial Wavelet Morlet. La tabla contiene los valores obtenidos tanto en entrenamiento como en prueba.

Analito	m		b		R		Error (%)	
	tr	ts	tr	ts	tr	ts	tr	ts
AA	0.992 ± 0.027	0.913 ± 0.057	-3.739 ± 8.021	20.846 ± 14.211	0.989 ± 0.008	0.920 ± 0.067	20.142 ± 6.688	13.995 ± 3.963
4-Aph	0.961 ± 0.016	0.929 ± 0.066	3.399 ± 15.801	17.177 ± 19.891	0.989 ± 0.005	0.971 ± 0.019	15.58 ± 4.348	23.289 ± 10.598
Pct	0.944 ± 0.037	0.958 ± 0.021	11.802 ± 14.797	11.579 ± 9.076	0.983 ± 0.009	0.969 ± 0.049	12.192 ± 2.390	34.576 ± 7.473

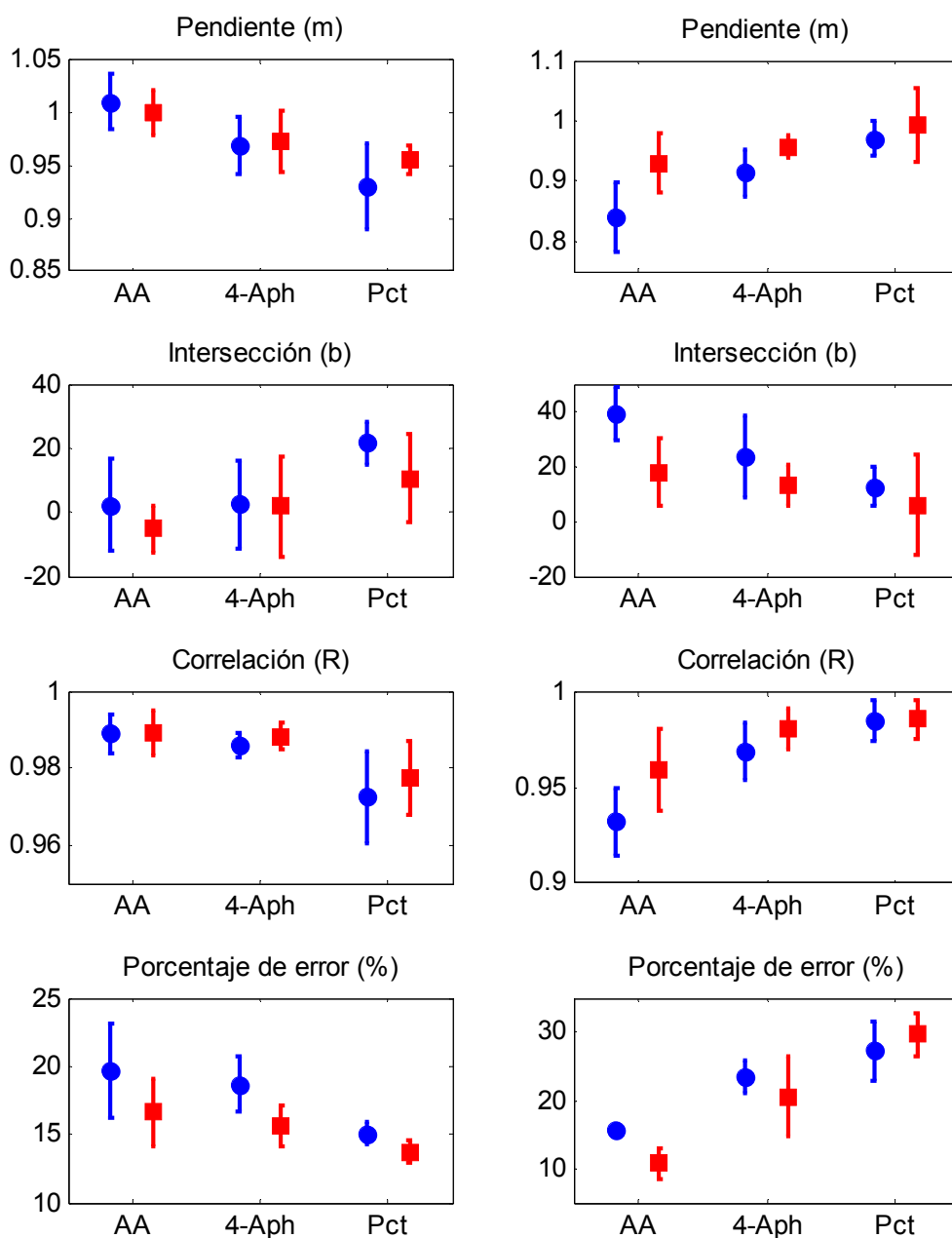


Fig. 26. Representación visual de la información contenida en las tablas 7 y 8. Los marcadores circulares corresponden a la red con 3 neuronas, los marcadores cuadrados a la red con 5 neuronas. Las gráficas de la columna izquierda son los resultados obtenidos en entrenamiento y las de la columna derecha son los resultados obtenidos en prueba.

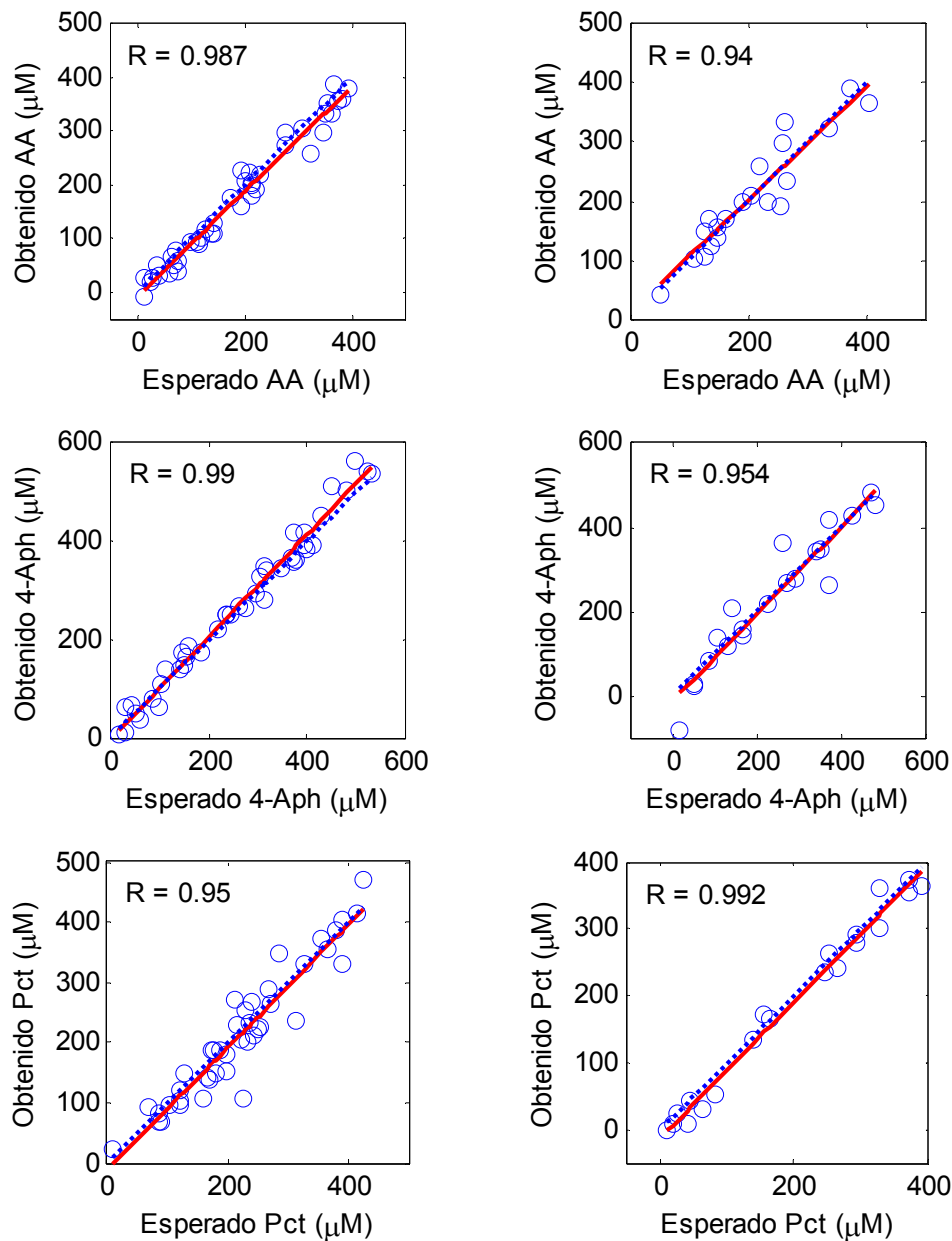


Fig. 27. Gráficas comparativas de regresión lineal entre los valores esperados y los obtenidos con una WNN con función Morlet basada en el producto tensorial. Los resultados corresponden a una red con 3 neuronas ocultas. Las gráficas de la columna derecha son los resultados obtenidos en entrenamiento y las gráficas de la columna izquierda son los resultados obtenidos en prueba.

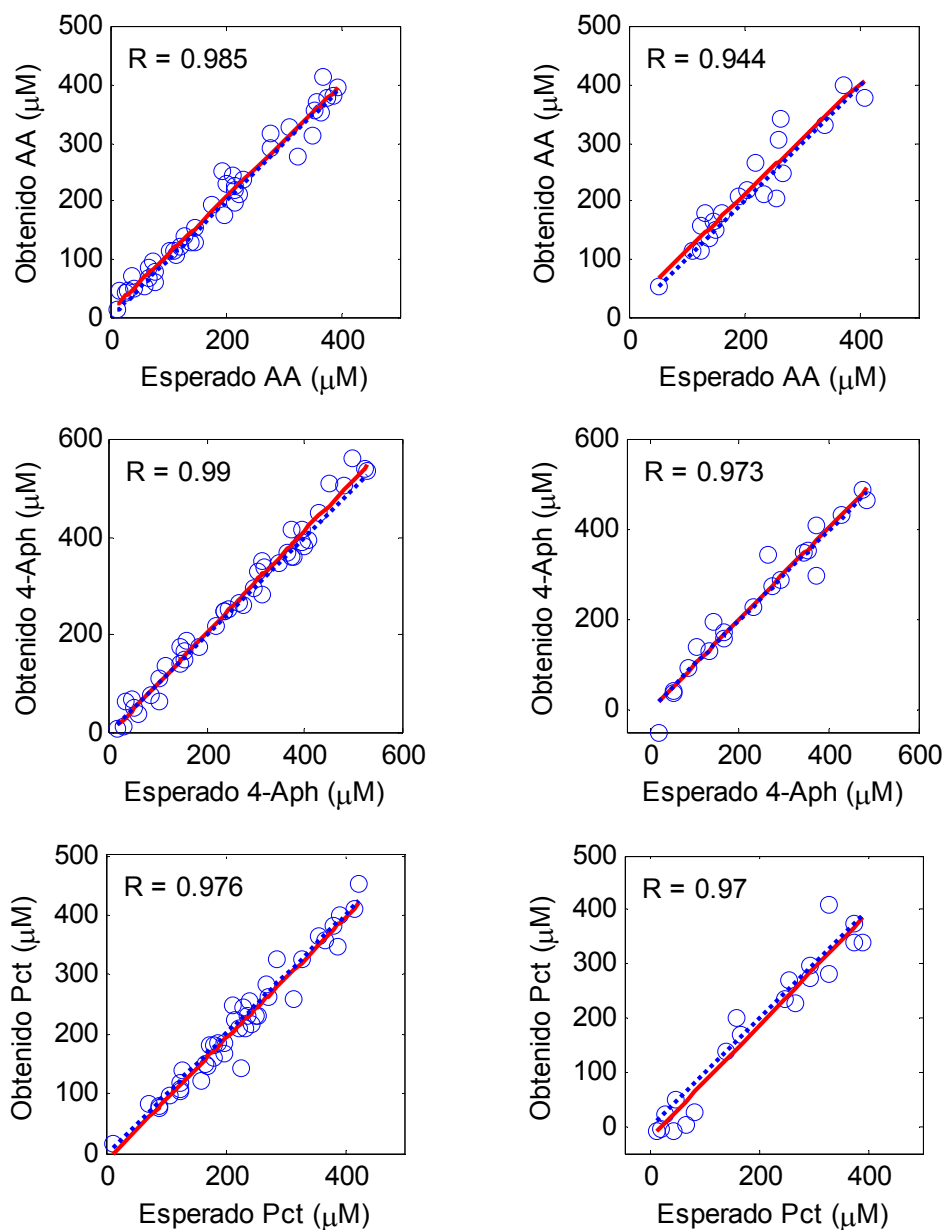


Fig. 28. Gráficas comparativas de regresión lineal entre los valores esperados y los obtenidos con una WNN con función Morlet basada en el producto tensorial. Los resultados corresponden a una red con 5 neuronas ocultas. Las gráficas de la columna derecha son los resultados obtenidos en entrenamiento y las gráficas de la columna izquierda son los resultados obtenidos en prueba.

De los resultados mostrados en las Tablas 5-8 y condensados en las Figs. 23 y 26, se observa que, en general, las redes con funciones de transferencia basadas en el producto tensorial tienen un buen desempeño para cualquiera de las funciones Wavelet aquí empleadas, aunque los resultados obtenidos con la red de 5 neuronas ocultas son ligeramente mejores en ambos casos en

comparación con los obtenidos con la red de 3 neuronas ocultas. Los porcentajes de error obtenidos en la etapa de prueba con las dos estructuras yacen entre el 10% y 30% en promedio para las tres salidas, aún cuando los valores de correlación estuvieron por arriba de 0.9. Este elevado porcentaje de error puede atribuirse al elevado número de funciones wavelet que se crean en cada neurona oculta (110 por cada una de ellas). La gran cantidad de grados de libertad (110 parámetros de escalamiento y 110 parámetros de traslación por cada neurona wavelet) permite obtener un buen ajuste en entrenamiento, sin embargo, la posibilidad de introducir funciones redundantes afecta negativamente el desempeño en la etapa de prueba, lo que impide un desempeño global mejorado.

8.4 Modelos MARS para la predicción de parámetros de control en la producción de azúcar a partir de la remolacha azucarera

Esta sección se desarrolló con datos públicos de prueba, concretamente provenientes del control de calidad en la producción del azúcar a partir de la remolacha, con el objeto de comprobar las prestaciones de los algoritmos ensayados con datos trilineales. Existe una necesidad en la industria azucarera de racionalizar y mejorar varios aspectos relacionados con el proceso de producción y control de calidad. Uno de ellos es conseguir un mejor entendimiento de la química involucrada en el proceso para mejorar tanto la orientación que se da a los productores de remolacha azucarera como al proceso de producción. Las investigaciones anteriores que se han realizado se centraron principalmente en determinar qué analitos químicos están presentes en el azúcar y productos intermedios; sin embargo, este tipo de información rara vez conduce a sugerencias concluyentes respecto a un proceso complicado como lo es la producción de azúcar.

Una opción que brinda un enfoque más exploratorio es el basado en el uso del análisis de espectrofotometría de fluorescencia para monitorizar el proceso de producción de azúcar a partir de la remolacha, desde las materias primas hasta el producto final pasando por los productos intermedios. La principal ventaja en el uso de datos espectrales para extraer información es

que este tipo de datos hacen posible construir modelos multivariantes más eficientes y robustos.

Los datos para el análisis de parámetros de calidad del azúcar que se usan en esta sección experimental pertenecen a una base de datos de carácter público existentes en la página del *Departamento de Ciencias de Alimentos de la Universidad de Copenhague* y que pueden descargarse a través de la dirección electrónica <http://www.models.kvl.dk/datasets>. Estos datos fueron adquiridos durante 3 meses de operación de una planta de azúcar en Escandinavia tomando muestras de azúcar directamente de la operación final de su proceso (centrifugado) y de manera continua durante ocho horas para obtener una muestra promedio representativa [4]. El número total de muestras obtenidas fue 268, de las cuales se descartaron 3 muestras por ser valores discrepantes.

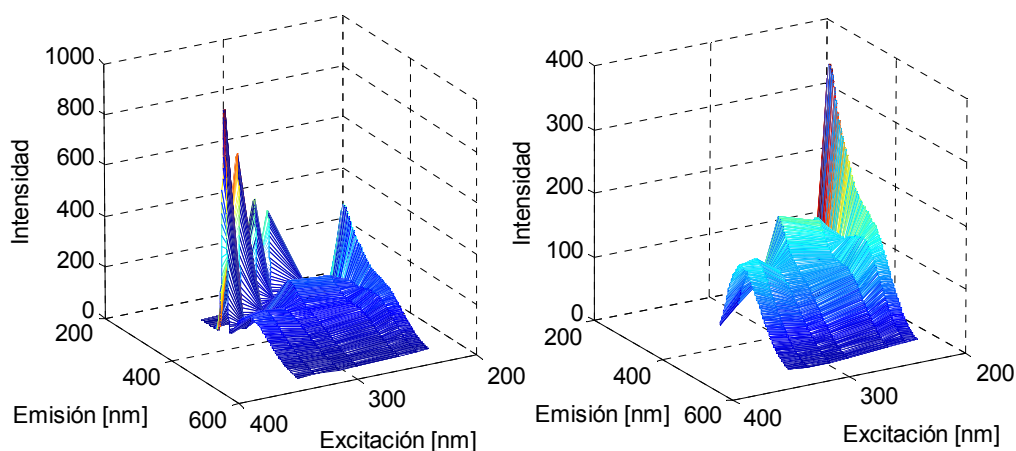


Fig. 29. Registro espectrofotométrico de fluorescencia obtenido con una de las muestras de azúcar. La gráfica de la izquierda corresponde al registro original y la de la derecha es el mismo registro luego de haber quitado las emisiones por debajo de las excitaciones.

Para preparar las muestras de análisis se disolvió el azúcar en agua en una proporción de 2.25g/15ml, y la solución resultante se midió en un espectrofluorímetro Perkin Elmer modelo LS50B. El espectro de emisión que se midió por cada muestra estuvo en el intervalo 275–560nm en intervalos de 0.5nm (571 longitudes de onda) para siete longitudes de onda de excitación (230, 240, 255, 290, 305, 325 y 340 nm). Como resultado de estas mediciones

se obtuvo un tensor de datos de 3 modos y tamaño $265 \times 571 \times 7$. El primero de los modos de este tensor se refiere al número de muestras, el segundo al número de longitudes de onda de emisión, y el tercero al número de longitudes de onda de excitación. La Fig. 29 muestra los espectros obtenidos con una de las muestras.

Junto con los datos espectrofotométricos también se obtuvieron indicadores de la calidad del azúcar producido. Estos indicadores fueron el contenido de cenizas y el color. El contenido de cenizas se determinó mediante conductividad y es una medida de la cantidad de impurezas inorgánicas en el azúcar refinado. Este valor está dado en porcentajes. El color se determina como la absorción a 420nm de una solución de azúcar filtrada por membrana y ajustada para tener un pH de 7. El color es una unidad derivada de la absorbancia, donde 45 es el máximo color permitido para el azúcar estándar. Este color es tan bajo que no es de importancia para el consumidor pero es de interés para el proceso de control y para los pequeños distribuidores. Los valores de ambos parámetros de calidad están graficados en la Fig. 30 para las 265 muestras.

El objetivo de esta sección experimental es usar MARS para construir, a partir de arreglos multidimensionales espectrofotométricos usados como predictores, modelos de calibración que permitan predecir el porcentaje de ceniza y color de las muestras de azúcar. Los datos descargados están formados por un total de 3997 descriptores (571 longitudes de onda de emisión por cada una de las 7 longitudes de onda de excitación) que se usaron para construir los modelos MARS que permitan predecir, simultáneamente, el contenido de cenizas y el color del azúcar. En la construcción del modelo se permitió la interacción de hasta dos variables, lo que dio origen a términos de segundo orden. La construcción de un modelo MARS se realiza en dos etapas, en la primera se obtiene una primera aproximación sobreajustada formada por un número máximo de términos establecido por el usuario, y en la segunda se *recorta* el modelo para optimizarlo. El criterio para detener la primera etapa del proceso MARS fue obtener un primer modelo con hasta 50 términos. Durante la ejecución de la segunda etapa para eliminar términos y obtener un modelo más pequeño se usó el criterio de la validación cruzada generalizada (GCV) y se probaron 3 valores del parámetro d (2, 3, 4). La triada de valores elegidos para

d contiene el valor recomendado por Jerome Friedman como bastante eficaz (3).

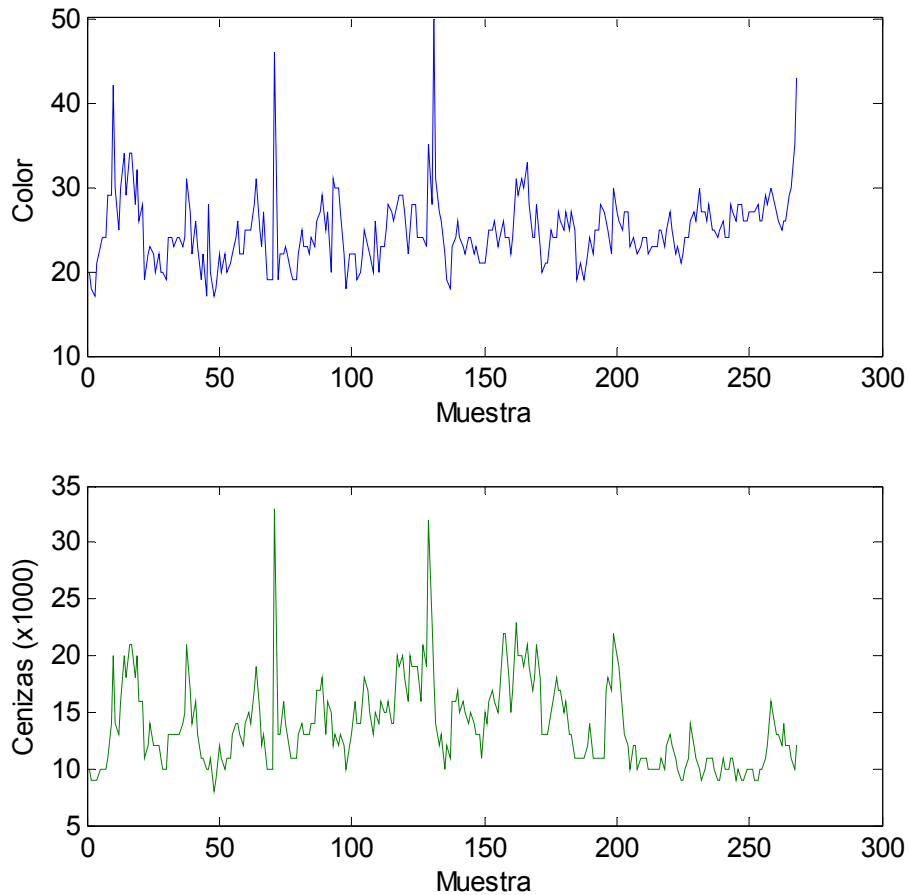


Fig. 30. Valores de color y contenido de cenizas medidas con las 265 muestras de azúcar.

Al igual que en los experimentales anteriores, se tomaron dos terceras partes de las 265 muestras para construir el modelo y la tercera parte restante para probarlo. Las muestras e intensidades con índice $[3n-2, 3n-1, \dots, 265]_{n=1..89}$ se usaron para construir el modelo, y las correspondientes a los índices $[3n, \dots, 265]_{n=1..89}$ para probarlo. La metodología usada en MARS hace que, dado el conjunto de predictores y respuestas, el modelo obtenido sea único y repetible, sin importar las veces que se realice el proceso; esto permite construir el modelo sólo una vez debido a que no hay parámetros que se inicialicen con valores aleatorios.

Los modelos MARS finalmente obtenidos tuvieron 30, 23 y 18 términos para valores de $d = 2, 3$ y 4 , respectivamente. Se obtienen modelos con más términos para valores menores de d ya que los valores mayores de d permiten obtener modelos con menos nodos y más suavizados.

En la Tabla 9 se muestra un resumen de las funciones base B_1, B_2, \dots, B_{33} junto con sus coeficientes a_i para el modelo con 18 términos, donde también se indican las interacciones entre variables. El modelo está formado por un término constante B_1 , 13 términos de primer orden ($B_2, B_3, B_4, B_5, B_6, B_7, B_8, B_9, B_{10}, B_{11}, B_{12}, B_{14}$), y 4 términos de segundo orden ($B_{13}, B_{15}, B_{16}, B_{18}$). Cada uno de los términos que forman el modelo queda descrito por una o dos longitudes de onda de emisión y su correspondiente intensidad que da origen a la spline truncada.

Tabla 9. Funciones base que forman el modelo MARS. La letra E con un subíndice numérico y etiqueta nm se refiere a la longitud de onda de emisión. El valor numérico de cada término entre paréntesis es la intensidad medida en la longitud de onda de emisión indicada. El valor numérico de la columna etiquetada Entrada se refiere a una de las siete longitudes de onda de excitación. Los valores en las columnas a 's son los coeficientes de cada término para el color y las cenizas.

Función	Descripción	Entrada	a_{color}	a_{cenizas}
B_1	1	-	0,83	0,49
B_2	$(E_{383,5\text{nm}} - 240,56)_+$	7	24,69	4,00
B_3	$(240,56 - E_{383,5\text{nm}})_+$	7	-31,56	-32,97
B_4	$(E_{374,5\text{nm}} - 41,53)_+$	4	-93,88	34,30
B_5	$(41,53 - E_{374,5\text{nm}})_+$	4	826,92	-213,65
B_6	$(E_{341,5\text{nm}} - 91,774)_+$	6	-1323,40	-1636,00
B_7	$(91,774 - E_{341,5\text{nm}})_+$	6	-0,30	-4,00
B_8	$(E_{355,5\text{nm}} - 78,77)_+$	1	-367,74	42,39
B_9	$(78,77 - E_{355,5\text{nm}})_+$	1	1749,00	2730,00
B_{10}	$(E_{375,5\text{nm}} - 96,17)_+$	5	-532,45	863,03
B_{11}	$(74,87 - E_{321\text{nm}})_+$	3	3230,80	-1166,80
B_{12}	$(E_{363\text{nm}} - 40,17)_+$	2	4072,40	-1145,00
B_{13}	$(E_{326,5\text{nm}} - 89,04)_+$ $(240,56 - E_{383,5\text{nm}})_+$	6, 7	266,83	-362,67
B_{14}	$(0,59 - E_{301\text{nm}})_+$	1	2300,80	159,30
B_{15}	$(E_{370,5\text{nm}} - 111,89)_+$ + ($91,774 - E_{341,5\text{nm}})_+$	5, 6	7724,40	-6232,80
B_{16}	$(111,89 - E_{370,5\text{nm}})_+$ $(91,774 - E_{341,5\text{nm}})_+$	5-6	-2645,50	1635,50
B_{17}	$(130,94 - E_{358\text{nm}})_+$	5	132,00	20,16
B_{18}	$(E_{313\text{nm}} - 252,26)_+$ + ($E_{374,5\text{nm}} - 41,53)_+$	3, 4	-146,81	21,98

Para evaluar la eficiencia de los 3 modelos de calibración MARS que se construyeron, se usó la tercera parte de los datos que se reservaron para este fin. Los parámetros obtenidos del análisis de regresión lineal junto con los correspondientes valores de errores porcentuales resultantes de la comparación entre las salidas de los modelos de calibración y los correspondientes valores reales se condensan en la Tabla 10 para el color y el Tabla 11 para el contenido de cenizas.

Las gráficas comparativas de la Fig. 31 entre los valores reales y los obtenidos con el modelo MARS de 18 términos muestran que existe un término en la etapa de prueba que puede etiquetarse como outlayer tanto en la predicción del color como en el contenido de cenizas. Este valor, cuya predicción se aleja mucho del valor real, también se presentó en los restantes modelos de calibración con 23 y 30 términos.

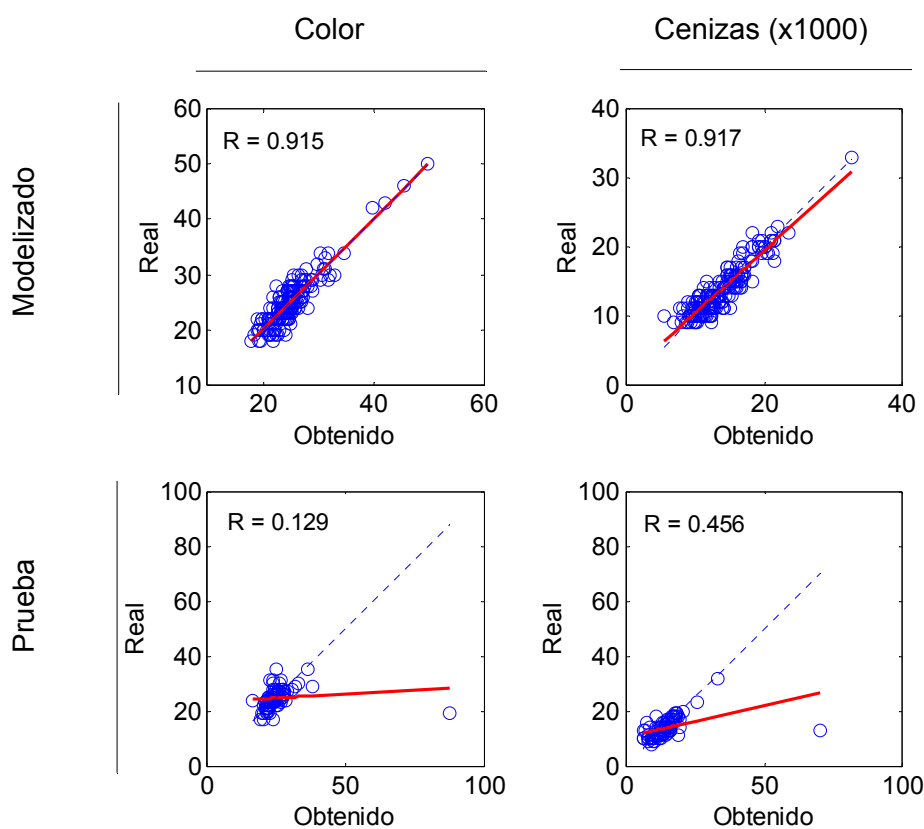


Fig. 31. Gráficas de regresión lineal entre los valores reales y los obtenidos con un modelo MARS de 18 términos para el color del azúcar y contenido de cenizas. En las gráficas de prueba se observa el término identificado como un outlayer adicional.

Tabla 10. *Parámetros de regresión lineal y error porcentual obtenidos al comparar los valores reales de color vs. los valores obtenidos con los modelos MARS al usar todas las muestras, incluido el outlayer.*

B's	m		b		R		EPAP (%)	
	tr	ts	tr	ts	tr	ts	tr	ts
18	0.836	0.268	4.079	18.595	0.914	0.128	6.034	12.352
23	0.868	0.299	3.287	17.855	0.931	0.145	5.513	12.713
30	0.921	0.330	1.954	17.252	0.960	0.156	4.166	12.766

Tabla 11. *Parámetros de regresión lineal y error porcentual obtenidos al comparar los valores reales del contenido de cenizas vs. los valores obtenidos con los modelos MARS al usar todas las muestras, incluido el outlayer.*

B's	m		b		R		EPAP (%)	
	tr	ts	tr	ts	tr	ts	tr	ts
18	0.929	0.917	0.972	1.587	0.916	0.455	9.455	18.271
23	0.960	0.989	0.539	0.619	0.934	0.490	8.549	18.581
30	0.956	1.015	0.598	0.392	0.959	0.493	6.442	18.699

La exclusión del término identificado como un outlayer y la reevaluación de los modelos usando el nuevo conjunto de datos de prueba permitieron mejorar los parámetros de regresión lineal obtenidos y disminuir también el error porcentual. La Fig. 32 muestra gráficas comparativas para el modelo MARS con 18 términos excluyendo el outlayer, y la Tabla 12 condensa los resultados de los análisis de regresión lineal realizados en los tres modelos MARS, junto con los errores porcentuales obtenidos en la predicción del color. La Fig. 33 y la Tabla 13 muestran los correspondientes resultados para la predicción del contenido de cenizas.

Tabla 12. *Parámetros de regresión lineal y error porcentual obtenidos al comparar los valores reales de color vs. los valores obtenidos con los modelos MARS una vez excluido el outlayer.*

B's	m		b		R		EPAP (%)	
	tr	ts	tr	ts	tr	ts	tr	ts
18	0.836	0.603	4.079	9.577	0.914	0.637	6.034	8.384
23	0.868	0.626	3.287	9.062	0.931	0.632	5.513	8.853
30	0.921	0.668	1.954	8.171	0.960	0.672	4.166	8.785

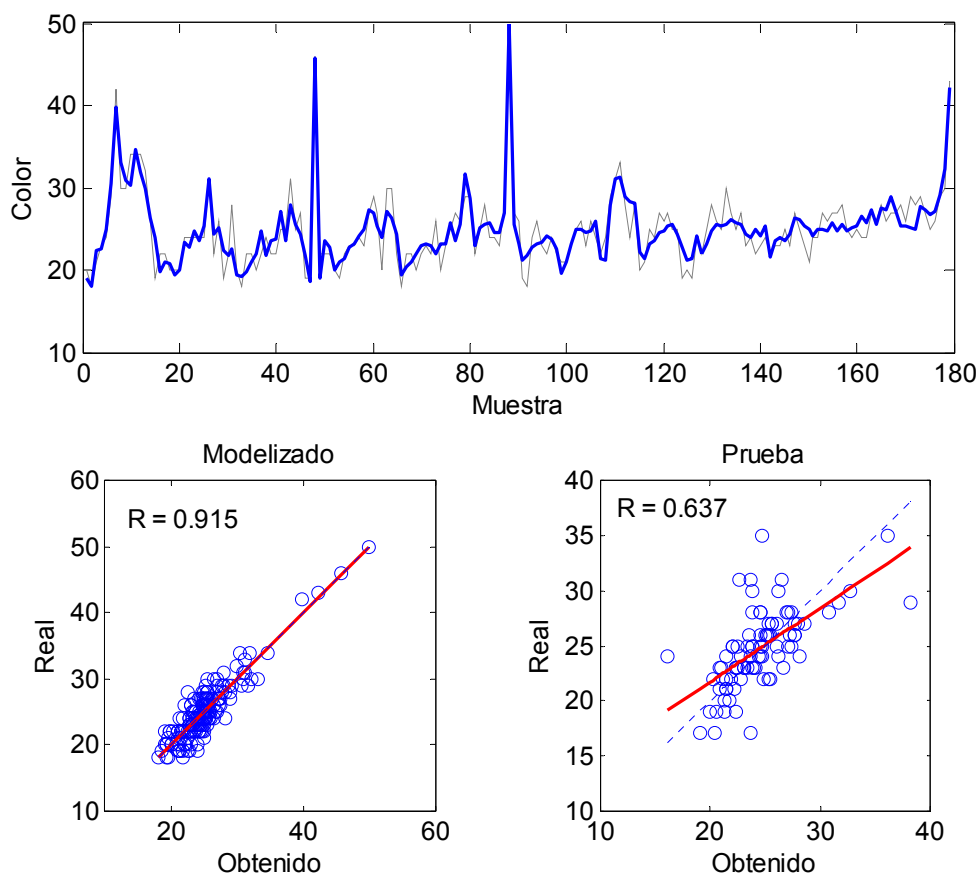


Fig. 32. Gráficas comparativas y de regresión lineal entre los valores reales de color del azúcar y los valores obtenidos con un modelo MARS de 18 términos. El trazo con una línea delgada en la figura superior corresponde a los valores reales, el trazo con una línea gruesa corresponde a los valores obtenidos.

Tabla 13. Parámetros de regresión lineal y error porcentual obtenidos al comparar los valores reales del contenido de cenizas vs. los valores obtenidos con los modelos MARS una vez excluido el outlier.

$B's$	m		b		R		$EPAP$ (%)	
	tr	ts	tr	ts	tr	ts	tr	ts
18	0.929	0.952	0.972	0.453	0.916	0.819	9.455	13.474
23	0.960	1.024	0.539	-0.485	0.934	0.830	8.549	13.915
30	0.956	1.051	0.598	-0.744	0.959	0.851	6.442	13.886

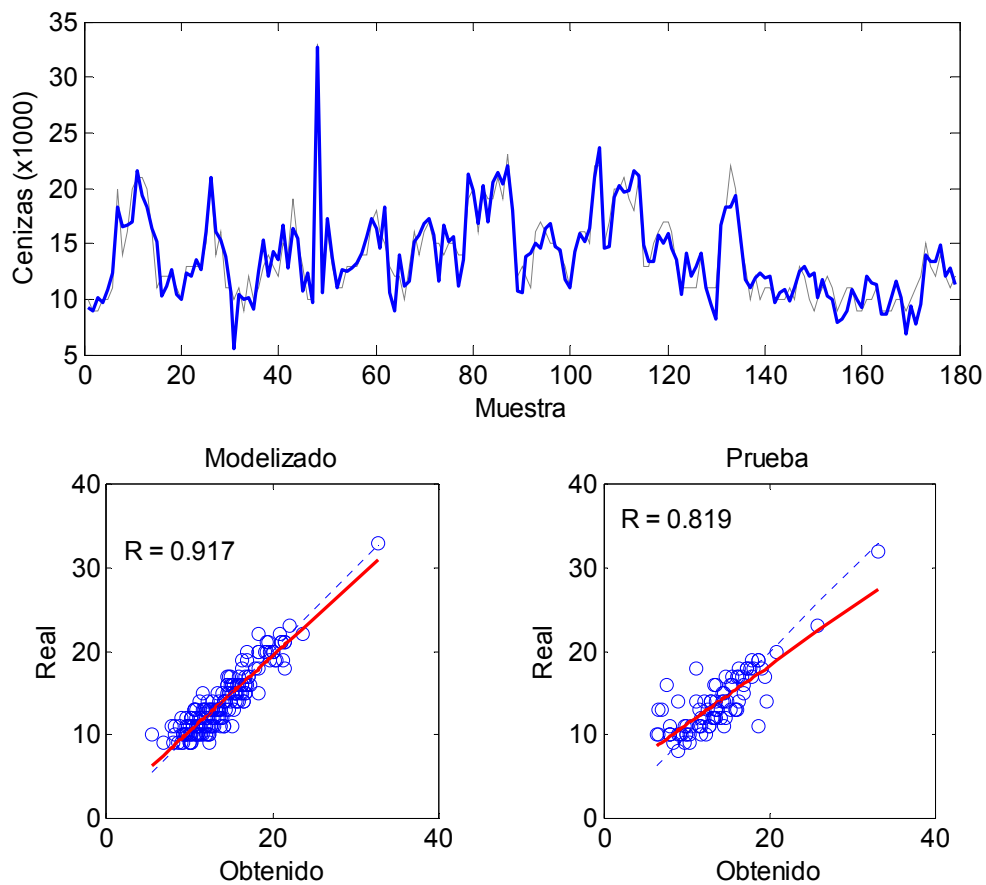


Fig. 33. Gráficas comparativas y de regresión lineal entre los valores reales del contenido de cenizas del azúcar y los valores obtenidos con un modelo MARS de 18 términos. El trazo con una línea delgada en la figura superior corresponde a los valores reales, el trazo con una línea gruesa corresponde a los valores obtenidos.

Los ajustes obtenidos con MARS para esta prueba resultan bastante satisfactorios. La efectividad del uso de funciones spline truncadas en la construcción de modelos de regresión queda demostrada con la predicción simultánea de dos variables que guardan estrecha relación entre ellas y con la obtención de parámetros de regresión y errores porcentuales dentro de márgenes aceptables, a pesar de que la predicción del contenido de cenizas en el azúcar fue mejor que la predicción del color.

8.5 Modelos B-MARS en la predicción de parámetros de control en la producción de azúcar a partir de la remolacha azucarera

La capacidad de creación de modelos B-MARS se evaluó usando el mismo conjunto de datos espectrofotométricos obtenidos durante el proceso de producción del azúcar de remolacha del apartado anterior. La construcción de modelos B-MARS es semejante a MARS en el sentido de que ambos se realizan en dos etapas, aunque el número de funciones B-spline que se evalúan para construir el modelo B-MARS es restringido. En MARS cada uno de los predictores da origen a dos funciones spline truncadas, mientras que en B-MARS el número de funciones B-spline a evaluar se calcula por las escalas y secuencia de nodos que se determinan en base al número total de valores de salida a ajustar y el grado de la función base.

Se construyeron modelos usando funciones B-spline de primero, segundo y tercer grado, y se permitió la interacción de hasta dos variables durante la búsqueda de los términos del modelo. El número total de muestras a predecir que se usó en la construcción de los modelos B-MARS se redujo en una unidad (264 muestras en total) debido a la eliminación del outlayer adicional que se identificó durante la construcción de los anteriores modelos MARS. La separación de la información en subconjuntos para modelizado y prueba se hizo igual que en las anteriores pruebas: dos terceras partes se usaron para construir el modelo y la tercera parte restante para probarlo. A pesar de esta separación, el algoritmo escrito para la construcción de los modelos B-MARS hace uso del número original de muestras (264 en este caso) para obtener más escalas y nodos, lo que permite ampliar la familia de funciones B-spline que se evalúan para construir el modelo e incluir funciones con un soporte cada vez más estrecho para ajustar detalles finos.

La función B-spline de primer grado permitió obtener una secuencia de 9 escalas y un total de 513 nodos para la escala más alta. El número de funciones que se evaluaron varió desde 2 para la escala más baja hasta 257 para la escala más alta (Fig. 34). El parámetro d usado en la validación cruzada generalizada (GCV), una vez hallados los coeficientes a_j por regresión de mínimos cuadrados, tomó valores de 2, 3 y 4 para poder comparar los modelos B-MARS contra los obtenidos usando MARS.

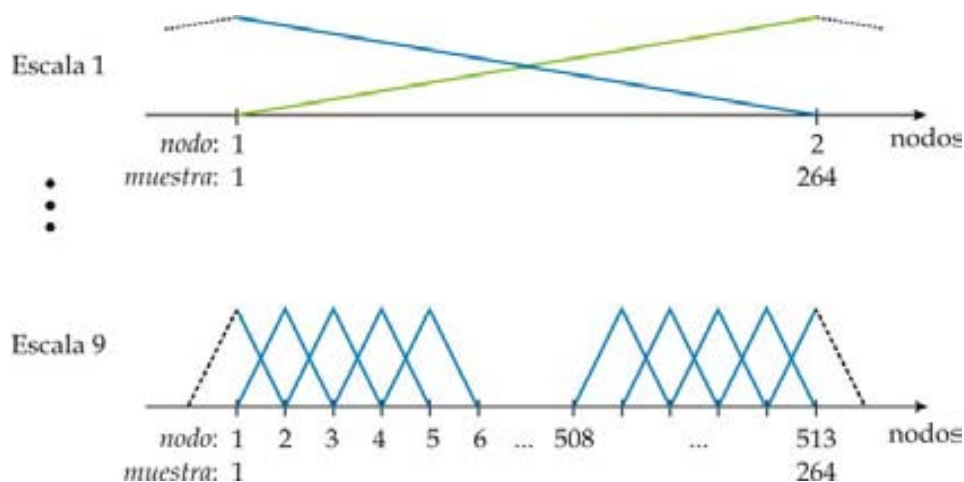


Fig. 34. Esquema de las familias de B-splines de primer grado construidas en base a los 513 nodos obtenidos con 9 escalas para las 264 muestras del experimento.

Los modelos B-MARS que se obtuvieron con funciones base de primer grado tuvieron 35, 27 y 23 términos para valores de $d=2$, 3 y 4, respectivamente. Los resultados obtenidos del análisis de regresión lineal junto con el error porcentual de las diferencias entre los valores esperados y los obtenidos con los modelos B-MARS están condensados en las Tablas 14 y 15. Los resultados de la Tabla 14 corresponden al color del azúcar, y la Tabla 15 al porcentaje de cenizas hallado al final de su proceso de producción.

Para visualizar la estructura de los modelos B-MARS, la Tabla 16 contiene un resumen de las funciones base y coeficientes a_j que se obtuvieron para el modelo con 23 términos ($d=2$). En la misma tabla se muestran también los términos con interacciones entre variables.

Tabla 14. Parámetros de regresión lineal y error porcentual obtenidos al comparar los valores reales de color vs. los valores obtenidos con los modelos B-MARS con funciones B-spline de primer grado.

B 's	m		b		R		$EPAP$ (%)	
	tr	ts	tr	ts	tr	ts	tr	ts
23	0.941	0.791	1.468	4.986	0.928	0.769	5.566	7.851
27	0.965	0.787	0.854	5.043	0.944	0.711	5.064	8.734
35	0.973	0.806	0.652	4.728	0.965	0.739	3.885	8.437

Tabla 15. Parámetros de regresión lineal y error porcentual obtenidos al comparar los valores reales del contenido de cenizas vs. los valores obtenidos con los modelos B-MARS con funciones B-spline de primer grado.

B's	m		b		R		EPAP (%)	
	tr	ts	tr	ts	tr	ts	tr	ts
23	0.930	0.997	0.957	-0.203	0.924	0.858	9.072	12.401
27	0.973	0.998	0.369	-0.163	0.942	0.861	8.165	12.276
35	0.964	1.135	0.490	-1.908	0.965	0.895	5.861	13.352

En la Fig. 35 se muestran los valores reales y los obtenidos para el color y contenido de cenizas obtenidos con los dos tercios del total de muestras de azúcar usadas en la construcción del modelo B-MARS de 23 términos. La Fig. 36 muestra las gráficas correspondientes para los resultados obtenidos en la etapa de prueba.

Tabla 16. Funciones base del modelo B-MARS con B-splines de primer grado. En la tabla se indica la escala, la posición de la función, la entrada (longitudes de onda de excitación), la variable (longitudes de onda de emisión), la interacción con otra variable y los coeficientes de regresión para el color y el contenido de cenizas.

Función	Escala	Posición	Entrada	Variable	Interacción	a_{color}	$a_{cenizas}$
B_1	-	-	-	-	-	-0.03	0.37
B_2	5	12	7	17	-	-32.98	-18.29
B_3	2	1	5	195	B_2	74.37	28.28
B_4	1	1	6	303	-	-59.81	7.26
B_5	3	4	1	291	-	-260.24	143.86
B_6	3	0	1	212	B_2	1615,70	1153,20
B_7	1	1	2	152	-	1178,30	61,10
B_8	6	46	1	84	B_7	212,13	290,12
B_9	2	3	3	11	-	171,90	-4218,10
B_{10}	3	5	1	405	-	-721,40	-169,10
B_{11}	2	4	3	95	-	-100,51	-52,52
B_{12}	2	3	2	31	-	1720,50	-268,20
B_{13}	7	20	3	194	B_9	1107,80	808,10
B_{14}	4	10	4	90	-	-279,07	40,52
B_{15}	3	7	1	24	-	616,81	-30,82
B_{16}	1	2	4	36	-	805,70	-1686,50
B_{17}	2	3	3	19	-	1,37	-22,82
B_{18}	6	36	1	70	-	-320,48	5,54
B_{19}	3	3	5	100	B_9	1628,10	1897,50
B_{20}	2	2	5	372	-	-168,30	-28,68
B_{21}	4	8	6	248	-	148,39	249,28
B_{22}	3	2	4	450	-	-20,99	-17,97
B_{23}	7	95	1	87	B_2	-60,11	-80,49

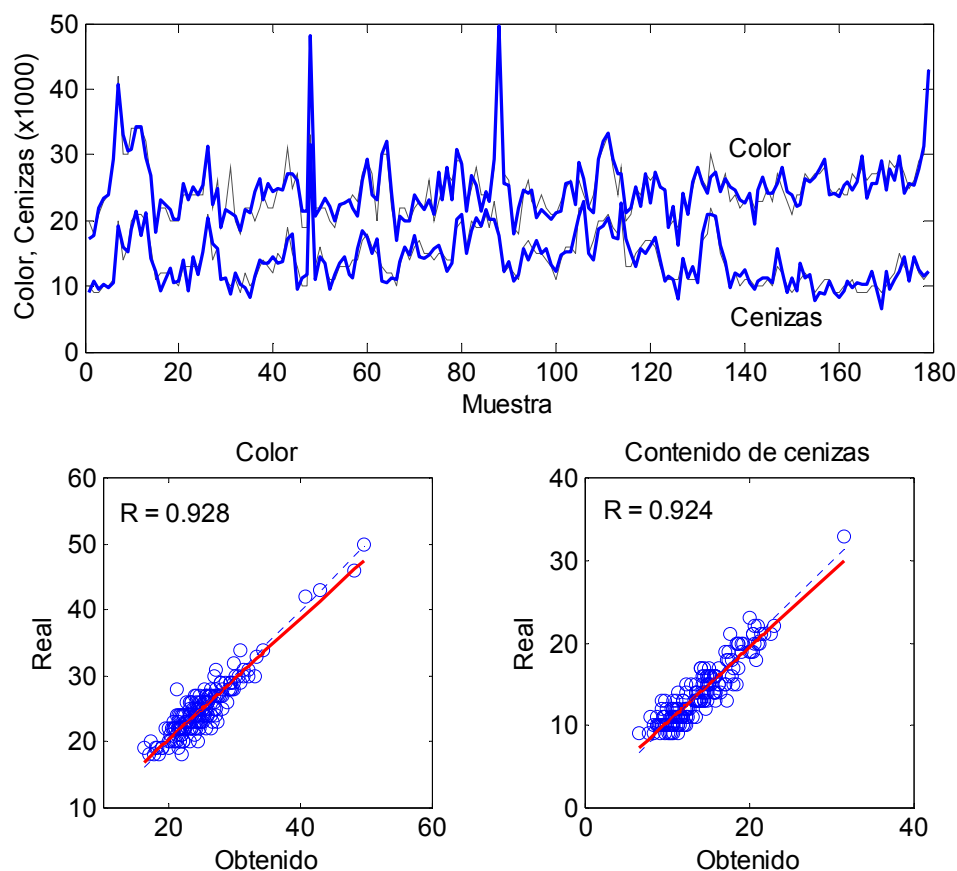


Fig. 35. Gráficas comparativas entre los valores reales de color del azúcar y cenizas y los correspondientes valores obtenidos en la etapa de modelizado con B-MARS de primer grado y 23 términos. La línea delgada de la gráfica superior corresponde a los valores reales y la línea gruesa a los valores obtenidos. Las dos figuras de la fila inferior son las correspondientes gráficas de regresión lineal

En los casos de modelizado donde se usaron funciones B-spline de segundo grado, el número total de muestras y el grado de la función permitió obtener una secuencia de 6 escalas y un total de 730 nodos para la sexta escala. El número de funciones que se evaluaron varió desde 3 para la escala más baja hasta 731 para la escala más alta (Fig. 37). El parámetro d usado en la GCV tomó los mismos valores que en el caso anterior ($d=2, 3$ y 4). Los modelos que se obtuvieron con B-splines de segundo grado tuvieron 27, 20 y 15 términos para los tres valores de d que se evaluaron, respectivamente.

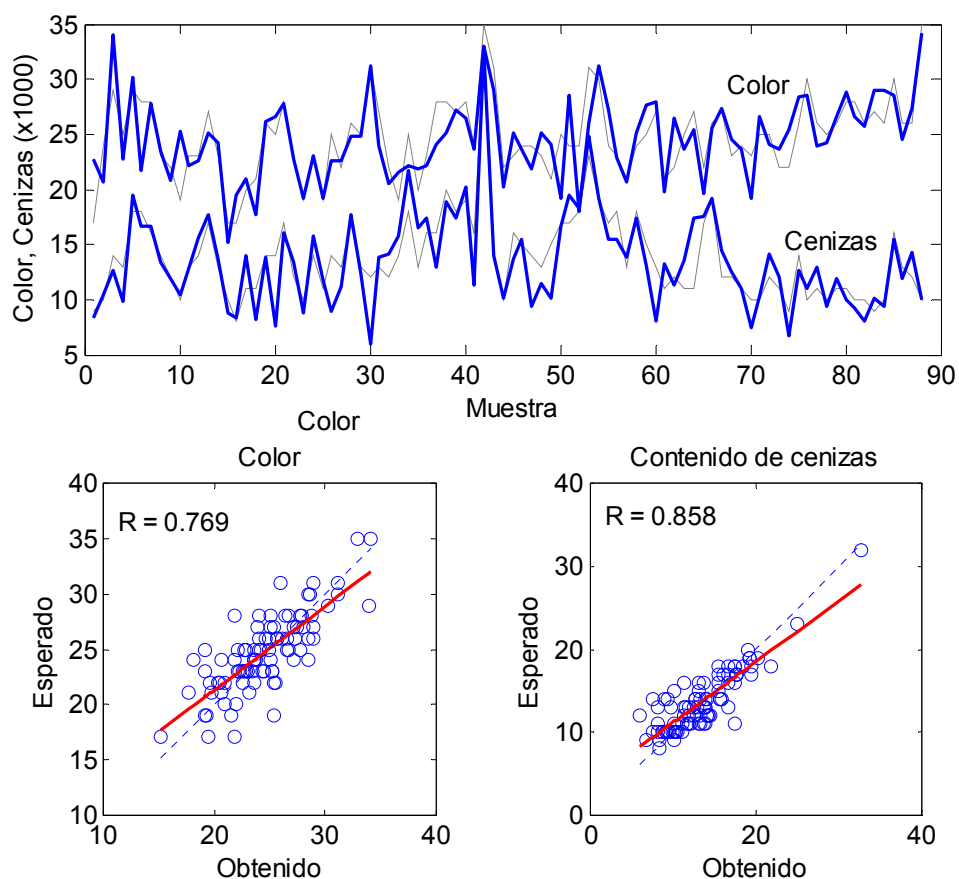


Fig. 36. Gráficas comparativas entre los valores reales de color del azúcar y cenizas y los correspondientes valores obtenidos en prueba con un modelo B-MARS de primer grado y 23 términos. La línea delgada de la gráfica superior corresponde a los valores reales y la línea gruesa a los valores obtenidos. Las dos figuras de la fila inferior son las correspondientes gráficas de regresión lineal

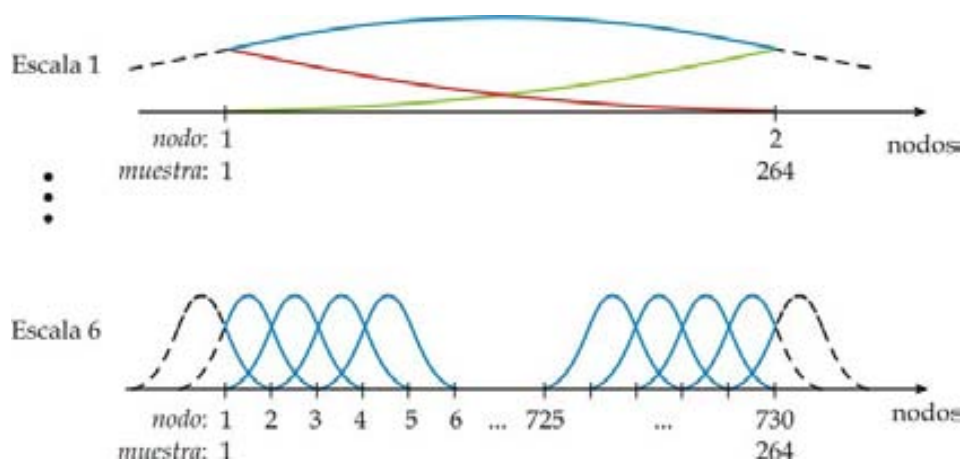


Fig. 37. Esquema de las B-splines de segundo grado evaluadas en base a los 730 nodos obtenidos con las 6 escalas creadas con las 264 muestras.

El error porcentual de las diferencias entre los valores reales y los obtenidos con las redes para el color del azúcar y su contenido de cenizas, más los parámetros de regresión lineal resultantes de la comparación entre los dos conjuntos de valores resultantes en las etapas de modelizado y prueba se condensan en las Tablas 17 y 18.

Para mostrar el desempeño de los modelos construidos con funciones B-spline de segundo grado, la Fig. 38 muestra gráficas comparativas entre los valores reales y los resultados obtenidos en la etapa de modelizado con el modelo B-MARS de 15 términos, la Fig. 39 muestra los correspondientes resultados obtenidos en prueba tanto para el color como para el contenido de cenizas.

La última función que se probó en la construcción de los modelos B-MARS fue la B-spline de tercer grado. El número de escalas y secuencias de nodos que se obtuvieron con estas funciones fueron 5 y 1025, respectivamente, lo que permitió evaluar hasta 1027 B-splines de tercer grado en la quinta escala. La distribución de las funciones en las cinco escalas sigue un patrón similar al presentado en las Figs. 34 y 37 para las B-spline de grados uno y dos. Los modelos B-MARS que se construyeron tuvieron 23, 18 y 12 términos para valores de $d=4$, 3 y 2, respectivamente.

Los resultados obtenidos de las comparaciones entre los valores reales de color del azúcar y contenido de cenizas, y los valores obtenidos con los modelos B-MARS de tercer grado, se condensan en las Tablas 19 y 20 junto con los errores porcentuales resultantes de las diferencias entre ambos conjuntos de valores. Para propósitos de visualización, las Figs. 40 y 41 muestran el desempeño del modelo B-MARS con 12 términos.

Tabla 17. Parámetros de regresión lineal y error porcentual obtenidos al comparar los valores reales de color vs. los valores obtenidos con los modelos B-MARS con funciones B-spline de segundo grado.

$B's$	m		b		R		$EPAP$ (%)	
	tr	ts	tr	ts	tr	ts	tr	ts
15	0.925	1.051	1.853	-1.615	0.931	0.730	5.507	10.035
20	0.942	1.068	1.435	-1.968	0.944	0.726	5.075	10.839
27	0.967	0.967	0.820	0.728	0.964	0.711	3.988	10.729

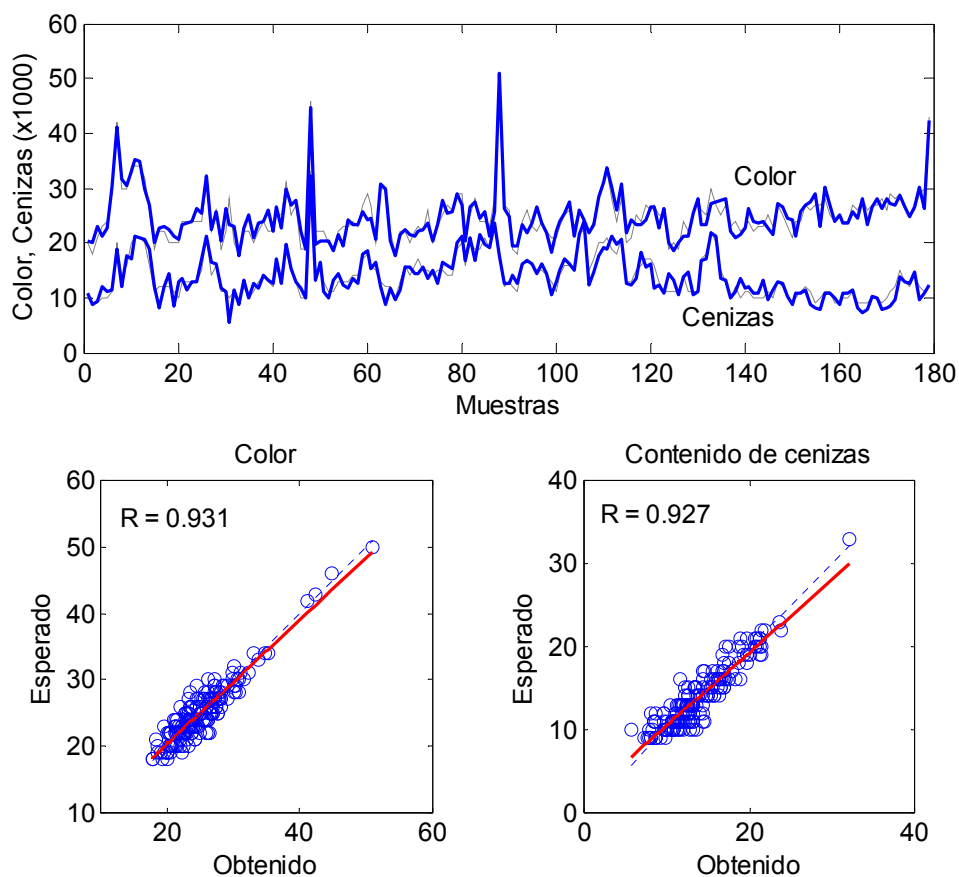


Fig. 38. Gráficas comparativas entre los valores reales de color del azúcar y contenido de cenizas vs. los correspondientes valores obtenidos en el modelizado con un modelo B-MARS de segundo orden y 15 términos. La línea delgada de la gráfica superior corresponde a los valores reales y la línea gruesa a los valores obtenidos. Las dos figuras de la fila inferior son las correspondientes gráficas de regresión lineal

Tabla 18. Parámetros de regresión lineal y error porcentual obtenidos al comparar los valores reales del contenido de cenizas vs. los valores obtenidos con los modelos B-MARS con funciones B-spline de segundo grado.

B's	m		b		R		EPAP (%)	
	tr	ts	tr	ts	tr	ts	tr	ts
15	0.974	0.959	0.354	0.2674	0.927	0.741	8.811	16.069
20	1.009	0.978	-0.125	0.074	0.944	0.761	7.982	17.808
27	1.016	0.953	-0.234	0.584	0.967	0.781	6.059	16.211

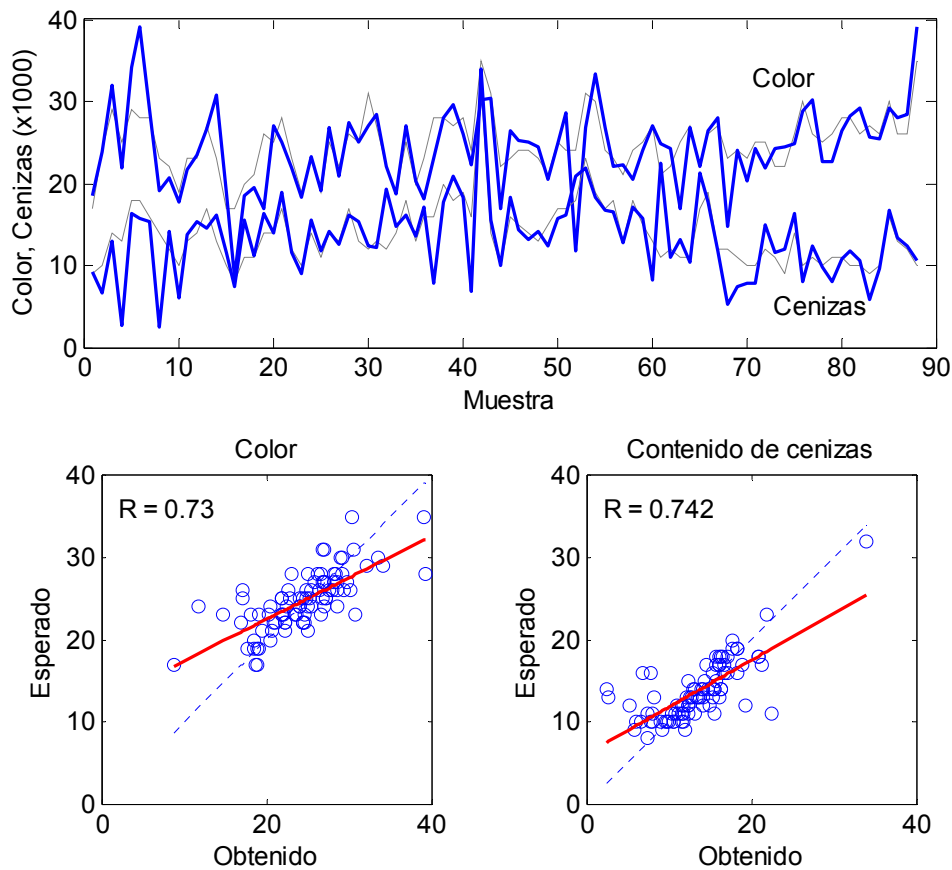


Fig. 39. Gráficas comparativas entre los valores reales de color del azúcar y contenido de cenizas vs. los correspondientes valores obtenidos en prueba con un modelo B-MARS de segundo grado y 15 términos. La línea delgada de la gráfica superior corresponde a los valores reales y la línea gruesa a los valores obtenidos. Las dos figuras de la fila inferior son las correspondientes gráficas de regresión lineal

Tabla 19. Parámetros de regresión lineal y error porcentual obtenidos al comparar los valores reales de color vs. los valores obtenidos con los modelos B-MARS con funciones B-spline de tercer grado.

B's	m		b		R		EPAP (%)	
	tr	ts	tr	ts	tr	ts	tr	ts
12	0.936	0.874	1.583	2.786	0.925	0.700	6.034	9.338
18	0.946	0.889	1.337	2.488	0.943	0.695	5.198	9.776
23	0.945	0.919	1.365	1.931	0.960	0.732	4.174	9.353

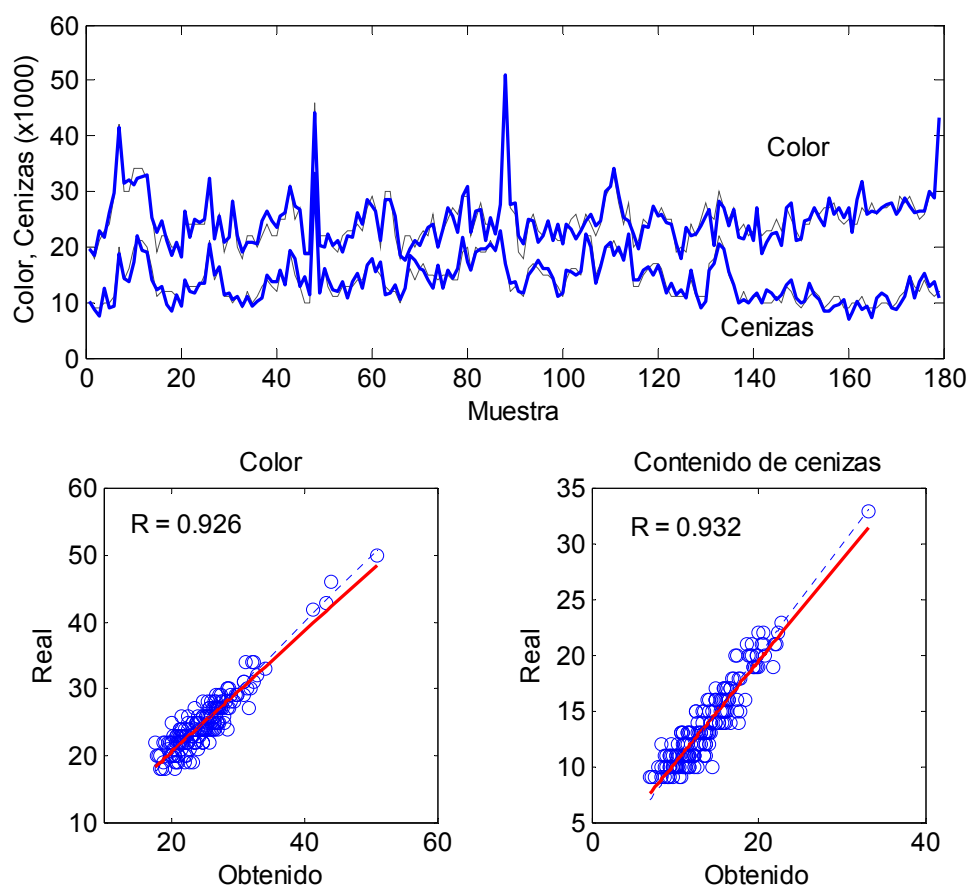


Fig. 40. Gráficas comparativas entre los valores reales de color del azúcar y contenido de cenizas vs. los correspondientes valores obtenidos en el modelizado con un modelo B-MARS de tercer grado y 12 términos. La línea delgada de la gráfica superior corresponde a los valores reales y la línea gruesa a los valores obtenidos. Las dos figuras de la fila inferior son las correspondientes gráficas de regresión lineal

Tabla 20. Parámetros de regresión lineal y error porcentual obtenidos al comparar los valores reales del contenido de cenizas vs. los valores obtenidos con los modelos B-MARS con funciones B-spline de tercer grado.

$B's$	m		b		R		$EPAP$ (%)	
	tr	ts	tr	ts	tr	ts	tr	ts
12	0.944	1.064	0.764	-1.149	0.932	0.831	8.763	14.419
18	0.956	0.967	0.597	0.239	0.947	0.779	7.679	16.344
23	0.968	0.855	0.435	1.930	0.965	0.779	5.890	15.016

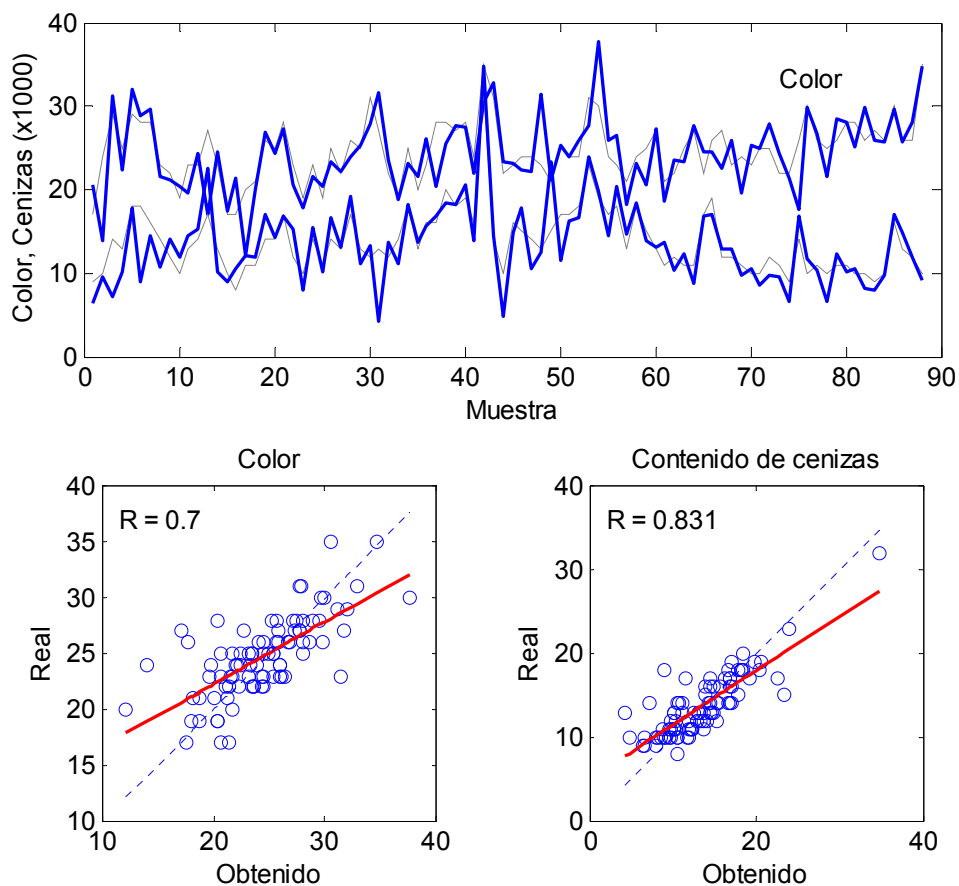


Fig. 41. Gráficas comparativas entre los valores reales de color del azúcar y contenido de cenizas vs. los correspondientes valores obtenidos en prueba con un modelo B-MARS de tercer grado y 12 términos. La línea delgada de la gráfica superior corresponde a los valores reales y la línea gruesa a los valores obtenidos. Las dos figuras de la fila inferior son las correspondientes gráficas de regresión lineal

Los resultados obtenidos con los modelos B-MARS basados en funciones B-spline de primero, segundo y tercer grado se compararon contra los resultados obtenidos con los modelos MARS basados en splines truncadas. En la Fig. 42 están graficados el número de términos de cada modelo construido y el error porcentual relativo.

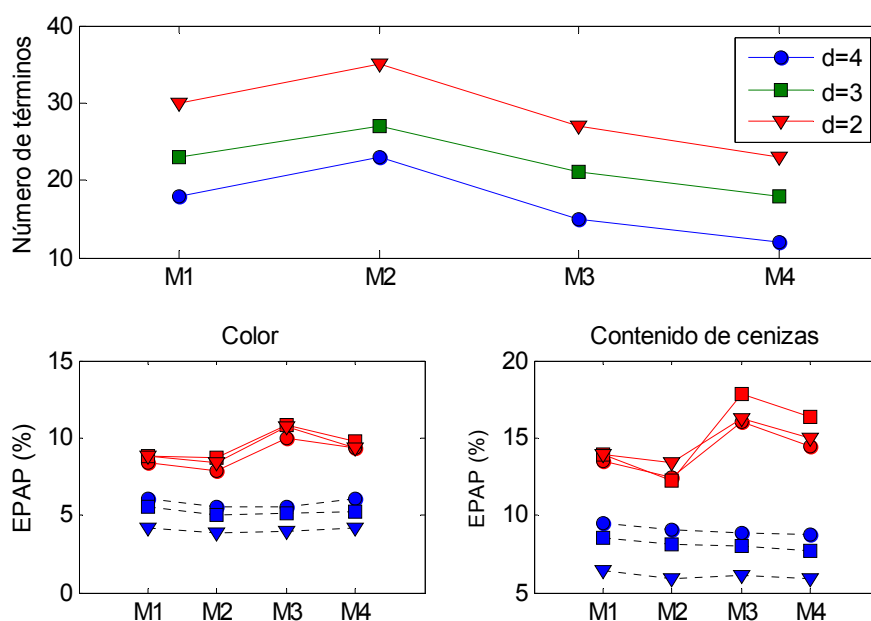


Fig. 42. Gráfica superior: número de términos en cada modelo en función del parámetro de suavizado d . Gráficas inferiores: comparativas del error porcentual obtenido con los modelos MARS y B-MARS. Las líneas discontinuas corresponden al EPAP obtenido en modelizado y las líneas continuas al obtenido en prueba. Las etiquetas en los ejes x para todas las gráficas corresponden a M1=modelo MARS, M2=modelo B-MARS grado 1, M3=modelo B-MARS grado 2, y M4=modelo B-MARS grado 3. Los marcadores de la gráfica superior son iguales para las gráficas inferiores.

En la Fig. 42 puede observarse que el número de términos en los modelos B-MARS disminuye conforme aumenta la complejidad de las funciones base para un valor de d fijo. Las gráficas comparativas del EPAP muestran que los modelos de regresión tuvieron un mejor desempeño en la predicción del color que en la predicción del contenido de cenizas. Así mismo, la diferencia entre los valores obtenidos y los esperados tendió a disminuir conforme aumentaba la complejidad de las funciones usadas en la construcción del modelo para ambos parámetros de control de calidad del azúcar.

En el caso de los modelos B-MARS de primer grado, el error obtenido en prueba es menor que el obtenido en los modelos MARS porque el soporte y forma de las B-spline de primer grado permiten que los valores obtenidos con los modelos se acerquen más a los valores esperados. El aumento del error para el caso de los modelos con funciones B-spline de segundo grado lo adjudicamos a la suavidad de las funciones base usadas en el ajuste, además de la disminución del número de términos que forman el modelo. Esta hipótesis se ve reforzada por los resultados obtenidos con el modelo con funciones base de tercer grado, cuyo intervalo de soporte se vuelve más estrecho y permite obtener valores más cercanos a los esperados.

8.6 Aplicación del preprocesamiento con la Transformada Wavelet Discreta y modelado con Redes Neuronales Artificiales

Los sensores químicos se usan normalmente en condiciones de estado estacionario, aunque recientes aplicaciones han explorado las ventajas de usar señales dinámicas adquiridas usando estos mismos sensores. Las señales dinámicas pueden obtenerse a partir de la cinética química o de la inmersión de los sensores en soluciones en flujo. Esta característica adicional representa una ventaja ya que la información suministrada por la respuesta dinámica de un sensor después de la inserción de muestras líquidas con un perfil escalón mejora la habilidad de discriminar entre la especie primaria de interés y las especies interferentes

Esta sección experimental describe el uso de señales transitorias con perfil escalón adquiridas con sensores potenciométricos acoplado a un sistema SIA. Este enfoque toma ventaja tanto de la sensibilidad cruzada en los sensores no específicos como de la respuesta dinámica para resolver las mezclas de analitos. La construcción de modelos de calibración a partir de respuestas dinámicas es difícil debido a que las señales aumentan en complejidad al añadir la dimensión cinética. La propuesta de solución a esta dificultad consiste en el preprocesamiento de los registros potenciométricos usando la Transformada Wavelet Discreta para reducir su dimensión y extraer características significativas antes de construir el modelo de calibración usando

Redes Neuronales Artificiales. Se eligió la DWT como técnica de preprocesamiento por el éxito obtenido en casos de estudio similares [5, 6]. La DWT permite reducir el tamaño de cada registro mediante una combinación de coeficientes obtenidos del procesamiento que permiten reconstruir la forma básica de la señal original sin incluir detalles finos, y que además hacen posible un modelado con ANNs.

Tanto el preprocesamiento mediante la DWT como el modelado con ANNs se hicieron usando los correspondientes Toolboxes de MATLAB. El esquema del acoplamiento entre las etapas de preprocesamiento y prueba que se ha probado en este experimental se muestra en la Fig. 43.

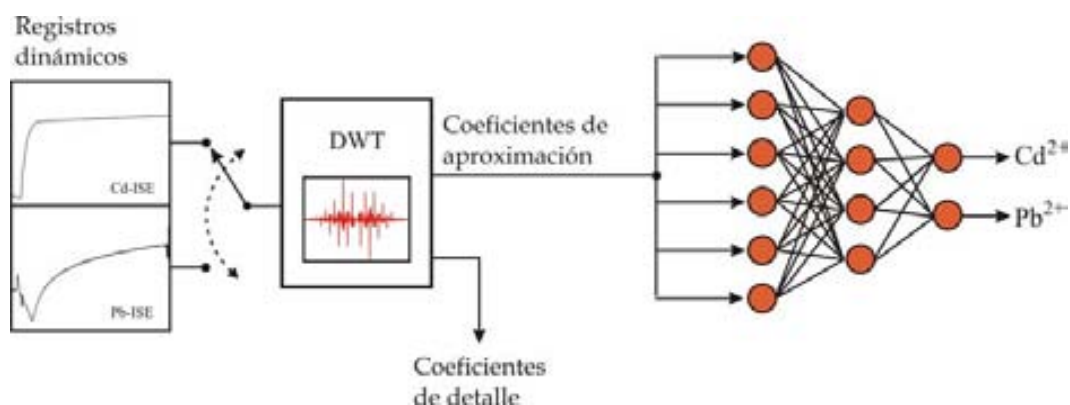


Fig. 43. Esquema del procesamiento Wavelet acoplado al modelado con ANN. Los coeficientes de aproximación obtenidos de la descomposición se usan para alimentar la red y construir el modelo de calibración.

8.6.1 Cuantificación de mezclas binarias de metales pesados a partir de la respuesta cinética de un sensor potenciométrico. Doble caso de estudio.

Para desarrollar este experimental se generaron aleatoriamente un total de 45 diferentes combinaciones de Cd^{2+} y Pb^{2+} para ser preparadas automáticamente por el sistema SIA y luego inyectadas a una celda de medida. Todas las soluciones se prepararon usando agua doblemente destilada y reactivos de grado analítico (o similar). Los estándares se prepararon a partir de una solución tampón de ácido acético y acetato con concentración de 0.05 M y pH de 4.5. Los analitos estuvieron en los intervalos $0\text{--}2.44 \times 10^{-4}$ M

para el Cd^{2+} y $0\text{--}4.97 \times 10^{-5}$ M para el Pb^{2+} . Los límites de las concentraciones quedaron determinados por el intervalo de trabajo óptimo de los sensores que se usaron [7].

La celda de medida estuvo formada por dos ISEs con respuesta cruzada a los metales pesados bajo estudio, además de un electrodo de Ag-AgCl como referencia. Uno de los ISE se construyó usando una membrana potenciométrica de Pb^{2+} con trioctilfosfina óxido como ionóforo, la otra se construyó usando un vidrio calcogenado selectivo a Cd^{2+} . Los ISEs se fabricaron en el Departamento de Química de la Universidad de San Petersburgo.

Las mezclas binarias de analitos se inyectaron al interior de la celda de medida con un perfil escalón, y la respuesta transitoria de cada ISE se adquirió de manera simultánea durante 60s en intervalos de 0.1s, por lo que al final se obtuvieron dos conjuntos de potenciales, uno por cada ISE. El universo inicial de datos estuvo formado por un tensor \mathbf{X} construido con dos matrices de entrada (una por cada ISE) formados por los registros transitorios (600 datos por registro) obtenidos con las 45 muestras, más una matriz \mathbf{Y} formada por las combinaciones binarias de concentraciones de metales pesados. En la Fig. 44 se muestra la distribución de las concentraciones de cadmio y plomo, y en la Fig. 45 se muestran los registros potenciométricos obtenidos con los ISEs además del detalle de dos registros (uno por cada sensor) donde se puede ver que el ISE con membrana de vidrio (el electrodo selectivo a Cd^{2+}) muestra una respuesta más rápida que el ISE con membrana polimérica (el electrodo selectivo a Pb^{2+}).

Las matrices de potenciales obtenidas con cada ISE se preprocesaron usando la TWD para reducir el número de variables antes de entrenar la ANN. Para escoger el procesamiento Wavelet óptimo para esta aplicación se probaron las 4 familias de Wavelets Discretas disponibles en MATLAB (Daubechies, Coiflets, Symlets y Splines Biortogonales) con niveles de descomposición que variaron de 1 a 9. El objetivo fue hallar la combinación *familia-nivel* que permitiese retener la mayor parte de la información original usando el menor número de coeficientes posible. El número total de combinaciones evaluadas fue 342, considerando que cada familia de Wavelets tiene miembros con varios órdenes.

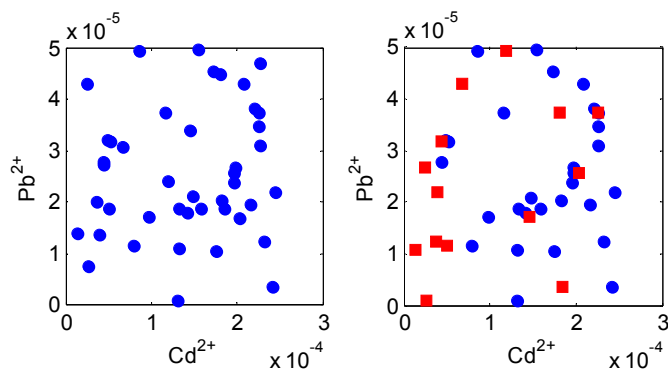


Fig. 44. Gráfica izquierda: distribución aleatoria de las concentraciones de iones Pb^{2+} y Cd^{2+} usados en esta sección experimental. Gráfica derecha: separación de concentraciones en elementos de modelizado (círculos azules) y prueba (cuadros rojos).

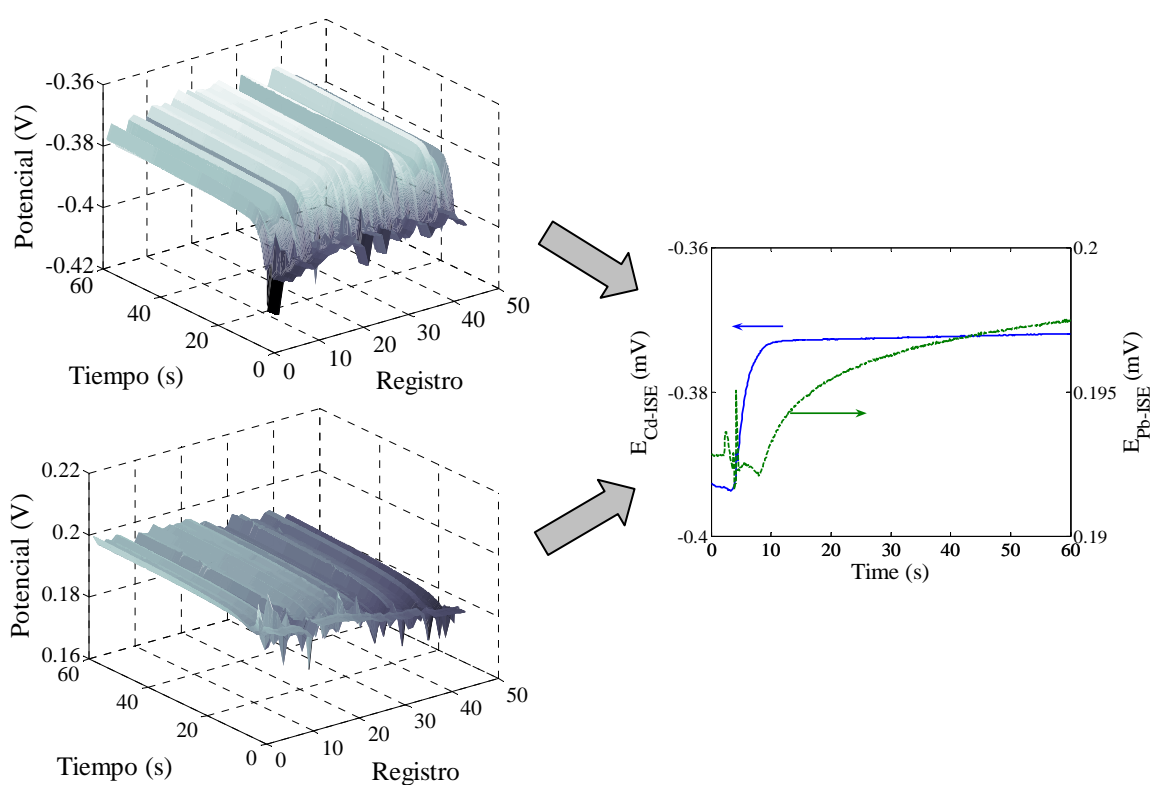


Fig. 45. En las gráficas a la izquierda se muestran los registros potenciométricos de cada ISE, en la gráfica derecha se muestran dos registros, uno por cada sensor, obtenidos con la mezcla de concentraciones $85.7\mu M$ para el Pb^{2+} y $49.4\mu M$ para el Cd^{2+} .

La combinación de Wavelet y nivel de descomposición usados en nuestra aplicación se eligió basándonos en la relación de compresión, porcentaje de energía total retenida por los coeficientes de aproximación después de cada descomposición y grado de similitud entre el registro original y el reconstruido usando los coeficientes retenidos. La relación de compresión es la relación entre la longitud original de los datos y el número de coeficientes de aproximación obtenidos después de la descomposición. La energía total se define como la suma de los coeficientes de detalle al cuadrado obtenidos en todas las escalas de descomposición más los coeficientes de aproximación al cuadrado obtenidos del último nivel de descomposición realizado. Se ha usado este segundo parámetro para evaluar la compresión debido a la pérdida de información que se tiene al desprestigiar coeficientes de detalle. Las gráficas de barras en la Fig. 46 muestran el número de coeficientes de aproximación obtenido con cada una de las 342 combinaciones descritas anteriormente. El grado de similitud entre las señales originales y reconstruidas se evaluó realizando una prueba que consiste en calcular la relación entre el área intersectada por uno de los registros de potencial y el área total bajo los registros. El resultado de la comparación es un número que varía entre 0 y 1 dependiendo del grado de similitud, es 0 cuando las dos señales no tienen nada en común y aumenta hacia 1 conforme aumenta la similitud entre ellas. Expresado como una operación de teoría de conjuntos y usando las variables A y B como las áreas bajo cada curva, el factor de comparación puede expresarse como $f_c = (A \cap B) / (A \cup B)$. El grado de similitud obtenido entre las señales originales y reconstruidas varió de 0.746 para los registros procesados con la Wavelet Coiflets de orden uno, hasta 0.999 para los registros procesados con la Wavelet Spline Biortogonal.

A partir de las transformaciones realizadas se determinó que los coeficientes de aproximación obtenidos con cualquier función Wavelet para los niveles 1 a 4 y 6 a 9 no se usarían en el modelado ya que *i*) la gran cantidad de coeficientes obtenidos con los niveles 1 a 4 hicieron que el entrenamiento de la red fuese muy tardado, *ii*) los coeficientes del nivel 6 no hicieron que la red convergiera al error deseado en entrenamiento, y *iii*) las señales reconstruidas con los coeficientes de los niveles 6 a 9 tuvieron grados de similitud por debajo de 0.95. Este último valor se impuso como umbral mínimo para asegurar una representación fiable de los registros a través de los coeficientes de

aproximación. Debido a esta restricción solamente se usaron los coeficientes de descomposición del nivel 5 para el entrenamiento de las redes. La función Wavelet que tuvo el mejor desempeño fue la Wavelet Spline Biortogonal de orden 3 para la descomposición y orden 1 para la reconstrucción (*bior3.1*, de acuerdo a la nomenclatura de MATLAB) que produjo un número de coeficientes de aproximación de 23. Las gráficas de la Fig. 47 muestran el grado de similitud entre los registros potenciométricos originales y reconstruidos usando Wavelets de orden 3, el cual varió de 0.985 para el ISE de Cd^{2+} hasta 0.991 para el ISE de Pb^{2+} . Solamente se muestra el tercer orden en las gráficas ya que la Wavelet elegida para la compresión es de orden 3 para el proceso de descomposición.

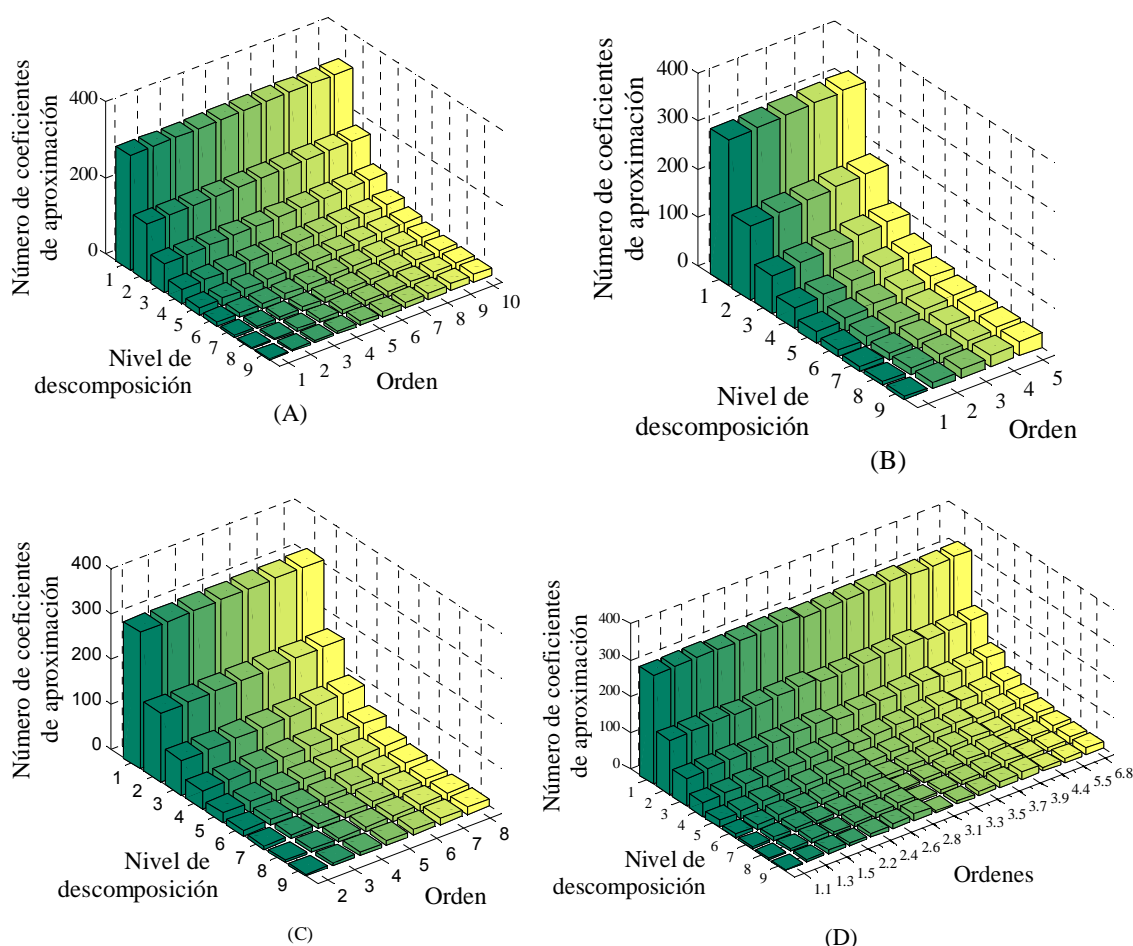


Fig. 46. Coeficientes de aproximación obtenidos con nueve niveles de descomposición usando las familias de Wavelet Discretas (A) Daubechies, (B) Coiflets, (C) Symlets y (D) Biortogonal. Todas las Wavelets se probaron en todos los órdenes disponibles. En la familia de Splines Biortogonales el primer número del orden corresponde al filtro de descomposición y el segundo al de reconstrucción.

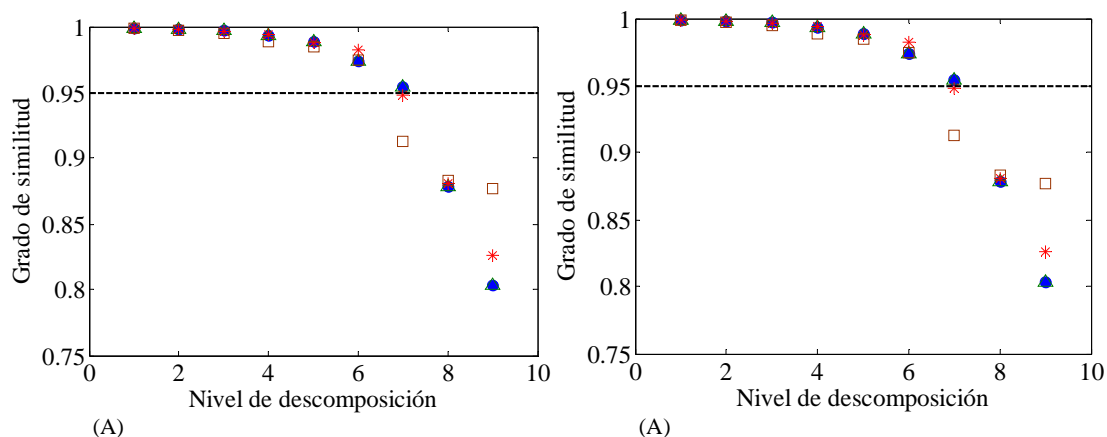


Fig. 47. Grados de similitud entre los registros reconstruidos y los originales obtenidos con (A) el sensor de cadmio y (B) el sensor de plomo. La línea segmentada marca el valor de 0.95 impuesto como límite. Los marcadores señalizan la familia Wavelet: (●) Daubechies, (▲) Symlets, (+) Coiflets, (□) Spline Biortogonal.

Para construir el modelo de calibración se probaron varias estructuras de redes neuronales del tipo feedforward con el objetivo de hallar la óptima para nuestra aplicación. El algoritmo de entrenamiento usado en todas las redes fue el de regularización Bayesiana. Los coeficientes de aproximación obtenidos del procesamiento Wavelet se usaron como entradas a la red y las concentraciones de cadmio y plomo como objetivos a modelizar. Las redes que se entrenaron para construir el modelo de calibración tuvieron estructuras con una o dos capas ocultas y una capa de salida. Las funciones de activación en la capa oculta fueron del tipo sigmoideal mientras que la de la capa de salida fue lineal. El número de neuronas para la estructura de una capa oculta varió de 6 a 30, y para el caso de dos capas ocultas la combinación de neuronas varió de 6-24 a 24-6. Estas combinaciones se obtuvieron incrementando en uno el número de neuronas en la primera capa oculta a la vez que se decrementaba en uno el número de neuronas en la segunda capa oculta. Se propusieron también estructuras con una sola neurona de salida para modelar independientemente cada ión metálico, así como estructuras con dos neuronas de salida para cuantificar simultáneamente el cadmio y el plomo.

Los conjuntos de coeficientes Wavelet y las concentraciones se dividieron en dos subconjuntos para entrenamiento y prueba. El 75% de la información se usó para determinar los parámetros del modelo y el resto se usó para probar la capacidad de generalización. La convergencia del entrenamiento se evaluó monitorizando la Suma de Errores al Cuadrado de la diferencia entre los

valores esperados y los obtenidos con la red en cada iteración. Como resultados de experiencias anteriores en la construcción de modelos de calibración, el error a alcanzar se estableció en 0.09, que es equivalente a una tolerancia de hasta el 5% en las diferencias entre los valores obtenidos y los esperados.

Ninguna de las redes con dos neuronas de salida (estructuras con una y dos capas ocultas) tuvieron un buen desempeño en la etapa de prueba aunque el 50% de ellas alcanzaron el error programado en entrenamiento. En estos casos, el coeficiente de correlación obtenido en prueba estuvo por debajo de 0.6 y en algunos casos fue negativo. Las redes con una salida tuvieron un mejor desempeño que las redes con dos salidas. La estructura que consiguió el mejor modelado estuvo construida con 10 neuronas en la primera capa oculta y 4 neuronas en la segunda capa oculta, ambas con función de transferencia sigmooidal tangencial, y una neurona en la capa de salida con función de transferencia lineal. La determinación simultánea de analitos se realizó usando dos redes en paralelo, ambas con la estructura anteriormente mencionada.

La efectividad de las redes entrenadas se evaluó haciendo un análisis de regresión lineal entre los valores obtenidos y esperados tanto para las redes entrenadas con la información extraída usando el electrodo de Cd^{2+} como para las entrenadas con la información extraída usando el electrodo de Pb^{2+} . La Fig. 48 condensa en cuatro gráficas los resultados obtenidos con el modelo de calibración para el electrodo de Cd^{2+} , la Fig. 49 contiene las gráficas equivalentes para el electrodo de Pb^{2+} .

La efectividad de la metodología se corroboró con 25 entrenamientos adicionales de validación cruzada por cada electrodo. Los datos de entrada y salida se dividieron aleatoriamente en cinco partes, cuatro quintos fueron tomados para entrenamiento y la quinta parte restante para prueba. Como en el primer caso del modelado, los datos de entrenamiento se normalizaron primero antes de normalizar los datos de prueba por cada uno de los entrenamientos. El error programado en el entrenamiento se alcanzó en todos los casos. Las Tablas 21 y 22 muestran los valores promedio de SSE y los coeficientes de correlación obtenidos con los ISEs de Cd^{2+} y Pb^{2+} , respectivamente. La incertidumbre que se indica en las Tablas corresponde al 95% del intervalo de confianza. Los SSEs obtenidos en prueba variaron desde 0.259 ± 0.029 para la

predicción de plomo hasta 0.725 ± 0.049 para la predicción de cadmio, ambos obtenidos usando el ISE de Pb^{2+} .

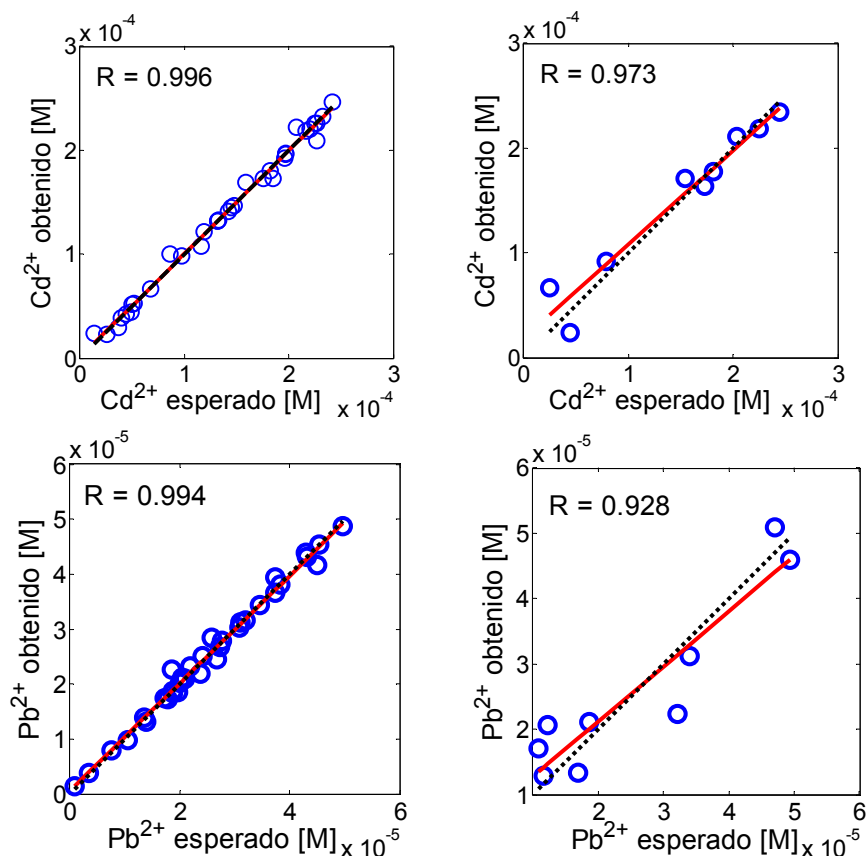


Fig. 48. Gráficas comparativas de los valores de concentración obtenidas vs. esperadas para el cadmio (gráficas superiores) y plomo (gráficas inferiores) usando los registros potenciométricos obtenidos con el ISE de Cd^{2+} . La línea punteada corresponde al ajuste ideal y la línea sólida al ajuste obtenido.

Tabla 21. Valores promedio de la SSE y coeficiente de correlación (R) obtenidos con el procedimiento de validación cruzada usando el ISE de Cd^{2+} .

Analito	Entrenamiento		Prueba	
	SSE	R	SSE	R
Cd^{2+}	$0.0980 \pm 3 \times 10^{-4}$	$0.996 \pm 9.9 \times 10^{-4}$	0.544 ± 0.062	0.917 ± 0.019
Pb^{2+}	$0.0977 \pm 5 \times 10^{-4}$	$0.995 \pm 1 \times 10^{-4}$	0.715 ± 0.044	0.844 ± 0.029

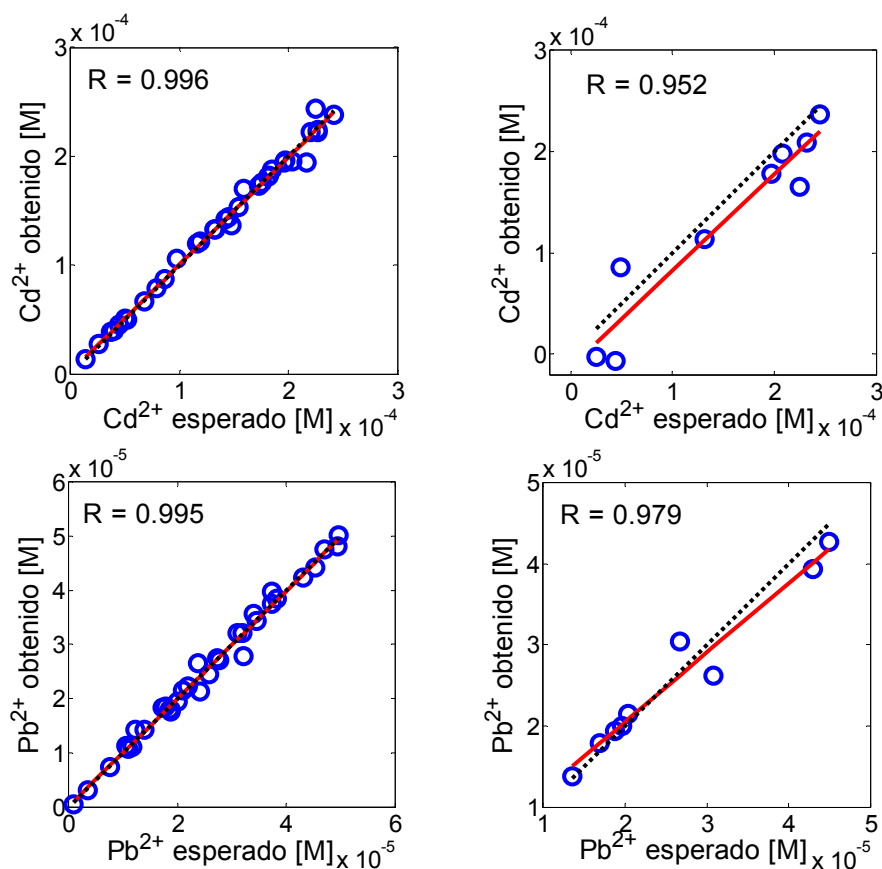


Fig. 49. Gráficas comparativas de los valores de concentración obtenidas vs. esperadas para el cadmio (gráficas superiores) y plomo (gráficas inferiores) usando los registros potenciométricos obtenidos con el ISE de Pb^{2+} . La línea punteada corresponde al ajuste ideal y la línea sólida al ajuste obtenido.

Tabla 22. Valores promedio de la SSE y coeficiente de correlación (R) obtenidos con el procedimiento de validación cruzada usando el ISE de Pb^{2+} .

Analito	Entrenamiento		Prueba	
	SSE	R	SSE	R
Cd^{2+}	$0.0978 \pm 5 \times 10^{-4}$	$0.996 \pm 3 \times 10^{-4}$	0.725 ± 0.049	0.926 ± 0.004
Pb^{2+}	$0.0976 \pm 4 \times 10^{-4}$	$0.995 \pm 1 \times 10^{-4}$	0.259 ± 0.029	0.934 ± 0.031

La combinación del procesamiento de registros electroquímicos usando la Transformación Wavelet Discreta, y la posterior construcción de un modelo de calibración a partir de la información obtenida de este procesamiento, ha demostrado ser eficiente. Por un lado, la reducción de la información usada para entrenar la red permite reducir el costo computacional, por otro lado, la

información cinética de la interferencia producida en los sensores se ha aprovechado para poder hacer la cuantificación bi-analito a partir de un solo sensor. La interferencia se ve reflejada en cada uno de los sensores cuando miramos los valores de SSE obtenidos en prueba. Para el sensor de Cd^{2+} , el SSE mayor se obtuvo en la detección de Pb^{2+} , y para el sensor de Pb^{2+} , el mayor SSE ha resultado de la detección de Cd^{2+} .

8.6.2 Cuantificación de mezclas ternarias de metales pesados a partir de la respuesta cinética de una matriz de sensores potenciométricos

En este apartado se presenta la construcción de una lengua electrónica para la cuantificación simultánea de triadas de iones Cd^{2+} , Cu^{2+} y Pb^{2+} a partir de registros transitorios obtenidos con un arreglo de tres ISEs sensibles a estos iones y sometidos a la inyección de muestras con un perfil escalón [55]. Se ha usado de nuevo la respuesta dinámica de los sensores por su probada utilidad en la discriminación de iones primarios de especies interferentes debido a la resolución cinética añadida al sistema.

Los estándares de este experimental también se prepararon con el sistema SIA usando un tampón de ácido acético y acetato con una concentración 0.05M y pH 4.5 pero los analitos estuvieron en los intervalos 1.05 μM a 29.8 μM para el Cd^{2+} , 0.05 μM a 1.5 μM para el Cu^{2+} , y 0.55 μM a 14.7 μM para el Pb^{2+} ; estos límites fueron determinados por los intervalos óptimos de trabajo de los sensores usados. Todas las soluciones (sales de nitrato) se prepararon usando agua doblemente destilada y reactivos de grado analítico o similar.

La celda de medida estuvo integrada por tres ISEs con respuesta cruzada a los iones metálicos bajo estudio más un electrodo de referencia de Ag-Ag/Cl . Los electrodos sensibles a Cu^{2+} y Cd^{2+} se fabricaron usando membranas de vidrio calcogenuro específicas (Universidad de San Petersburgo, Rusia), mientras que el electrodo sensible a Pb^{2+} se fabricó usando una membrana de PVC con óxido de trioctilfosfina como ionóforo.

Al igual que en el experimental anterior, la respuesta transitoria obtenida de cada ISE se adquirió simultáneamente durante 60 s por muestra en intervalos de tiempo de 0.1 s. De estas mediciones se obtuvieron tres matrices de potenciales formadas cada una de ellas por 81 registros de potenciales transitorios con 600 puntos de voltaje cada uno, además de la matriz de concentraciones formada por las 81 triadas de iones metálicos. En la Fig. 50 se muestran todas las concentraciones de iones metálicos generadas de manera aleatoria para este estudio. En la Fig. 51 se muestran las matrices de registros potenciométricos obtenidos con los tres sensores, además del detalle de un registro por cada ISE para mostrar su comportamiento. La división de las concentraciones de iones metálicos y sus correspondientes registros potenciométricos se hizo como se describe a continuación: el subconjunto formado por parejas con un índice originalmente par se usó para construir el modelo mientras que el subconjunto de las parejas con un índice originalmente impar se usó para probar la capacidad de generalización. No se hicieron mayores consideraciones para dividir los datos debido a que los valores de las concentraciones fueron generados inicialmente de manera aleatoria.

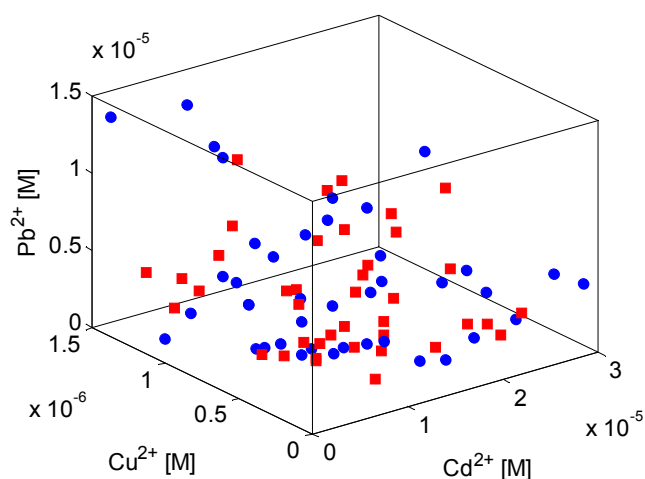


Fig. 50. Distribución espacial de las 81 concentraciones de iones metálicos obtenidos aleatoriamente para ser preparados por el sistema SIA. Las triadas usadas para construir el primer modelo de calibración (círculos azules) son diferenciadas de las triadas usadas para probar la capacidad de generalización (cuadros rojos). Los puntos de calibración y prueba usados en la validación cruzada son diferentes de los que se muestran aquí.

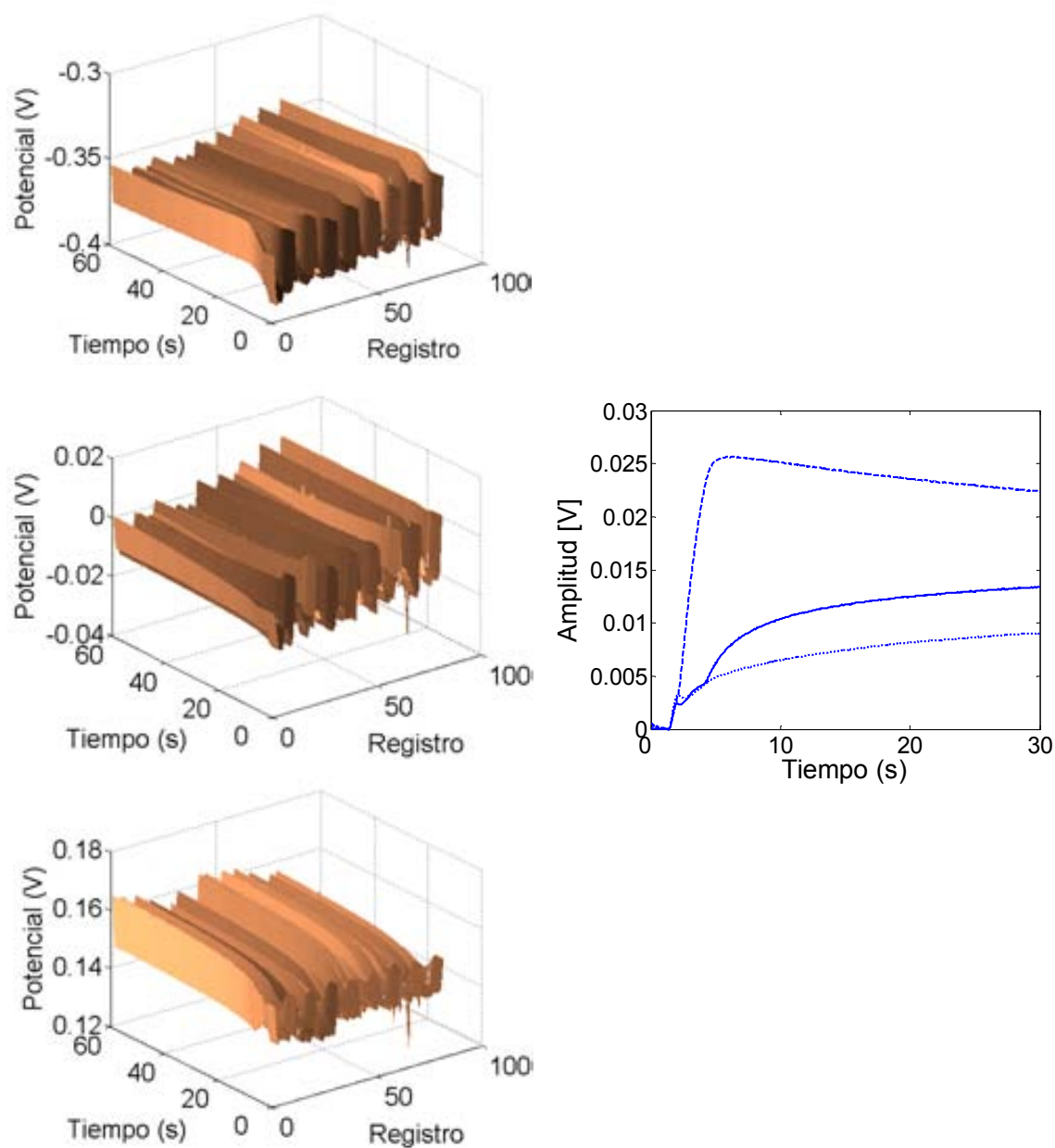


Fig. 51. Ejemplo de registros transitorios obtenidos con los ISE de Cd (línea punteada), Cu (línea continua) y Pb (puntos) después de la inserción de una muestra al interior de la celda con un perfil escalón. Los registros corresponden a las concentraciones de analitos $[2.65 \mu\text{M}, 1.065 \mu\text{M}, 4.7 \mu\text{M}]$ para Cd^{2+} , Cu^{2+} and Pb^{2+} , respectivamente. Los trazos han sido escalados y desplazados a un cero común para propósitos de comparación.

Para comprimir la información mediante la DWT se probaron las familias de Wavelets discretas Dubechies (db), Coiflets (coif), Splines Biortogonales (bior) y Symlets (sym) de diferentes órdenes y con niveles de descomposición

que variaron de 1 a 8, siendo los tres niveles más altos para ciertas Wavelets los que produjeron la menor combinación de coeficientes Wavelet y la más eficiente. La inspección visual de las señales originales y reconstruidas indican que una combinación de hasta 20 coeficientes de aproximación + detalle son suficientes para alcanzar el propósito de una reconstrucción con una elevada confiabilidad. La elección en el número de coeficientes Wavelet se confirmó calculando el error entre las señales originales y reconstruidas. La Tabla 23 resume la información relacionada con las Wavelets y los niveles de descomposición que permitieron una reconstrucción fiable con un total de hasta 20 coeficientes. Los registros electroquímicos finalmente se procesaron usando la Wavelet Spline Biortogonal de orden 2 para la descomposición y orden 2 para la reconstrucción (*bior2.2*, de acuerdo a la nomenclatura de MATLAB) y llevando la descomposición hasta el nivel 7. Esta combinación de Wavelet/nivel de descomposición permitió minimizar la diferencia entre los registros originales y los reconstruidos usando pocos coeficientes. Por cada registro potenciométrico procesado se retuvieron dieciséis coeficientes (9 de aproximación y 7 de detalle), lo que permitió obtener una relación de compresión de 37.5. La matriz de datos final usada como entrada para la construcción del modelo de calibración se obtuvo concatenando los coeficientes Wavelet obtenidos de cada matriz de datos de los sensores Cu^{2+} , Cd^{2+} y Pb^{2+} . El tamaño final de la matriz de coeficientes fue de [48×81].

Las Redes Neuronales Artificiales entrenadas fueron del tipo feedforward con una capa oculta y una capa de salida. Se probaron estructuras con diferente número de neuronas en la capa oculta y dos tipos de funciones de transferencia sigmoidales (logarítmica y tangencial). La capa de salida tuvo tres neuronas para modelar simultáneamente los analitos y función de transferencia lineal. El nuevo conjunto de datos formado por los coeficientes Wavelet y las concentraciones se dividió en dos subconjuntos con el 50% del total de la información cada uno para propósitos de entrenamiento y prueba. El subconjunto formado por las parejas coeficientes/concentraciones con índice impar se usó para entrenar la red mientras que el subconjunto con índice par se usó para probar la capacidad de generalización. No se hicieron mayores consideraciones para dividir los datos debido a que los valores de las concentraciones fueron generados inicialmente de manera aleatoria. El entrenamiento se realizó con el algoritmo de regulación Bayesiana.

Tabla 23. RMSEs obtenidos al comparar los registros originales /vs./ los reconstruidos usando unos pocos coeficientes Wavelet. En la tabla se muestran solamente las combinaciones de Wavelet/Orden/Nivel que produjeron un RMSE usando un máximo de 20 coeficientes de aproximación/detalle.

Wavelet	Orden	Nivel	Número de coeficientes	RMSE
Daubechies (db)	2	6	19	$1.35 \times 10^{-4} \pm 1.62 \times 10^{-5}$
		7	16	$1.47 \times 10^{-4} \pm 1.59 \times 10^{-5}$
	3	6	20	$1.60 \times 10^{-4} \pm 2.02 \times 10^{-5}$
		7	19	$1.61 \times 10^{-4} \pm 2.02 \times 10^{-5}$
		8	20	$1.70 \times 10^{-4} \pm 2.00 \times 10^{-5}$
Symlets (sym)	2	6	19	$1.35 \times 10^{-4} \pm 1.62 \times 10^{-5}$
		7	16	$1.47 \times 10^{-4} \pm 1.59 \times 10^{-5}$
		8	18	$1.46 \times 10^{-4} \pm 1.59 \times 10^{-5}$
	3	6	20	$1.60 \times 10^{-4} \pm 2.02 \times 10^{-5}$
		7	19	$1.61 \times 10^{-4} \pm 2.02 \times 10^{-5}$
Coiflets (coif)	1	6	19	$1.17 \times 10^{-4} \pm 1.15 \times 10^{-5}$
		7	18	$1.27 \times 10^{-4} \pm 1.13 \times 10^{-5}$
		8	19	$1.41 \times 10^{-4} \pm 1.11 \times 10^{-5}$
Biortogonal (bior)	2.2	6	19	$1.15 \times 10^{-4} \pm 1.28 \times 10^{-5}$
		7	16	$1.26 \times 10^{-4} \pm 1.27 \times 10^{-5}$
	3.1	6	15	$2.22 \times 10^{-4} \pm 2.83 \times 10^{-5}$
		7	12	$2.24 \times 10^{-4} \pm 2.88 \times 10^{-5}$
		8	13	$2.26 \times 10^{-4} \pm 2.94 \times 10^{-5}$
	3.3	6	20	$1.78 \times 10^{-4} \pm 2.04 \times 10^{-5}$
		7	19	$1.48 \times 10^{-4} \pm 1.87 \times 10^{-5}$
		8	20	$1.61 \times 10^{-4} \pm 1.89 \times 10^{-5}$

La convergencia de la red durante el entrenamiento se monitorizó usando la suma de errores al cuadrado (SSE) obtenido con las diferencias entre los valores esperados y los obtenidos. Se probaron estructuras de ANNs con un número de neuronas ocultas que varió desde 3 hasta 20 con un SSE a alcanzar en entrenamiento de 2.1. Todas las redes alcanzaron el error programado en entrenamiento además que no se hallaron tendencias o diferencias significativas en el error de prueba para las estructuras con diferente número de neuronas ocultas y diferentes funciones de transferencia no-lineales (Fig. 52). El modelo de calibración final se construyó con 10 neuronas en la capa oculta usando ambas funciones de transferencia sigmoidales en la capa oculta. El número de neuronas y el error en entrenamiento se eligieron para poder comparar los resultados obtenidos con este modelado contra los resultados obtenidos con un estudio posterior basado en la herramienta *N-PLS2*. Las gráficas de la Fig. 53 resumen el

comportamiento de la red neuronal con función de transferencia logarítmica sigmoideal; las gráficas de la Fig. 54 corresponden a la red con función de transferencia tangencial sigmoideal.

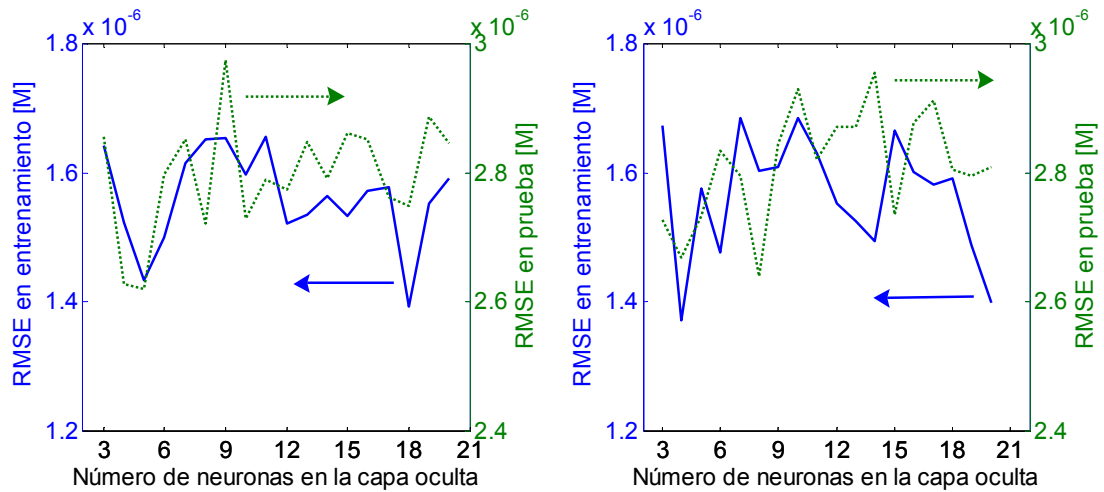


Fig. 52. Valores de RMSE obtenidos con las ANNs con número variable de neuronas ocultas desde 3 hasta 20 y funciones de transferencia logarítmica sigmoideal (izquierda) y tangencial sigmoideal (derecha).

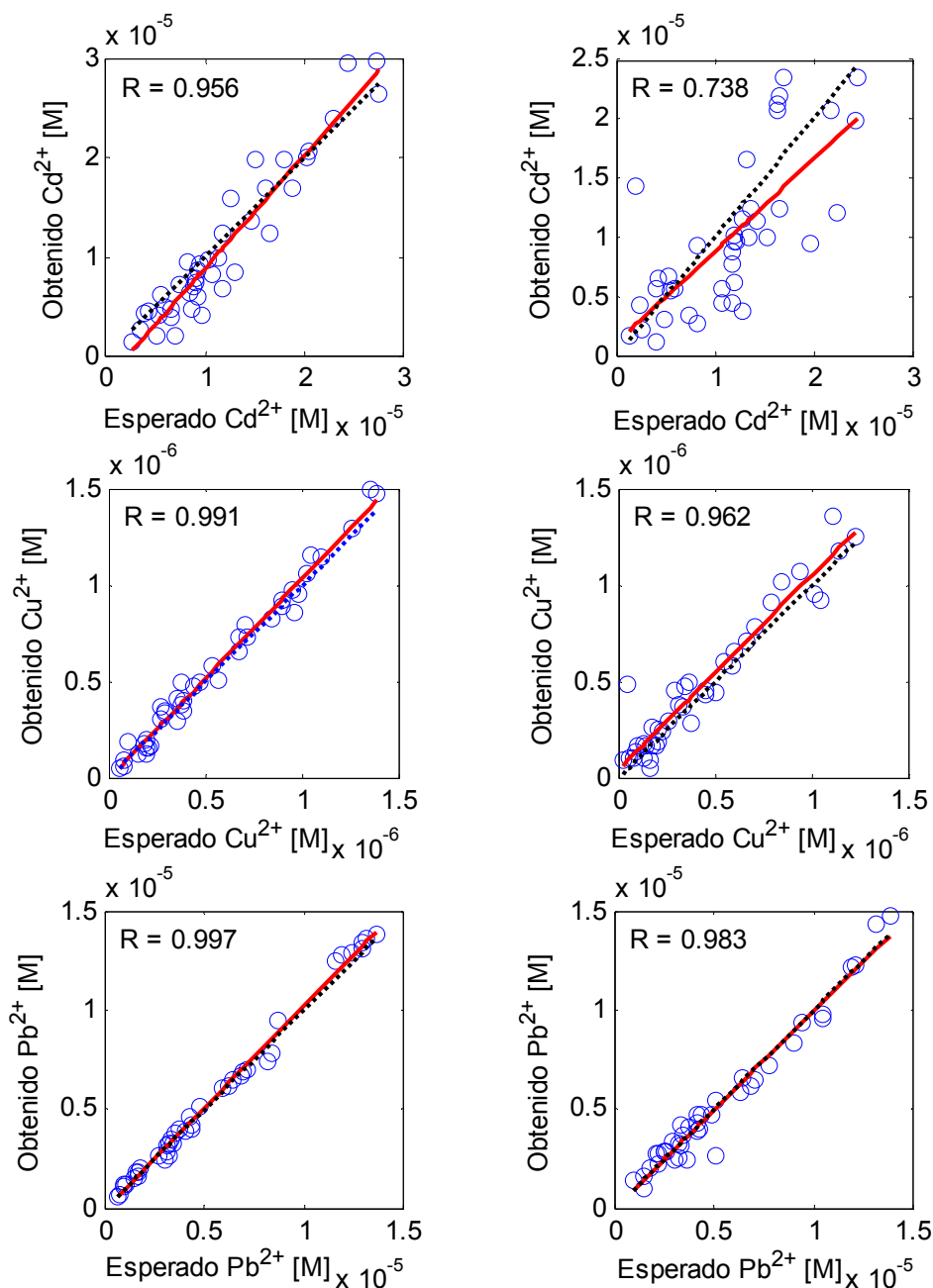


Fig. 53. Gráficas comparativas de los valores de concentraciones esperados vs obtenidos con la ANN con función de transferencia logarítmica sigmoial. La línea punteada corresponde al ajuste ideal y la línea sólida al ajuste obtenido. Las gráficas de la columna izquierda corresponden al entrenamiento, las de la columna derecha a la prueba.

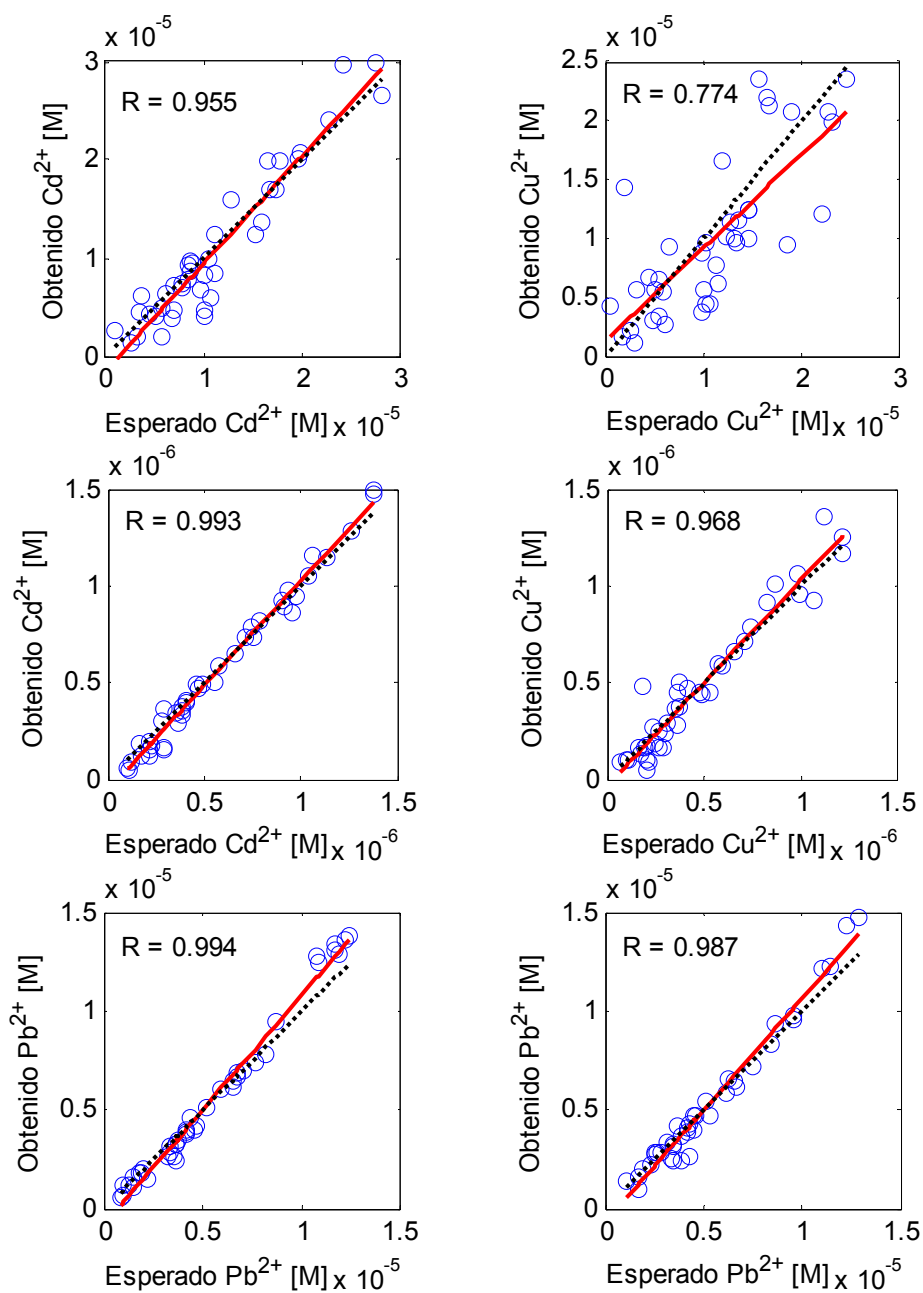


Fig. 54. Gráficas comparativas de los valores de concentraciones esperados vs obtenidos con la ANN con función de transferencia tangencial sigmoideal. La línea punteada corresponde al ajuste ideal y la línea sólida al ajuste obtenido. Las gráficas de la columna izquierda corresponden al entrenamiento, las de la columna derecha a la prueba.

Para corroborar el desempeño de la estructura escogida se realizaron 15 entrenamientos adicionales de validación cruzada. En cada uno de los entrenamientos se dividieron aleatoriamente las parejas de *coeficientes Wavelet/triadas de concentraciones* en dos subconjuntos para propósitos de

entrenamiento y prueba. La división se hizo tomando 41 parejas para entrenamiento y 40 para prueba. Los índices que identificaban la posición original de las parejas en cada nuevo subconjunto creado durante el proceso de validación cruzada se guardaron para ser usados en la siguiente sección donde se describirá la calibración con *N-PLS2*. El error a alcanzar se estableció en 2.1 como en los casos anteriores y se alcanzó en todos los entrenamientos realizados. Los valores RMSE promedios obtenidos en los entrenamientos de las ANNs fueron $2.42 \pm 0.15 \mu\text{M}$, $61.0 \pm 6.7 \text{nM}$ y $0.51 \pm 0.08 \mu\text{M}$ para el Cu^{2+} , Cd^{2+} y Pb^{2+} , respectivamente, para las redes con función de transferencia logarítmica sigmoideal en las neuronas de la capa oculta. La capacidad de generalización de las mismas redes entregó valores de RMSE promedios para el subconjunto de prueba de $5.76 \pm 0.43 \mu\text{M}$, $162 \pm 25 \text{nM}$ y $0.91 \pm 0.097 \mu\text{M}$ para el Cu^{2+} , Cd^{2+} y Pb^{2+} , respectivamente. Los valores de RMSE obtenidos en entrenamiento con las redes con función de transferencia tangencial sigmoideal fueron de $2.41 \pm 0.16 \mu\text{M}$, $53.4 \pm 7.3 \text{nM}$ y $0.47 \pm 0.01 \mu\text{M}$ para el Cu^{2+} , Cd^{2+} y Pb^{2+} , respectivamente; y de $5.69 \pm 0.42 \mu\text{M}$, $162 \pm 23 \text{nM}$ y $0.88 \pm 0.094 \mu\text{M}$ en prueba para el Cu^{2+} , Cd^{2+} y Pb^{2+} , respectivamente. Todas las incertidumbres que se han indicado fueron calculadas al 95% del intervalo de confianza. Los valores de concentración obtenidos con las redes entrenadas se compararon contra los correspondientes valores de concentraciones esperados realizando un análisis de regresión lineal. Los valores promedios de pendientes, intercepciones y coeficientes de correlación obtenidos de la validación cruzada para los procesos de entrenamiento y prueba están contenidos en la Tabla 24.

La determinación tri-analito realizada en este experimental refuerza la efectividad de la combinación formada por el procesamiento wavelet seguido de la modelización con redes neuronales. El procedimiento permitió obtener buenas correlaciones entre los valores obtenidos y esperados para los tres analitos, especialmente para Cu^{2+} y Pb^{2+} , cuyas gráficas en las Figs. 46 y 47, muestran menor dispersión que las gráficas para Cd^{2+} . Esta dispersión puede atribuirse a la interferencia que los iones de Cu^{2+} y Pb^{2+} causan en el sensor de Cd^{2+} , el menos selectivo de los tres ISEs.

Tabla 24. Valores promedios de pendiente (*m*), intercepción (*b*) y coeficiente de correlación (*R*) de las gráficas de comparación de concentraciones esperadas /vs./ obtenidas con la validación cruzada para el modelado DWT-ANN. Las etiquetas 'log' y 'tan' representan las funciones de transferencia logarítmica sigmoideal y tangencial, respectivamente. Los intervalos de confianza fueron calculados al 95%.

Analito		Modelado			Prueba		
		<i>m</i>	<i>b</i>	<i>R</i>	<i>m</i>	<i>b</i>	<i>R</i>
<i>Cd</i> ²⁺	log	0.74±0.205	(2.85±2.8)×10 ⁻⁰	0.95±0.007	0.55±0.039	(4.85±0.69)×10 ⁻¹	0.66±0.055
	tan	0.76±0.020	(3.01±0.34)×10 ⁻¹	0.95±0.006	0.56±0.045	(4.96±0.67)×10 ⁻¹	0.67±0.056
<i>Cu</i> ²⁺	log	0.90±0.201	(5.22±1.4)×10 ⁻⁰	0.99±0.001	0.83±0.050	(7.86±2.2)×10 ⁻⁰	0.90±0.027
	tan	0.91±0.023	(4.74±1.3)×10 ⁻⁰	0.99±0.001	0.85±0.052	(7.29±2.3)×10 ⁻⁰	0.91±0.027
<i>Pb</i> ²⁺	log	0.92±0.019	(3.14±1.3)×10 ⁻¹	0.99±0.001	0.88±0.031	(5.48±2.1)×10 ⁻¹	0.97±0.006
	tan	0.92±0.026	(3.06±1.1)×10 ⁻¹	0.99±0.001	0.88±0.036	(5.44±1.9)×10 ⁻⁰	0.97±0.006

8.7 Construcción de redes neuronales B-spline y Catmull-Rom para determinar compuestos fenólicos en la monitorización de la calidad del agua

Las redes neuronales B-spline y Catmull-Rom permiten construir modelos de calibración a partir de datos de entrada multidimensionales. Estas redes se probaron usando los conjuntos de datos descritos previamente en las Secciones 8.1 y 8.5.2. En estos conjuntos de datos, las mediciones de corriente o voltaje obtenidas con cada una de las muestras que forman el experimento forman un vector n -dimensional. La construcción de una red neuronal de este tipo que pueda cubrir un intervalo acotado $[a, b]$ por dimensión requiere la evaluación de $\prod_{k=1}^{N_k} I_k$ neuronas ocultas (Sección 4.4). Intentar construir una red neuronal B-spline o Catmull-Rom para 42 o 600 variables de entrada implicaría construir un sistema con una representación extremadamente compleja, por esa razón, debemos realizar primero una reducción del número de variables usando PARAFAC2 o Tucker3. Ambas herramientas son consideradas como métodos de descomposición de datos multi-modales (Sección 2.3). La ventaja de usar PARAFAC2 o Tucker3 respecto a la matrización de los datos y posterior descomposición en sus componentes principales usando PCA, es que los modelos obtenidos tienen menos grados de libertad, son más sencillos de interpretar y más pequeños que los obtenidos usando PCA.

El funcionamiento de la red B-spline se probó usando el conjunto de compuestos fenólicos descrito en la Sección 8.2. El arreglo \underline{X} formado por las dos matrices de corriente obtenidas con técnicas voltamperométricas usando los sensores de Pd y AuPd se preprocesaron con PARAFAC2 y Tucker3 para extraer los vectores de loadings de cada uno de los modos que expliquen la mayor parte de la variación contenida en \underline{X} . Los porcentajes de variación obtenidos con la extracción de 1 a 3 componentes están contenidos en la Tabla 25, donde se puede ver que la descomposición obtenida a partir de 1 componente permite explicar más del 90% de la variación existente en el arreglo \underline{X} . Las ligeras diferencias entre ambos métodos de descomposición son atribuibles a las diferencias en los grados de libertad entre ambos modelos; esta diferencia es visualmente imperceptible (Fig. 55).

Tabla 25. Porcentajes de variación explicada con las descomposiciones de 1 a 3 componentes hechas con PARAFAC2 y Tucker3.

Número de componentes	Porcentaje de variación explicada (%)	
	PARAFAC2	Tucker3
1	92.950	92.514
2	98.634	98.347
3	99.378	98.469

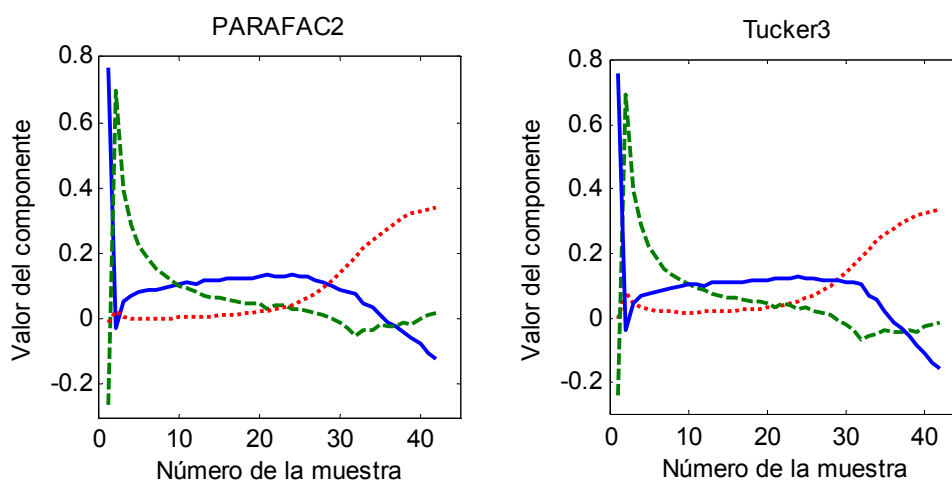


Fig. 55. Gráficas de los 3 componentes obtenidos del primer modo con las descomposiciones PARAFAC2 y Tucker3. La línea continua es el componente 1, la línea segmentada es el componente 2, y la línea punteada es el componente 3.

Debido a que la diferencia entre los componentes obtenidos con los dos métodos de descomposición es mínima, resulta prácticamente indistinto usar uno u otro método de descomposición. Se optó por usar los componentes obtenidos con PARAFAC2 para el entrenamiento de redes B-spline y se reservó la descomposición Tucker3 para el entrenamiento de la red spline Catmull-Rom.

Primero se construyeron redes neuronales B-spline mono-dimensionales con funciones base de grado 1, 3 neuronas de salida para la cuantificación simultánea de los 3 analitos, y un número variable de neuronas B-spline para intentar conseguir el ajuste del modelo. Se construyó primero una red mono-dimensional porque el primer componente obtenido de la descomposición es capaz de explicar más del 90% de la variación en el arreglo de predictores. Tomando como referencia las gráficas de la Sección 8.2 de los resultados obtenidos en entrenamientos con el mismo conjunto de datos, los nuevos

entrenamientos con las redes B-spline se hicieron con un valor de error a alcanzar en entrenamiento menos estricto, de un 15% de margen de error por cada analito predicho para permitir un ajuste más relajado de los datos de entrenamiento y a la vez intentar conseguir reducir las diferencias entre los resultados obtenidos y los esperados de los datos de prueba.

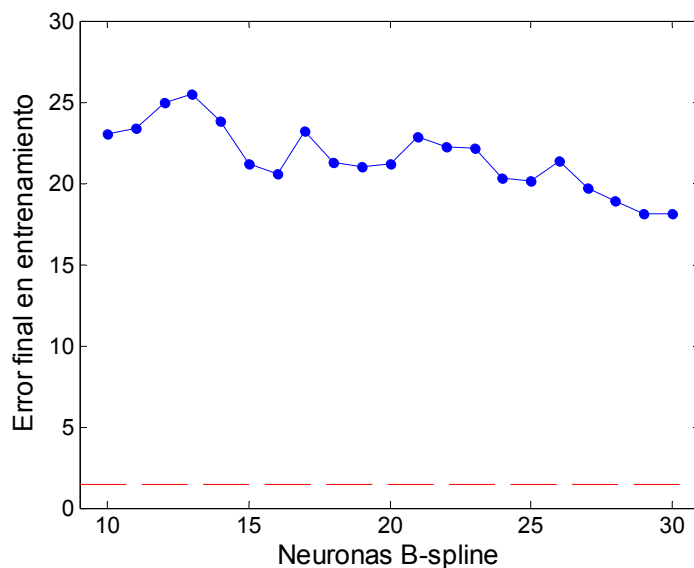


Fig. 56. Evolución del error obtenido en entrenamiento al finalizar las iteraciones programadas en las redes neuronales B-spline con función de primer grado, desde 10 hasta 30 neuronas. La línea segmentada fue el error programado para el entrenamiento.

Para el caso de la función B-spline de primer grado, las neuronas en la capa oculta variaron en número de 10 a 30 en intervalos de 1 neurona. Los entrenamientos hechos con estas redes no alcanzaron el error programado en ninguno de los casos. Los resultados del error obtenido con cada una de estas redes se muestran en la Fig. 56. No se presentan resultados de análisis de regresión lineal por la carencia de modelizados realizados con éxito.

La falta de éxito en los entrenamientos realizados podría ser el resultado del tipo de función B-spline usada en la aproximación o de la sencillez de la solución propuesta al intentar modelar simultáneamente 3 salidas usando un solo vector de entrada.

La relación entre predictores y respuestas que se intentó modelar en esta prueba fue una relación del tipo $\mathbb{R} \rightarrow \mathbb{R}^3$. Para evaluar si la escasez de grados de libertad del primer modelo es el principal motivo del desempeño del primer modelo, se construyeron redes bi- y tri-dimensionales con funciones B-spline de primero y segundo grado. Las entradas a estas redes fueron los dos y tres componentes obtenidos de la descomposición PARAFAC2.

Debido a las limitaciones de memoria para el manejo de las variables en Matlab V7.1 ejecutándose en Windows Vista OS, sólo se pudieron construir redes B-spline con las neuronas indicadas en la Tabla 26. Los intentos de aumentar el número de neuronas en las estructuras propuestas provocaron errores de falta de memoria.

Tabla 26. Estructuras que se pudieron implementar en Matlab debido a las limitaciones de memoria.

Grado de la función	Dimensiones de los datos de entrada a la red		
	1	2	3
	2	10-30	10-27
3	10-30	5-15	3-10

Todas las redes neuronales con funciones, dimensiones y número de neuronas descritas en la Tabla 26 tuvieron un desempeño deficiente en entrenamiento y ninguna consiguió alcanzar el error programado, aunque se observó una tendencia en la disminución del error conforme aumentaba la dimensión de entrada. Los resultados obtenidos con todas las estructuras descritas en la Tabla 26 se condensan en la Fig. 57.

La ligera mejora en el error de modelización que se observa en las gráficas de la Fig. 57 es el resultado de la combinación de incrementar la dimensión de los datos de entrada y de usar una función B-spline de grado más elevado. Sin embargo, el ajuste a los datos de prueba no consigue minimizarse porque la superficie e hipersuperficie descritas por las funciones bi- y tri-dimensionales, respectivamente, no se acercan a las concentraciones de prueba por la irregularidad de la distribución.

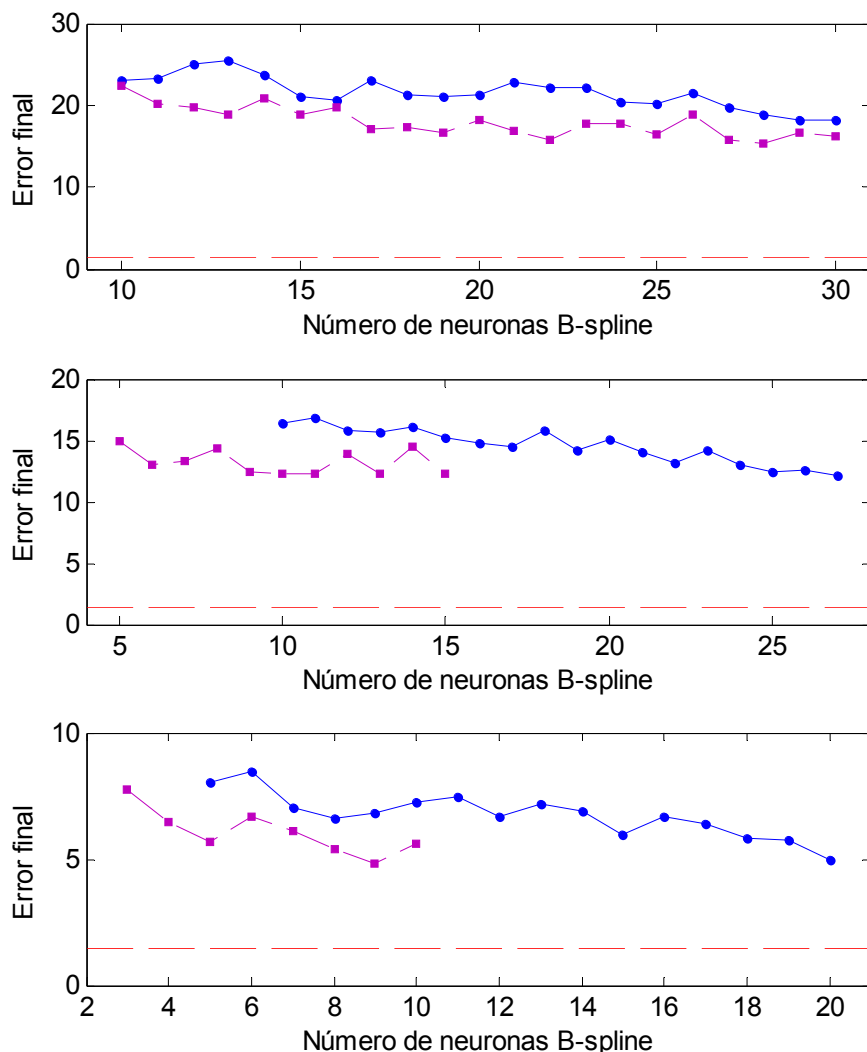


Fig. 57. Evolución del error en los entrenamientos de las redes B-spline. Los marcadores circulares corresponden a las redes con funciones de segundo grado, los marcadores cuadrados corresponden a las redes con funciones de tercer grado. La gráfica superior es la obtenida con las redes monodimensionales, la del medio es para redes bidimensionales y la inferior para redes tridimensionales. La línea segmentada es el error objetivo a alcanzar en entrenamiento.

En el entrenamiento de la red spline Catmull-Rom se usaron componentes de descomposición extraídos del conjunto de mezclas ternarias de metales de la Sección 8.5.2. El arreglo \underline{X} formado por las dos matrices de potenciales adquiridas con los electrodos de Cd^{2+} , Cu^{2+} , y Pb^{2+} se preprocesaron usando Tucker3. La información referente al porcentaje de variación explicada con los componentes obtenidos con la descomposición Tucker3 se encuentra en la Tabla 27. La descomposición se realizó hasta obtener 3 componentes por modo, aunque el primer componente ya permite

explicar el 99.95% de la variación contenida en el arreglo \underline{X} . Aún cuando el objetivo fue usar Tucker3 para la descomposición del arreglo \underline{X} , también se realizó la descomposición PARAFAC2 con el único propósito de comparar posibles diferencias sustanciales.

Tabla 27. Porcentajes de variación explicada con las descomposiciones de 1 a 3 componentes hechas con PARAFAC2 y Tucker3.

Número de componentes	Porcentaje de variación explicada (%)	
	PARAFAC2	Tucker3
1	99.957	99.945
2	99.987	99.985
3	99.996	99.996

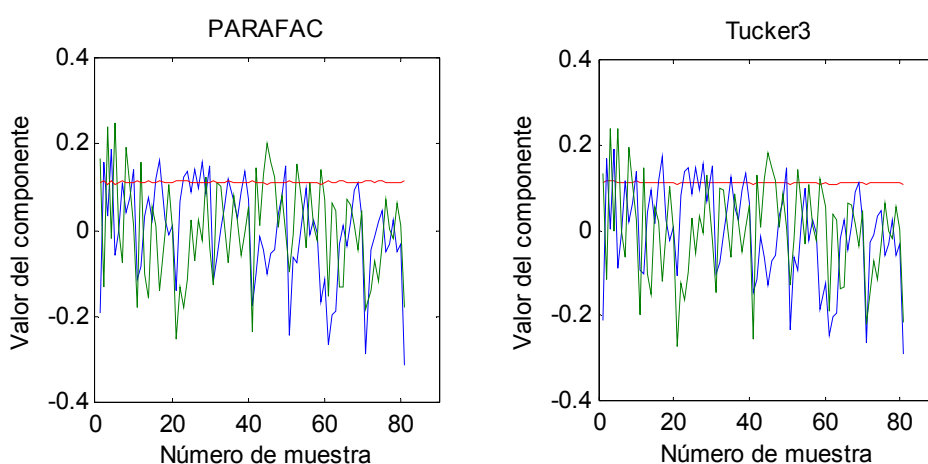


Fig. 58. Gráficas de los 3 componentes obtenidos del primer modo con las descomposiciones PARAFAC2 y Tucker3. La línea continua es el componente 1, la línea segmentada es el componente 2, y la línea punteada es el componente 3.

Los porcentajes de variación explicada de la Tabla 27 sugieren el uso de un solo componente para entrenar la red Catmull-Rom, por lo que se construyeron redes neuronales mono-dimensionales. Se dividió el intervalo acotado $[-1, 1]$ de entrada en un número variable de segmentos, desde 5 hasta 30, y se asignó una neurona por cada segmento. En las redes Catmull-Rom cada neurona intenta ajustar un segmento de las curvas descritas por el conjunto de concentraciones. El entrenamiento de las redes se hace con los valores de entrada, pesos en las conexiones de salida y puntos de control. Debido a la normalización de las concentraciones a predecir, también en el

intervalo $[-1, 1]$, los puntos de control se inicializaron a valores aleatorios en este mismo intervalo.

El desempeño de las redes Catmull-Rom presentó ligeras mejoras respecto a las redes B-spline en el sentido de que el error en entrenamiento se acercó más al valor programado al inicio del entrenamiento, aunque tampoco fue alcanzado. La evolución de la minimización del error puede visualizarse en la Fig. 59. No se presenta la evaluación del error con los datos de prueba debido a que los entrenamientos no se completaron con éxito.

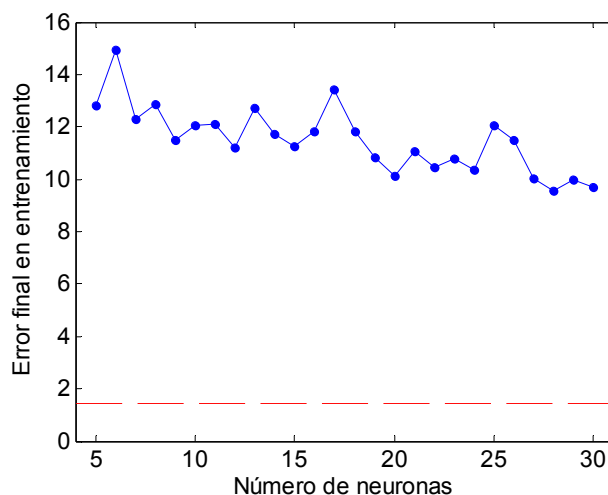


Fig. 59. Evolución del error obtenido en entrenamiento al finalizar las iteraciones programadas en las redes neuronales Catmull-Rom de 5 a 30 neuronas. La línea segmentada es el error objetivo a alcanzar en entrenamiento.

A pesar de que la gráfica de la Fig. 59 muestra claramente que el error en entrenamiento de la red Catmull-Rom tiende a disminuir conforme aumenta el número de neuronas en la capa oculta, las redes que se programaron para la predicción de iones metálicos no consiguieron alcanzar el error objetivo fijado en entrenamiento. Esto puede ser el resultado de dos condiciones. La primera condición es el reducido número de elementos usados como entrada para el entrenamiento de las redes. Usar sólo un componente de la descomposición Tucker3 incrementa la dificultad de poder realizar una calibración multianálito porque se intenta establecer una relación del tipo $\mathbb{R} \rightarrow \mathbb{R}^3$ con pocos grados de libertad. La segunda condición a la que puede deberse la falta de ajuste de la red es la carencia de pesos en las salidas de las neuronas ocultas. Si tomamos

en cuenta que durante el proceso de entrenamiento de una red Catmull-Rom lo que se modifica son los puntos de control y los pesos de entrada, entonces el ajuste simultáneo de tantas salidas como analitos se pretendan predecir resulta en un proceso lento que sería posible alcanzar si se sobreesatura con funciones spline el dominio del espacio de entrada.

8.8 Modelos N-PLS2 en el procesamiento multimodo de datos

En este apartado se describe la construcción de lenguas electrónicas multimodo usando la herramienta de calibración *N-PLS2* descrita en la Sección 2.3.3. La construcción de estos modelos de calibración a partir de N-PLS2 es posible porque los datos medidos nos permitieron ordenar la información en un tensor de tres modos descrito por las variables *Muestra_Potencial_Electrodo* o *Muestra_Corriente_Electrodo* dependiendo si la lengua electrónica construída es del tipo potenciométrico o voltamperométrico, respectivamente.

8.8.1 Lengua electrónica potenciométrica para la cuantificación de metales pesados

La primera lengua que se presenta permite la cuantificación simultánea de triadas de iones Cd^{2+} , Cu^{2+} y Pb^{2+} a partir de registros transitorios obtenidos con un arreglo de tres ISEs sensibles a estos iones. El desarrollo experimental para la obtención de la información fue detallado en la sub-sección 8.5.2, por lo que a continuación sólo describiremos la construcción del modelo, el proceso de validación y los resultados obtenidos.

Las tres matrices de registros potenciométricos obtenidas con cada uno de los ISEs se ordenaron formando un tensor de predictores \mathbf{X} de tamaño $81 \times 600 \times 3$ (81 muestras \times 600 puntos de voltaje \times 3 electrodos). Junto con la matriz de predictores también se formó una matriz \mathbf{Y} de tamaño 81×3 (81 muestras \times 3 iones metálicos). Los datos se procesaron con MATLAB 7.1 usando "*The N-way Toolbox for MATLAB V 3.1*", desarrollado por R. Bro de la Universidad de Copenhague.

Antes de construir los modelos con *N-PLS2* se verificó la existencia de trilinealidad en el arreglo de predictores. Para hacerlo se reordenó el tensor \underline{X} en tres matrices: una extendida por filas (81×1800), otra por columnas (48300×3) y una tercera por tubos (243×600). Cada matriz se factorizó usando la descomposición en valores singulares para evaluar la trilinealidad, si ésta existe entonces el número de autovalores significativos debe ser el mismo en cada matriz extendida [43]. Los tres primeros autovalores obtenidos de las matrices extendidas fueron [86.54; 1.47; 0.86], [86.53; 1.79; 0.84] y [86.55; 1.47; 0.86]. Se observa que de cada matriz aumentada se obtienen dos autovalores significativos, lo que confirma la trilinealidad de los datos. Los arreglos \underline{X} y \underline{Y} se dividieron en dos sub-conjuntos formados por parejas alternadas de predictores y respuestas. El subconjunto formado por parejas con índice par se usó para construir el modelo *N-PLS2* mientras que el subconjunto de las parejas con índice impar se usó para probar la capacidad de generalización. No se hicieron mayores consideraciones para dividir los datos debido a que los valores de las concentraciones fueron generados inicialmente de manera aleatoria.

El conjunto de datos usado para construir el modelo *N-PLS2* primero se preprocesó centrándolo a lo largo del modo uno, de acuerdo al procedimiento descrito por Gurden et al [44]. Los valores promedios obtenidos de esta etapa de preprocesamiento se usaron para centrar el subconjunto de datos de prueba. No se hizo escalamiento en ninguno de los modos.

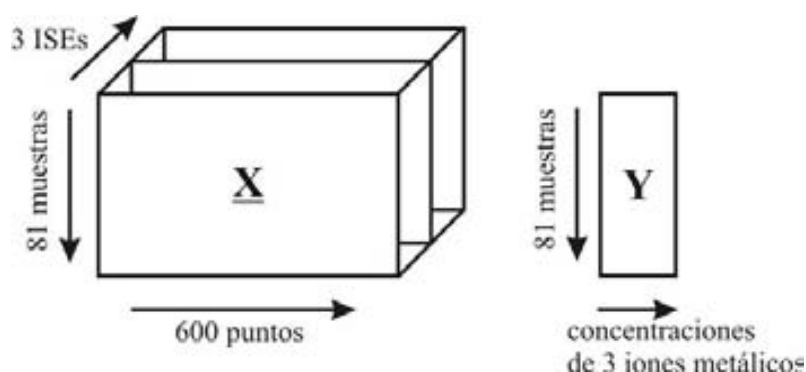


Fig. 60. Representación esquemática de los registros y las triadas de concentraciones ordenados en un arreglo de tres modos y una matriz, respectivamente, para el modelado con *N-PLS*.

Se probaron modelos de regresión multivariable contruidos con 1 a 15 términos. El porcentaje de variación explicada para los predictores y respuestas se usó como parámetro para hallar el número significativo de términos que darían origen a un modelo confiable. En la Fig. 61 se grafica el porcentaje de variación de \underline{X} y \underline{Y} explicado por los términos 1 a 15. A partir de esta figura se puede observar que para \underline{X} sólo se necesitan 2 términos para explicar más del 90% de la variación, mientras que para \underline{Y} se necesitan 10 términos para explicar un porcentaje similar. El modelo final de regresión *N-PLS2* usado en esta aplicación se construyó usando 10 términos ya que es el mínimo necesario para explicar el 90% de varianza del conjunto de concentraciones. Esta elección de 10 términos fue la motivación para la construcción de ANNs con 10 neuronas ocultas en el apartado 8.5. Las diferencias (RMSE) entre los valores esperados y los obtenidos con el modelo *N-PLS2* usando el subconjunto de modelado fueron 2.53 μM , 74.8 μM y 0.64 μM para el Cd^{2+} , Cu^{2+} y Pb^{2+} , respectivamente, lo que corresponde al 8.5%, 5.0% y 4.4% de las máximas concentraciones empleadas.

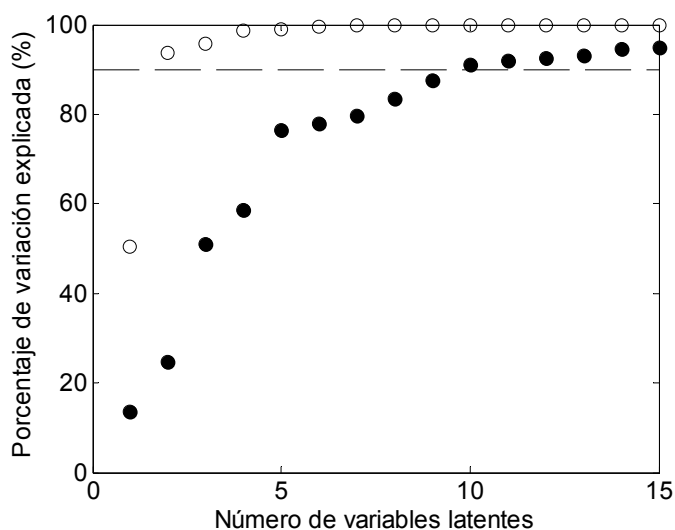


Fig. 61. Porcentaje de variación explicada por los modelos *N-PLS2* desde 1 hasta 15 términos. Los círculos vacíos corresponden a \underline{X} y los círculos llenos a \underline{Y} . La línea punteada marca el porcentaje límite que determina el número óptimo de términos.

Para evaluar el desempeño del modelizado se realizaron gráficas comparativas de concentraciones esperadas vs. obtenidas. La Fig. 62 muestra las gráficas obtenidas con los subconjuntos de modelizado y prueba, donde se

puede observar una buena correlación entre los valores obtenidos y esperados para los tres iones, especialmente para el Cu^{2+} y Pb^{2+} , cuyas gráficas muestran menos dispersión que la gráfica del Cd^{2+} . Esta dispersión puede atribuirse a la interferencia que los iones de Cu^{2+} y Pb^{2+} provocan al sensor de Cd^{2+} , el menos selectivo de los tres ISEs. Los resultados de los factores de correlación, pendientes e intercepciones obtenidos de las comparaciones de los tres iones mediante el análisis de regresión lineal están contenidos en la Tabla 28.

La eficiencia de la metodología basada en *N-PLS2* se corroboró haciendo una validación cruzada basada en 15 modelos adicionales con 10 términos cada uno. Los conjuntos de datos \mathbf{X} y \mathbf{Y} se dividieron en subconjuntos de modelizado y prueba cada vez que se creaba un nuevo modelo *N-PLS2*. Todos los subconjuntos se construyeron usando los índices que se reservaron del anterior modelizado basado en procesamiento DWT + ANNs con el fin de establecer elementos comunes para propósitos de comparación. Los RMSEs totales que se obtuvieron (calculados para los tres analitos a la vez) fueron $1.54 \pm 0.08 \mu\text{M}$ y $2.46 \pm 0.21 \mu\text{M}$ para modelado y prueba, respectivamente. La Tabla 28 resume los análisis de regresión lineal y en ella se condensan los valores promedios de pendientes, intercepciones y coeficientes de correlación obtenidos entre los valores de concentraciones esperados vs. obtenidos.

Estos resultados se compararon contra el enfoque de la Sección 8.6 basada en el procesamiento con Wavelets Discretas seguido de modelado con ANNs. Los errores obtenidos en ambos procedimientos se compararon usando las gráficas de barras que se muestran en la Fig. 63. De ellas se observa que aunque la metodología formada por la combinación DWT-ANN produjo mejores resultados en el proceso de calibración para ambas funciones de transferencia sigmoidales en la capa oculta, los modelos obtenidos con *N-PLS2* fueron mejores en su capacidad de generalización que los obtenidos con las redes.

Tabla 28. Valores promedios de pendientes (*m*), intercepciones (*b*) and coeficientes de correlación (*R*) de las gráficas de comparación de las concentraciones esperadas /vs./ obtenidas con la regresión N-PLS2 (*mr*) y su validación cruzada (*vc*). Los intervalos de incertidumbre fueron calculados al 95% del intervalo de confianza.

Analito		Modelado			Prueba		
		m	b	R	m	b	R
Cd ²⁺	mr	0.893	1.14×10^{-6}	0.945	0.842	1.31×10^{-6}	0.83
	vc	0.87±0.016	$(1.33 \pm 0.14) \times 10^{-6}$	0.934±0.009	0.82±0.042	$(2.30 \pm 0.54) \times 10^{-6}$	0.821±0.039
Cu ²⁺	mr	0.965	1.96×10^{-8}	0.983	0.957	2.41×10^{-9}	0.966
	vc	0.967±0.008	$(1.70 \pm 0.48) \times 10^{-8}$	0.983±0.004	0.939±0.033	$(2.85 \pm 0.17) \times 10^{-8}$	0.973±0.005
Pb ²⁺	mr	0.976	1.30×10^{-7}	0.988	0.916	6.84×10^{-7}	0.974
	vc	0.969±0.006	$(1.58 \pm 0.27) \times 10^{-7}$	0.984±0.003	0.976±0.028	$(1.01 \pm 1.8) \times 10^{-7}$	0.973±0.004

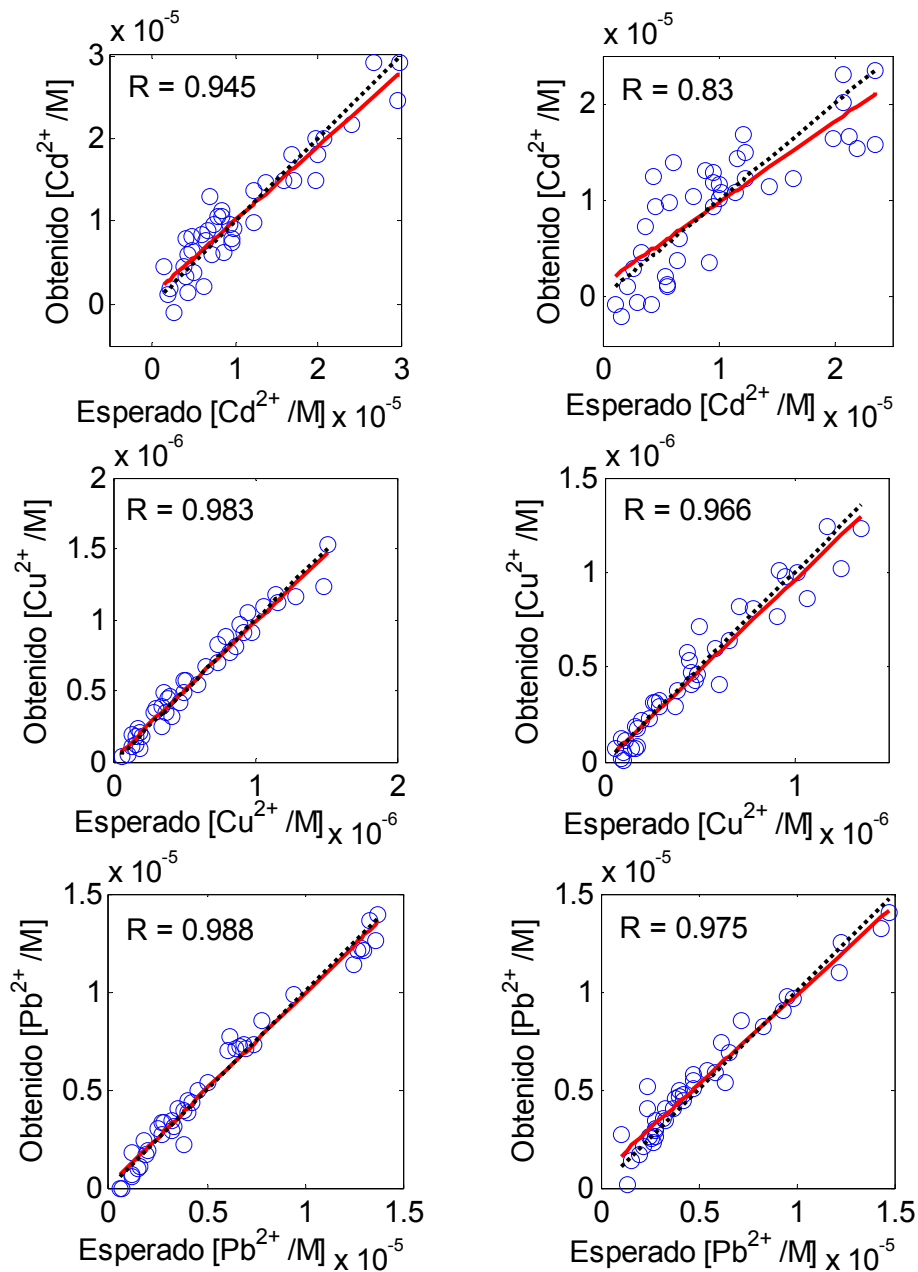


Fig. 62. Comparación de concentraciones obtenidas vs. esperadas para Cd²⁺ (superior), Cu²⁺ (mitad) y Pb²⁺ (inferior) usando procesamiento N-PLS2. La línea punteada corresponde a la idealidad mientras que la línea sólida corresponde al ajuste obtenido. Las gráficas a la izquierda corresponden al modelado y las derechas a la prueba.

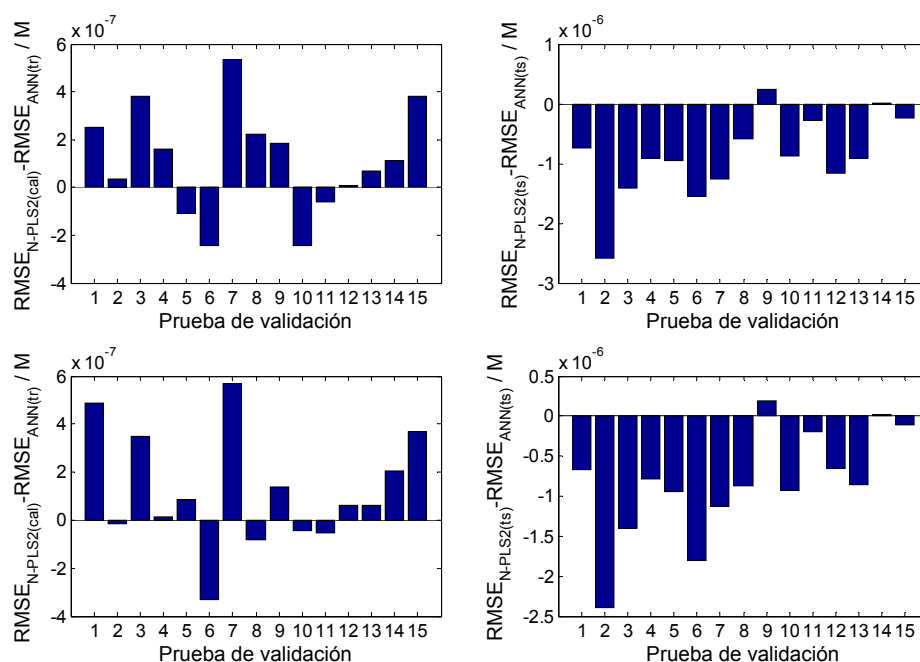


Figura 63. Diferencias promedio de RMSE entre N-PLS2 y DWT-ANN para calibración (gráficas a la derecha) y pruebas (gráficas a la izquierda). Las etiquetas cal, tr, y ts en los subíndices de las etiquetas en el eje y indican calibración, entrenamiento y prueba, respectivamente. Las gráficas en la fila superior corresponden a las redes con función de transferencia sigmoideal logarítmica, mientras que las gráficas en la fila inferior corresponden a redes con función de transferencia tangencial sigmoideal.

8.8.2 Resolución de mezclas de compuestos fenólicos antioxidantes usando una bio-lengua electrónica del tipo voltamperométrico

En esta sección se presenta el una bio-lengua electrónica construida para resolver mezclas de compuestos fenólicos. Se ha usado el término bio-lengua electrónica porque la matriz de electrodos que ahora se emplea está formada por un arreglo de biosensores enzimáticos amperométricos basados que han sido modificados para poder detectar un conjunto de polifenoles identificados como antioxidantes y que se pueden encontrar en frutas, vegetales y bebidas como el vino.

Los compuestos fenólicos bajo estudio fueron catecol, ácido cafeico y catequina, presentes en el vino y que, en conjunto con otros compuestos fenólicos, le confieren al vino propiedades antioxidantes. Los intervalos de concentraciones estudiados fueron los mismos que pueden hallarse en estas

bebidas. Las muestras sintéticas con los compuestos fenólicos usados en este experimental se prepararon con el sistema SIA usando un tampón fosfato al 0.1 M y pH 6.5, junto con una solución de KCl al 0.1M para asegurar la conductividad eléctrica. Las concentraciones de cada uno de los fenoles bajo estudio estuvieron en los intervalos [0-200] μM . Todas las soluciones se prepararon usando agua des-ionizada y reactivos de grado analítico o similar de acuerdo a un diseño experimental con distribución cúbica.

La celda de medida estuvo integrada por 4 diferentes biosensores voltamperométricos de epoxi-grafito, un electrodo de Ag/AgCl de doble unión usado como electrodo de referencia, y un electrodo de platino usado como electrodo auxiliar. De los 4 electrodos de trabajo, uno de ellos fue modificado con tirosina, otro con lacasa, otro más con nanopartículas de cobre y el cuarto fue un sensor sin modificar. Los detalles de la construcción de los electrodos activos pueden consultarse en [8]. Las mediciones voltamperométricas se hicieron usando un equipo de electroquímica AUTOLAB PGSTAT 20 de 6 canales.

Las mediciones se realizaron usando voltamperometría cíclica de barrido. El potencial aplicado estuvo en el intervalo [0.8, -0.4]V, con una tasa fija de barrido de 100mV/s y un escalón de potencial de 9mV. Se prepararon un total de 37 muestras de las cuales 27 fueron usadas para el modelizado y 10 para la prueba. Las concentraciones de las muestras de prueba se determinaron aleatoriamente pero sus valores estuvieron dentro del intervalo de dominio de las concentraciones de modelizado.

Con este desarrollo experimental se obtuvieron cuatro matrices de corriente formadas por 37 registros con 268 valores de corriente cada una de ellas, además de la matriz de concentraciones formada por las 37 triadas de compuestos fenólicos. Estas matrices se reordenaron en un arreglo **X** de 3 modos antes de construir los modelos. Para mostrar el tipo de señales electroquímicas obtenidas con las soluciones bajo estudio, en la Fig. 64 se muestran cuatro gráficos con dos registros cada uno adquiridos con los biosensores amperométricos de la matriz de sensores. La Fig. 65 muestra la distribución de las triadas de concentraciones tanto de modelizado como prueba.

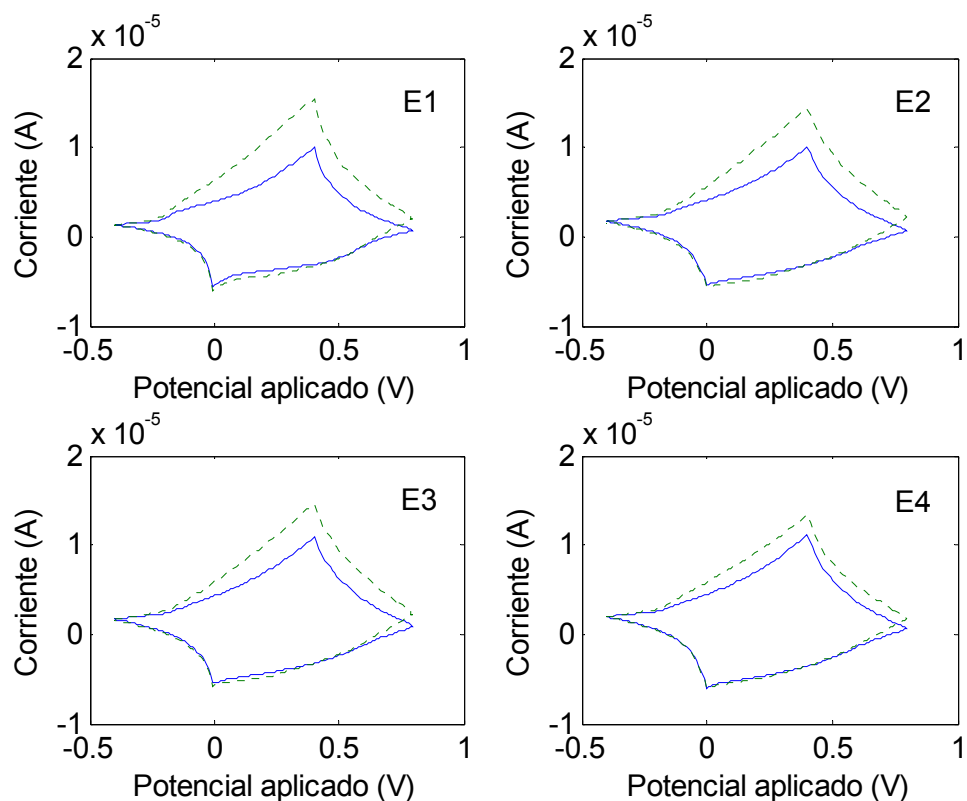


Fig. 64. Ejemplos de voltamperogramas adquiridos con la matriz de biosensores. Las etiquetas E1, E2, E3 y E4 identifican a los electrodos 1, 2, 3 y 4, respectivamente. Los voltamperogramas corresponden a las concentraciones [74,10; 168.29; 164.72] μM para la línea continua, y [190,43; 119,81; 149,23] μM para la línea segmentada.

La matrización del arreglo \underline{X} en tres matrices aumentadas por columnas, por filas y por tubos permitió su factorización a través de la descomposición en valores singulares para verificar si existe la trilinealidad en el arreglo antes de aplicar N-PLS2. Los cuatro primeros autovalores de las tres matrices aumentadas fueron [842,84; 94,92; 43,16; 23,28], [848,37; 40,49; 23,63; 13,29], y [820,89; 81,25; 36,28; 19,55], lo que verifica la trilinealidad de los datos en \underline{X} . En la construcción del modelo N-PLS2 no se preprocesaron los datos de entrenamiento.

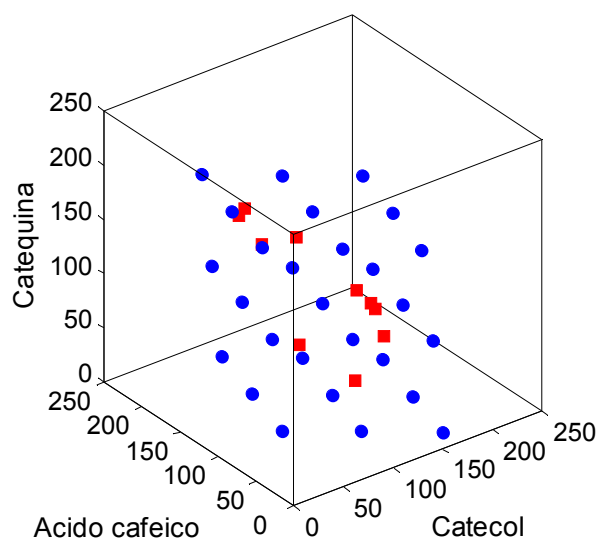


Fig. 65. Distribución especial de las triadas de compuestos fenólicos. Los círculos azules

Basándonos en la experiencia de la Sección 8.6.1, se construyeron y probaron modelos construidos con 1 a 15 términos. Los porcentajes de variación explicados para cada uno de los arreglos \underline{X} y \underline{Y} por los modelos construidos se muestran en la Fig. 66. De acuerdo a la gráfica en esta figura, un modelo N-PLS2 de 5 términos permitiría explicar el 91.10% de la variación en \underline{Y} y 99.85% de la variación en \underline{X} . La raíz de los errores cuadráticos promedios (RMSE) entre las concentraciones esperadas y las reales con el modelo N-PLS2 usando el subconjunto de modelado fueron 25.45 μM , 48.29 μM y 33.59 μM para el catecol, ácido cafeico y catequina, respectivamente, lo que corresponde al 12.72%, 24.14% y 16.79% de las máximas concentraciones empleadas. Los porcentajes de error obtenidos con el subconjunto de prueba fueron 9.19%, 22.95% y 15.41% para los tres fenoles en el mismo orden.

Para mejorar los resultados del modelo de calibración con 5 términos, se evaluó el desempeño del resto de los modelos N-PLS2 en términos del RMSE obtenido tanto en modelizado como en prueba para escoger el mejor modelo. La Fig. 67 resume los valores de RMSE obtenidos con los subconjuntos de modelizado y prueba. El error para modelizado siempre disminuye a medida que aumenta el número de términos, sin embargo el error de prueba tiene un punto de inflexión en el modelo con 8 términos. Esto puede deberse a que a

medida que aumenta el número de términos, el modelo se ajusta mejor a los datos de modelizado y sea menos flexible a comportarse como un modelo de predicción.

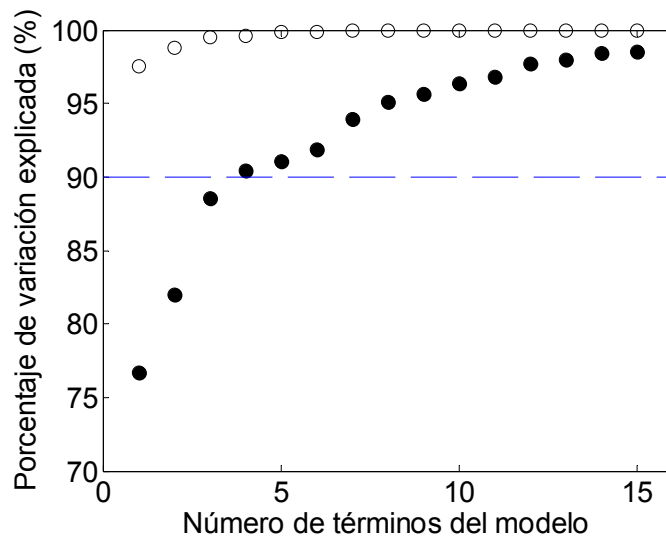


Fig. 66. Porcentaje de variación explicada por los modelos N-PLS2 desde 1 hasta 15 términos. Los círculos vacíos corresponden a X y los círculos llenos a Y . La línea punteada marca el porcentaje límite del 90%.

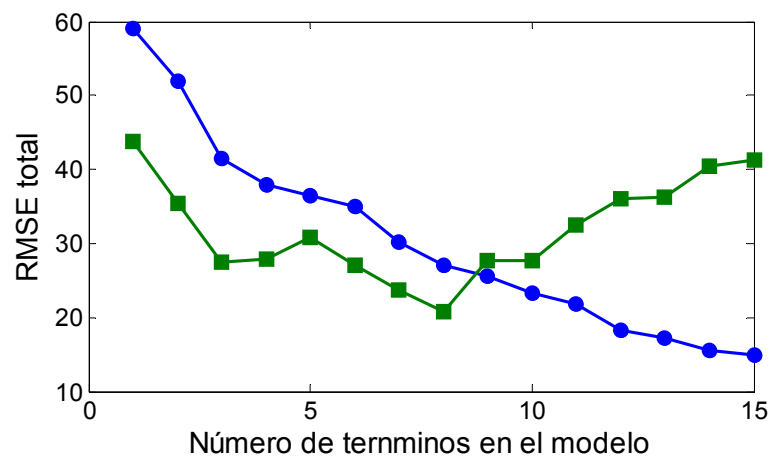


Fig. 67. Valores de RMSE obtenidos en modelizado (marcadores circulares) y prueba (marcadores cuadrados) con los modelos N-PLS2 de 1 a 15 términos.

Se verificó el desempeño del modelo N-PLS2 de 8 términos usando los subconjuntos de modelizado y prueba. Los resultados del análisis de regresión

lineal para ambos conjuntos se muestran gráficamente en la Fig. 68 y se condensan en la Tabla 1 junto con los valores de RMSE obtenidos.

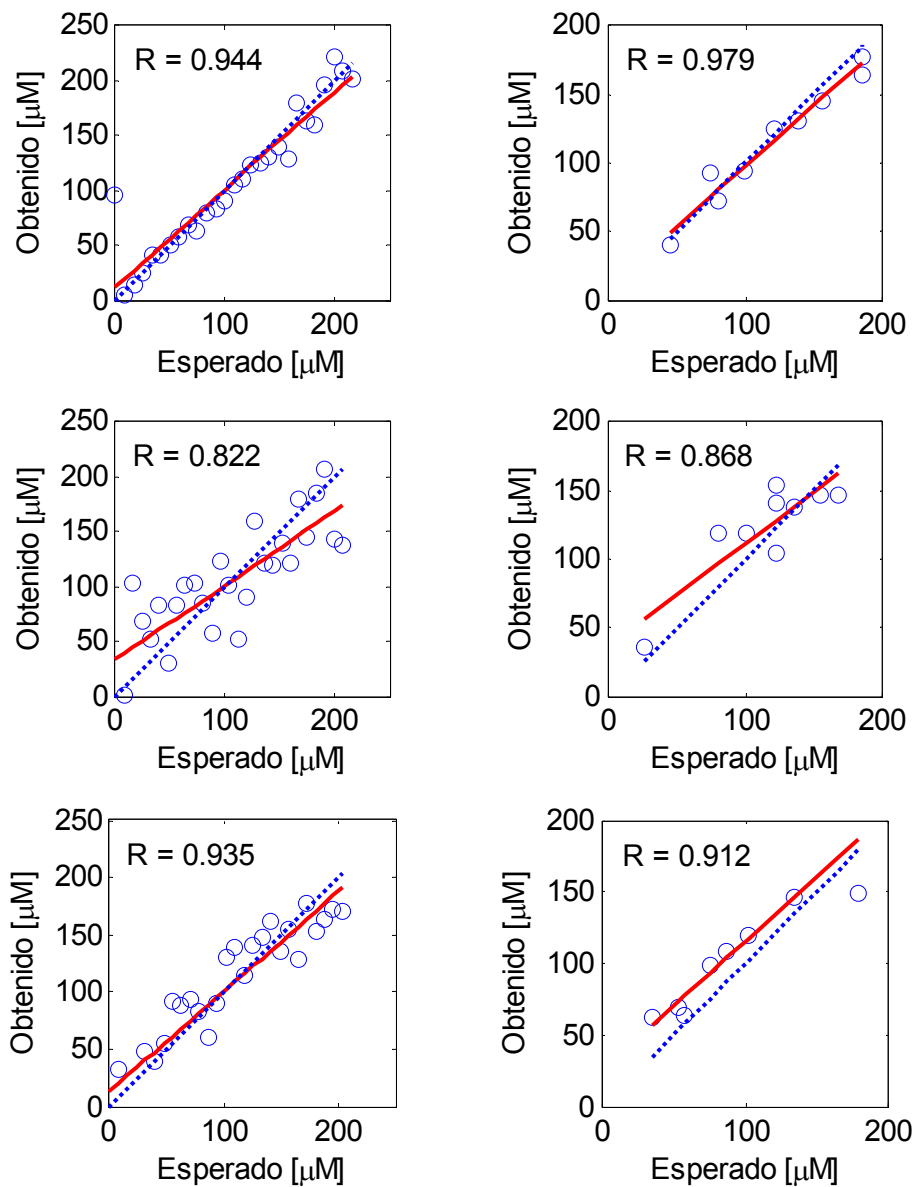


Fig 68. Gráficas de análisis de regresión lineal entre los resultados reales y los obtenidos con el modelo N-PLS2 de 8 términos. Las gráficas en la columna de la izquierda corresponden al entrenamiento y las de la derecha a la prueba. La fila superior corresponde al catecol, la del medio al ácido cafeico y la inferior a la catequina

Tabla 29. Valores de pendiente (*m*), intercepción (*b*) and coeficiente de correlación (*R*) obtenidos con los análisis de regresión lineal entre las concentraciones esperadas y las/ obtenidas con el modelo N-PLS2 de 8 términos. En la tabla también se incluyen los RMSE obtenidos por cada uno de los compuestos fenólicos..

Analito	Modelado				Prueba			
	m	b	R	RMSE	m	b	R	RMSE
Catecol	0.890	11.85	0.944	21.64	0.877	9.77	0.978	12.25
Ácido cafeico	0.677	33.33	0.821	36.07	0.739	37.60	0.868	22.66
Catequina	0.872	13.01	0.934	22.07	0.899	25.68	0.911	16.94

A partir de los valores de la Tabla 29 se puede concretar que N-PLS2 permitió construir un buen modelo de calibración para la resolución de tres compuestos fenólicos antioxidantes a partir de los datos obtenidos con un conjunto de biosensores voltamperométricos. El traslape de las señales electroquímicas que se muestran en la Fig. 64, y que no permiten la diferenciación de los compuestos bajo estudio, pudo ser resuelto correctamente usando un modelo de regresión multi-modo, aunque el modelado y predicción del ácido cafeico presentó mayores dificultades.

La importancia del modelo final conseguido con N-PLS2 para esta aplicación es de tal relevancia que podría compararse con técnicas analíticas más complejas tal como HPLC [8]

Conclusiones

En esta memoria se ha presentado un conjunto de herramientas enfocadas a la implementación de modelos de calibración multivariable para la construcción de lenguas electrónicas. Las conclusiones que se expresan en este apartado, y que se derivan de los objetivos presentados al inicio de esta memoria y los resultados presentados en el Capítulo 8, se han agrupado en dos categorías: i) las relacionadas con los algoritmos propuestos e implementados en su totalidad, y ii) las relacionadas con la herramienta de calibración N-PLS2, que es un método de calibración desarrollada por terceros.

Para cada una de las aplicaciones descritas a lo largo de los Capítulos 3 a 7, se estudiaron y presentaron los fundamentos teóricos que permitieron sus implementaciones. En relación a los algoritmos propuestos en la primera categoría, particularizaremos los comentarios porque el desempeño fue diferente para cada uno de los algoritmos propuestos, inclusive entre las aplicaciones basadas en redes neuronales artificiales que tuvieron dos vertientes, las redes con funciones de activación sigmoideas y las ANNs con funciones de activación de poco o nulo uso en el área química. La correspondencia entre los datos experimentales y las pruebas de los algoritmos siguen una línea de tiempo, por lo que no se reportaron resultados para todas las herramientas y experimentales existentes.

A diferencia de las redes neuronales artificiales con funciones de activación del tipo sigmoideal, que han sido estudiadas en profundidad y han demostrado ser efectivas en varias áreas de la tecnología para la identificación de relaciones lineales y no-lineales entre variables de entrada y salida, las redes con funciones de activación diferente a las sigmoideas son un campo en ciernes. En esta tesis se incursionó en este tipo de redes y se probaron funciones Wavelet, splines, B-splines y funciones sigmoideas como funciones de activación en las neuronas de la capa oculta. En base al desempeño

conseguido con las ANNs implementadas con estas funciones de transferencia podemos concluir lo siguiente:

I. *Redes Neuronales Wavelet con función de activación basada en la norma Euclidiana.*

Las funciones Wavelet multidimensionales basadas en la norma Euclidiana son apropiadas para la construcción de los modelos de calibración porque permiten la evaluación simultánea de múltiples variables de entrada, que en nuestros trabajos experimentales se corresponden con los valores de potenciales y corrientes medidos con sensores potenciométricos y voltamperométricos, respectivamente. El uso de este tipo de funciones representaba una ventaja en la construcción de los modelos de calibración porque el elevado número de grados de libertad disponibles para el posicionamiento de la función en el hiperespacio (no así para su escalamiento) significaba una buena opción para el ajuste de la relación entre los registros electroquímicos de entrada y las concentraciones químicas de salida.

A pesar de que estas funciones resultaron ser eficaces en la construcción de los modelos de calibración y satisficieron nuestras necesidades de mapeo $\mathbb{R}^n \rightarrow \mathbb{R}$ entre las variables de entrada y salida (la posterior extensión a múltiples variables de salida se consigue con la red neuronal), hemos llegado a la conclusión de que la forma de la función limita mucho su eficacia en la etapa de predicción porque la función forma una hipersuperficie que, dependiendo de los parámetros finales en el entrenamiento de la red, podría abarcar una región muy estrecha del hiperespacio formado por las variables de entrada y volverse muy específica a los datos de modelizado, o abarcar una región muy amplia e interferir negativamente con otras funciones Wavelet colindantes.

Las anteriores conclusiones se ven reforzadas con los resultados obtenidos tanto con la estructura unitaria para el modelizado de un solo sensor, como la formada por dos redes en disposición paralela para la calibración simultánea de dos sensores, donde ambas estructuras fueron eficaces para el modelizado pero ineficientes en la predicción.

II. *Redes Neuronales Wavelet con función de activación basada en el producto tensorial.*

Las funciones Wavelet multidimensionales formadas por el producto tensorial de funciones monodimensionales fueron más eficaces (en comparación con las anteriores funciones) en la construcción de modelos de calibración usando ANNs. La ventaja de usar este tipo de función proviene del hecho de que se crea una wavelet monodimensional por cada variable de entrada y cada una de ellas se ve afectada por su propio parámetro de escalamiento, lo que permite aumentar los grados de libertad para el ajuste de la ANN a la vez que permite conseguir un modelo con un buen ajuste a los datos de entrenamiento y errores bastante aceptables en la etapa de predicción.

La posibilidad de que la separación inicial de la información en subconjuntos de entrenamiento y prueba haya podido influenciar positivamente los modelos que se construyeron en el experimental de la Sección 8.1, se hicieron entrenamientos adicionales de validación cruzada que arrojaron resultados igualmente sobresalientes, por lo que podemos concluir que la separación de información en datos de entrenamiento y prueba no es una tarea crítica y que la verdadera efectividad de la herramienta radica en las funciones usadas.

A pesar de que esta estructura solamente fue probada con la información proveniente de un solo sensor, pensamos que es posible realizar una aplicación con redes paralelas para recibir como una entrada los datos provenientes de una matriz de electrodos. Esta conclusión está basada en el hecho de que en una estructura común de ANN, las neuronas en la capa de salida son las encargadas de combinar la información proveniente de las neuronas de la capa oculta, también en el supuesto de que existen funciones redundantes en la red debido a que se usó una Wavelet por cada variable de entrada.

III. *Modelos MARS y B-MARS*

Los datos usados en esta sección experimental se obtuvieron con un espectrofotómetro de fluorescencia, por lo que su naturaleza es totalmente diferente al tipo de información obtenida en el Grupo de Sensores y Biosensores, que es obtenida usando la electroquímica.

En la Sección 8.3 se construyeron modelos MARS y B-MARS, los cuales son modelos de calibración lineales construidos haciendo una búsqueda exhaustiva de funciones base que permiten tener un modelo óptimo. La forma final de estos modelos es una suma de términos desde orden 0 hasta un máximo orden preestablecido por el usuario.

De los resultados obtenidos con MARS y B-MARS podemos concluir que la naturaleza y dimensión de los datos a partir de los cuales se construyen los modelos de calibración, tienen una gran influencia en la eficiencia de los mismos, además de la forma de las funciones usadas para construir los modelos. Los datos usados en las Secciones 8.5 y 8.6 fueron obtenidos con un espectrofotómetro y guardan una relación lineal entre ellos, al igual que los modelos MARS y B-MARS, por lo que los modelos que se construyeron fueron consistentes y con buena capacidad de predicción. También debe tomarse en cuenta que la cantidad de información de la que se dispuso para la búsqueda de los términos del modelo fue superior en comparación a los demás experimentales, lo que permitía hacer una búsqueda más exhaustiva del modelo óptimo.

IV. *Redes neuronales B-spline y spline Catmull-Rom*

Los resultados obtenidos con este tipo de red no fueron satisfactorios. Estas funciones adolecen del mismo problema que las funciones Wavelet multidimensionales basadas en la norma Euclidiana y no pueden usarse para el cumplimiento de nuestro objetivo. Las neuronas con funciones B-spline y spline Catmull-Rom también tienden a formar una superficie que se vuelve muy específica a los valores de entrenamiento y que es difícil poder generalizar a los datos de prueba. El número de neuronas ocultas que se requieren para hacer un modelizado multivariable es mayor en estas estructuras que en el caso de las WNN basadas en la norma Euclidiana. Esta tendencia quedó demostrada con las pruebas que se hicieron en la Sección 8.7, donde se incrementó paulatinamente el número de neuronas ocultas a la par que disminuía el error de entrenamiento. Otros puntos que no favorecen el uso de redes B-spline y Catmull-Rom son el excesivo uso de memoria en el ordenador y la incapacidad de manejar muchas variables de entrada, por lo que no pueden usarse datos crudos de modelizado en estas redes y debe hacerse primero un preprocesamiento para reducir las

dimensiones. Debido a que no se pudieron obtener resultados satisfactorios con estas redes, podemos concluir que no son una buena opción para construir modelos de calibración.

V. *Preprocesamiento TWD seguido del modelado con ANN*

Se tomó ventaja de la habilidad de las ANNs para el modelado de relaciones entre variables de entrada y salida. Para disminuir el costo computacional del modelado y reducir también el tamaño de la estructura de la ANN se preprocesaron los registros electroquímicos usando la TWD.

La eficacia del uso de los coeficientes Wavelet obtenidos del procesamiento de los registros quedó validada con el entrenamiento correcto de la red neuronal, que pudo ser usada también para la predicción de nuevos valores de concentraciones dentro de intervalos de error aceptables. Esta metodología para la construcción de un modelo de calibración ha sido muy eficiente porque nos permitió resolver mezclas de analitos usando un solo sensor o una matriz de ellos.

De esta aplicación podemos concluir que la extracción de características significativas usando la TWD no sólo permite reducir la dimensión de los registros, sino también eliminar términos redundantes de ellos. Esta actividad podría significar un poco más de esfuerzo por tener que realizar un cribado de las funciones Wavelet y niveles de descomposición que reduzcan considerablemente el tamaño de los registros, sin embargo, este esfuerzo adicional se ve compensado en una estructura de red más sencilla que podría llevarse a una implementación electrónica en un sistema dedicado.

VI. *Modelado con N-PLS2*

La segunda categoría de las herramientas reportadas en esta memoria sólo está formada por N-PLS2. La efectividad de la regresión multimodo por mínimos cuadrados parciales ha sido validada en publicaciones anteriores, pero a la fecha de nuestra publicación es la primera vez que se reporta para la obtención de un modelo de calibración multivariable en una lengua electrónica.

A diferencia de las anteriores herramientas descritas en esta memoria, donde se hace el tratamiento matricial de la información, N-PLS2 es una herramienta para el tratamiento multimodo de datos. La única condición que se debe cumplir para usar esta aplicación es la verificación de la trilinealidad de la información de entrada antes de construir los modelos. En nuestro caso, esta condición se cumplió para las dos aplicaciones reportadas en la Sección 8.8, la cual se verificó a través de una descomposición en valores singulares de los arreglos matricizados de registros potenciométricos y voltamperométricos, siendo este último tipo de información la que había presentado más complejidad de modelado y que se resolvió satisfactoriamente usando N-PLS2.

Otra bondad de esta herramienta es que sus modelos son igualmente posibles de exportar a un sistema electrónico dedicado una vez que han sido obtenidos con un ordenador.

En términos generales podemos establecer que varias de las aplicaciones que se desarrollaron en un 100% para esta memoria no han sido optimizadas para ser usadas como modelos de predicción. La capacidad de algunas de ellas de ajustar correctamente a los datos de modelizado es un indicador de la posibilidad de convertirlos en herramientas de calibración. La linealidad de la información juega un papel importante, y dependiendo de la herramienta, también lo juega la cantidad de información que se dispone para construir los modelos. De todas las aplicaciones reportadas las que mejores resultados tuvieron fueron aquellas donde se demostró trilinealidad en los registros electroquímicos, aunque no se probó que esta sea una condición excepcional para el buen funcionamiento de los algoritmos.

Artículos publicados

Multivariate calibration model from overlapping voltammetric signals employing wavelet neural networks

A. Gutés^a, F. Céspedes^a, R. Cartas^a, S. Alegret^a, M. del Valle^{a,*},
J.M. Gutierrez^b, R. Muñoz^b

^a *Grup de Sensors i Biosensors, Departament de Química, Universitat Autònoma de Barcelona, 08193 Bellaterra, Catalunya, Spain*

^b *Sección de Bioelectrónica, Departamento de Ingeniería Eléctrica, CINVESTAV, Ciudad de México, México*

Received 26 July 2005; received in revised form 3 March 2006; accepted 9 March 2006

Available online 19 April 2006

Abstract

This work presents the use of a Wavelet Neural Network (WNN) to build the model for multianalyte quantification in an overlapped-signal voltammetric application. The Wavelet Neural Network is implemented with a feedforward multilayer perceptron architecture, in which the activation function in hidden layer neurons is substituted for the first derivative of a Gaussian function, used as a mother wavelet. The neural network is trained using a backpropagation algorithm, and the connection weights along with the network parameters are adjusted during this process. The principle is applied to the simultaneous quantification of three oxidizable compounds namely ascorbic acid, 4-aminophenol and paracetamol, that present overlapping voltammograms. The theory supporting this tool is presented and the results are compared to the more classical tool that uses the wavelet transform for feature extraction and an artificial neural network for modeling; results are of special interest in the work with voltammetric electronic tongues.

© 2006 Elsevier B.V. All rights reserved.

Keywords: Wavelet Neural Network; Wavelet transform; Voltammetric analysis; Oxidizable compounds

1. Introduction

There is no doubt of the use of multivariate signals as a consolidated trend in analytical chemistry. To work with these signals, the application of appropriate chemometrical tools is mandatory. In the field of electrochemical sensors for liquids, there is the recent approach known as electronic tongue [1]. These systems use both a non-specific sensor array that responds non-selectively to a series of chemical species [2] and some of the existent signal processing techniques. These systems have made possible to discriminate different types of drinks [3,4], to monitor milk quality by measuring microorganisms' growth [5], the classification of clinical samples and food [6,7] and the quantification of ionic concentrations in aqueous solutions [8,9], among other applications. Two main variants exist, which are

those using arrays of potentiometric sensors [6–9], and those using voltammetric electrodes [3–5]. Among the latter, authors have proposed the use of a number of different metallic electrodes, or a number of modified electrodes. Conceptually, a voltammetric system with a single electrode brings a first level of complexity, as there is measured information of high order data fed to the chemometric tool.

In this way, the proposed approach entails the contribution of two parts, being the first one the sensor array, or the electrochemical technique itself; this, as in the application presented in this work, provides a complete multidimensional signal for each experiment [3], in our case a voltammogram. In a multicomponent environment, the sensor array produces complex signals, which contain information about different compounds plus other features; hence, the second part needed in the electronic tongue is the multivariate signal processing tool. Commonly, signals coming from voltammetric procedures are serious overlapping records having non-stationary characteristics. Additionally, voltammograms contain hundreds of measures related with the

* Corresponding author. Tel.: +34 93 5811017; fax: +34 93 5812379.

E-mail address: manel.delvalle@uab.es (M. del Valle).

sample that demands a preprocessing stage intended for the feature extraction prior to the use of chemometric tools. A tool that has already demonstrated its power and versatility in voltammetry, is the Artificial Neural Network (ANN), specially useful for the modeling and calibration of complex analytical signals [10].

The processing of raw voltammograms by ANNs has been reported in the literature. Bessant and Saini [11] used ANNs for calibration with voltammograms acquired from aqueous solutions having mixtures of different organic compounds. In that work, no data reduction was performed, so one input neuron was required for each point of the voltammogram. Gutiérrez et al. [12] developed a bio-electronic tongue based on voltammetry and ANNs for quantifying phenolic compounds. As in Saini's work, no data reduction was performed to process the voltammograms. Even though the results reported by Saini or Gutiérrez for ANNs calibrated with voltammetric data were good, it is often necessary to reduce the length of input data to an ANN in order to gain advantages such as the reduction in training time and avoiding of repetition and redundancy of input data. This can potentially yield more accurate networks, since successful data compression may improve the generalization ability of the ANN, may enhance the robustness and may simplify the model representation [13].

The most popular method for data compression in chemometrics is principal component analysis (PCA). When voltammograms are compressed by PCA, one must be aware of some theoretical limitations. PCA is a linear projection method that fails to preserve the structure of a non-linear data set. If there is some non-linearity in voltammograms, this non-linearity can appear as a small perturbation on a linear solution and will not be described by the first PCs as in a linear case [10]. Alternatively, it is possible to use Wavelet analysis to pre-process voltammetric signals before ANN modeling.

For non-stationary signals, the Wavelet Transform (WT) has become an interesting method in the chemical field [14,15] because of its ability to compress, filter and smooth signals. The coefficients obtained from wavelet decomposition, which are the voltammograms' extracted features, were fed to an ANN to attain successful calibration models for voltammetric analysis [16,17]. The preprocessing by WT reduces the size of the data set being input to an ANN and also its noise content. However, to develop this strategy, a huge effort is required in order to get a proper wavelet-ANN combination that yields acceptable results. Part of this effort consists on determining the mother wavelet function and the maximum decomposition level that best represents the original signal without significant loss of information. To compact electrochemical signals, Palacios-Santander et al. [18] tested 110 possible combinations made with 22 mother wavelets and five consecutive decomposition levels. The final combination was chosen considering the reconstruction error of the signals as well as the number of approximation coefficients obtained. Other criteria to select these parameters have been used [19–21], such as the analysis of wavelet coefficients by PLS, variance or correlation.

In order to reduce the tasks described above for getting the appropriate wavelet-ANN set, a new class of neural network

that makes use of wavelets as activation functions has been developed [22]. These state-of-the-art networks, known as Wavelet Neural Networks (WNN), have demonstrated remarkable results in the prediction, classification and modeling of different non-linear signals [23–25]. Among the few reports of WNN applied in the chemical area, the modeling and prediction of chemical properties is the main theme, gathering the complexation equilibria of organic compounds with α -cyclodextrins [26], the chromatographic retention times of naphthas [27] or the QSPR relationships for critical micelle concentration of surfactants [28]. The application of WNN in chemical process control [29] is also mentioned. A single application was found in analytical chemistry, the oscillographic chronopotentiometric determination of mixtures of Pb^{2+} , In^{3+} and Zn^{2+} [30], where a discrete WNN was used to build the calibration model.

In this work, a WNN is used as a signal processing tool in a voltammetric calibration model devised for multidetermination purposes. Specifically, a three-component study case is selected, the simultaneous determination of the oxidizable compounds Ascorbic acid (AA), 4-Aminophenol (4-Aph) and Paracetamol (Pct) that present overlapped responses. The information entered to the WNN is a set of raw voltammograms obtained with a carbon-based electrode and an automated Sequential Injection Analysis (SIA) system. The previously described system [31] generates effortlessly the set of experimental points needed for training the network. The WNN was built departing from the Multilayer Perceptron Network architecture, with wavelets as activation functions in its hidden layer neurons. The set of parameters to be adjusted during training now include the translation and scaling parameters of the Wavelet, as well as the weights between neurons. The performance of the WNN in our calibration model for voltammetry was compared to the already validated WT-ANN coupling [17].

2. Theory

2.1. Artificial neural networks

Artificial Neural Networks are computing systems made up with a large number of simple, highly interconnected processing elements (called nodes or artificial neurons) that abstractly emulate the structure and operation of the biological nervous system. There are many different types and architectures of neural networks that vary fundamentally in the way they learn. The architecture of the WNN implemented in this work is based on a Multilayer Perceptron (MLP) network.

The basic MLP network has an input, a hidden and an output layer. The input layer has neurons with no activation function and is only used to distribute the input data. The hidden layer (which can be more than one) has neurons with continuously differentiable non-linear activation function; finally, the output layer has neurons with either linear or non-linear activation functions. The data entered to the network move through it towards the output layer where the results are obtained. These outputs are compared with expected values, and if a difference exists, then the connection weights between neurons are changed according to the rules of some learning error algorithm.

2.2. Wavelet Transform (WT)

The WT is an important tool for the analysis and processing of non-stationary signals (whose spectral components vary in time) because it provides an alternative to the classical analysis made with the Short Time Fourier Transform (STFT) [32]. The advantage of WT over STFT is the good localization properties obtained in both time and frequency domains. The main idea of wavelet theory consists on representing an arbitrary signal $f(x)$ by means of a family of functions that are scaled and translated versions of a single main function known as the mother wavelet. The relationship between these functions is represented by Eq. (1):

$$\Psi_{s,t}(x) = \frac{1}{\sqrt{|s|}} \Psi\left(\frac{x-t}{s}\right) \quad s, t \in \mathfrak{R} \quad (1)$$

Where $\Psi(x)$ is the mother wavelet, $\Psi_{s,t}(x)$ is the derived wavelet family known as the daughter wavelets, s is the scale parameter and t is the translation parameter. The factor $s^{-1/2}$ normalizes the family of wavelets in order to keep the unity energy. For a detailed analysis the reader can consult Ref. [33].

The Wavelet Transform of $f(x)$ is given by Eq. (2):

$$W_f(s, t) = \int_{-\infty}^{\infty} f(x) \overline{\Psi_{s,t}(x)} dx \quad (2)$$

where $\overline{\Psi_{s,t}(x)}$ is the complex conjugate of $\Psi_{s,t}(x)$.

The inversion formula of WT is given by Eq. (3)

$$f(x) = \frac{1}{C_\Psi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W_f(s, t) \frac{1}{\sqrt{|s|}} \Psi\left(\frac{x-t}{s}\right) \frac{ds dt}{s^2} \quad (3)$$

where C_Ψ is a constant that depends only on $\Psi(x)$ and is defined as follows:

$$C_\Psi = \int_{-\infty}^{\infty} \frac{|\hat{\Psi}(\omega)|^2}{\omega} d\omega < \infty \quad (4)$$

This last equation is known as the admissibility condition and depends only on the mother wavelet. The term $\hat{\Psi}(\omega)$ in Eq. (4) is the Fourier transform of $\Psi(x)$. For $C_\Psi < \infty$, $\Psi(x)$ must be such that:

$$|\hat{\Psi}(\omega)| < \infty, \quad \text{for any } \omega \quad (5)$$

and $\hat{\Psi}(0) = 0$, implying that

$$\int \Psi(x) dx = 0 \quad (6)$$

meaning that $\Psi(x)$ cannot have offset values [34].

2.3. Wavelet neural network

ANNs can use different non-linear activation functions as well as diverse training algorithms, being the most popular the sigmoidal functions and the backpropagation algorithm, respectively. The WNN is a relatively new class of network that uses wavelets with adjusted position and scale parameters as ac-

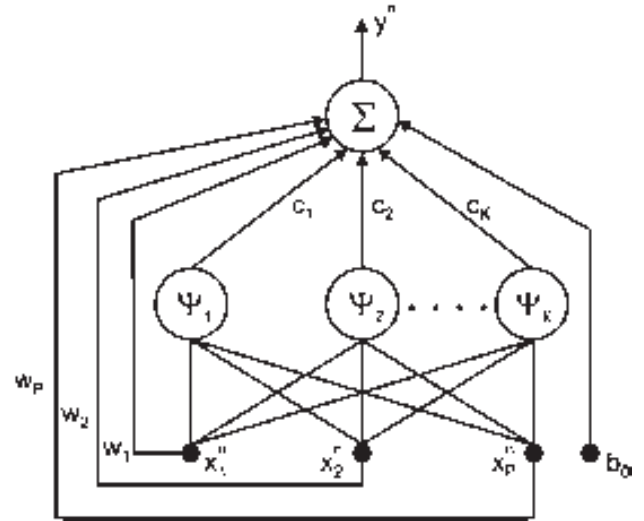


Fig. 1. Architecture of the WNN proposed as a processing tool in the voltammetric e-tongue. x_j^i denotes the j -th intensity value of the i -th voltammogram, and y^i the sought information in it, viz. one component's concentration value.

tivation functions. This first approach to a WNN model makes sense if the inversion formula for the WT is seen like a sum of the products between the wavelet coefficients and the family of daughter wavelets [35]. The WNN is based on the similarity found between the inverse WT Strömberg's equation and a hidden layer MLP network [36]. Combining wavelets and neuronal networks can hopefully remedy the weakness of each other, resulting in networks with efficient constructive methods and capable of handling problems of moderately large dimension [37].

2.4. WNN model

For developing a WNN, frames are less complex to use than orthogonal wavelet functions. The family of wavelets generated from a mother wavelet Ψ can be represented as a continuous frame M_c by Refs. [35,38]:

$$M_c = \left\{ \frac{1}{\sqrt{|s_i|}} \Psi\left(\frac{x-t_i}{s_i}\right), t_i, s_i \in \mathbb{Z}, s_i > 0 \right\} \quad (7)$$

that must fulfill the next requirement

$$A \|f(x)\|^2 \leq \sum_{s,t} |\langle f(x), \Psi_i(x) \rangle|^2 \leq B \|f(x)\|^2 \quad (8)$$

with $A > 0, B < +\infty$

The family of frames described by Eqs. (7), (8) belongs to the Hilbert space $L^2(\mathfrak{R})$ and has been successfully used as the base approach tool in the design of WNNs [39].

The WNN model proposed is shown in Fig. 1. The model corresponds to a feedforward MLP architecture with a single output. The output y^n (where n is an index, not a power) depends on the connection weights c_i between the output of each neuron and the output of the network, the connection weights w_j between the input data and the output, an offset value b_0 useful when adjusting functions that have a mean value other

than zero, the n -th input vector x^n and the wavelet function Ψ_i of each neuron. The model depicted in Fig. 1 can be represented by Eq. (9).

$$y^n = \sum_{i=1}^K c_i \Psi_i(x^n) + b_o + \sum_{j=1}^P w_j x_j^n \quad \{i, j, K, P\} \in \mathbb{Z} \quad (9)$$

where subindexes i and j stand for the i -th neuron in the hidden layer and the j -th element in the input vector x^n , respectively, K is the number of wavelet neurons and P is the number of elements in input vector x^n . With the model just described, a P -dimensional space can be mapped to a monodimensional space ($R^P \rightarrow R$), letting it to predict the value of the output y^n when the n -th voltammogram x^n is input to the trained network.

The basic neuron in this architecture is a multidimensional wavelet, Ψ_i , which is built with the product of P monodimensional wavelets, $\Psi(a_{ij})$, of the form:

$$\Psi_i(x^n) = \prod_{j=1}^P \Psi(a_{ij}) \quad \text{where} \quad a_{ij} = \frac{x_j^n - t_{ij}}{s_{ij}} \quad (10)$$

whose scaling (s_{ij}) and translation (t_{ij}) coefficients are the adjustable parameters of the i -th wavelet neuron. With this mathematical model for the wavelet neuron the network's output becomes a linear combination of several multidimensional wavelets [22,40–42].

Here, we use the first derivative of a Gaussian function defined by $\Psi(x) = xe^{-0.5x^2}$ as a mother wavelet, which has demonstrated to be an effective function for the implementation of WNN [22].

2.5. Training algorithm

The error backpropagation method proposed by Rumelhart [43] is widely used as a training rule in multilayer perceptron networks. This process is based on the derivation of the delta rule, which allows the weights of the network to be updated

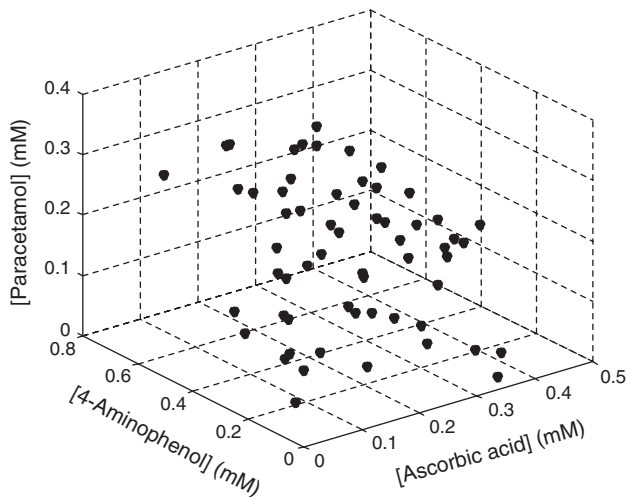


Fig. 2. Three dimensional space of the automatically generated training standards. Each point corresponds to the triad of concentrations of the three oxidizable components.

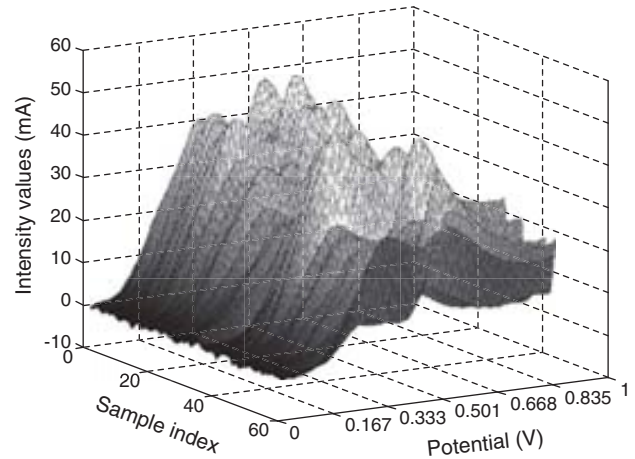


Fig. 3. Measured signals obtained with the generated standards.

whenever a training vector is input to the network. A variation of this rule is called the least minimum square rule; in this variation the weights of the network are updated when all the training vectors have been input to the network. The training algorithm is aimed to diminish the difference between the outputs of the network and the expected values. This difference is evaluated according to the Mean Squared Error (MSE) function defined by Eq. (11):

$$J(\Omega) = 1/2 \sum_{n=1}^N (y_{\text{exp}}^n - y^n)^2 = 1/2 \sum_{n=1}^N (e^n)^2 \quad (11)$$

where y^n is the output of the network and y_{exp}^n is the expected output value related to the input vector x^n .

Since the proposed model is of multi-variable character, we define:

$$\Omega = \{b_0, w_j, c_i, t_{ij}, s_{ij}\} \quad (12)$$

as the set of parameters that will be adjusted during training.

These parameters must change in the direction determined by the negative of the output error function's gradient.

$$-\frac{\partial J}{\partial \Omega} = \sum_{n=1}^N e^n \frac{\partial y^n}{\partial \Omega} \quad \text{where} \quad \frac{\partial y^n}{\partial \Omega} = \frac{\partial y}{\partial \Omega} \Big|_{x=x^n} \quad (13)$$

In addition, we propose to average these changes with the number, N , of input vectors, in order to obtain a weighted error.

$$-\frac{\partial J}{\partial \Omega} = \frac{1}{N} \sum_{n=1}^N e^n \frac{\partial y^n}{\partial \Omega} \quad (14)$$

The partial derivatives of y^n for the set of parameters in Ω are indicated in Eqs. (15)–(19)

$$\frac{\partial y^n}{\partial b_0} = 1 \quad (15)$$

$$\frac{\partial y^n}{\partial w_j} = x_j^n \quad (16)$$

$$\frac{\partial y^n}{\partial c_i} = \Psi_i(x^n) \quad (17)$$

$$\frac{\partial y^n}{\partial t_{ij}} = -\frac{c_i}{s_{ij}} \frac{\partial \Psi_i}{\partial a_{ij}} \Big|_{x=x^n} \quad (18)$$

$$\frac{\partial y^n}{\partial s_{ij}} = -\frac{c_i}{s_{ij}} a_{ij} \frac{\partial \Psi_i}{\partial a_{ij}} \Big|_{x=x^n} \quad (19)$$

in Eqs. (18) and (19), $\frac{\partial \Psi_i}{\partial a_{ij}} \Big|_{x=x^n} = \Psi'(a_{i1}^n) \Psi(a_{i2}^n) \dots \Psi(a_{ij}^n) \dots \Psi(a_{iP}^n)$ where $\Psi'(a_{i1}^n)$ is the value taken by the derivative of the mother wavelet at point a_{i1}^n .

The changes in network parameters are calculated at each iteration according to $\Delta\Omega = \mu \frac{\partial J}{\partial \Omega}$, where μ is a positive real value known as the *learning rate*. With these changes the variables contained in Ω are updated using:

$$\Omega_{\text{new}} = \Omega_{\text{old}} + \Delta\Omega \quad (20)$$

where Ω_{old} represents the current values, $\Delta\Omega$ represents the changes and Ω_{new} corresponds to the new values after each iteration.

The algorithm has two conditions that stop the training process when any of them is accomplished. These conditions are the number of training epochs and the convergence error.

2.6. Initialization of network parameters

An important point in the training process is the proper initialization of the network parameters, because the convergence of the error depends on it. In particular, for our network model, the initialization reported by Oussar [39] is appropriate. Considering a range in input vectors defined by the domain $[x_{j,\text{min}}, x_{j,\text{max}}]$, then the initial values of the i -th neuron for translation and scaling parameters are set to $t_{ij} = 0.5(x_{j,\text{min}} +$

$x_{j,\text{max}})$ and $s_{ij} = 0.2(x_{j,\text{max}} - x_{j,\text{min}})$, respectively, to guarantee no focusing of wavelets in localities of the input universe. The weights are proposed to have random starting values since its initialization is less critical than the translation and scaling variables.

3. Experimental

3.1. Reagents

All reagents employed were analytical grade unless specified. Doubly distilled water was used throughout. Determinations of ascorbic acid, 4-aminophenol and 4-acetamidophenol (paracetamol) in the 0.01–0.5 mM range were developed from stock solutions 10^{-3} M prepared by dissolving pure substances (Fluka) in KCl 0.1 M background solution.

3.2. Apparatus

The automated voltammetric electronic tongue is based on a SIA system, developed and validated previously [31,44]. It employs a bi-directional microburette (Precision Syringe drive/2 Module, Hamilton, Switzerland) equipped with a syringe of 2.5 ml (Hamilton) plus a motorised MVP selection valve (Hamilton) with a 6 way valve head, HVXM 6-5 (Hamilton). PTFE tubing (Bioblock, France) with 1 mm i.d. was used to connect the different SIA components and as holding coil. Connections were completed with low-pressure chromatography fittings. A specially designed mixing cell, made in Perspex, was used in the sample preparation. Automatic stirring was used to assure homogeneity. The whole SIA system was controlled with an in-house software programmed in BASIC (Quick-Basic, Microsoft, USA).

Measurements were developed using an Autolab/PGSTAT20 electrochemical system (Ecochemie, Netherlands), with a specially designed voltammetric cell [45]. Reference electrode consisted in an Ag/AgCl electrode (Model 52-40 Crison

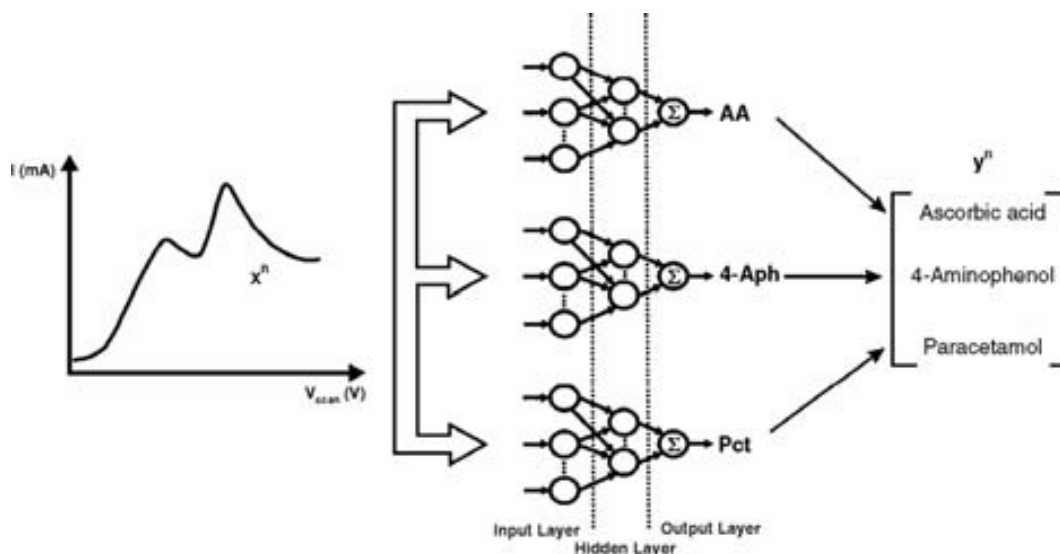


Fig. 4. The voltammogram is entered in parallel to the networks in order to obtain the corresponding concentration values of each component present.

Instruments, Spain). Stainless steel used as a constitutional part of the measuring cell was used as counter electrode. The working electrode consisted in a home-made epoxy–graphite transducer, of general use in our laboratories [46]. Voltammograms were obtained with the linear sweep voltammetric technique, with scan potentials from 0 to 1.0 V at 0.1 V/s in steps of 10 mV.

3.3. Data generation

Three oxidizable components are analyzed by the proposed voltammetric e-tongue: Ascorbic Acid (AA), 4-Aminophenol (4-Aph) and Paracetamol (Pct). The sensitivity of the method is generally high; nevertheless, the selected case presents a

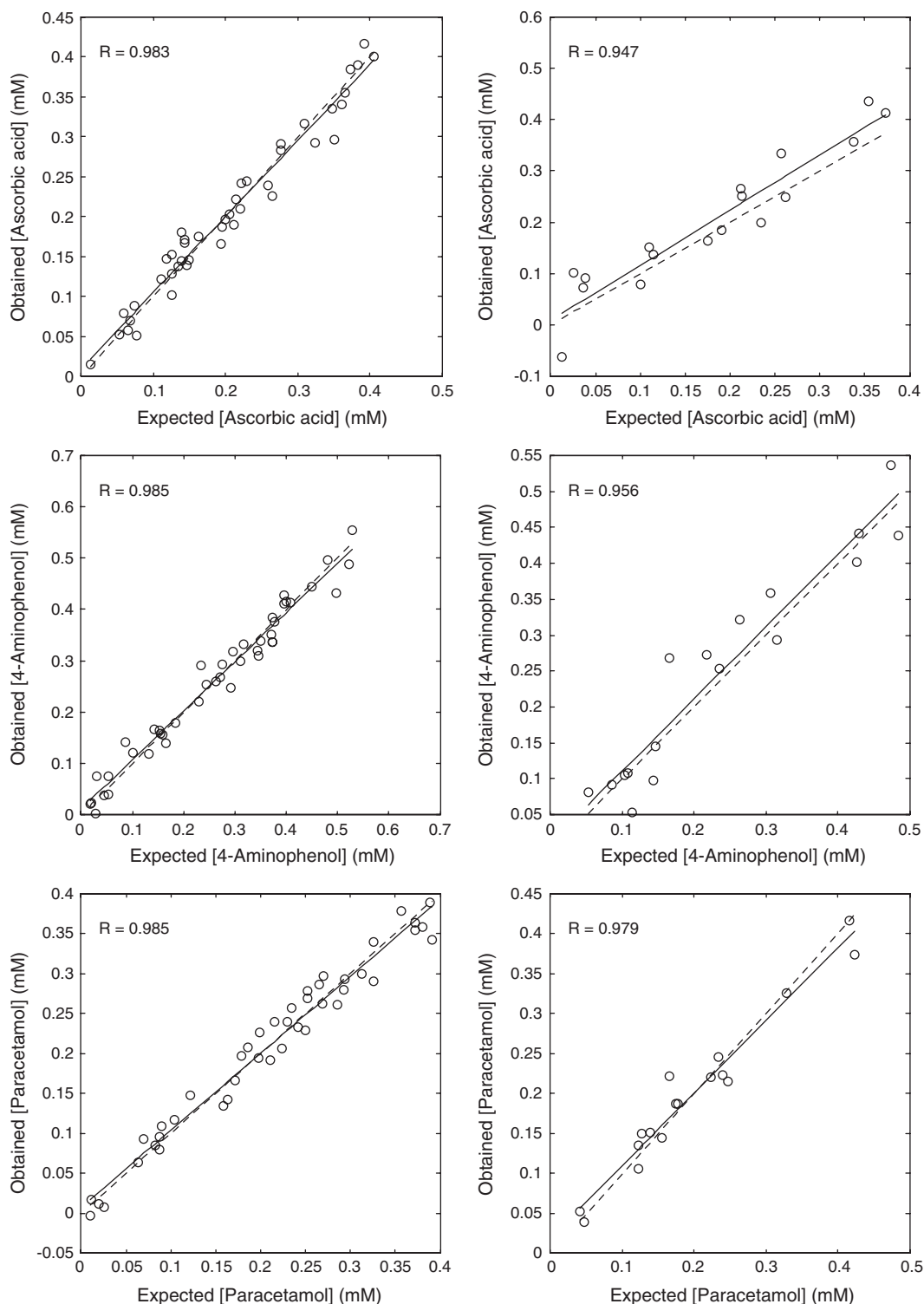


Fig. 5. Comparison between the expected results and those obtained with the WNN with 3 neurons. The graphs correspond to the three species under study. The dashed line corresponds to ideality ($y=x$) and the solid line is the regression of the comparison data. Plots at left correspond to training and plots at right to testing.

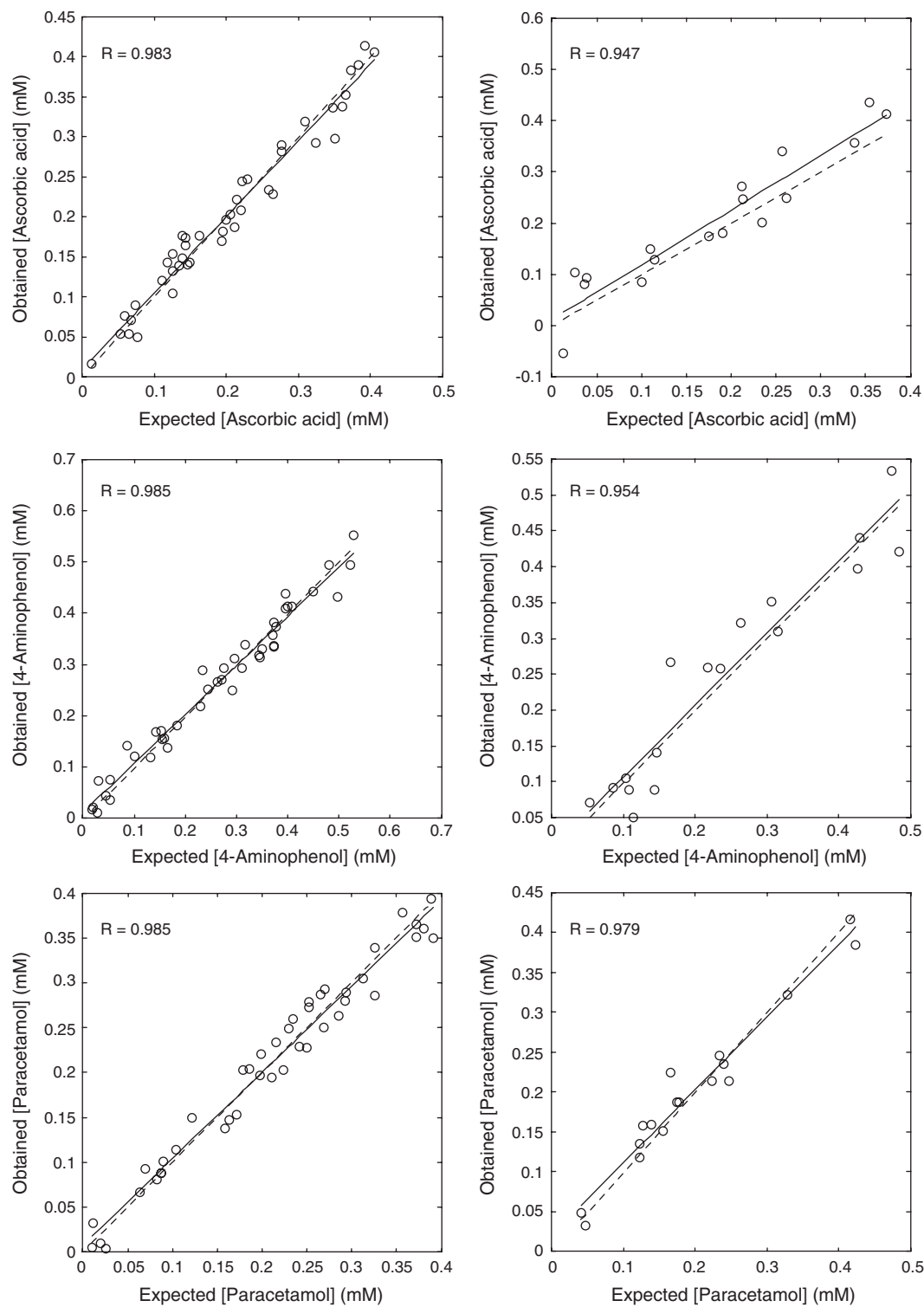


Fig. 6. Comparison between the expected results and those obtained with the WNN with 5 neurons. The graphs correspond to the three species under study. The dashed line corresponds to ideality ($y=x$) and the solid line is the regression of the comparison data. Plots at left correspond to training and plots at right to testing.

voltammetric signal with a high degree of overlapped response, making difficult the determination of each component.

The SIA system was used to prepare individual standards by mixing, diluting and homogenizing prefixed volumes of the stock solutions. Next, the prepared standard was pumped into the mixing cell and the scanning measurement performed. The

prepared standards ranged their concentrations in the intervals (mM) [0.012,0.373] for AA, [0.017,0.529] for 4-Aph and [0.010,0.424] for Pct. Fig. 2 plots the distribution of standards in a three-dimensional space, where each point in the plot represents a triad of concentrations under study. A set of 60 standards was prepared, and for each one a voltammogram of 101 intensity

Table 1
Linear regression parameters corresponding to the comparison lines of obtained versus expected concentration values using the WNNs

		3 neurons		5 neurons	
		<i>m</i>	<i>b</i>	<i>m</i>	<i>b</i>
AA	train	0.951±0.056	0.010±0.013	0.952±0.056	0.010±0.013
	test	1.076±0.200	0.008±0.042	1.067±0.200	0.012±0.042
4-Aph	train	0.956±0.053	0.012±0.016	0.956±0.053	0.012±0.016
	test	1.002±0.170	0.011±0.047	1.005±0.048	0.006±0.174
Pct	train	0.964±0.053	0.008±0.013	0.963±0.053	0.078±0.013
	test	0.906±0.105	0.019±0.024	0.913±0.105	0.020±0.024

Train and test correspond to the training and testing processes, respectively. Confidence intervals calculated at the 95% confidence level.

values was obtained. The set of voltammograms is plotted in Fig. 3, each voltammogram corresponds to one point in the three-dimensional space of concentrations (Fig. 2). Measured intensities were in the interval $[-1.4, 52.4]$ mA.

3.4. Programming

As the voltammetric matrix contains information related to the concentrations of the oxidizable components under study, it constitutes the input data for training and testing the WNN, whereas the concentrations of AA, 4-Aph and Pct constitute the targets to be modeled. The WNN will map a voltammogram represented by x^n to a point of the three-dimensional space of concentrations identified by y^n . To accomplish this, three WNN with 3 or 5 neurons in its hidden layer were programmed and trained for modeling each compound (Fig. 4). Structures of greater dimension were not tested because the training process became very slow due to the network and input data sizes and because the results obtained with the mentioned dimensionality were satisfactory.

3.5. Information preprocessing

Being 60 the number of generated standards, the input data is a voltammetric matrix of dimension $[101, 60]$, and the target is the concentration matrix of dimension $[3, 60]$ (AA, 4-Aph and Pct). For training convenience, the input data and targets were normalized to an interval of $[-1, 1]$ and randomly split into two groups, 70% of the total information was taken for training and the rest for testing.

3.6. Discrete Wavelet Transform (DWT) coupled with ANN

In order to evaluate the WNN, results were compared to ANNs trained with approximation coefficients obtained by pre-

vious discrete wavelet transformation of the signal. The behavior of DWT—ANN combination has already been tested by the authors and the details are given in Ref. [17].

3.7. Software

The functions listed in the algorithm that describes the WNN structure and the gradients for each variable were written in Matlab 7 (Math Works, Natick, MA) using an Intel Pentium III processor desktop computer at 1.1 GHz with 512 Mbytes of RAM.

4. Results and discussion

4.1. Training

The networks were programmed to reach an error of 0.01, evaluated by $\frac{1}{N}J(\Omega)$, where $J(\Omega)$ is the MSE defined in Eq. (11); we denote this error as the *Mean Squared Training Error*. The initialization of weights, translation and scaling parameters were according to the description given above in the training algorithm. The learning rate and the maximum number of training epochs were set to 0.001 and 20,000, respectively. In all cases, the training error was reached in a number of iterations less than this maximum. Greater values of learning rate caused the network output to oscillate and not to converge; on the other hand, smaller values yielded a slow convergence.

4.2. Testing

Figs. 5 and 6 show the comparative graphs between the expected and predicted concentrations for the three oxidizable compounds when using networks with 3 and 5 neurons. Table 1 summarizes this information for both variants. The slope (*m*) and intercept (*b*) that defines the comparison line $y=mx+b$ that best

Table 2
Recovery percentages obtained for the three trainings of the WNN with 3 hidden neurons

Oxidizable compound	Training					Testing				
	Case 1	Case 2	Case 3	Mean	RSD (%)	Case 1	Case 2	Case 3	Mean	RSD (%)
AA	101.94	101.94	102.72	102.20	0.44	103.82	103.82	103.56	103.73	0.14
4-Aph	103.67	103.43	104.03	103.71	0.29	106.33	104.74	105.93	105.67	0.78
Pct	96.82	97.32	97.70	97.28	0.45	105.83	103.59	102.88	104.10	1.48

Along with the recovery percentages are the mean and Relative Standard Deviation (RSD%) for the three cases.

Table 3
Recovery percentages obtained for the three trainings of the WNN with 5 hidden neurons

Oxidizable compound	Training					Testing				
	Case 1	Case 2	Case 3	Mean	RSD (%)	Case 1	Case 2	Case 3	Mean	RSD (%)
AA	102.09	102.43	102.50	102.34	0.21	109.56	103.38	104.83	105.92	3.05
4-Aph	103.42	103.43	103.74	103.53	0.18	102.91	104.74	103.55	103.73	0.90
Pct	101.52	97.32	97.67	98.84	2.36	103.12	103.59	103.79	103.50	0.33

Along with the recovery percentages are the mean and Relative Standard Deviation (RSD%) for the three cases.

fits the data altogether with the uncertainty interval for a 95% of confidence interval are shown for each one of the networks. The ideal case implies lines with $m=1$ and $b=0$, which is fulfilled in all cases at the 95% confidence level, except a slight bias in the 5-neurons training case for Pct.

Two more trainings for each WNN structure were made with random initialization of weights to check if the final model is consistent. To compare the accuracy of the predicted information, a *Recovery Percentage* (RP) was calculated for each trained WNN. The RP is defined by Eq. (21).

$$RP = \frac{\sum_{i=1}^N 100 \cdot \left(1 + \frac{y_i - y_{\text{exp}_i}}{y_{\text{exp}_i}}\right)}{N} \quad (21)$$

where y_i is the i -th obtained value, y_{exp_i} is the i -th expected value and N is the number of targets. The results are contained in Tables 2 and 3 along with the mean and variance for each studied substance.

From Tables 2 and 3, it is evident that for the WNN with 3 neurons, the majority of the recovered information is within $\pm 5\%$ of the ideal recovery percentage. Nevertheless, although some recovered information exceeds the expected values, the distribution of these values around an average fall into the expected recovery interval. This situation is different for the case of the WNN with 5 neurons, where the dispersion of the data is out of the 95% interval. From these results we can conclude that an increase in the number of neurons in the hidden layer does not improve the performance of the network. This can be explained if we consider that for each neuron added to the network a total of 101 monodimensional wavelets are added too, increasing the possibility of having redundant wavelets in the model. This redundancy affects negatively the performance of the network and hence the recovery percentage.

Finally, for a further validation of the presented results, 10 replicate training processes were performed on the selected 3-neuron architecture, with a 17-fold cross validation, selecting the test set each time at random from the total set of experiments. The different results were recorded, and the comparison lines

constructed for each case. Stating y and x for the obtained and expected values, for the training case, the average comparison lines were $y=(0.9513 \pm 0.0015) \cdot x + (0.0098 \pm 0.0002)$, $y=(0.9548 \pm 0.0001) \cdot x + (0.0119 \pm 0.0001)$ and $y=(0.9643 \pm 0.0004) \cdot x + (0.0075 \pm 0.0001)$ for AA, 4-Aph and Pct, respectively. For the testing case, the comparison lines were $y=(1.0789 \pm 0.0398) \cdot x + (0.0068 \pm 0.0052)$, $y=(0.9803 \pm 0.0192) \cdot x + (0.0133 \pm 0.0037)$ and $y=(0.9052 \pm 0.0027) \cdot x + (0.0207 \pm 0.0011)$ in the same order as before. Uncertainties indicated for each parameter correspond to the 95% confidence interval of the 10-replicate distribution. Analogously, average correlation coefficients (corresponding to the 10 replicate training cases) were 0.983, 0.985 and 0.985 for AA, 4-Aph and Pct. For the test comparison lines, average correlation coefficients were 0.937, 0.947 and 0.971, in the same order as above. From these values, it is deduced that the case presented before in graphic details is an average situation from the infinite training possibilities.

4.3. DWT coupled with ANN

Several mother wavelets (daubechies, coiflets, symlets and biorthogonal) and four successive decomposition levels were tested. At each level, only the approximation coefficients were retained and used to reconstruct the voltammograms. In order to choose the combination of mother wavelet, order and decomposition level that yielded a good recovery with the less possible number of approximation coefficients, the original and recovered voltammograms were compared by correlation analysis. The combination that fulfilled our purpose was obtained with the Daubechies wavelet of eighth order and decomposition level number three. The number of approximation coefficients finally used was 16, getting a compression of 84% and a correlation factor of 0.987 between the original and reconstructed signals.

The matrices of the approximation coefficients and concentration values were used as inputs and targets for training and testing an ANN with three outputs, and a set of 3 parallel single output ANNs. Network's topology was feedforward trained with Bayesian regularization algorithm. The ANN with three outputs had two hidden layers with 10 and 5 neurons, respectively; each

Table 4
Linear regression parameters obtained with the training and testing data sets using the ANN with three outputs trained with the wavelet coefficients

Three outputs network	Training			Testing		
	m	b	R	m	b	R
AA	0.994 \pm 0.008	2.07e-4 \pm 0.016	0.999	0.817 \pm 0.274	0.0294 \pm 0.154	0.785
4-Aph	0.995 \pm 0.007	-2.75e-4 \pm 0.011	0.999	0.843 \pm 0.246	-0.0587 \pm 0.135	0.843
Pct	0.994 \pm 0.012	-5.38e-5 \pm 0.021	0.998	0.886 \pm 0.29	0.00259 \pm 0	0.929

Table 5
Linear regression parameters obtained with the training and testing data sets using three ANNs with one output trained with the wavelet coefficients

Parallel networks	Training			Testing		
	m	b	R	m	b	R
AA	0.992±0.01	-6.20e-4±0.019	0.998	0.644±0.147	-0.033±0.262	0.772
4-Aph	0.937±0.038	-6.7e-3±0.064	0.978	0.732±0.113	-0.030±0.206	0.869
Pct	0.997±0.005	6.20e-4±0.009	0.999	0.954±0.109	0.012±0.173	0.938

parallel network had two hidden layers with 6 and 24 neurons, and a single neuron in output layer. Both structures had non-linear functions in the hidden layers and linear function in the output layer.

The set of input and output matrices were normalized to $[-1,1]$ and split into training and testing subsets (two third parts were taken for training). The error goal for training was set to 0.01, to be reached in 300 epochs of training or less and was tracked by using the sum of squared errors (SSE).

Trained networks were evaluated with training and testing input data, and a linear regression analysis between the outputs of the networks and the expected values for each component was done. The correlation factors obtained, along with the parameters of the straight line that best fits the corresponding set of data are contained in Tables 4 and 5.

4.4. Comparison between WNN and DWT—ANN

In this work, wavelet analysis and neural network have been combined in two manners, DWT—ANN and WNN. In the first one, the wavelet theory is decoupled from the neural networks. The voltammograms are decomposed with the help of wavelet transform and the approximation coefficients obtained are, in a second stage, furnished to a neural network for modeling purposes. In the second one, wavelet theory and neural networks are combined into a single method. In this methodology, the translation and the scaling parameters, along with the weights of neuron's connections are adjusted during training. As it is observed by the results, the outputs of the DWT—ANN combination produces a correlation factor greater than 0.9 when the training data is applied to the input, however, this correlation diminishes when the testing data is used, particularly in the recovery of ascorbic acid (AA) where $R < 0.8$.

The performance of the proposed WNN architecture was better than that of the DWT—ANN combination. In WNN, R was greater than 0.94 for training and testing when determining the three chemical species. To explain the good performance of the WNN the following factors can be considered:

- Whereas in the DWT—ANN method each voltammogram is represented by 16 approximation coefficients that retain the spectral information contained at low frequencies, in the WNN the complete data set is fed into the network. In this case a quasi-continuous transformation takes place due to the way in which the wavelets parameters are adjusted; on the other hand, when DWT is applied, the approximation coefficients are obtained by digital filtering and subsampling, which is interpreted as a more discrete process.

- In the WNN there is one weight w_i for each i -th data point contained in the set of voltammograms x_i^t . After the network was trained, the set $\{w_i\}$ establishes which points in a voltammogram contribute more to the quantification of the chemical species.

A network with high modeling performance, as it is needed to solve complex learning related problems is correctly built with the proposed WNN architecture. This network is expected to be especially valuable when the input data is irregularly spaced and/or overlapped, being the latter the usual case with multicomponent voltammetric signals.

The proper choice of the number of hidden neurons and the use of continuous multidimensional frames for decomposing the input data, allow the network to map the voltammograms with their respective concentration values by simply adjusting the scale and translation parameters.

5. Conclusions

An innovative neural network which intrinsically uses wavelet functions has been developed. The design merges the artificial neural networks and wavelet theory to give rise to a new class of network known as the Wavelet Neural Network (WNN). We have described here an application aimed to quantitatively determine the concentration of chemical species based on information obtained from voltammetric sensors. The WNN results in a proper multivariate modeling tool for voltammetry that performs better than the sequential WT—ANN combination.

As the wavelet transform has proven its ability for capturing essential features in the time-frequency behavior of a signal, it seems reasonable to represent a non-stationary signal by these functions. The strategy for training the WNN lets us define the appropriate parameters for a family of wavelet functions that best fits the voltammetric signals solved in this work. Multi-resolution analysis offers to the WNN a unique characteristic in prediction tasks doing an appropriate selection of a mother wavelet and the number of hidden units, with which the overfitting problem can be effectively avoided. The accuracy and stability of the WNN can be further improved by the implementation of other wavelet functions and increasing the number of network outputs.

Acknowledgments

Financial support for this work was provided by the MCyT (Madrid, Spain) through the project CTQ2004-08134, by

CONACYT (Mexico) through the project 43553 and by the Department of Universities and the Information Society (DURSI) from the Generalitat de Catalunya.

References

- [1] Y. Vlasov, A. Legin, *Fresenius J. Anal. Chem.* 361 (1998) 255–260.
- [2] Y. Vlasov, L. Andrey, A. Rudnitskaya, *Sens. Actuators, B* 44 (1997) 532–537.
- [3] F. Winquist, P. Wide, I. Lundström, *Anal. Chim. Acta* 357 (1997) 21–31.
- [4] A. Legin, A. Rudnitskaya, Y.G. Vlasov, C. Di Natale, E. Mazzane, A. D'Amico, *Sens. Actuators, B* 44 (1997) 291–296.
- [5] F. Winquist, C. Krantz-Rülcker, P. Wide, I. Lundström, *Meas. Sci. Technol.* 9 (1998) 1937–1946.
- [6] A. Legin, A. Smirnova, A. Rudnitskaya, L. Lvova, E. Suglobova, Y. Vlasov, *Anal. Chim. Acta* 385 (1999) 131–135.
- [7] C. Di Natale, R. Paolesse, A. Macagnano, A. Mantini, A. D'Amico, A. Legin, L. Lvova, A. Rudnitskaya, Y. Vlasov, *Sens. Actuators, B* 64 (2000) 15–21.
- [8] J. Gallardo, S. Alegret, R. Muñoz, M. De-Roman, L. Leija, P.R. Hernández, M. del Valle, *Anal. Bioanal. Chem.* 377 (2003) 248–256.
- [9] J. Gallardo, S. Alegret, M.A. de Roman, R. Muñoz, P.R. Hernández, L. Leija, M. del Valle, *Anal. Lett.* 36 (2003) 2893–2908.
- [10] F. Despagne, D.L. Massart, *Analyst* 123 (1998) 157R–178R.
- [11] C. Bessant, S. Saini, *Anal. Chem.* 71 (1999) 2806–2813.
- [12] A. Gutiérrez, F. Céspedes, S. Alegret, M. del Valle, *Biosens. Bioelectron.* 20 (2005) 1668–1673.
- [13] E. Richards, C. Bessant, S. Saini, *Chemometr. Intell. Lab. Syst.* 61 (2002) 35–49.
- [14] A.K.-M. Leung, F. Chau, J. Gao, *Chemometr. Intell. Lab. Syst.* 43 (1998) 165–184.
- [15] S. Xue-Guang, A. Kai-Man, C. Foo-Tim, *Acc. Chem. Res.* 36 (2003) 276–283.
- [16] M. Cocchi, J.L. Hidalgo-Hidalgo-de-Cisneros, I. Naranjo-Rodríguez, J.M. Palacios-Santander, R. Seeber, A. Ulrici, *Talanta* 59 (2003) 735–749.
- [17] L. Moreno-Barón, R. Cartas, A. Merkoçi, S. Alegret, M. del Valle, L. Leija, P.R. Hernández, R. Muñoz, *Sens. Actuators, B* 113 (2006) 487–499.
- [18] J.M. Palacios-Santander, A. Jiménez-Jiménez, L.M. Cubillana-Aguilera, I. Naranjo-Rodríguez, J.L. Hidalgo-Hidalgo-de-Cisneros, *Microchim. Acta* 142 (2003) 27–36.
- [19] D.J. Rimbaud, B. Walczak, R.J. Poppi, O.E. De Noord, D.L. Massart, *Anal. Chem.* 69 (1997) 4317–4323.
- [20] L. Eriksson, J. Trygg, E. Johansson, R. Bro, S. Wold, *Anal. Chim. Acta* 420 (2000) 181–195.
- [21] B.K. Alsberg, A.M. Woodward, M.K. Winson, J.J. Rowland, D.B. Kell, *Anal. Chim. Acta* 368 (1998) 29–44.
- [22] Q. Zhang, A. Benveniste, *IEEE Trans. Neural Netw.* 3 (1992) 889–898.
- [23] B.R. Bhakshi, G. Stephanopoulos, *AIChE J.* 39 (1993) 57–81.
- [24] L. Cao, Y. Hong, H. Fang, G. He, *Physica, D* 85 (1995) 225–238.
- [25] H. Szu, B. Telfer, J. Garcia, *Neuronal Netw.* 9 (1996) 695–708.
- [26] Q.-X. Guo, L. Liu, W.-S. Cai, Y. Jiang, Y.-C. Liu, *Chem. Phys. Lett.* 290 (1998) 514–518.
- [27] X. Zhang, J. Qi, R. Zhang, M. Liu, Z. Hu, H. Xue, B. Fan, *Comput. Chem.* 25 (2001) 125–133.
- [28] Z. Kardanpour, B. Hemmateenejad, T. Khayamian, *Anal. Chim. Acta* 531 (2005) 285–291.
- [29] J. Zhao, B. Chen, J. Shen, *Comput. Chem. Eng.* 23 (1998) 83–92.
- [30] H. Zhong, J. Zhang, M. Gao, J. Zheng, G. Li, L. Chen, *Chemometr. Intell. Lab. Syst.* 59 (2001) 67–74.
- [31] A. Gutiérrez, F. Céspedes, S. Alegret, M. del Valle, *Talanta* 66 (2005) 1187–1196.
- [32] O. Rioul, M. Vetterli, *IEEE Signal Process.* 8 (1991) 14–38.
- [33] G. Kaiser, *A Friendly Guide to Wavelets*. Ed. Birkhäuser, Boston MA, 1994, p. 300.
- [34] Y.T. Chan, *Wavelet Basics*, Kluwer Publishers, Boston MA, 1995, p. 134.
- [35] M. Akay (Ed.), *Time Frequency and wavelets in Biomedical Signal Processing*. IEEE Press Series on Biomedical Engineering, Wiley—IEEE Press, Piscataway NJ, 1997, p. 739.
- [36] Y. Meyer, *Wavelets: Algorithms and Applications*, Society for Industrial and Applied Mathematics, SIAM, Philadelphia, PA, 1993, p. 133.
- [37] Q. Zhang, *IEEE Trans. Neural Netw.* 8 (1997) 227–236.
- [38] C.E. Heil, D.F. Walnut, *SIAM Rev.* 31 (1989) 628–666.
- [39] Y. Oussar, I. Rivals, L. Personnaz, G. Dreyfus, *Neurocomputing* 20 (1998) 173–188.
- [40] M. Cannon, J.E. Slotine, *Neurocomputing* 9 (1995) 293–342.
- [41] S.G. Mallat, *IEEE Trans. Pattern Anal. Mach. Intell.* 11 (1989) 674–693.
- [42] J. Zhang, G.G. Walter, Y. Miao, W.N.W. Lee, *IEEE Trans. Signal Process.* 43 (1995) 1485–1497.
- [43] D.E. Rumelhart, G.E. Hilton, R.J. Williams, *Learning internal representation by error propagation*, *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, vol. 1, MIT Press, Cambridge MA, 1986, Chapter 8.
- [44] A. Gutiérrez, F. Céspedes, S. Alegret, M. del Valle, *Anal. Bioanal. Chem.* 382 (2005) 471–476.
- [45] X. Llopis, A. Merkoçi, M. del Valle, S. Alegret, *Sens. Actuators, B* 107 (2005) 742–748.
- [46] F. Céspedes, S. Alegret, *Trends Anal. Chem.* 19 (2000) 276–285.



Two analyte calibrations from the transient response of a single potentiometric sensor employed with the SIA technique

Raul Cartas^a, Aitor Mimendia^a, Andrey Legin^b, Manel del Valle^{a,*}

^a Sensors and Biosensors Group, Chemistry Department, Universitat Autònoma de Barcelona, Edifici Cn, 08193 Bellaterra, Barcelona, Spain

^b Chemistry Dept, St. Petersburg University, Universitetskaya nab. 7/9, 199034 St. Petersburg, Russia

ARTICLE INFO

Article history:

Received 29 May 2009

Received in revised form

21 September 2009

Accepted 25 September 2009

Available online 3 October 2009

Keywords:

Wavelet transform

Artificial neural networks

Multi-analyte calibration

SIA

Potentiometry

ABSTRACT

Simultaneous quantification of Cd^{2+} and Pb^{2+} in solution has been correctly targeted using the kinetic information from a single non-specific potentiometric sensor. Dual quantification was accomplished from the complex information in the transient response of an electrode used in a Sequential Injection Analysis (SIA) system and recorded after step injection of sample. Data was firstly preprocessed with the Discrete Wavelet Transform (DWT) to extract significant features and then fed into an Artificial Neural Network (ANN) for building the calibration model. DWT stage was optimized regarding the wavelet function and decomposition level, while the ANN stage was optimized on its structure. To simultaneously corroborate the effectiveness of the approach, two different potentiometric sensors were used as study case, one using a glass selective to Cd^{2+} and another a PVC membrane selective to Pb^{2+} .

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

Automated systems based on flow-injection techniques, namely Flow-Injection Analysis (FIA) and Sequential Injection Analysis (SIA), have been used in analytical chemistry during the last three decades [1,2]. With these, species detection for either qualitative or quantitative purposes can be accomplished by using different detection principles; when the systems employ sensors at the detection stage, specially simple and efficient analytical systems can be devised. Selectivity of used sensors plays an important role given it determines discrimination capability [3]. Early applications based on flow-injection techniques were developed using highly selective sensors, but the lack of availability of specific sensors for all needs fostered the proposal of a new approach. This alternate solution to the lack of sensor variety is the use of sensors with less restrictive response characteristics and the attempt to take advantage of multidimensional information that they can provide [4].

Detection of a single analyte in the simpler cases does not represent a challenge for low-selective sensors; but for multi-species detection in complex matrices, these sensors provide signals with multiple analyte dependence that makes mandatory the use of chemometric tools for deconvolution. This multicomponent aspect can be derived either from a high dimensional signal [5,6] or from the use of sensor arrays [7].

Calibration models for multiple component determinations have been built with different multivariate tools such as Multiple Linear Regression [8–10], Multivariate Curve Resolution [11,12], Principal Component Regression [13–15], Partial Least Squares [13,16–18], Non-linear Iteratively Partial Least Squares [10], Non-linear Least Squares [9], Support Vector Machines [19] and Artificial Neural Networks (ANNs) [20–23]. From this set of tools, ANNs have outstood due to its ability to model both linear and non-linear responses (with respect to the analytes). Models based on ANNs are sometimes created from high-dimensionality information, for example a mass spectra or a voltammogram. In such cases, preprocessing must be done to compress the signals while preserving enough amount of relevant information. Usual approaches for data preprocessing are Principal Component Analysis (PCA) [2,13,15,18,24], Fourier Transform [20] and Discrete Wavelet Transform (DWT) [25]. Among these tools, DWT has certain interesting advantages because, besides to compression, the technique also filters and smoothes signals.

Analytical use of sensors is carried out normally in steady state conditions, but recent applications have explored the advantages of using their dynamic signals. These can be derived from chemical kinetics or from their use immersed in flowing solutions. This additional characteristic represents an advantage, since the richer content of information supplied by the dynamic response of a sensor after, e.g. a step profile insertion of the sample improves the discrimination ability between primary and interfering species [13,26].

* Corresponding author. Tel.: +34 93 5811017; fax: +34 93 5812379.
E-mail address: manel.delvalle@uab.es (M. del Valle).

From this departure point, FIA systems combined with spectrophotometric detection were applied to resolve ternary mixtures of metallic cations [14] or binary mixtures of antibiotics [16]. Dynamic signals recorded from an array of potentiometric sensors coupled to an SIA system were reported for the resolution of ternary and quaternary mixtures of cations in water samples [20]. Heavy metals have also been the goal of multi-species determination based on transient signals. Relevant to this communication is the work reported in [26], where a flow-injection system coupled to a sensor array was used for the simultaneous quantification of two metallic ions in solution from the dynamic sensor recordings. Similarly, dynamic signals obtained with two different biosensor-based FIA systems have been used for the binary determinations of glucose/pH or urea/pH pairs [21]. The same methodology was successfully applied later for the quantification of several concentrations of penicillin [22]. In all the described cases, transient signal is directly processed, without any data pre-treatment.

This work describes the use of a potentiometric sensor in a SIA system, which after a chemometric treatment of its transient signal, permits to simultaneously quantify two analytes. The advanced approach suggested takes advantage of the dynamic response and cross-sensitivity contained in a single non-specific sensor to resolve a binary mixture. As case study, quantification of cadmium and lead is performed departing from the signal recorded from one sensor only, after the high reproducible step injection of a sample employing the SIA system.

In the data treatment aspect, it is difficult to build calibration models from dynamic responses, given the increased complexity of the signals after addition of the kinetic dimension. In this work, the difficulty was surpassed by first preprocessing the recordings using DWT to reduce their size and extract significant features [25]. Compressed information was later fed to an Artificial Neural Network (ANN) to finally build the calibration model. The utility of the approach is shown by modelling both analytes in two different study cases: two potentiometric sensors from different nature, PVC or glass membrane, were used to illustrate the concept. The final aim of the work is to enrich the type of information to be used by electronic tongues with multiple sensors, to better resolve the most complex cases.

2. Experimental

2.1. Equipment

Quantification of two heavy metals in solution was the goal of this application. A total of 45 different combinations for cad-

mium and lead were randomly generated using Excel and then programmed for being automatically prepared (by dilution from stocks) and injected into a cell by a SIA system available at our laboratory [20]. Composition of mixed solutions is available for downloading as [Supplementary information](#). The SIA system shown in Fig. 1 is formed by a fluidic stage and a measurement stage. The former consists of an automatic microburette (Crison 2030 microburette, Crison, Spain) equipped with a 5 ml syringe (Hamilton, Switzerland), a holding coil (5 ml volume, 1 mm \varnothing PTFE tube, Bioblock, France), an 8 way Hamilton MVP valve (Hamilton, Switzerland) and a 7 ml home-made Perspex mixing cell with a magnetic stirrer. The measurement stage comprises the sensors, a reference electrode (Thermoelectron 900200, MA, USA) and an 8-channel signal conditioning circuit connected to a Data Acquisition System (National Instruments NI 6221 Multifunction DAQ, TX, USA). All active elements involved in the SIA system are controlled using a virtual instrument developed in LabView [20] through the RS-232 port of a PC Pentium III at 600 MHz.

2.2. Reagents and solutions

Standards were prepared by the SIA system using a 0.05 M acetic/acetate buffer at pH 4.5 as carrier and diluting solution. Analytes were in ranges $0\text{--}2.44 \times 10^{-4}$ M for Cd^{2+} and $0\text{--}4.97 \times 10^{-5}$ M for Pb^{2+} . These ranges were determined by the optimal working range of the sensors used. Studied concentrations, although low for conventional potentiometric sensors, are higher than those permitted by the Environmental Protection Agency. It is also worth noticing that for heavy metals these permitted levels are extremely low (75 nM for lead and 45 nM for cadmium). All solutions carrier were prepared using doubly distilled water and analytical grade (or similar) reagents.

2.3. Sensors and procedure

The measuring cell incorporated two ion selective electrodes (ISEs) with cross response to the tested heavy metals, and a Ag–AgCl reference electrode. One ISE was based on a potentiometric Pb^{2+} PVC membrane with trioctylphosphine oxide as ionophore, and the other was based on a chalcogenide glass selective to Cd^{2+} . ISEs were fabricated at the Chemistry Department of St. Petersburg University.

Each ISE was characterized in static and flow conditions. Batch analysis measurements were firstly done in distilled water and later in single ion solutions at different concentrations. Linear working

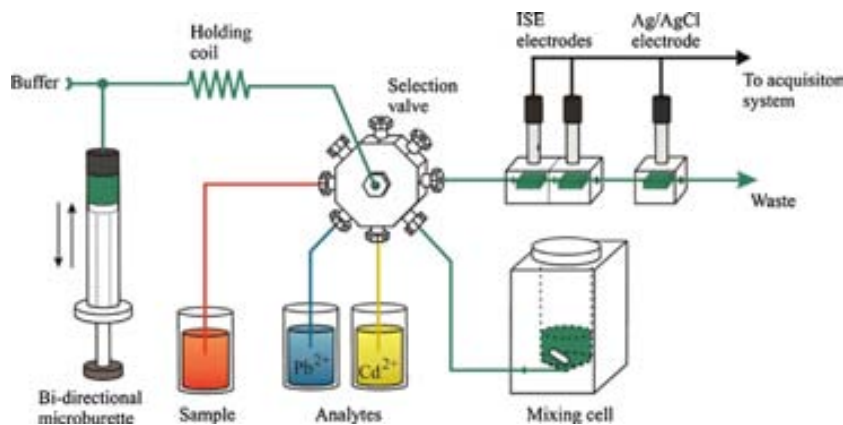


Fig. 1. Scheme of the SIA system employed. The measurement part comprises the ISEs, the reference electrode and the acquisition system (not shown in the figure). The fluidic part is formed by the microburette, the holding coil, the selection valve and the mixing cell with magnetic stirrer.

Table 1

Summary of data calibration results for the Cd-ISE and Pb-ISE in static conditions according to Nernstian model. Slope and intercept values obtained from the linear fit of the linear response zone.

Sensor	Response to distilled water (mV)	Linear fit data	
		Intercept (mV)	Slope (mV/dec)
Cd ²⁺	−365.9	−272.19	16.55
Pb ²⁺	138.7	321.29	26.45

zone was determined from the calibration curves for each sensor. Slope and intercept describing the linear response for each sensor are contained in Table 1. Characterization in flow condition was done by firstly examining the sensors in distilled water, then in acetate buffer and lastly integrating them in the flow cells to determine the responses to single metal solutions automatically prepared by the SIA system. Detection limit was taken at the point of intersection of the two asymptotic behaviours of calibration curves, as recommended by IUPAC. Calibration results are summarized in Table 2. Characterization shows that cadmium electrode response to Cd²⁺ ion presents poor response, having a sensitivity value lower than the theoretical one for double charged ions (29 mV/dec). Despite this bad response, the electrode was not replaced given that was the only one available.

After characterization, the cell containing the set of electrodes was subject to injections of binary mixtures of heavy metals. Samples were injected into the cell with a step profile and the transient response for each ISE was simultaneously recorded during 60 s per sample in time steps of 0.1 s, obtaining two data sets of potentials, one per ISE.

2.4. Information processing and modelling

Both matrices of potentials were preprocessed by DWT to reduce the number of variables before entering the ANN. Preprocessing and modelling were done using MATLAB. DWT was carried out with the Wavelet Toolbox. DWT is implemented in MATLAB using Mallat's pyramidal algorithm (Fig. 2). This signal processing technique operates over a single discrete signal by applying two digital complementary filters performing low-pass and high-pass filtering followed by downsampling. The result of the decomposition is two sets of coefficients named approximation and detail coefficients, each one correlated with the low- and high-frequency content of the signal, respectively. The size of each set after transforming the signal is approximately half the original size. The process can be repeatedly applied on either set of coefficients, taking the decomposition procedure one level down and obtaining two more sets of coefficients (approximation and detail) each time the transformation is carried out. In this work, compression was accomplished by retaining the approximation coefficients resulting from this transformation and discarding the detail coefficients.

The full set of discrete wavelets available in MATLAB's toolbox (the four families Daubechies, Coiflets, Symlets and Biorthogonal Spline) was evaluated with decomposition levels ranging from 1 up to 9. The goal was to find the combination that performed the best in terms of retaining most of the original information with the fewer number of components for correctly modelling the next step. The number of combinations evaluated was 324, considering each Wavelet family was also assayed at different orders.

Calibration model was based on an ANN built with the Neural Network Toolbox. Several ANNs were also attempted in order to find the optimum structure. In all cases, ANNs were feedforward

Table 2

Summary of data calibration results for the Cd²⁺ and Pb²⁺ ISEs in flowing conditions. Sensitivities expressed as slopes obtained from linear fit of the linear working zone.

Sensor	Sensitivity to Cd ²⁺ ion (mV/dec)	Sensitivity to Pb ²⁺ ion (mV/dec)	Potentiometric selectivity coefficient $\log(K_{x,y}^{pot})^a$	Detection limit (M) ^b
Cd ²⁺	8.51	6.05	1.2 ± 0.1	2.00 × 10 ^{−6}
Pb ²⁺	8.97	20.58	−3.2 ± 0.2	4.48 × 10 ^{−5}

^a Uncertainty intervals calculated at 95% of confidence level.

^b Determined as the intercept between the linear Nernstian behaviour and the horizontal line in absence of response.

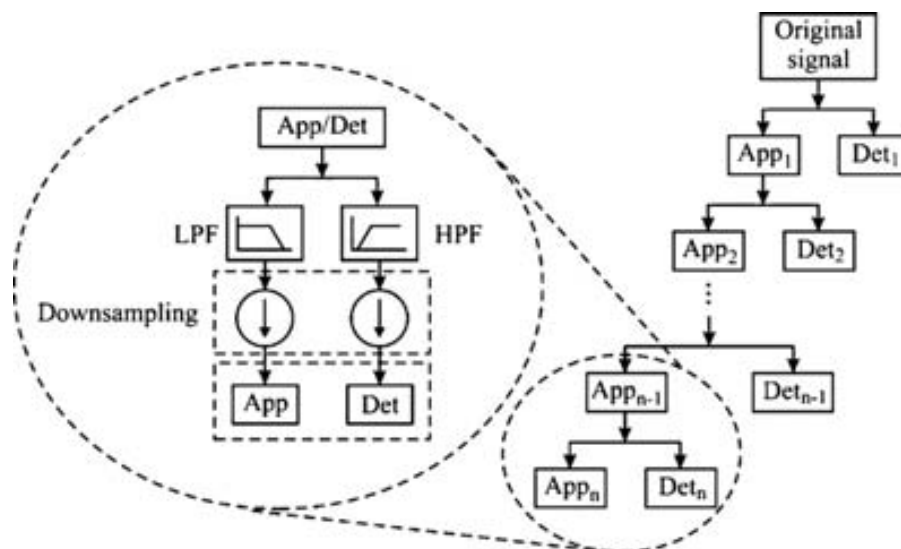


Fig. 2. Mallat's pyramidal algorithm implemented in MATLAB's Wavelet Toolbox. Approximation and detail coefficients are indicated by App and Det, respectively. Subindexes denote decomposition level up to an n -th level. LPF and HPF stand for low-pass and high-pass filters, respectively. Downsampling is represented by a circle with a centred arrow pointing downwards.

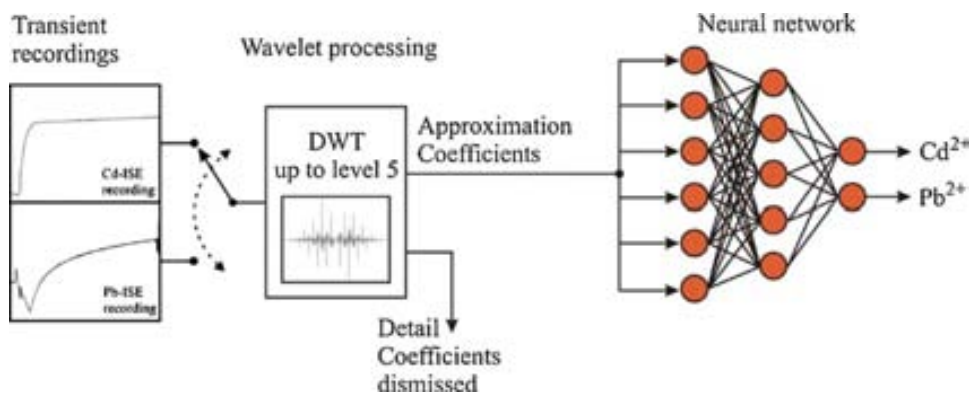


Fig. 3. Scheme depicting both the information processing and modelling stages. Recordings are processed by the DWT and the approximation coefficients obtained from the decomposition are used to feed the ANN for building the calibration model.

type trained with the Bayesian regularization algorithm for better fitting the test set. Approximation coefficients were taken as inputs, and concentrations of cadmium and lead as targets. Sets of inputs and targets were split into two subsets for training and testing purposes. Structures with one and two hidden layers, different number of neurons (from 6 up to 30) and different non-linear transfer functions in their hidden layers were tested. A scheme showing the coupling of preprocessing and modelling stages is depicted in Fig. 3.

3. Results

The departure universe of data consisted of two input matrices (one per ISE) formed by the transient recordings (600 data values) obtained with the 45 samples, plus a target matrix formed by the corresponding concentrations of each binary mixture of heavy metals. The size of each recording made mandatory the compression before building the calibration model with the ANN. As reference, two recordings are shown in Fig. 4, one per each ISE, where one can visualize that the glass membrane ISE (the Cd^{2+} electrode) displays a faster response than the polymeric membrane type (the Pb-ISE).

The combination of wavelet and decomposition level used in our application was chosen based on the compression ratio, percentage of total energy retained by approximation coefficients after each decomposition and degree of similarity between the original recording and the one reconstructed using only the coefficients retained. Compression ratio is the relationship between the original data size to the number of approximation coefficients

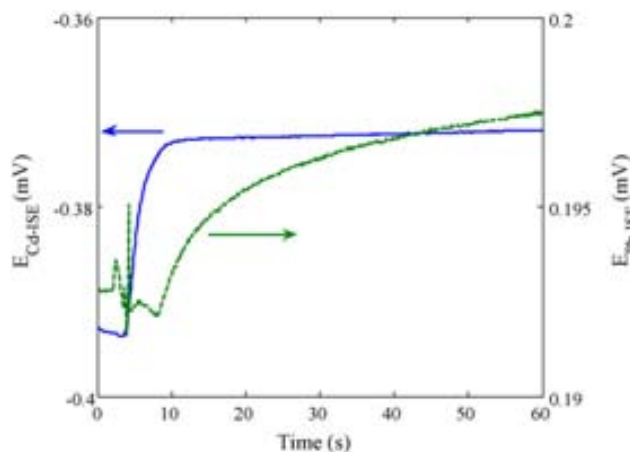


Fig. 4. Example of two transient signals recorded with the ISEs after step insertion of a sample into the cell. Analyte mixture corresponds to concentrations (lead and cadmium) 85.7 and 49.4 μM , respectively.

obtained after decomposition. Total energy is defined as the sum of all squared detail coefficients over all scales plus the square of the remaining approximation coefficients. Finally, comparisons between the original and reconstructed recordings were carried out at all levels of decomposition. This evaluation is proposed since we have loss of information after dismissing the detail coefficients.

Compression ratios reported values as low as 1.9 for Coiflets Wavelet of fifth order and decomposition level 1, to as high as 300 for Daubechies Wavelet of first order and decomposition level 9. Bar graphs in Fig. 5 show the number of approximation coefficients obtained by DWT for the 326 combinations detailed above. Energy retained by approximation coefficients was over 99% of total energy contained in the complete set of coefficients for all decomposition levels, meaning that almost no high-frequency components are contained in raw recordings (most of the high-frequency components are noise).

Degree of similarity between original and reconstructed signals was evaluated by running a test described in [25]. The test compares two discrete signals of the same size and yields a result between 0 and 1 depending on how they match. The result is 0 when the two signals have nothing in common and increases with its matching. Results along the different tests done ranged from 0.746 for signals processed with the Coiflets Wavelet of first order, up to 0.999 for signals processed with the set of second order filters (for decomposition and reconstruction) contained in the Biorthogonal Spline Wavelet, being the latter the family showing best performance.

Approximation coefficients from decomposition levels 1 to 4 and 6 to 9 for all wavelets were not considered for ANN modelling given that (i) the large amount of coefficients from levels 1 to 4 made the network training last too long, (ii) the coefficients from level 6 did not helped the network to converge to the desired error during training and (iii) signals reconstructed with coefficients from levels 6 to 9 yielded similarity values below 0.95. This threshold, imposed to assure a faithful representation in the compression stage, was the reason that only approximation coefficients obtained at decomposition level 5 were used for building the calibration model.

Among the tests, the wavelet function that best performed was the Biorthogonal Spline Wavelet of order 3 for decomposition and order 1 for reconstruction ('bior3.1', according to MATLAB's nomenclature). The number of approximation coefficients retained after decomposition was 23, yielding a final data matrix of size 23×45 for each set of transient signals. Mean degree of similarity was 0.985 for Cd^{2+} ISE and 0.991 for Pb^{2+} ISE. For comparison purposes, Fig. 6 shows two plots for the degree of similarity obtained when original signals from Cd-ISE and Pb-ISE were compared against their corresponding signals reconstructed using the approximation coefficients after wavelet decomposition. Only wavelets of order 3 are

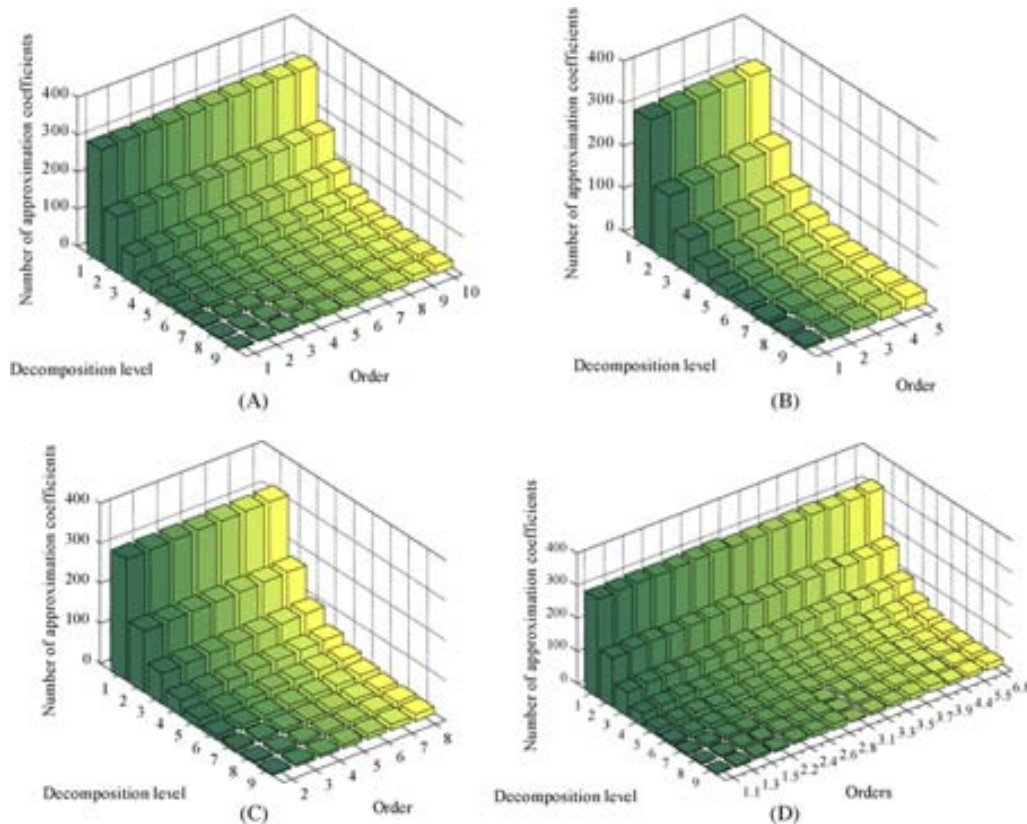


Fig. 5. Bar graphs showing the number of approximation coefficients obtained with nine decomposition levels using discrete wavelet families (A) Daubechies, (B) Coiflets, (C) Symlets and (D) Biorthogonal Spline. All wavelets were tested at all orders available. For the case of the Biorthogonal Spline family, the format used to indicate the orders is as follows, the first number corresponds to the order for decomposition and the second number to the order of reconstruction.

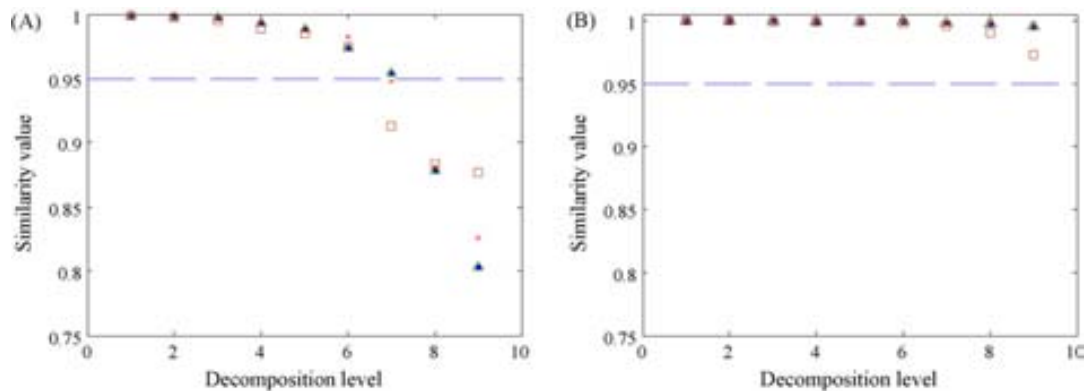


Fig. 6. Plots for similarity values from reconstructed and original recordings obtained with (A) Cd-ISE and (B) Pb-ISE. The dashed line marks the 0.95 value imposed as limit. The markers identify the wavelet family: (●) Daubechies, (△) Symlets, (+) Coiflets, (□) Biorthogonal Spline. Plots were obtained with wavelets of third order for comparison purposes.

indicated since the chosen wavelet is of order 3 for decomposition.

Regarding the calibration model, all trained networks had structures with one or two hidden layers and one output layer. Non-linear activation functions of sigmoidal type were used in hidden layers while linear activation function was used in the output layer. Neurons for the one hidden layer case varied from 6 up to 30. Neurons for the two hidden layers case varied from 6–24 up to 24–6. These combinations were obtained by increasing in one the neurons in the first hidden layer while decreasing in one the neurons in the second hidden layer. Structures with one output for independently modelling each metal ion were proposed, as well as structures with two outputs for simultaneous quantification of cadmium and lead.

Wavelet coefficients and concentrations were randomly split into two subsets for training and testing purposes, 75% of total information was used to determine the parameters of the model while the remaining was used for testing its generalization capability. No third subset was used for internal validation, given the training algorithm used (Bayesian regularization) prevents overfitting by other means. In order to facilitate convergence, input and output data in training subset were firstly normalized in the interval $[-1, 1]$. Data normalization of testing subset was done according to maximum and minimum values obtained from training subset. The sum of squared errors (SSE), from the differences between the expected and calculated values, was used for tracking convergence during training. Error goal was set to 0.09 since this value had proven to avoid overfitting in our work and yielded low dispersion

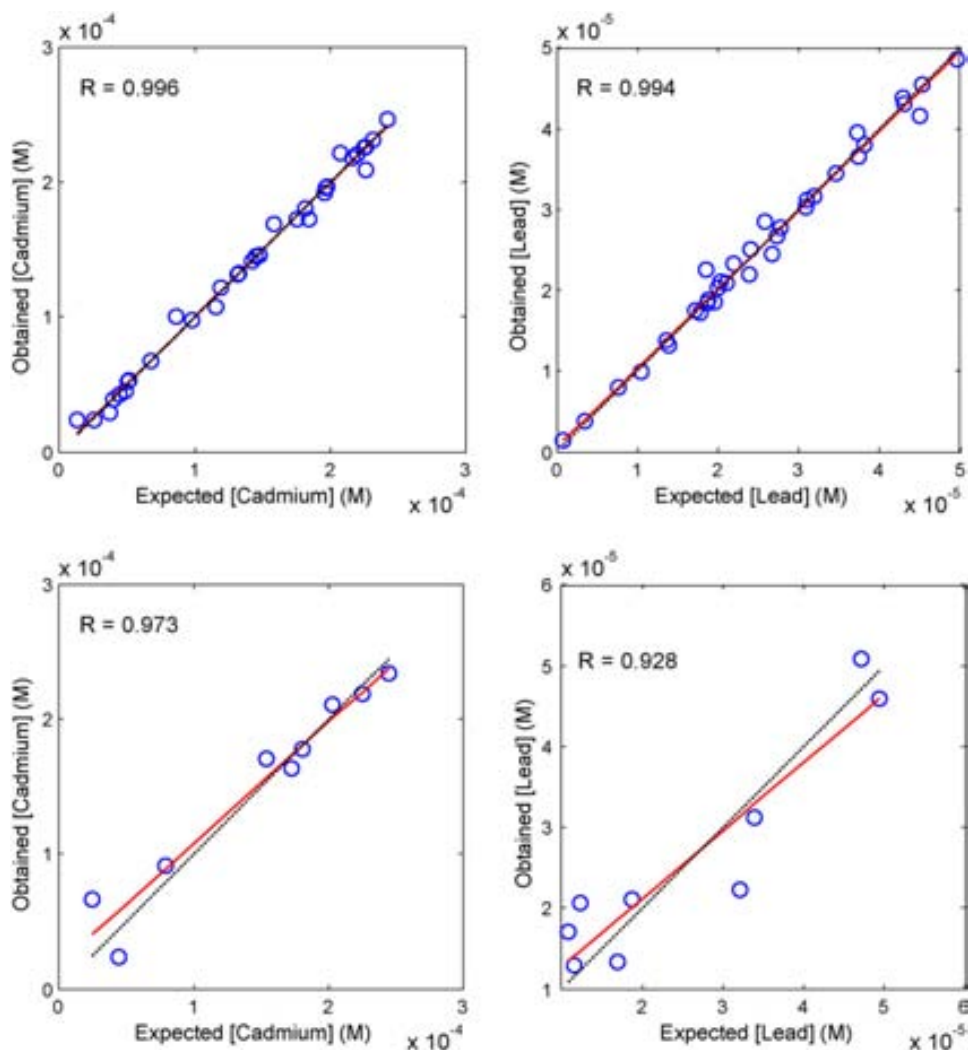


Fig. 7. Comparison of obtained vs. expected concentration for cadmium (left) and lead (right) using the recordings from the Cd-ISE. Dashed line corresponds to ideality and solid line corresponds to the obtained fit. Plots in the upper row are the results for training, plots in the lower row are the results for testing.

when linear regression analysis is performed between the obtained and expected values. SSE and correlation coefficients of the previous comparison were also used as figures of merit when the trained networks were tested.

None of the network structures with two outputs, as well as structures with one hidden layer and two outputs performed well in testing although training was fulfilled. For these cases, error goal was met in almost 50% of training runs but correlation coefficients obtained in testing were below 0.6 and even negative.

Networks with one output for single analyte quantification were those showing better performance. The structure that worked the best had 10 neurons in the first hidden layer, 4 neurons in the second hidden layer, and one neuron in its output layer, both for Cd and for Pb sensor signals. Hyperbolic tangent sigmoid function was used as transfer function in both hidden layers, and linear transfer function in the output layer. Simultaneous analyte determination was accomplished using two parallel networks with the aforementioned structure, one per metal ion and fed with the same DWT reduced transient. Fig. 7 shows the regression plots between the expected and predicted concentration for cadmium and lead obtained with the Cd²⁺ electrode, Fig. 8 are the equivalent plots for the Pb²⁺ electrode, an ISE with a completely different membrane type, and very different intrinsic response rate. Cd²⁺ sensor employed a chalcogenide glass and was fast responding, while Pb²⁺ sensor used a potentiometric PVC membrane and was much slower.

Effectiveness of the data processing methodology was corroborated by doing 25 additional trainings with a 9-fold cross validation over each electrode's recordings. Input and output data were randomly split into five parts, four fifths were taken for training and one fifth for testing. As previously, training data was firstly normalized and then testing data. Error goal was met in all cases. Results obtained yielded mean SSE in testing as low as $0.259 \pm 0.029 \text{ mol}^2 \text{ l}^2$ for lead prediction using the Pb-ISE, to as high as $0.725 \pm 0.049 \text{ mol}^2 \text{ l}^2$ for cadmium prediction with the same electrode. Average values for SSEs and correlation coefficients obtained with Cd-ISE are shown in Table 3; Table 4 presents the results obtained with Pb-ISE. Indicated uncertainties correspond to the 95% confidence interval for the 25 replicate training cases.

The proposed methodology was compared against the approach based on PCA preprocessing followed by ANN modelling to provide some reference. PCA was performed on both sets of recordings using the MATLAB's Statistics Toolbox. The analysis of the decomposition showed that for any of the two data sets more than 98% of variance was accounted for by the first two principal components. Two matrices were built with the two principal components from each data set and independently used for modelling. When neural networks with equivalent structures to those above were trained with this PCA preprocessed matrix, none of the networks converged to the desired error goal during training. These results, we believe, highlight the difficulties of the chemometric treatment.

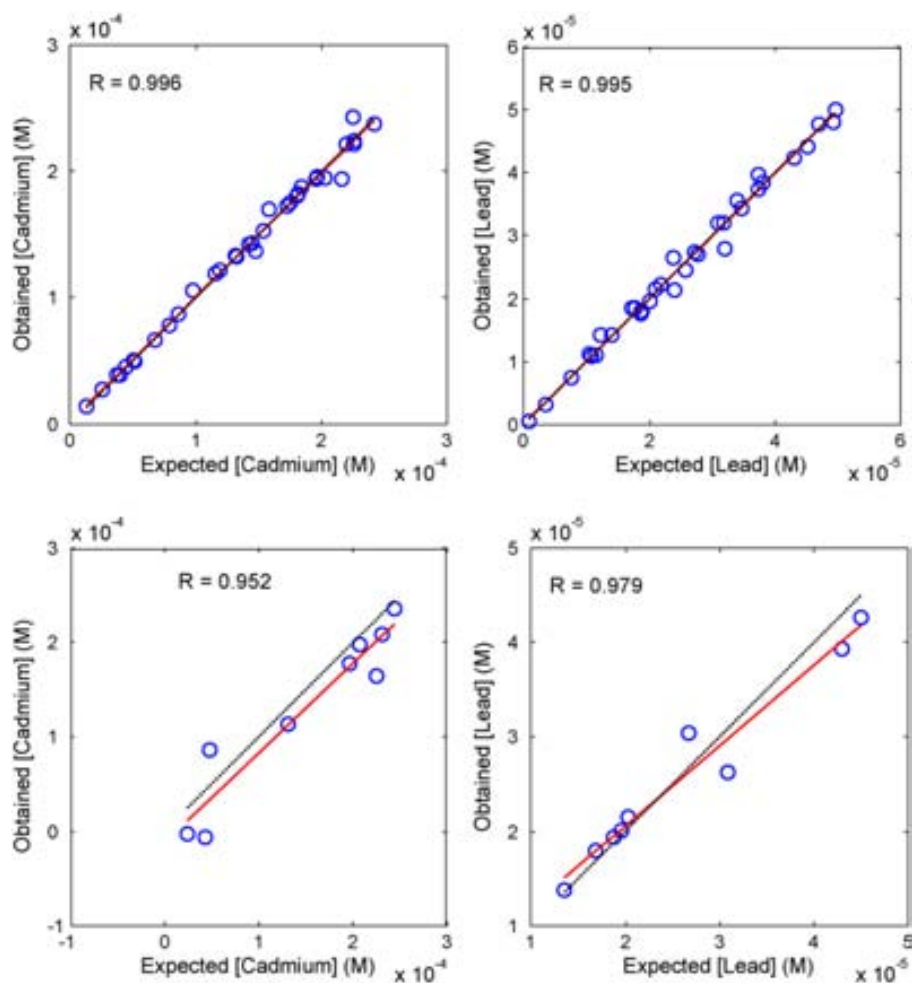


Fig. 8. Comparison of obtained vs. expected concentration for cadmium (left) and lead (right) using the recordings from the Pb-ISE. Dashed line corresponds to ideality and solid line corresponds to the obtained fit. Plots in the upper row are the results for training, plots in the lower row are the results for testing.

Table 3
Mean values for the SSE and correlation coefficient (R) obtained with the cross validation process using the Cd-ISE. Uncertainty intervals calculated at 95% confidence level.

Analyte	Training		Testing	
	SSE ($\text{mol}^2 \text{l}^{-2}$)	R	SSE ($\text{mol}^2 \text{l}^{-2}$)	R
Cd^{2+}	$0.0980 \pm 3 \times 10^{-4}$	$0.996 \pm 9.9 \times 10^{-4}$	0.544 ± 0.062	0.917 ± 0.019
Pb^{2+}	$0.0977 \pm 5 \times 10^{-4}$	$0.995 \pm 1 \times 10^{-4}$	0.715 ± 0.044	0.844 ± 0.029

Table 4
Mean values for the SSE and correlation coefficient (R) obtained with the cross validation process using the Pb-ISE. Uncertainty intervals calculated at 95% of confidence interval.

Analyte	Training		Testing	
	SSE ($\text{mol}^2 \text{l}^{-2}$)	R	SSE ($\text{mol}^2 \text{l}^{-2}$)	R
Cd^{2+}	$0.0978 \pm 5 \times 10^{-4}$	$0.996 \pm 3 \times 10^{-4}$	0.725 ± 0.049	0.926 ± 0.004
Pb^{2+}	$0.0976 \pm 4 \times 10^{-4}$	$0.995 \pm 1 \times 10^{-4}$	0.259 ± 0.029	0.934 ± 0.031

4. Conclusions

The present work shows a procedure for simultaneous quantitative determination of a binary mixture using the transient recording from a single electrode. The procedure was carried out twice with sensors of different nature to demonstrate its applicability. Our approach takes advantage of the kinetic information of the interference produced on a sensor to develop the calibration models; in the study, any of the Cd^{2+} or Pb^{2+} electrode could be used to resolve a binary mixture of the two heavy metals.

The calibration model is built by first extracting significant features employing the DWT, then evaluating the information obtained after wavelet transformation for getting a good compression/reconstruction relationship, and finally building the calibration model with an ANN. The DWT processing of the recordings has permitted the reduction of the acquired data from each electrode in a factor ca. 26. The aim of the developed application is to improve performance of existing electronic tongue systems, which may be employing signals from fewer sensors but exploiting their dynamic resolution.

Acknowledgements

This work was supported by Spanish Ministry of Science and Innovation, through project TEC2007-68012-c03-02/MIC and by CONACyT (México) through PhD scholarship 144044 for R. Cartas.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.talanta.2009.09.048.

References

- [1] A. Gutiérrez, F. Céspedes, M. del Valle, *Anal. Chim. Acta* 600 (2007) 90–96.
- [2] J. Ruzicka, E.H. Hansen, *Anal. Chem.* 72 (2000) 212A–217A.
- [3] Yu. Vlasov, A. Legin, A. Rudnitskaya, C. Di Natale, A. D'Amico, *Pure Appl. Chem.* 77 (2005) 1965–1983.
- [4] J. Gallardo, S. Alegret, M. del Valle, *Talanta* 66 (2005) 1303–1309.
- [5] J. Saurina, S. Hernández-Cassou, *Anal. Chim. Acta* 438 (2001) 335.
- [6] A. Ruiz-Medina, M.L. Fernández-de Córdoba, M.J. Ayora-Cañada, M.I. Pascual-Reguera, A. Molina-Díaz, *Anal. Chim. Acta* 404 (2000) 131–139.
- [7] P. Ciosek, Z. Brzózka, W. Wróblewski, *Sens. Actuators B* 118 (2006) 454–460.
- [8] R. Schindler, M. Watkins, R. Vonach, B. Lendl, R. Kellner, *Anal. Chem.* 70 (1998) 226–231.
- [9] C. Di Natale, A. Macagnano, F. Davide, A. D'Amico, A. Legin, Y. Vlasov, A. Rudnitskaya, B. Selezenev, *Sens. Actuators B* 44 (1997) 423–428.
- [10] E. Richards, C. Bessant, S. Saini, *Electroanalysis* 14 (2002) 1533–1542.
- [11] P. Ciosek, W. Wróblewski, *Talanta* 71 (2007) 738–746.
- [12] A. Pasamontes, M.P. Callao, *Anal. Sci.* 22 (2006) 131–135.
- [13] V. Pravidová, M. Pravda, G.G. Guilbault, *Anal. Lett.* 35 (2002) 2389–2419.
- [14] M. Blanco, J. Coello, H. Iturriaga, S. MasPOCH, M. Redón, *Anal. Chem.* 67 (1995) 4477–4483.
- [15] L. Lvova, S.S. Kim, A. Legin, Y. Vlasov, J.S. Yang, G.S. Cha, H. Nam, *Anal. Chim. Acta* 468 (2002) 303–314.
- [16] J. Polster, G. Prestel, M. Wollenweber, G. Kraus, G. Gauglitz, *Talanta* 42 (1995) 2065–2072.
- [17] J. Mortensen, A. Legin, A. Ipatov, A. Rudnitskaya, Y. Vlasov, K. Hjuler, *Anal. Chim. Acta* 403 (2000) 273–277.
- [18] M.J. Culzoni, H.C. Goicoechea, *Anal. Bioanal. Chem.* 389 (2007) 2217–2225.
- [19] P. Ciosek, K. Brudzewski, W. Wróblewski, *Meas. Sci. Technol.* 17 (2006) 1379–1384.
- [20] D. Calvo, A. Durán, M. del Valle, *Sens. Actuators B* 131 (2008) 77–84.
- [21] B. Hitzmann, A. Ritzka, R. Ulber, T. Scheper, K. Schügerl, *Anal. Chim. Acta* 348 (1997) 135–141.
- [22] B. Hitzmann, A. Ritzka, R. Ulber, K. Schöngarth, O. Broxtermann, *J. Biotechnol.* 65 (1998) 15–22.
- [23] L.S. Ferreira, M.B. De Souza Jr., J.O. Trierweiler, O. Broxtermann, R.O.M. Folly, B. Hitzmann, *Comput. Chem. Eng.* 27 (2003) 1165–1173.
- [24] R.M. de Carvalho, C. Mello, L.T. Kubota, *Anal. Chim. Acta* 420 (2000) 109–212.
- [25] L. Moreno-Barón, R. Cartas, A. Merkoçi, S. Alegret, M. del Valle, L. Leija, P.R. Hernández, R. Muñoz, *Sens. Actuators B* 113 (2006) 487–499.
- [26] A.V. Legin, A.M. Rudnitskaya, K.A. Legin, A.V. Ipatov, Yu.G. Vlasov, *Russ. J. Appl. Chem.* 78 (2005) 89–95.

Multiway Processing of Data Generated with a Potentiometric Electronic Tongue in a SIA System

Raul Cartas,^a Aitor Mimendia,^a Andrey Legin,^b Manel del Valle^{*a}

^a Sensors and Biosensors Group, Chemistry Dept., Universitat Autònoma de Barcelona, Edifici Cn, 08193 Bellaterra, Barcelona, Spain

tel. ++34-93-5811017, fax ++34-93-5812379

^b Chemistry Dept, St. Petersburg University, Universitetskaya nab. 7/9, 199034 St. Petersburg, Russia

*e-mail: manel.delvalle@uab.es

Received: October 15, 2010

Accepted: November 28, 2010

Abstract

In this work a potentiometric electronic tongue based on an array of three Ion Selective Electrodes (ISEs) as sensor elements and multiway partial least squares (N-PLS2) regression method as processing tool has been developed for simultaneous quantification of Cd²⁺, Cu²⁺ and Pb²⁺ in aqueous solution mixtures. The sensor array was formed by two chalcogenide glass sensors selective to Cd²⁺ and Cu²⁺ plus one PVC membrane sensor selective to Pb²⁺. Preparation, measurement and injection of samples in a step profile were performed by an automated SIA system. Concentration range covered was 1.05 μM to 29.8 μM for Cd²⁺, 0.05 μM to 1.5 μM for Cu²⁺ and 0.55 μM to 14.7 μM for Pb²⁺. Potentiometric recordings were arranged into a three-way array and treated by N-PLS2 regression method to obtain the calibration model. Prediction ability yielded RMSE values as low as 4.20 ± 0.37 μM, 0.094 ± 0.007 μM and 0.915 ± 0.006 μM, and correlation coefficients from linear regression analysis of 0.82 ± 0.038, 0.97 ± 0.005 and 0.97 ± 0.004 for Cd²⁺, Cu²⁺ and Pb²⁺, respectively.

Keywords: Potentiometry, SIA, Multivariate calibration, Multiway, N-PLS2

DOI: 10.1002/elan.201000642

1 Introduction

A recent trend to improve the performance of sensor analysis systems is to combine them with chemometrics data treatment for correcting artefacts, discriminating the signal of interest or resolving analytes in mixtures with interferences [1]. This trend has consolidated the use of electronic tongues (e-tongues), which are bioinspired analytical systems for liquid media, formed by an array of sensors with cross-selectivity along with a chemometric processing technique; this is needed to extract the sought information from the complex signals generated due to multiple analyte dependence of the sensors [2,3,4].

Measurements carried out for each sample by a sensor system can be of zero-order (e.g. the equilibrium potential of one electrode), first-order (e.g. a vector of potentials measured by one electrode at J different time intervals) or second-order (e.g. a matrix formed by vectors of potentials measured at J different time intervals by an array of K electrodes). As an experiment commonly will be formed by several samples, when the measured responses from the experiment are of zero-order, these can be arranged in a vector and give rise to a one-way structure; if the responses are of first-order, these can be disposed into a two-way structure; and for second-order responses, these can be arranged into a tensor or three-way array. In general, a collection of N -th order data recorded

from a set of samples creates an $(N+1)$ -way structure. These higher order structures are termed N -way or multiway arrays.

Calibration models for zero-order data are commonly obtained by applying ordinary least squares regression method. For first-order data, these models can be created by using standard linear or non-linear chemometric calibration tools such as multiple linear regression (MLR), principal component regression (PCR), partial least squares (PLS) regression, non-linear partial least squares (NLPLS) regression or artificial neural network (ANN) [5–8]. For second-order data, calibration models can be also built using any of the aforementioned first-order calibration tools, by previously performing a reduction in modes. The three-way array of measurements needs to be unfolded first into a two-way array. As disadvantages, this matricization creates discontinuities from one recording to the next, causes to lose the original relationship and creates limitations for first-order calibration [9]. Besides, models obtained by applying first-order methods to data whose multiway structure is ignored can be less robust and hard to interpret, on the opposite to models based on multiway methods, which also give better predictions and possibilities for exploring the data [10–12].

Nowadays, multiway data are increasingly being used in analytical chemistry, as it originate directly by instrumental development in hyphenated techniques, or tech-

niques like fluorescence spectroscopy, chromatography, flow injection analysis, magnetic resonance, near infrared analysis (NIR) and some others [13]. Building single- or multivariate calibration models from multiway data has been lately fulfilled by applying *N*-way Partial Least Squares (N-PLS) regression [1,14]. This tool is an extension of standard PLS regression and can be used to regress on data structures of varying number of ways. Both in PLS and N-PLS a number is appended at the end of the acronym to identify the number of modes of the responses, for PLS it is restricted up to 2. In recent years N-PLS regression has been used in a wide variety of data such as molecular descriptors to in the QSAR determination of biological activities of antifungals [15] or benzamides [16]; ultraviolet spectrophotometric or fluorescence recordings from an optosensing system to quantify pollutants in wastewater [17] and drinking water [18]; spectrophotometric or fluorimetric matrices to characterize olive oils [19], to determine kerosene fraction in diesel [20], xanthine and hypoxanthine in human urine [21], and to quantify folic acid [22,23] as well as fluoroquinolones [24] in serum samples and human urine; gas chromatograms to detect adulteration of commercial gasoline [25] and, along with sensory panelists data, to determine quality of balsamic vinegars of Modena [26]; AC voltammograms to determine concentrations of a suppressor additive in a copper metallization plating bath used in semiconductor manufacturing [27]; stripping voltammograms to quantify pesticides in river water samples [28]; and other kind of data used to predict quality and detect faulty operation in an industrial fermentation process [29].

Relevant to this communication are the works published by E. Chow et al. [30] and A. V. Legin et al. [3], both related to multiway e-tongue systems. The first one reports the use of voltammograms recorded from a set of four peptide-modified gold electrodes immersed in ternary mixtures of Cd^{2+} , Cu^{2+} and Pb^{2+} . The calibration models for their quantification were built with N-PLS. The second one proposes an e-tongue for quantifying mixtures of Zn^{2+} and Pb^{2+} from dynamic potentiometric responses recorded from a set of seven polymeric membrane sensors placed in a flow-through cell. This approach is rather different from the one reported in a previous work [31], where binary mixtures of Cd^{2+} and Pb^{2+} in aqueous solution were simultaneously quantified from transient responses recorded from a single potentiometric ISE either selective to cadmium or lead. The recordings from each ISE were arranged into a matrix and treated by the classical methodology applied in our laboratory based on Wavelet processing followed by Artificial Neural Network (ANN) modelling, since no multiway array was involved.

In the present work, we report an e-tongue to simultaneously quantify triads of heavy metal ions Cd^{2+} , Cu^{2+} and Pb^{2+} . These are targeted from transient recordings extracted from an array of three flow-through Ion Selective Electrodes (ISEs) to Cd^{2+} , Cu^{2+} and Pb^{2+} . The recordings correspond to the responses of the ISEs when

subjected to a step change of sample, supplied by a Sequential Injection Analysis (SIA) system, used for automation purposes. Sensors dynamics were used since they have proven to be helpful on differentiating primary ions from interfering species thanks to the kinetic resolution added to the system [3]. For the data treatment, N-PLS2 was used to build the regression model, as the measurements gave rise to a three-way array (tensor) defined by $\text{SAMPLE} \times \text{TIME} \times \text{ELECTRODE}$. The results obtained by multiway processing were compared against the most classical methodology applied in our laboratory based on the aforementioned methodology. Thus, this work represents one of the first attempts in which an e-tongue is devised employing the multiway N-PLS2 processing strategy.

2 Theory

2.1 Notation

Different notations will be used from here on to represent data. Vectors (either row or column) are denoted by bold lower case letters. Matrices are denoted by capital bold letters. Three- or higher way arrays are represented by underlined bold capital letters. Positive integer numbers are denoted by capital italic letters. The superscripts 't' and '+' denote a transposed matrix/vector and the Moore–Penrose pseudoinverse, respectively. The symbol \otimes denotes the Kronecker product, and the hat '^' on top of vectors or arrays denote a predicted variable. Finally, independent and dependent variables are termed predictors and responses, respectively.

2.2 PLS

Partial Least Squares (PLS) regression is a common tool in chemometrics used for building calibration models based on data sets of predictors and responses [8,32,33]. The goal of the methodology is to predict single (\mathbf{y}) or multiple responses (\mathbf{Y}), from a matrix of predictors \mathbf{X} , and to describe the common structure underlying the two variables [34].

In general, PLS simultaneously decomposes \mathbf{X} and \mathbf{y} into a set of latent variables that maximises the covariance between the independent and dependent data. For a univariate regression, \mathbf{X} and \mathbf{y} are decomposed into a common scores matrix \mathbf{T} and individual loading vectors \mathbf{P} and \mathbf{q} [6]. \mathbf{T} acts as the inner relationship element between the linear equations such that $\mathbf{X} = \mathbf{TP}^t + \mathbf{E}_x$ and $\mathbf{y} = \mathbf{Tq}^t + \mathbf{e}_y$, where \mathbf{E}_x and \mathbf{e}_y represent the residual matrix and vector for \mathbf{X} and \mathbf{y} , respectively. To predict values of the dependent variable, the multivariate regression model $\hat{\mathbf{y}} = \mathbf{Xb} + \mathbf{e}_y$ is applied, where the regression coefficients \mathbf{B} are derived from the model parameters [35].

2.3 N-PLS

N-PLS regression is a generalization of the two-way PLS regression method applied to higher order data sets [10,36]. The methodology was introduced by Bro in 1996 and later improved in 2001 [37]. Several publications describing the mathematical foundations of this methodology have been published [10–12,38,39] and only a brief description will here be given.

As an extension to two-way PLS regression, N-PLS regression builds a calibration model incorporating a relationship between the sets of predictors and responses based on the multiway structure of the arrays. The main difference between N-PLS compared to two-way PLS regression is the use of larger dimensional loading matrices for decomposing the multiway arrays.

The goal of the methodology is to simultaneously fit multilinear models formed by score and loading vectors from both predictors and responses arrays, along with a regression model relating the two decomposition models. According to the underlying theory of PLS regression, the model is fitted such that the score vectors have maximum covariance with the unexplained part of the dependent variable [10].

For three-way structures $\underline{\mathbf{X}}$ and $\underline{\mathbf{Y}}$ with sizes $(I \times J \times K)$ and $(I \times L \times M)$ respectively, the methodology models the arrays of predictors and responses in a multilinear PAR-AFAC-type decomposition. The models for the matrixized arrays are defined by $\mathbf{X} = \mathbf{T}\mathbf{G}_X(\mathbf{P}^J \otimes \mathbf{P}^K)^t + \mathbf{E}_X$ and $\mathbf{Y} = \mathbf{U}\mathbf{G}_Y(\mathbf{Q}^L \otimes \mathbf{Q}^M)^t + \mathbf{E}_Y$, respectively, where \mathbf{T} and \mathbf{U} are score vectors, \mathbf{P} and \mathbf{Q} are loading vectors, and \mathbf{E}_X and \mathbf{E}_Y are residual matrices. Superscripts J , K , L and M define to which way the loading vector refers to.

\mathbf{G}_X is a matrixized core array defined by $\mathbf{G}_X = \mathbf{T}^+ \mathbf{X}(\mathbf{P}^J \otimes \mathbf{W}^K)^+$ that in the improved N-PLS version solves various problems related with the modelling of $\underline{\mathbf{X}}$ [37]. \mathbf{G}_Y is defined likewise for $\underline{\mathbf{Y}}$.

Using the trilinear models previously obtained, the relationship between $\underline{\mathbf{X}}$ and $\underline{\mathbf{Y}}$ is found by a regression model of the kind $\mathbf{U} = \mathbf{T}\mathbf{B} + \mathbf{E}_U$. Regression coefficients \mathbf{B} for the N-PLS model can be found following the procedures described in [38,39].

3 Experimental

3.1 Equipment

A SIA system available at our laboratory was used to automatically prepare the ternary mixtures of heavy metal ions by dilution from stocks. The SIA system is formed by a fluidic stage consisting on an automatic microburette equipped with a syringe, a holding coil, an 8 way Hamilton MVP selection valve and a home-made Perspex mixing cell with a magnetic stirrer; a measurement stage is also used comprising the sensors, a reference electrode and an 8-channel signal conditioning circuit connected to a Data Acquisition System. Details on models, sizes and volumes of fluidic devices along with a figure depicting

the system can be found elsewhere [31,40]. All active elements involved in the SIA system are controlled using a virtual instrument developed in LabView [41] through the RS-232 ports of a PC Pentium III at 600 MHz.

3.2 Reagents and Solutions

Standards were prepared by the SIA system using a 0.05 M acetic/acetate buffer at pH 4.5 as carrier and diluting solution. Triads of concentration standards were randomly generated to be automatically prepared by the SIA system. Analytes were in ranges 1.05 μM to 29.8 μM for Cd^{2+} , 0.05 μM to 1.5 μM for Cu^{2+} and 0.55 μM to 14.7 μM for Pb^{2+} , determined by the optimal lower working range of the sensors used. All solutions (nitrate salts) were prepared using doubly distilled water and analytical grade (or similar) reagents.

3.3 Sensors

Three ion selective electrodes (ISEs) with cross response to the tested metal ions plus an Ag-AgCl reference electrode were integrated into the measuring cell. Electrodes responding to Cu^{2+} and Cd^{2+} used chalcogenide glass membranes and the one responding to Pb^{2+} used a PVC membrane with trioctylphosphine oxide as ionophore. ISEs were fabricated at the Chemistry Department of St. Petersburg University [42].

3.4 Procedure

The cell containing the set of electrodes was subjected to injections of standards with ternary mixtures of the heavy metals studied. Samples were injected into the cell with a step profile and the transient response for each ISE was simultaneously recorded during 60 s per sample in time steps of 0.1 s.

3.5 Data

Three data matrices of potentials were obtained, one per ISE. Each matrix was formed by 81 transient recordings of potentials with 600 data values each. The matrices were arranged into a three-way array of predictors $\underline{\mathbf{X}}$. Also, to be used is the matrix $\underline{\mathbf{Y}}$ of responses formed by 81 triads of concentrations for the three heavy metals studied.

3.6 Software

Data was handled using Matlab 7.1 version (The Mathworks, Matick, USA). Preprocessing and N-PLS regression model was carried out using “*The N-way Toolbox for MATLAB*”, 3.1 version, developed by R. Bro, University of Copenhagen [36].

4 Results and Discussion

4.1 Characterization of Sensors Responses

Results for sensitivities, limit of detection (*LOD*) for single ion solution, potentiometric selectivity coefficients obtained for two metal mixtures according to Nikolsky–Eisenmann equation and details on experimental points used for selectivity experiments of the three sensors employed in the array are given in previous work [42]. As shown in there, where an e-tongue application was developed, but employing ANNs, the different degree of response of each electrode to the three metal ions considered assures the cross-sensitivity condition, necessary in every e-tongue system.

4.2 N-PLS2 Treatment

Recordings from electrodes and concentration triads were arranged into a tensor \mathbf{X} of size (81 samples \times 600 data points \times 3 electrodes) and a matrix \mathbf{Y} of size (81 samples \times 3 metal ions) prior to modelling and testing (Figure 1).

Inner trilinearity of the data set was checked before N-PLS2 modelling. The tensor \mathbf{X} was unfolded into a row-wise (81 \times 1800), a column-wise (48600 \times 3) and a tube-wise (243 \times 600) set of matrices and Singular Value Decomposition was performed on each of them. If trilinearity exists then the number of significant eigenvalues must be the same on each unfolded matrix [43]. The first three eigenvalues obtained from column-, row- and tube-wise matrices are [86.54; 1.47; 0.86], [86.53; 1.79; 0.84] and [86.55; 1.13; 0.16], respectively. It is seen that from each augmented matrix we obtained 2 significant eigenvalues, thus confirming in this way the trilinearity of the data.

Data was split into two subsets by taking alternate recordings-concentrations pairs. Odd indexed subset was used for building the N-PLS2 model while even indexed subset was used for testing the generalization capability. Given the initial random generation of concentrations, no special treatment was given to this data splitting.

Illustrated in Figure 2 are three recordings, one for each ISE, displayed to visualize sensors behaviour. From the plots it is possible to visualize the faster and higher

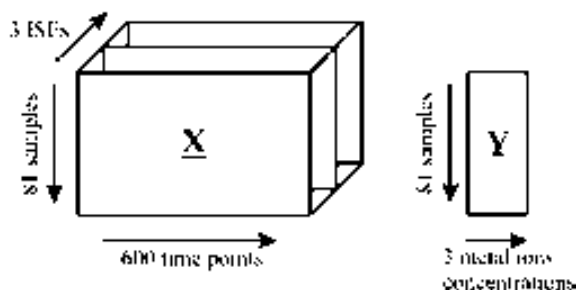


Fig. 1. Schematic representation of recordings and triads of concentrations arranged into a three-way array and a matrix, respectively, for N-PLS2 modelling.

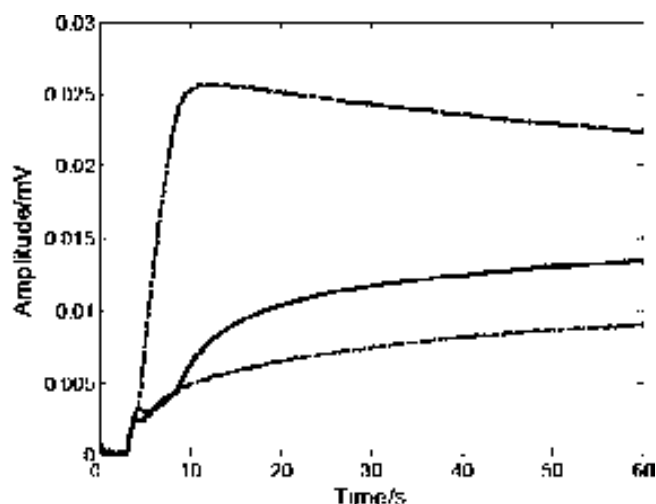


Fig. 2. Example of transient response signals recorded with the Cd- (dashed line), Cu- (continuous line) and Pb-ISE (points) after step insertion of a sample into the cell. Recordings correspond to analyte concentrations [2.65 μ M, 1.065 μ M, 4.7 μ M] for Cd^{2+} , Cu^{2+} and Pb^{2+} , respectively. Graphs are scaled and displaced to a common zero offset for comparison purposes.

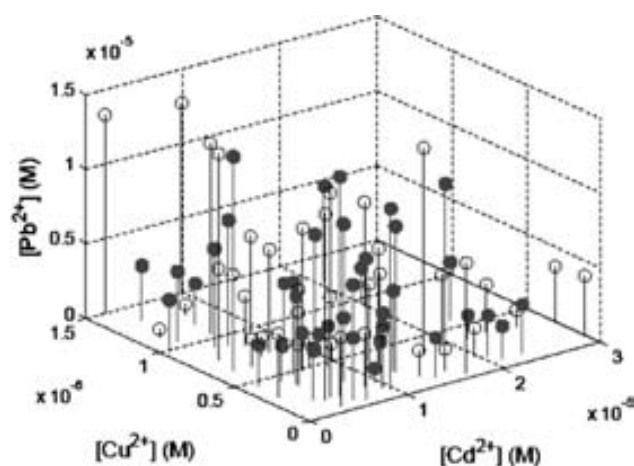


Fig. 3. Space distribution of the 81 metal ion concentrations randomly generated for being prepared by the SIA system. Triads used for building the first calibration model (\circ) are differentiated from those used to test the generalization capability (\bullet). Calibration and testing points used in bootstrapping process are different from those shown here.

responses of the glass membrane ISEs, compared to the polymeric membrane types. Randomly generated concentrations of metal ions used in this study are plotted in Figure 3, where modelling and testing subsets are visually differentiated.

Modelling subset used for building the N-PLS2 model were firstly mean centred along the first way according to Gurden et al. [44]. Mean values obtained from this pre-processing stage were later used for centring the testing data sets. No scaling process was performed on any of the modes.

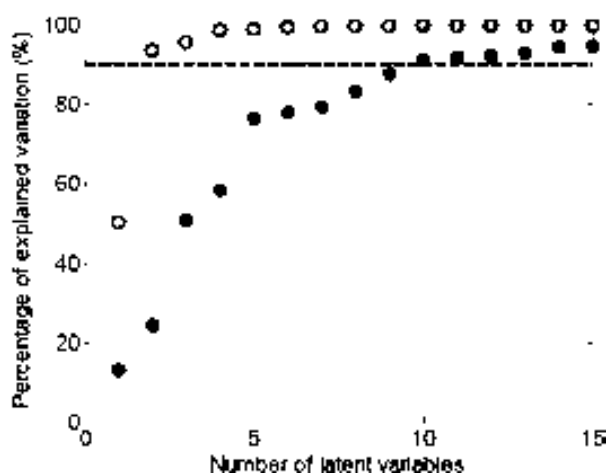


Fig. 4. Percentage of variation explained by latent variables for N-PLS models fitted from 1 up to 15 latent variables. Empty circles correspond to **X** and filled circles to **Y**. Dashed line marks the percentage limit that determines the optimal number of LVs.

Multivariate regression models were built using 1 to 15 latent variables (LV). Percentage of explained variation for predictors and responses was used to find the significant number of LVs yielding a reliable model. The number of LVs versus the explained variance for **X** and **Y** is plotted in Figure 4. From this figure it is seen that, for **X**, 2 LVs are enough to explain more than 90% of variance while for **Y**, 10 LVs are needed for explaining a similar percentage. The final NPLS-2 regression model was built with 10 LVs, since it is the minimum number of terms needed to explain 90% of variance from the concentrations set. The fitting error (*RMSE*) obtained with modelling subset were 2.53 μM , 74.8 nM and 0.64 μM for Cd^{2+} , Cu^{2+} and Pb^{2+} , respectively, which correspond to 8.5%, 5.0% and 4.4% of maximal concentrations employed. Generalization capability was tested with the subset previously reserved for this purpose. Tests yielded *RMSE* values of 3.85 μM , 96.4 nM and 0.84 μM for Cu^{2+} , Cd^{2+} and Pb^{2+} , respectively, which correspond to 12.9%, 6.4% and 5.7% of maximal concentrations employed.

Comparison graphs of expected vs. obtained concentrations were built to check performance. Plots obtained with modelling and testing subsets are shown in Figure 5, where good correlation between expected and obtained

values is clear for the three ions and both subsets, especially for Cu^{2+} and Pb^{2+} whose plots show much lower dispersion than that for Cd^{2+} . This dispersion may be attributable to the interference effect that the Cu^{2+} and Pb^{2+} ions caused to the Cd^{2+} sensor, the least selective of the three ISEs. Results of correlation factors, slopes and intercepts obtained from the linear regression analysis for the comparisons of the three ions are summarized in Table 1.

The efficiency of the proposed methodology was corroborated by a bootstrapping procedure based on fifteen additional fittings with 10 LVs each. Data sets **X** and **Y** were randomly split into modelling and testing subsets each time a new model was created. Splitting was done by taking 41 experimental points for modelling and 40 for testing. Indexes identifying both subsets were saved for further comparison work. Total *RMSE* obtained were $1.54 \pm 0.08 \mu\text{M}$ and $2.46 \pm 0.21 \mu\text{M}$ for calibration and testing, respectively. Average values for slopes, intercepts and correlation coefficients of the expected versus obtained concentration values obtained by linear regression analysis for modelling and testing cases are also in Table 1. These results were compared with the most classical approach used in our laboratory based on Discrete Wavelet Processing followed by ANN modelling.

4.3 Comparison with Wavelet Processing Followed by ANN Modelling (DWT-ANN)

The transient ISE signals were Wavelet processed by using the Matlab's *Wavelet Toolbox*. This treatment was chosen as reference as it has shown interesting results on related study cases [31,45,46]. It helps to reduce the size of each recording by choosing a combination of few Wavelet coefficients that permits to rebuild the basic shape of the original signal without fine details, and makes possible an ANN model.

A compromise exists between the wavelet function, the number and type of coefficients picked up (only approximation or approximation plus detail) and the information retained by them. The fewer the number of coefficients kept for a fixed wavelet function, the smoother the reconstructed signal is given that information is lost when detail coefficients are dismissed.

Table 1. Mean values for slope (*m*), intercept (*b*) and correlation coefficient (*R*) of the comparison graphs of obtained/vs./expected concentrations, as calculated by linear regression analysis from the first N-PLS2 regression model (fm) and its bootstrapping validation process (bs). Uncertainty intervals calculated at 95% confidence level.

Analyte		Modelling			Testing		
		<i>m</i>	<i>b</i>	<i>R</i>	<i>m</i>	<i>b</i>	<i>R</i>
Cd^{2+}	fm	0.893	1.14×10^{-6}	0.945	0.842	1.31×10^{-6}	0.83
	bs	0.87 ± 0.016	$(1.33 \pm 0.14) \times 10^{-6}$	0.934 ± 0.009	0.82 ± 0.042	$(2.30 \pm 0.54) \times 10^{-6}$	0.821 ± 0.039
Cu^{2+}	fm	0.965	1.96×10^{-8}	0.983	0.957	2.41×10^{-9}	0.966
	bs	0.967 ± 0.008	$(1.70 \pm 0.48) \times 10^{-8}$	0.983 ± 0.004	0.939 ± 0.033	$(2.85 \pm 0.17) \times 10^{-8}$	0.973 ± 0.005
Pb^{2+}	fm	0.976	1.30×10^{-7}	0.988	0.916	6.84×10^{-7}	0.974
	bs	0.969 ± 0.006	$(1.58 \pm 0.27) \times 10^{-7}$	0.984 ± 0.003	0.976 ± 0.028	$(1.01 \pm 1.8) \times 10^{-7}$	0.973 ± 0.004

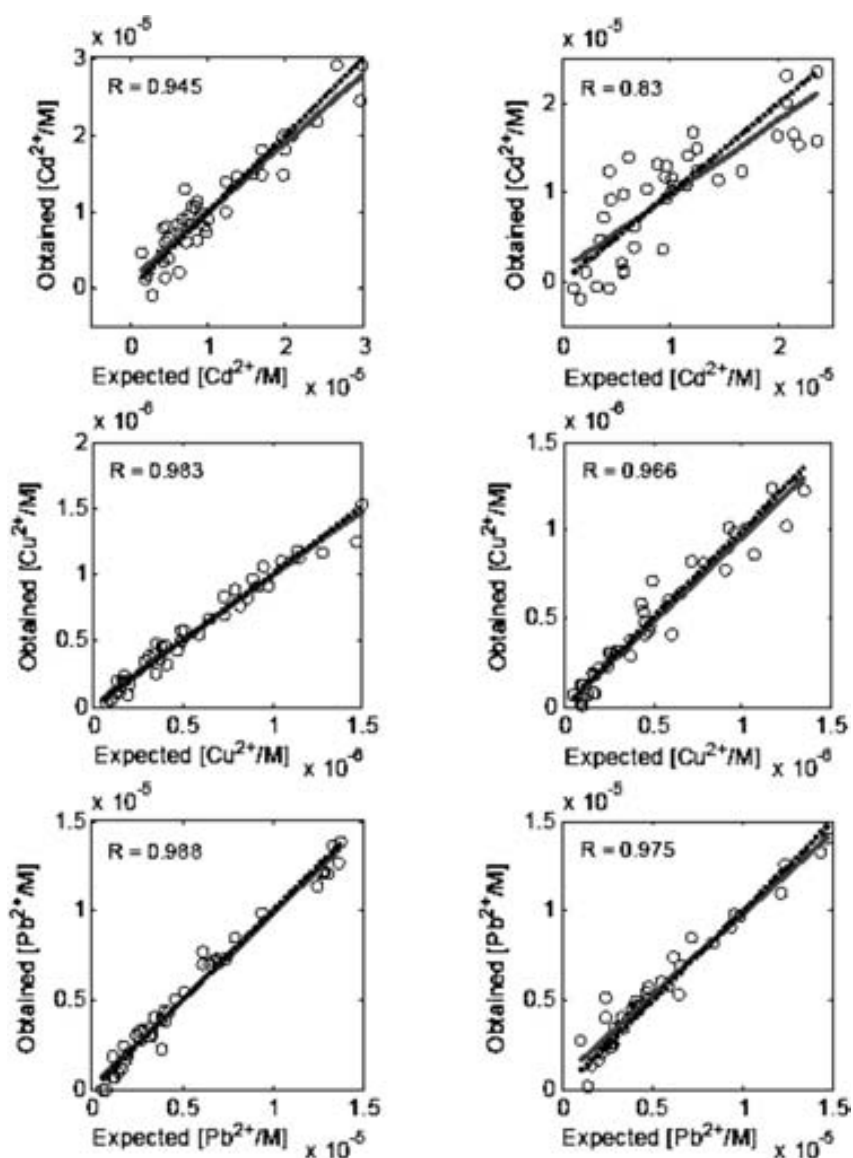


Fig. 5. Comparison of obtained vs. expected concentration for Cd^{2+} (upper), Cu^{2+} (middle) and Pb^{2+} (lower) using N-PLS2 processing. Dashed line corresponds to ideality and solid line corresponds to the obtained fit. Plots at left correspond to modelling and plots at right to testing.

Discrete Wavelet families tested were Daubechies (db), Coiflets (coif), Biorthogonal (bior) and Symlets (sym) of various orders and decomposition levels ranging from 1 up to 8, being the last three highest levels those yielding the smallest efficient combination of coefficients for certain wavelets. Visual inspection of original vs. reconstructed signals indicated that combinations up to 20 approximation plus detail coefficients are enough for fulfilling the purpose of an accurate reconstruction. This statement was later confirmed by calculating the error between the original and reconstructed signals. Table 2 briefs the information related with the Wavelets and decomposition levels yielding a total of up to 20 coefficients. Signals were finally compressed by using the Wavelet bior2.2 and taking the decomposition up to level 7. This wavelet/decomposition level combination allowed for the lowest

RMSE with the fewer coefficients. Sixteen coefficients (9 approximation coefficients plus 7 detail coefficients) were kept from each recording, obtaining a compression ratio of 37.5. The data matrix generated after processing was of size $[48 \times 81]$, formed by sequentially joining the wavelet coefficients obtained from electrodes Cu^{2+} , Cd^{2+} and Pb^{2+} .

Artificial Neural Networks, fed with the calculated coefficients for each sensor, were feedforward type with one hidden layer and one output layer. Structures with varying number of neurons in its hidden layer and two different sigmoidal transfer functions (logarithmic and tangential) were tested. The output layer had three neurons with linear transfer function for simultaneously modelling the analytes. The complete data set was split into two subsets for training and testing purposes; odd num-

Table 2. Root mean squared errors obtained as the result of comparing original recordings /vs./ those reconstructed by using a few Wavelet coefficients, as the compression procedure prior to ANN modelling. Shown in the table are only those Wavelet/Order/Level combinations yielding low RMSE with a total of up to 20 approximation/detail coefficients.

Wavelet	Order	Level	Number of coefficients	RMSE
Daubechies (db)	2	6	19	$1.3 \times 10^{-4} \pm 1.6 \times 10^{-5}$
		7	16	$1.4 \times 10^{-4} \pm 1.5 \times 10^{-5}$
	3	6	20	$1.6 \times 10^{-4} \pm 2.0 \times 10^{-5}$
		7	19	$1.6 \times 10^{-4} \pm 2.0 \times 10^{-5}$
Symlets (sym)	2	8	20	$1.7 \times 10^{-4} \pm 2.0 \times 10^{-5}$
		6	19	$1.3 \times 10^{-4} \pm 1.6 \times 10^{-5}$
		7	16	$1.4 \times 10^{-4} \pm 1.5 \times 10^{-5}$
	3	8	18	$1.4 \times 10^{-4} \pm 1.5 \times 10^{-5}$
		6	20	$1.6 \times 10^{-4} \pm 2.0 \times 10^{-5}$
Coiflets (coif)	1	6	19	$1.6 \times 10^{-4} \pm 2.0 \times 10^{-5}$
		7	19	$1.1 \times 10^{-4} \pm 1.1 \times 10^{-5}$
		7	18	$1.2 \times 10^{-4} \pm 1.1 \times 10^{-5}$
		8	19	$1.4 \times 10^{-4} \pm 1.1 \times 10^{-5}$
Biorthogonal (bior)	2.2	6	19	$1.1 \times 10^{-4} \pm 1.2 \times 10^{-5}$
		7	16	$1.2 \times 10^{-4} \pm 1.2 \times 10^{-5}$
		6	15	$2.2 \times 10^{-4} \pm 2.8 \times 10^{-5}$
	3.1	6	15	$2.2 \times 10^{-4} \pm 2.8 \times 10^{-5}$
		7	12	$2.2 \times 10^{-4} \pm 2.8 \times 10^{-5}$
		8	13	$2.2 \times 10^{-4} \pm 2.9 \times 10^{-5}$
	3.3	6	20	$1.7 \times 10^{-4} \pm 2.0 \times 10^{-5}$
		7	19	$1.4 \times 10^{-4} \pm 1.8 \times 10^{-5}$
8		20	$1.6 \times 10^{-4} \pm 1.8 \times 10^{-5}$	

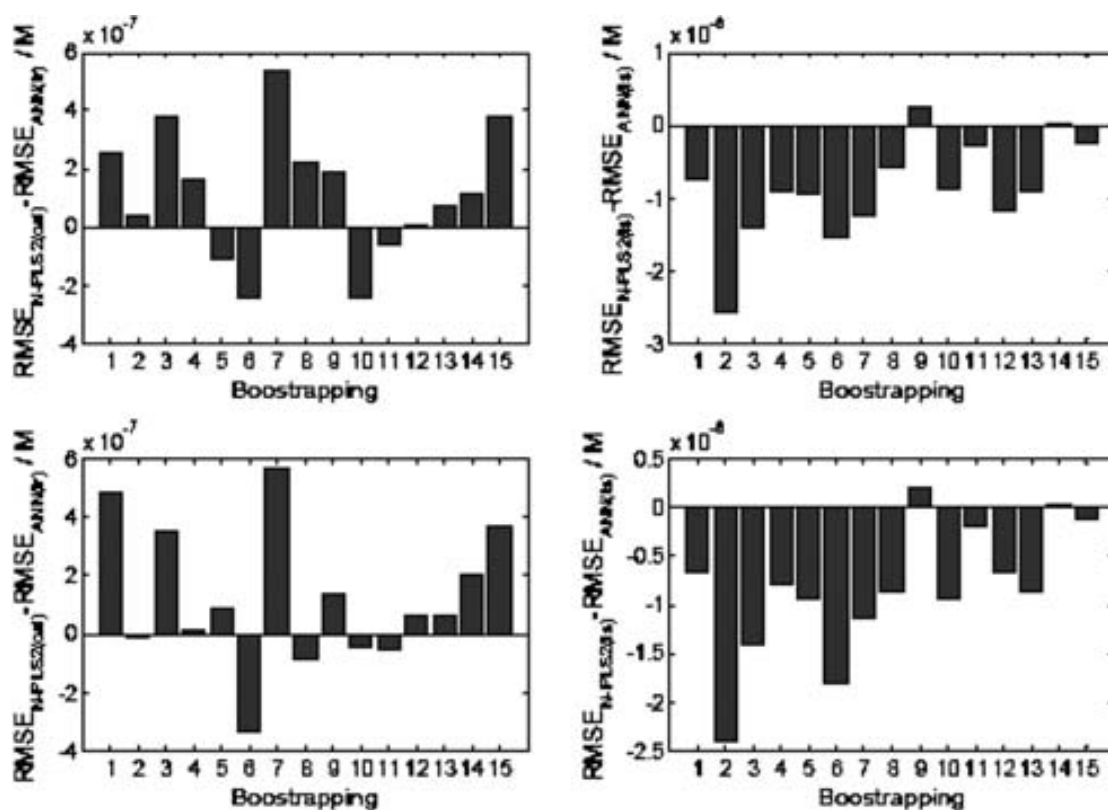


Fig. 6. Mean RMSE differences between N-PLS2 and DWT-ANN results for calibration (plots at right) and testing (plots at left). Labels *cal*, *tr*, and *ts* in subscripts on 'y' axis stand for calibration, training and testing, respectively. Plots in the upper row correspond to DWT-ANNs with logarithmic sigmoidal transfer functions, and plots in the lower row to DWT-ANNs with tangential sigmoidal transfer function.

bered subset of coefficients and concentrations was used to determine the parameters of the model while even

numbered subset was used for testing its generalization capability. Training was accomplished with the Bayesian

regulation algorithm. In order to facilitate convergence, input and output data in training subset were firstly normalized in the interval $[-1, 1]$, using max and min values obtained from the training subset.

The sum of squared errors (*SSE*) from the differences between the expected and calculated values was used for tracking convergence during training. Error goal was set to 2.1, equivalent to the total RMSE obtained by N-PLS2. *SSE* was later converted into RMSE for comparison purposes.

Before bootstrapping, several training runs were executed in order to evaluate the performance of the network for an increasing number of hidden neurons, from 3 up to 20. All networks reached the error goal for training, and no decreasing trends or significant differences were found regarding testing error for an increasing number of hidden neurons or use of non-linear transfer function. For better comparison of procedures, a bootstrapping methodology was also followed. Networks with 10 hidden neurons were used as equivalence to the 10 terms model obtained with N-PLS2. Wavelet coefficients and triads of concentrations were split into training and testing subsets according to the random indexes previously saved from the N-PLS2 bootstrapping process. As previously, training and testing subsets were normalized. Error goal for training was set to 2.1 as before and was met in all cases. Average RMSE values obtained in training for ANNs with logarithmic sigmoidal transfer function in their hidden layer were $2.42 \pm 0.15 \mu\text{M}$, $61.0 \pm 6.7 \text{ nM}$ and $0.51 \pm 0.08 \mu\text{M}$ for Cd^{2+} , Cu^{2+} and Pb^{2+} , respectively. Generalization capability of the same networks yielded average RMSEs values of test subset of $5.76 \pm 0.43 \mu\text{M}$, $162 \pm 25 \text{ nM}$ and $0.91 \pm 0.097 \mu\text{M}$ for Cu^{2+} , Cd^{2+} and Pb^{2+} , respectively. The corresponding RMSE training values for networks with tangential sigmoidal transfer function were $2.41 \pm 0.16 \mu\text{M}$, $53.4 \pm 7.3 \text{ nM}$ and $0.47 \pm 0.01 \mu\text{M}$ for Cd^{2+} , Cu^{2+} and Pb^{2+} , respectively; and $5.69 \pm 0.42 \mu\text{M}$, $162 \pm 23 \text{ nM}$ and $0.88 \pm 0.094 \mu\text{M}$ for testing with Cd^{2+} , Cu^{2+} and Pb^{2+} , respectively. All previously indicated uncertainties correspond to the 95% confidence interval. Concentration values obtained with the trained ANNs were compared against the corresponding expected values by linear regression analysis. The average values for slopes, intercepts and correlation coefficients obtained for compari-

son graphs – for modelling and testing cases – are summarized in Table 3.

The bar graph comparing errors between both procedures, $\overline{RMSE}_{N-PLS2} - \overline{RMSE}_{DWT-ANN}$ and obtained during the bootstrapping process is shown in Figure 6. From there it is clear that although the *DWT-ANN* yielded better calibration results for both sigmoidal transfer functions, the N-PLS2 models outstood those built with ANNs in terms of generalization capability.

5 Conclusions

Presented work has shown a potentiometric electronic tongue to simultaneously quantify ternary mixtures of heavy metal ions in aqueous solution. The system was developed by using a three ISEs array and, for the first time in the literature, with multiway N-PLS2 data treatment. Along with the advantage of using the kinetic information from the response of the sensors to develop the multivariate calibration models, the proposal also takes profit of the three-way characteristic of the signals, created when synchronous measurements were performed on the three electrodes for a given experiment.

When the calibration model was built employing N-PLS2, no reduction of original recordings was necessary; only mean centring on one of the modes was performed. This is an important reduction of computing effort if compared to other methodologies involving feature extraction prior to modelling.

Goals accomplished using N-PLS2 with the e-tongue were: not to dismiss the multiway nature of the data, to create a more robust response model, to avoid discontinuities created when data is unfolded into a matrix and to improve the performance of existing e-tongue systems by exploiting sensors' dynamic resolution. Results obtained with this tool were validated by a bootstrapping procedure and compared against a established methodology based on *DWT-ANN* modelling. Performance of the proposed model overcame those obtained by ANN approaches. The future development of this tool on a dedicated electronic system is feasible from a computational point of view given that N-PLS models, intrinsically linear in nature, are much simpler than those based on ANN.

Table 3. Mean values for slope (*m*), intercept (*b*) and correlation coefficient (*R*) of the comparison graphs of obtained /vs./ expected concentrations, obtained from the bootstrapping validation process for the *DWT-ANN* modelling. Labels 'log' and 'tan' stand for logarithmic and tangential sigmoidal transfer functions, respectively. Uncertainty intervals calculated at 95% confidence level.

Analyte	Modelling	Testing		
		<i>m</i>	<i>b</i>	<i>R</i>
Cd^{2+}	log	0.74 ± 0.205	$2.85 \times 10^{-6} \pm 2.8 \times 10^{-6}$	0.95 ± 0.007
	tan	0.76 ± 0.020	$3.01 \times 10^{-6} \pm 3.4 \times 10^{-7}$	0.95 ± 0.006
Cu^{2+}	log	0.90 ± 0.201	$5.22 \times 10^{-8} \pm 1.4 \times 10^{-8}$	0.99 ± 0.001
	tan	0.91 ± 0.023	$4.74 \times 10^{-8} \pm 1.3 \times 10^{-8}$	0.99 ± 0.001
Pb^{2+}	log	0.92 ± 0.019	$3.14 \times 10^{-7} \pm 1.3 \times 10^{-7}$	0.99 ± 0.001
	tan	0.92 ± 0.026	$3.06 \times 10^{-7} \pm 1.1 \times 10^{-7}$	0.99 ± 0.001

Acknowledgements

This work was supported by the *Spanish Ministry of Science and Innovation*, through Project TEC2007-68012-c03-02/MIC and by *CONACyT* (México) through PhD scholarship 144044 for R. Cartas.

References

- [1] B. K. Lavine, *Anal. Chem.* **2000**, 72, 91R.
- [2] Y. Vlasov, A. Legin, *Fresenius J. Anal. Chem.* **1998**, 361, 255.
- [3] A. V. Legin, A. M. Rudnitskaya, K. A. Legin, A. V. Ipatov, Yu. G. Vlasov, *Russ. J. Appl. Chem.* **2005**, 78, 89.
- [4] M. del Valle, *Electroanalysis* **2010**, 22, 1539.
- [5] J. H. Kalivas, *Anal. Lett.* **2005**, 38, 2259.
- [6] E. Richards, C. Bessant, S. Saini, *Electroanalysis* **2002**, 14, 1533.
- [7] V. Pravdová, M. Pravda, G. G. Guilbault, *Anal. Lett.* **2002**, 35, 2389.
- [8] M. Esteban, C. Ariño, J. M. Díaz-Cruz, *Crit. Rev. Anal. Chem.* **2006**, 36, 295.
- [9] N. M. Faber, J. Ferré, R. Boqué, J. H. Kalivas, *Chemom. Intell. Lab. Syst.* **2002**, 63, 107.
- [10] R. Bro, *J. Chemom.* **1996**, 10, 47.
- [11] R. Bro, *Ph.D. Thesis*, Royal Veterinary and Agricultural University, Copenhagen **1998**.
- [12] A. Smilde, R. Bro, P. Geladi, *Multiway Analysis with Applications in the Chemical Sciences*, Wiley, Chichester, West Sussex, UK **2004**.
- [13] R. Bro, *Crit. Rev. Anal. Chem.* **2006**, 36, 279.
- [14] M. M. Sena, R. J. Poppi, *J. Pharm. Biomed. Anal.* **2004**, 34, 27.
- [15] M. Goodarzi, M. P. Freitas, *Chemom. Intell. Lab. Syst.* **2009**, 96, 59.
- [16] M. P. Freitas, J. A. Martins, *Talanta* **2005**, 67, 182.
- [17] K. P. Singha, N. Basant, A. Malik, V. K. Singh, D. Mohan, *Anal. Chim. Acta* **2008**, 630, 10.
- [18] A. Valero-Navarro, P. C. Damiani, J. F. Fernández-Sánchez, A. Segura-Carretero, A. Fernández-Gutiérrez, *Talanta* **2009**, 78, 57.
- [19] F. Guimet, J. Ferré, R. Boqué, M. Vidal, J. Garcia, *J. Agric. Food Chem.* **2005**, 53, 9319.
- [20] O. Divya, A. K. Mishra, *Anal. Chim. Acta* **2007**, 592, 82.
- [21] J. M. Amigo, J. Coello, S. MasPOCH, *Anal. Bioanal. Chem.* **2005**, 382, 1380.
- [22] A. Jiménez Girón, I. Durán-Merás, A. Espinosa-Mansilla, A. Muñoz de la Peña, F. Cañada Cañada, A. C. Olivieri, *Anal. Chim. Acta* **2008**, 622, 94.
- [23] A. Muñoz de la Peña, I. Durán Merás, A. Jiménez Girón, *Anal. Bioanal. Chem.* **2006**, 385, 1289.
- [24] A. Espinosa-Mansilla, A. Muñoz de la Peña, D. González Gómez, F. Salinas, *Anal. Chim. Acta* **2005**, 531, 257.
- [25] M. Pozzobon Pedroso, L. A. Fonseca de Godoy, E. Correa Ferreira, R. J. Poppi, F. Augusto, *J. Chromatogr. A* **2008**, 1201, 176.
- [26] C. Durante, M. Cocchi, M. Grandi, A. Marchetti, R. Bro, *Chemom. Intell. Lab. Syst.* **2006**, 83, 54.
- [27] A. Jaworski, H. Wikiel, K. Wikiel, *Electroanalysis* **2009**, 21, 580.
- [28] T. Galeano-Díaz, A. Guiberteau-Cabanillas, A. Espinosa-Mansilla, M. D. López-Soto, *Anal. Chim. Acta* **2008**, 618, 131.
- [29] L. H. Chiang, R. Leardi, R. J. Pell, M. B. Seasholtz, *Chemom. Intell. Lab. Syst.* **2006**, 81, 109.
- [30] E. Chow, D. Ebrahimi, J. J. Gooding, D. B. Hibbert, *Analyst* **2006**, 131, 1051.
- [31] R. Cartas, A. Mimendia, A. Legin, M. del Valle, *Talanta* **2010**, 80, 1428.
- [32] R. G. Brereton, *Analyst* **2000**, 125, 2125.
- [33] V. Lengard, M. Kermit, *Food Qual. Prefer.* **2006**, 17, 234.
- [34] H. Abdi, in *Encyclopedia of Social Sciences Research Methods* (Eds: M. S. Lewis-Beck, A. Bryman, T. Futing), Sage Publications, Thousand Oaks **2003**, pp. 978–982.
- [35] P. Geladi, B. R. Kowalski, *Anal. Chim. Acta* **1986**, 185, 1.
- [36] C. A. Andersson, R. Bro, *Chemom. Intell. Lab. Syst.* **2000**, 52, 1.
- [37] R. Bro, A. K. Smilde, S. de Jong, *Chemom. Intell. Lab. Syst.* **2001**, 58, 3.
- [38] A. K. Smilde, *J. Chemom.* **1997**, 11, 367.
- [39] S. De Jong, *J. Chemom.* **1998**, 12, 77.
- [40] D. Calvo, A. Durán, M. del Valle, *Anal. Chim. Acta* **2007**, 600, 97.
- [41] A. Durán, M. Cortina, L. Velasco, J. A. Rodríguez, S. Alegret, M. del Valle, *Sensors* **2006**, 6, 19.
- [42] A. Mimendia, A. Legin, A. Merkoçi, M. del Valle, *Sens. Actuators B* **2010**, 146, 420.
- [43] M. Padilla, I. Montoliu, A. Pardo, A. Perera, S. Marco, *Sens. Actuators B* **2005**, 116, 145.
- [44] S. P. Gurden, J. A. Westerhuis, R. Bro, A. K. Smilde, *Chemom. Intell. Lab. Syst.* **2001**, 59, 121.
- [45] L. Moreno-Barón, R. Cartas, A. Merkoçi, S. Alegret, J. M. Gutiérrez, L. Leija, P. R. Hernández, R. Muñoz, M. del Valle, *Anal. Lett.* **2005**, 38, 2189.
- [46] L. Moreno-Barón, R. Cartas, A. Merkoçi, S. Alegret, M. del Valle, L. Leija, P. R. Hernández, R. Muñoz, *Sens. Actuators B* **2006**, 113, 487.

Apéndices

Apéndice 1

Algoritmo para la construcción y entrenamiento de una Red Neuronal Wavelet con funciones multidimensionales basadas en la norma Euclidiana. El entrenamiento se lleva a cabo con la técnica del gradiente conjugado. Las ecuaciones asociadas con este algoritmo son las presentadas en los apartados 3.2 y 3.2.1.

Inicio

$$p \leftarrow 0$$

$$\Theta^{(p)} \leftarrow \text{Inicialización} \{a_0, \mathbf{d}, \mathbf{w}, \mathbf{s}_j, \mathbf{m}_j\}$$

$Error_{esperado} \leftarrow$ Definido por el usuario

goto Evaluación_de_la_red

if $Error_{obtenido} > Error_{esperado}$ **then**

{

$$D^{(p)} \leftarrow -\nabla E_r(\Theta^{(p)})$$

$$\alpha^{(p)} \leftarrow -\frac{E'(\Theta^{(p)})^T \cdot D^{(p)}}{(D^{(p)})^T \cdot \text{diag}(E''(\Theta^{(p)})) \cdot D^{(p)}}$$

$$\Theta^{(p+1)} \leftarrow \Theta^{(p)} + \alpha^{(p)} D^{(p)}$$

}

else

Fin del algoritmo

end if

goto Evaluación_de_la_red

while $Error_{obtenido} > Error_{esperado}$

{

$$\beta^{(p+1)} \leftarrow \frac{(\nabla E_r(\Theta^{(p+1)}) - \nabla E_r(\Theta^{(p)}))^T \cdot \nabla E_r(\Theta^{(p+1)})}{\nabla E_r(\Theta^{(p)})^T \cdot \nabla E_r(\Theta^{(p)})}$$

$$D^{(p+1)} \leftarrow -\nabla E_r(\Theta^{(p+1)}) + \beta^{(p+1)} D^{(p)}$$

$$\alpha^{(p+1)} \leftarrow -\frac{E'(\Theta^{(p+1)})^T \cdot D^{(p+1)}}{(D^{(p+1)})^T \cdot \text{diag}(E''(\Theta^{(p+1)})) \cdot D^{(p+1)}}$$

$$p \leftarrow p + 1$$

$$\Theta^{(p+1)} \leftarrow \Theta^{(p)} + \alpha^{(p)} D^{(p)}$$

goto Evaluación_de_la_red

}

end while

Label *Evaluación_de_la_red*

```

{
  for n=1:N tensores de entrada
  {
    for j=1:Nj neuronas wavelet
    {
      
$$z_j^{(n)} \leftarrow \frac{\|\mathbf{x}^{(n)} - \mathbf{m}_j\|_2}{s_j}$$

      
$$\Psi_j(\mathbf{x}^{(n)}) \leftarrow \frac{1}{\frac{N_k}{s^2}} \psi(z_j^{(n)})$$

    }
    end for
    
$$\sigma^{(n)} \leftarrow \sum_k d_k x_k^{(n)}$$

  }
  end for
  
$$\hat{y}^{(n)} \leftarrow \sum_j w_j \Psi_j(\mathbf{x}^{(n)}) + \sigma^{(n)} + a_0$$

  
$$Error_{obtenido} \leftarrow \frac{1}{2} \sum_n (y^{(n)} - \hat{y}^{(n)})^2$$

}
Return
Fin del algoritmo

```

Apéndice 2

Algoritmo para la construcción y entrenamiento de una Red Neuronal Wavelet con funciones multidimensionales basadas en el producto tensorial de funciones wavelet monodimensionales. El entrenamiento se lleva a cabo con la técnica del gradiente conjugado. Las ecuaciones asociadas con la estructura de la red son las presentadas en el apartado 3.2 y las asociadas con el entrenamiento fueron presentadas en el apartado 3.2.2.

Inicio

$$p \leftarrow 0$$

$$\Theta^{(p)} \leftarrow \text{Inicialización} \{a_0, \mathbf{d}, \mathbf{w}, s_j, \mathbf{m}_j\}$$

$Error_{esperado} \leftarrow$ Definido por el usuario

GoTo Evaluación_de_la_red

if $Error_{obtenido} > Error_{esperado}$ **then**

{

$$D^{(p)} \leftarrow -\nabla E_r(\Theta^{(p)})$$

$$\alpha^{(p)} \leftarrow -\frac{E'(\Theta^{(p)})^T \cdot D^{(p)}}{(D^{(p)})^T \cdot \text{diag}(E''(\Theta^{(p)})) \cdot D^{(p)}}$$

$$\Theta^{(p+1)} \leftarrow \Theta^{(p)} + \alpha^{(p)} D^{(p)}$$

}

else

Fin del algoritmo

end if

GoTo Evaluación_de_la_red

while $Error_{obtenido} > Error_{esperado}$

{

$$\beta^{(p+1)} \leftarrow \frac{(\nabla E_r(\Theta^{(p+1)}) - \nabla E_r(\Theta^{(p)}))^T \cdot \nabla E_r(\Theta^{(p+1)})}{\nabla E_r(\Theta^{(p)})^T \cdot \nabla E_r(\Theta^{(p)})}$$

$$D^{(p+1)} \leftarrow -\nabla E_r(\Theta^{(p+1)}) + \beta^{(p+1)} D^{(p)}$$

$$\alpha^{(p+1)} \leftarrow -\frac{E'(\Theta^{(p+1)})^T \cdot D^{(p+1)}}{(D^{(p+1)})^T \cdot \text{diag}(E''(\Theta^{(p+1)})) \cdot D^{(p+1)}}$$

$$p \leftarrow p + 1$$

$$\Theta^{(p+1)} \leftarrow \Theta^{(p)} + \alpha^{(p)} D^{(p)}$$

```

    GoTo Evaluación_de_la_red
}
end while
Label Evaluación_de_la_red
{
    for n=1:N tensores de entrada
    {
        for j=1:Nj neuronas wavelet
        {
            
$$\mathbf{z}_j^{(n)} \leftarrow (\mathbf{x}^{(n)} - \mathbf{m}_j) / \mathbf{s}_j \ ; \ \mathbf{x}^{(n)} = \{x_{k,j}^{(n)}\}_{k=1}^{N_k}, \mathbf{m}_j = \{m_{k,j}\}_{k=1}^{N_k}, \mathbf{s}_j = \{s_{k,j}\}_{k=1}^{N_k}$$

            
$$\Psi_j(\mathbf{x}^{(n)}) \leftarrow \prod_k \psi(\mathbf{z}_j^{(n)}) \ , \ \mathbf{z}_j^{(n)} = \{z_{k,j}^{(n)}\}_{k=1}^{N_k}$$

        }
        end for
        
$$\sigma^{(n)} \leftarrow \sum_k d_k x_k^{(n)}$$

    }
    end for
    
$$\hat{y}^{(n)} \leftarrow \sum_j w_j \Psi_j(\mathbf{x}^{(n)}) + \sigma^{(n)} + a_0$$

    
$$Error_{obtenido} \leftarrow \frac{1}{2} \sum_n (y^{(n)} - \hat{y}^{(n)})^2$$

}
Return
Fin del algoritmo

```

Apéndice 3

Algoritmo para la construcción de un modelo de regresión del tipo MARS. El proceso consiste de dos pasos que pueden implementarse en ficheros separados. Las ecuaciones asociadas con la estructura de la red son las presentadas en el Capítulo 6.

Inicio

Paso 1

$$B_1(\mathbf{x}) \leftarrow 1$$

$$S \leftarrow 2$$

while $S \leq S_{\max}$

$$LoF_{(FW)} \leftarrow \infty$$

for $j = 1: S - 1$

for $k = 1: N_k$ **variables de entrada**

Elegir $\mathbf{x}_k \notin \left\{ \mathbf{x}_{k(i,j)} \right\} \mid 1 \leq i \leq \text{Máximo número de interacciones}$

for $n = 1: N$ **puntos de salida**

Elegir $t \in \left\{ \mathbf{x}_k^{(n)} \right\} \mid B_j(\mathbf{x}_k) > 0$

$$\hat{y} \leftarrow \sum_{s=1}^{S-1} a_s B_s(\mathbf{x}) + a_S B_j(\mathbf{x}) \left[+(\mathbf{x}_k - t_k^n) \right]_+ + a_{S+1} B_j(\mathbf{x}) \left[-(\mathbf{x}_k - t_k^n) \right]_+$$

Ajustar los valores de $\left\{ a_j \right\}_{j=1}^S$ para minimizar $y - \hat{y}$

$$LoF_{(referencia)} \leftarrow GCV(\hat{y})$$

if $LoF_{(referencia)} < LoF_{(FW)}$ **then**

Crear y actualizar los siguientes valores

$$LoF_{(FW)} \leftarrow LoF_{(referencia)}$$

$$j_{(FW)} \leftarrow j$$

$$k_{(FW)} \leftarrow k$$

$$t_{(FW)} \leftarrow t_k^{(n)}$$

end if

$$B_S(\mathbf{x}) \leftarrow B_{j_{(FW)}}(\mathbf{x}) \left[+(\mathbf{x}_{k_{(FW)}} - t_{(FW)}) \right]_+$$

$$B_{S+1}(\mathbf{x}) \leftarrow B_{j_{(FW)}}(\mathbf{x}) \left[-(\mathbf{x}_{k_{(FW)}} - t_{(FW)}) \right]_+$$

$$S \leftarrow S + 2$$

```

        end for
    end for
end while
Paso 2
 $J_{(BW)} \leftarrow \{1, 2, \dots, S_{\max}\}$ 
 $K_{(BW)} \leftarrow J_{(BW)}$ 
Ajustar los valores  $\{a_j\} \mid j \in J_{(BW)}$  para minimizar  $y - \hat{y}$ 

 $LoF_{(referencia)} \leftarrow GCV \left( \sum_{j \in J_{(BW)}} a_j B_j(\mathbf{x}) \right)$ 
for  $M = M_{\max} : 2$ 
     $\zeta \leftarrow \infty$ 
     $L \leftarrow K_{(BW)}$ 
    for  $m = 2 : M$ 
        Eliminar un elemento de  $K_{(BW)}$  a la vez haciendo  $K \leftarrow L - L\{m\}$ 
        Ajustar los valores de  $\{a_j\} \mid j \in K$  para minimizar  $y - \hat{y}$ 

         $LoF_{(BW)} \leftarrow GCV \left( \sum_{j \in K} a_j B_j(\mathbf{x}) \right)$ 

        if  $LoF_{(BW)} < \zeta$ 
             $\zeta \leftarrow LoF_{(BW)}$ 
             $K_{(BW)} \leftarrow K$ 
        end if
        if  $LoF_{(BW)} < LoF_{(referencia)}$ 
             $LoF_{(referencia)} \leftarrow LoF_{(BW)}$ 
             $J_{(BW)} \leftarrow K$ 
        end if
    end for
end for
Fin del Algoritmo

```

Referencias

1. Yu. Vlasov, A. Legin, A. Rudnitskaya, C. Di Natale, A. D'amico, Nonspecific Sensor Arrays ("Electronic tongue") for chemical analysis of liquids, *Pure Appl. Chem.*, Vol. 77, No. 11, pp. 1965–1983, 2005.
2. Holmberg, M., Eriksson, M., Krantz-Rülcker, C., Artursson, T., Winqvist, F., Lloyd-Spetz, A., Lundström, I.: Second workshop of the second network on artificial olfactory sensing (NOSE II). *Sens. Actuators B* 101, 213–223 (2004).
3. Yu.G. Vlasov a.), A.V. Legin a, A.M. Rudnitskaya a, A. D'Amico b, C. Di Natale, «Electronic tongue» — new analytical tool for liquid analysis on the basis of non-specific sensors and methods of pattern recognition, *Sensors and Actuators B* 65 _2000. 235–236
4. Gutiérrez, A., Céspedes, F., Cartas, R., Alegret, S., del Valle, M., Gutierrez, J.M., Muñoz, R., *Multivariate calibration model from overlapping voltammetric signals employing wavelet neural networks*. *Chemometr. Intell. Lab. Syst.* 83, 169–179 (2006)
5. M. del Valle, *Electronic tongues employing electrochemical sensors*, *Electroanalysis*, 14, 2010, 1539-1555.
6. Matthias Otto, J. D. R. Thomas, *Model studies on multiple channel analysis of free magnesium, calcium, sodium, and potassium at physiological concentration levels with ion-selective electrodes*, *Anal. Chem.* 1985, 57, 2647-2651.
7. Winqvist, F., Holmin, S., Krantz-Rülcker, C., Wide, P., Lundström, I.: A hybrid electronic tongue. *Anal. Chim. Acta* 406, 147–157 (2000).
8. Patrycja Ciosek, Wojciech Wróblewski *Sensor arrays for liquid sensing – electronic tongue Systems*, *Analyst*, 2007, 132, 963–978
9. E. Richards, C. Bessant, S. Saini, *Multivariate Data Analysis in Electroanalytical Chemistry*, *Electroanalysis* 14 (2002) 1533-1542.
10. V. Pravdová, M. Pravda, G. G. Guilbault, *Role of chemometrics for electrochemical sensors*, *Anal Letters* 35 (2002) 2389-2419.
11. G.M.Escandar, N.M. Faber, H.C. Goicoechea, A. Muñoz de la Peña, A.C. Olivieri, R.J. Poppi, *Second and third order multivariate calibration: data, algorithms and applications*, *Trends in Analytical Chemistry*, 26 (2007) 752-765.
12. R. Bro, *Multivariate calibration, What is in chemometrics for the analytical chemist?*, *Analytica Chimica Acta*, 2003. 500(1-2): 185–194.
13. B.K. Lavine, *Chemometrics*, *Anal. Chem.* 72 (2000) 91R-97R.
14. L. Mutihac, R. Mutihac, *Mining in Chemometrics*, *Analytica Chimica Acta* 612 (2008) 1-18.
15. E. Ribes-Gómez, S. McLoone, G. Irwin, *A taxonomy for wavelet neural network applied to nonlinear modelling*, *International Journal of System Science*, Vol. 39, No. 6, 2008, 607-627
16. L. Francis, *Neural Network Desmitified. Casualty Actuarial Society Forum*, Winter 2001, 253-320.
17. B. Warner, M. Misra, *Understanding Neural Networks as Statistical Tools. American Statistician*, November 1996, 284-293.

18. C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford New York, 1998
19. J. R. Stetter, W. R. Penrose, S. Yao, *Sensors, Chemical Sensors, Electrochemical Sensors, and ECS*, *J Electrochem Soc*, 150 (2003) S11-S16.
20. J. Wang, *Analytical Electrochemistry*, Third Edition, Wiley-VHC (2006).
21. J. Wang, *Electrochemical sensors for environmental monitoring: a review of recent technology*, Technical Report, National Exposure Research Laboratory, Office of Research and Development, U.S. Environmental Protection Agency
22. J. Janata, *Principles of Chemical Sensors*, Second Edition, Springer (2009).
23. C. C. Liu, "Electrochemical Sensors." *The Biomedical Engineering Handbook: Second Edition*. Ed. Joseph D. Bronzino. CRC Press LLC (2000)
24. C. G. Zoski, Editor, *Handbook of electrochemistry*, Elsevier, Amsterdam Holanda (2007).
25. F. Settle, Editor, *Handbook of instrumental techniques for analytical chemistry*, Prentice Hall PTR, Upper Saddle River, NJ (1997).
26. C. M. A. Brett, A. M. Oliveira Brett, *Electrochemistry principles, methods and applications*, Oxford University Press, NY (1994).
27. C. Krantz-Rülcker, M. Stenberg, F. Winqvist, I. Lundström, *Electronic tongues for environmental monitoring based on sensor arrays and pattern recognition: a review* *Anal. Chim. Acta* 426 (2001) 217–226
28. X. Zhang, H. Ju, J. Wang, Editores, *Electrochemical Sensors, Biosensors and their Biomedical Applications*, Academic Press, Elsevier (2008).
29. Verònica Gómez Cortés, *Sequential Injection Analysis Using Second-Order Calibration for the Development of Analytical Methods*, PhD Thesis, Universitat Rovira I Virgili, ISBN: 978-84-691-0990-8/D.L: T.2293 (2007)
30. M. Miró, V. Cerdà, J.M. Estela, *Multisyringe flow injection analysis: characterization and applications*. *Trends Anal. Chem.* 21 (2002) 199-210.
31. A. Gutés, F. Céspedes, M. del Valle, *Electronic tongues in flow analysis*, *Anal Chim Acta* 600 (2007) 90-96.
32. B. Karlberg, G. E. Pacey, *Flow Injection Analysis. A practical guide. Techniques and Instrumentation in Analytical Chemistry – Volume 10*. Elsevier Science Publishers B.V. (1989)
33. J. Ruzicka, E.H. Hansen, *Flow injection analysis. Part I. A new concept of fast continuous flow analysis*. *Anal. Chim. Acta* 78 (1975) 145
34. J. Ruzicka, E.H. Hansen, *Flow injection analysis, principles, applications and trends*, *Anal. Chim. Acta* 114 (1980) 19.
35. J. Ruzicka, G. D. Marshall, G. D. Christian, *Variable flow rates and a sinusoidal flow pump for flow injection analysis*, *Anal. Chem.* 62 (1990) 1861.
36. Alberto Pasamontes Fúnez, *Multivariate curve resolution applied to sequential injection data. Analysis of amoxicillin and clavulanic acid*, PhD Thesis, Universitat Rovira I Virgili.
37. E.H. Hansen, J. Wang, *The three generations of flow injection analysis*, *Anal. Letters* 37 (2004) 345-360.
38. J. Ruzicka, E.H. Hansen, *Flow injection analysis, principles, applications and trends*, *Anal. Chim. Acta* 114 (1980) 19.
39. V. Cerdà, A. Cerdà, A. Cladera, M.T. Oms, F. Mas, E. Gómez, F. Bauzá, M. Miró, R. Forteza, J.M Estela, *Monitoring of environmental parameters by sequential injection analysis*, *Trends Anal. Chem.* 20 (2001) 407-418.
40. K. Mervartová, M. Polásek, J. Martínez Calatayud, *Recent applications of flow-injection and sequential-injection analysis techniques to chemiluminescence determination of pharmaceuticals*, *J. Pharmaceut. Biomed. Anal.* 45 (2007) 367–381

41. P. D. Tzanavaras, D. G. Themelis, *Review of recent applications of flow injection spectrophotometry to pharmaceutical analysis*, *Anal. Chim. Acta* 588 (2007) 1–9
42. W. Xu, R. C. Sandford, P. J. Worsfold, A. Carlton, G. Hanrahan, *Flow Injection Techniques in Aquatic Environmental Analysis: Recent Applications and Technological Advances*, *Crit. Rev. Anal. Chem.* 35 (2005) 237–246
43. M. I. Evgenév, S. Yu. Garmonov, L. Sh. Shakirova, *Flow-Injection Analysis of Pharmaceuticals*, *J. Anal. Chem.* 56 (2001) 313–323.
44. R. Pérez-Olmos, J.C. Soto, N. Zárate, A.N. Araújo, M.C.B.S.M. Montenegro, *Sequential injection analysis using electrochemical detection: A review*, *Anal. Chim. Acta* 554 (2005) 1–16
45. W. Siangproh, W. Leesutthipornchai, W. Dungchai, O. Chailapakul, *Electrochemical Detection for Flow-based System: A Review*, *J. Flow Injection Anal.* 26 (2009) 5–25
46. R. B.R. Mesquita, A. O. S. S. Rangel, *A review on sequential injection methods for water analysis*, *Anal. Chim. Acta* 648 (2009) 7–22
47. R. Boqué, J. Ferré, *Using Second-Order Data in Chromatographic Analysis*, *LC-GC Chromatographyonline*, 17 (2004) 402–407.
48. A. Gutés, F. Céspedes, S. Alegret, M. del Valle, *Sequential injection system with higher dimensional electrochemical sensor signals: Part 1. Voltammetric e-tongue for the determination of oxidizable compounds*, *Talanta* 66 (2005) 1187–1196.
49. A. Gutés, F. Céspedes, S. Alegret, M. del Valle, *Simultaneous determination of phenolic compounds by means of an automated voltammetric “electronic tongue”*, *Anal. Bioanal. Chem.* 382 (2005) 471–476.
50. E. Sánchez, B. R. Kowalski, *Tensorial calibration: I. First-order calibration*, *J. Chemometrics* 2 (1988) 247–263.
51. K. S. Booksh, B.R. Kowalski, *Theory of Analytical Chemistry*, *Anal. Chem.* 66 (1994) 782A-791A.
52. M. J. Rodríguez Cuesta, *Limit of detection for second order calibration methods*, PhD Thesis, Universidad Rovira I Virgili (2006)
53. V. Pravdová, M. Pravda, G.G. Guilbault, *Anal. Letters* 35 (2002) 2389-2419.
54. A. V. Legin, A. M. Rudnitskaya, K. A. Legin, A. V. Ipatov, Yu. G. Vlasov, *Russ. J. Appl. Chem.* 78 (2005) 89-95.
55. R. Cartas, A. Mimendia, A. Legin, M. del Valle, *Multi-way processing of data generated with a potentiometric electronic tongue in a SIA system*, *Electroanalysis* 23 (2011) 953-961.
56. F. Winquist, P. Wide, I. Lundström, *An electronic tongue based on voltammetry*, *Anal. Chim. Acta*, 357 (1997) 21-31.
57. R. Bro, *Review on Multiway Analysis in Chemistry 2000-2005*, *Crit. Rev. Anal. Chem.* 30 (2006) 279-293
58. T. Poggio, F. Girosi, *Networks for approximation and learning*, *IEEE Proc* 78 (1990) 1481-1497.
59. K.S. Narendra, K. Parathasarathy, *Identification and control of dynamic systems using neural networks*, *IEEE Trans. Neural Networks*, 1 (1990) 4-27
60. J.V. Beck, K.J. Arnold, *Parameter estimation in engineering and science*, John Wiley & Sons, New York
61. A. Gutiérrez, S. Marco, Editores, *Multivariate Calibration Model for a Voltammetric Electronic Tongue Based on a Multiple Output Wavelet Neural Network*, in *Biologically Inspired Signal Processing for Chemical Sensing*, Springer (2009)

62. S.H. Huang, H-C Zhang, *Artificial Neural Networks in Manufacturing: Concepts, Applications, and Perspectives*, IEEE Trans. Compon. Packag. Manuf. Technol. – Part A, 17 (1994) 212-228
63. M.T. Hagan, H.B. Demuth, M. Beale, *Neural Network Design*, PWS Publishing Company (1996)
64. T.L. Fine, *Feedforward Neural Network Methodology*, Springer (1999).
65. L. Fausett, *Fundamentals of Neural Networks. Architectures, algorithms and applications*, Pearson Education (2006).
66. I. Aleksander, H. Morton, *An introduction to neural computing*, Chapman & Hall, 1992.
67. B. Kröse, P. van der Smagt, *An introduction to neural networks*, The University of Amsterdam, (1996).
68. M.A. Arbib, editor, *Handbook of brain theory and neural network*, The MIT Press (2003).
69. S. Haykin, *Neural Networks. A comprehensive foundation*, Prentice Hall International (1999)
70. C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press (1998)
71. R. Rojas, *Neural Networks. A systematic Introduction*, Springer-Verlag (1996).
72. C.M. Bishop, *Pattern recognition and machine learning*, Springer-Verlag (2006).
73. J. Zupan, J. Gasteiger, *Neural networks for chemists. An introduction*, VCH Verlagsgesellschaft, (1993).
74. J.A. Freeman, D.M. Skapura, *Neural Network. Algorithms, Applications and Programming Techniques*, Addison-Wesley Publishing Company (1991).
75. K. Gurney, *An introduction to neural Networks*, UCL Press (1999).
76. A.J. Skinner, J.Q. Broughton, *Neural networks in computational material science: training algorithms*, Modelling Simul. Mater. Sci. Eng., 3 (1995) 371-390
77. C. Charalambous, *Conjugate gradient algorithm for efficient training of artificial neural networks*, IEEE Proceedings G (Circuits, Devices and Systems) 3 (1992) 301-310
78. C.D. Doan, S.Y. Liong, *Generalization for Multilayer Neural Network: Bayesian Regularization or Early Stopping*, Proceedings of Asia Pacific Association of Hydrology and Water Resources 2nd Conference, Singapore, 2004.
79. P. Kumar, S.N. Merchant, U.B. Desai, *Improving performance in pulse radar detection using Bayesian regularization for neural network training*, Digital Signal Process., 14 (2004) 438-448.
80. MATLAB User Manual. The MathWorks Inc., 2000.
81. P.M. Kroonenberg, *Applied multiway data analysis*, John Wiley & Sons, Hoboken, New Jersey, 2008.
82. A. de Juan, R. Tauler, *Comparison of three-way resolution methods for non-trilinear chemical data sets*, J. Chemometrics, 15 (2001) 749-772.
83. A. Smilde, R. Bro, P. Geladi, *Multiway data analysis with applications in the Chemicals science*, John Wiley & Sons, West Sussex, England, 2004.
84. H.L. Wu, R.Q. Yu, K. Oguma, *Trilinear Component Analysis in Modern Analytical Chemistry*, Anal. Sci. 17 (2001) i481-i486.
85. R. Bro, *PARAFAC. Tutorial and applications*, Chemom. Intell. Lab. Sys. 38 (1997) 149-171.
86. R. A. Harshman, *Foundations of the PARAFAC procedure: Models and conditions for an explanatory multimodal factor analysis*, UCLA Working Papers in Phonetics, 16 (1970) 1-84. (University Microfilms, Ann Arbor, Michigan, No. 10,085).

87. C.A. Andersson, R. Bro, The N-way Toolbox for MATLAB, *Chemom. Intell. Lab. Sys.* 52 (2000) 1-4.
88. P.M. Kroonenberg, R.A. Harshman, T. Murakami, Analysing three-way profile data using the PARAFAC and Tucker3 models illustrated with views on parenting, *Applied Multivariate Research* 13 (2009) 5-41.
89. M. Esteban, C. Ariño, J.M. Díaz-Cruz, *Chemometrics in Electroanalytical Chemistry*, *Crit. Rev. Anal. Chem.* 36 (2006) 295-313
90. R. G. Brereton, Introduction to Multivariate Calibration in Analytical Chemistry, *Analyst*, 125 (2000) 2125-2154.
91. V. Lengard, M. Kermit, 3-Way and 3-block PLS regressions in consumer preference analysis, *Food Qual. Prefer.* 17 (2006) 234-242.
92. H. Abdi, in *Encyclopedia of Social Sciences Research Methods* (Eds: M. S. Lewis-Beck, A. Bryman, & T. Futing), Sage Publications, Inc., Thousand Oaks, (2003) 978-982.
93. P. Geladi, B. R. Kowalski, An Example of 2-Block Predictive Partial-Least Squares Regression with Simulated data, *Anal. Chim. Acta*, 1986, 185, 1.
94. R. Bro, Multiway calibration methods. Multilinear PLS, *J. Chemom.* 10 (1996) 47-61.
95. R. Bro, A. K. Smilde, S. de Jong, On the difference between low-rank and subspace approximation: improved model for multi-linear PLS regression, *Chemom. Intell. Lab. Syst.* 58 (2001) 3-13.
96. R. Bro, Multi-way Analysis in the Food Industry, Ph.D. Thesis, Royal Veterinary and Agricultural University, Copenhagen, DNK (1998).
97. A. K. Smilde, Comments On Multilinear PLS, *J. Chemom.* 11 (1997) 367-377.
98. S. De Jong, Regression Coefficients in Multilinear PLS, *J. Chemom.* 12 (1998) 77-81.
99. T.L. Fine, *Feedforward Neural Network Methodology*, Springer, 1999.
100. L. Fausett, *Fundamentals of Neural Networks. Architectures, algorithms and applications*, Pearson Education, 2006.
101. A.K. Deisingh, D.C. Stone, M. Thompson, Applications of electronic noses and tongues in food analysis, *Int. J. Food Sci. Technol.*, 39 (2004) 587-604.
102. A. Gutes, F. Céspedes, M. del Valle, Electronic tongues in flow analysis, *Anal. Chim. Acta*, 600 (2007) 90-96.
103. Y. Ni, S. Kokot, Does chemometrics enhance the performance of electroanalysis?, *Anal. Chim. Acta*, 626 (2008) 130-146.
104. L. Mutihac, R. Mutihac, Mining in chemometrics, *Anal. Chim. Acta*, 612 (2008) 1-18.
105. A. Riul Jr., C.A.R. Dantas, C.M. Miyazakic, O.N. Oliveira Jr., *Recent advances in electronic tongues*, *Analyst*, 135 (2010) 2453-2744.
106. A. Bratov, N. Abramova, A. Ipatov, *Recent trends in potentiometric sensors arrays-A review*, *Anal. Chim. Acta*, 678 (2010) 149-159.
107. M. del Valle, *Electronic tongues employing electrochemical sensors*, *Electroanalysis*, 22 (2010) 1539-1555.
108. P. Ciosek, W. Wróblewski, *Sensor arrays for liquid sensing-electronic tongue systems*, *The Analyst*, 132 (2007) 963-978.
109. Vitushkin A.G. and Henkin G.M., *Linear Superposition of Functions*, *Russian Math. Surveys*, 22 (1967) 77-125.
110. Q. Zhang, A. Benveniste, *Wavelet Networks*, *IEEE Trans. Neural Network*, 3 (1992) 889- 898.
111. Y. Fang, T.W.S. Chow, *Orthogonal Wavelet Neural Networks Applying to Identification of Wiener Model*, *IEEE Trans. Circuits Syst. Regul. Pap.* 47 (2000) 591-593.

112. J. Zhao, B. Chen, J. Shen, *Multi-dimensional non-orthogonal wavelet-sigmoid basis function neural network for dynamic process fault diagnosis*, *Comput. Chem. Eng.*, 23 (1998) 83-92.
113. Z. Garkani-Nejad, H. Rashidi-Nodeh, *Comparison of conventional artificial neural network and wavelet neural network in modeling the half-wave potential of aldehydes and ketones*, *Electrochim. Acta*, 55 (2010) 2597-2605.
114. R.M. Balabin, R.Z. Safieva, E.I. Lomakina, *Wavelet Neural Network (WNN) approach for calibration model building based on gasoline near infrared (NIR) spectra*, *Chemom. Intell. Lab. Syst.*, 93 (2008) 58-62.
115. K. Zarei, M. Atabati, *Principal component-wavelet neural network as a multivariate calibration method for simultaneous determination of iron, nickel, and cobalt*, *Anal. Letters*, 39 (2006) 2085-2094.
116. A.A. Ensafi, T. Khayamian, R. Tabaraki, *Simultaneous kinetic determination of thiocyanate and sulphide using eigenvalue ranking and correlation ranking in principal-component wavelet neural network*, *Talanta*, 71 (2007) 2021-2028.
117. A. Gutiérrez, F. Céspedes, R. Cartas, S. Alegret, M. del Valle, J.M. Gutiérrez, R. Muñoz, *Multivariate calibration model from overlapping voltammetric signals employing wavelet neural Networks*, *Chemom. Intell. Lab. Syst.*, 83 (2006) 169-179.
118. J.M. Gutiérrez, A. Gutiérrez, F. Céspedes, M. del Valle, *Wavelet neural network to resolve the overlapping signal in the voltammetric determination of phenolic compounds*, *Talanta*, 76 (2008) 373-381.
119. Q.X. Guo, W.S. Cai, Y.C. Liu, *Driving force prediction for inclusion complexation of alpha-cyclodextrin with benzene derivatives by a wavelet neural network*, *Chem. Phys. Lett.*, 290 (1998) 514-518.
120. K. Zarei, M. Atabati, M. Ebrahimi, *Quantitative structure-property relationship study of the solvent polarity using wavelet neural networks*, *Anal. Sci.*, 23 (2007) 937-942.
121. T. Khayamian, M. Esteki, *Prediction of solubility for polycyclic aromatic hydrocarbons in supercritical carbon dioxide using wavelet neural networks in quantitative structure property relationship*, *J. Supercrit. Fluids*, 32 (2004) 73-78
122. P.S. Addison, *The Illustrated Wavelet Transform Handbook*, Institute of Physics, Bristol UK (2002).
123. C.K. Chui, *An introduction to Wavelets*, Academic Press, San Diego, CA (1992)
124. L. Debnath, *Wavelet Transforms and Their Applications*, Birkhäuser Boston (2002)
125. C. Blatter, *Wavelets: A primer*, AK Peters Ltd, Natick MA (1998)
126. S. Mallat, *A wavelet tour of signal processing*, Academic Press, Elsevier, (1999)
127. I. Daubechies, *Ten lectures on Wavelets*, Society for Industrial and Applied Mathematics, Philadelphia PA (1992).
128. G. Bachman, L. Narici, E. Beckenstein, *Fourier and Wavelet Analysis*, Springer (2000).
129. G. Kaiser, *A Friendly Guide to Wavelets*, Birkhäuser, Cambridge MA (1994).
130. A. Gutiérrez, S. Marco, Editors, *Multivariate Calibration Model for a Voltammetric Electronic Tongue Based on a Multiple Output Wavelet Neural Network*, in *Biologically Inspired Signal Processing for Chemical Sensing*, Springer (2009)
131. R.M. Rao, A.S. Bopardikar, *Wavelet Transforms. Introduction to Theory and Applications*, Addison-Wesley (1998).
132. M. Misiti, Y. Misiti, G. Oppenheim, J.M. Poggi, *Wavelets and Their Applications*, ISTE (2007).
133. B. Walczak, Editor, *Wavelets in Chemistry*, Elsevier (2000).

134. T.P. Sarkar, C. Su, *A Tutorial on Wavelets From an Electrical Engineering Perspective, Part 2: The continuous case*, IEEE Antenn. Propag. Mag., 40 (1988) 36-49.
135. A. Boggess, F.J. Narcowick, *A First Course in Wavelets with Fourier Analysis*, Prentice Hall (2001)
136. T. Kugarajah, Q. Zhang, *Multi-dimensional wavelet frames*, IEEE Trans. Neural Networks, 6 (1995) 1552-1556.
137. I. Daubechies, A. Grossmann, Y. Meyer, *Painless nonorthogonal expansions*, J. Math. Phys. 27, 1271-1283 (1986)
138. I. Daubechies, *Ten Lectures on wavelets*, In: *CBMS-NSF Regional Conference Series In Applied Mathematics*, Philadelphia, PA. Society for Industrial and Applied Mathematics, vol. 61 (1992)
139. Y. Oussar, I. Rivals, L. Personnaz, G. Dreyfus, *Training Wavelet Networks for Nonlinear Dynamic Input-Output Modeling*, Neurocomputing (1998) 173-188.
140. T. Hallard, G.R. Spedding, *2-D wavelet transforms: generalisation of the Hardy space and application to experimental studies*, Eur. J. Mech. B/Fluids, 12 (1993) 107-134.
141. J.N. Hwang, S.R. Lay, M. Maechler, R.D. Martin, J. Schimert, *Regression modelling in backpropagation and projection pursuit learning*, IEEE Trans. Neural Networks, 5 (1994) 342-353.
142. F. Piazza, A. Uncini, M. Zenobi, *Artificial neural networks with adaptive polynomial activation functions (vol 2)*, Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, Italy, June 2000.
143. L. Vecci, P. Campolucci, F. Piazza, A. Uncini, *Approximation Capabilities of Adaptive Spline Activation Function*, Proc. of International Conference on Neural Networks ICNN'97, Houston TX, USA, June 1997.
144. Vecci L., Piazza F. and Uncini A., *Learning and Approximation Capabilities of Adaptive Spline Activation Function Neural Networks*, Neural Networks, 11 (1998) 259-270.
145. J. Zhou, L. Li, *Regularized B-spline network and its application to heart arrhythmia classification*, ACM Symposium on Applied Computing, 2004.
146. C. Bishop, *Improving the generalization properties of radial basis neural Networks*, Neural Comput., 3 (1991) 579-588.
147. E. Hartman, J.D. Keeler, *Predicting the future: Advantages of semilocal units*, Neural Comput., 3 (1991) 566-578.
148. T. Poggio, F. Girosi, *Regularization algorithms for learning that are equivalent to multilayer networks*, Science, 247 (1990) 978-982.
149. F. Girosi, M. Jones, T. Poggio, *Regularization theory and neural Networks architecture*, Neural Comput., 7 (1995) 219-269.
150. E. Süli, D. Mayers, *An introduction to numerical analysis*, Cambridge University Press (2003).
151. *Apuntes de interpolación clásica*. Facultad de Informática, Universidad Politécnica de Madrid, http://artico.lma.fi.upm.es/numerico/asigs/c_numerico/cuadernos/interp_clasica.pdf
152. *Análisis numérico I*, Ingeniería Técnica en Informática, Universidad de Huelva, <http://www.uhu.es/cristobal.garcia/>
153. H.C. Müller S.C., *Una introducción al análisis numérico*, Departamento de Matemáticas, Universidad Mayor de San Simón, https://www.u-cursos.cl/ingenieria/2007/2/MA33A/2/material_docente/.
154. R.L. Burden, J.D. Faires, *Numerical Analysis*, 8th Edition, Thompson Brooks/Cole (2005).

155. J.H. Ahlberg, E.N. Nilson, J.L. Walsh, *The theory of splines and their applications*, Academic Press (1967).
156. M. Paluszny, H. Prautzsch, W. Boehm, *Métodos de Bézier y B-splines*, Universitätsverlag Karlsruhe, Denmark (2005).
157. M. Unser, *Splines. A perfect fit for signal and image processing*, IEEE Signal Process. Mag., 16 (1999) 22-38
158. K. Höllig, *Finite element methods with B-splines*, SIAM Philadelphia (2003).
159. C. de Boor, *A practical guide to splines*, Springer (2001).
160. C. Habermann, F. Kidermann, *Multidimensional Spline Approximation: Theory and Applications*, Comput. Econ., 30 (2007) 153-169
161. P.M. Prenter, *Splines and variational methods*, John Wiley & Sons (1989)
162. D.S. Burdick, *An introduction to tensor products with applications to multiway data analysis*, Chem. Intell. Lab. Systems, 28 (1995) 229-237.
163. L. do Santos Coelho. M.W. Pessôa, *Nonlinear identification using a B-spline neural network and chaotic immune approaches*, Mech. Syst. Sig. Process., 23 (2009) 2418-2434.
164. S. Guarnieri, F. Piazza, A. Uncini, *Multilayer feedforward networks with adaptive spline activation function*, IEEE Trans. Neural Networks, 10 (1999) 672-683.
165. M. Solazzi, A. Uncini, *Regularising neural Networks using flexible multivariate activation function*, Neural Networks, 17 (2004) 247-260.
166. E. Catmull, R. Rom, *A class of local interpolating splines*, en Computer Aided Geometric Design, R. E. Barnhill and R. F. Reisenfeld, Eds. Academic Press, New York (1974).
167. S. Guarnieri, F. Piazza, A. Uncini, *Multilayer feedforward networks with adaptive spline activation function*, IEEE Trans. Neural Networks, 10 (1999) 672-683.
168. M. Solazzi, A. Uncini, *Regularising neural Networks using flexible multivariate activation function*, Neural Networks, 17 (2004) 247-260.
169. J. H. Friedman, *Multivariate Adaptive Regression Splines*, The Annals of Statistics, Vol. 19, No 1, 1-141, 1991.
170. E. Deconinck, M.H. Zhang, F. Petitet, E. Dubus, I. Ijjaali, D. Coomans, Y. Vander Heyden, *Boosted regression trees, multivariate adaptive regression splines and their two-step combinations with multiple linear regression on partial least squares to predict blood-brain barrier passage: A case study*, Anal. Chim. Acta, 609 (2008) 12-23.
171. P.A.W. Lewis, J.G. Stevens, *Nonlinear Modelling of Time Series Using Multivariate Adaptive Regression Splines (MARS)*, Journal of the American Statistics Association, Vol. 86, No. 416, 864-877, 1991.
172. S. Bakin, M. Hegland, M. Osborne, *Can MARS be improved with B-splines?*, Computational Techniques and Applications Conference CTAC97
173. T.P. Sarkar, C. Su, R. Adve, M. Salazar-Palma, L. García-Castillo, R.R. Boix, *A Tutorial on Waveletes From an Electrical Engineering Perspective, Part 1: Discrete Wavelet Techniques*, IEEE Antenn Propag Mag, 40 (1988) 49-70
174. S. Mallat, *A theory for multiresolution signal representation: The wavelet representation*, IEEE Tran Pattern Anal Mach Intell, 11 (1989) 674-693
175. Y. Meyer, *Wavelets. Algorithms and applications*, SIAM Philadelphia (1993)
176. G. Strang, T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Wellesley MA (1997)
177. M. Cocchi, R. Seeber, A. Ulrici, *Multivariate calibration of analytical signals by WILMA (Wavelet Interface to Linear Modelling Analysis)*, Journal of Chemometrics, Volume 17, Issue 8-9 (August-September 2003), pages 512-527.

178. L. Moreno-Barón, R. Cartas, A. Merkoçi, S. Alegret, J.M. Gutiérrez, L. Leija, P.R. Hernández, R. Muñoz, *Data compression for a voltammetric electronic tongue modelled with Artificial Neural Networks*, Analytical Letters, 38 (2005) 2189-2206
179. L. Moreno-Barón, R. Cartas, A. Merkoçi, S. Alegret, M del Valle, L. Leija, P.R. Hernández, R. Muñoz, *Application of the Wavelet Transform coupled with Artificial Neural Networks for quantification purposes in a voltammetric electronic tongue*, Sensors and Actuators B 113 (2006) 487–499
180. S.Alegret, J.Alonso, J.Bartrolí, F.Céspedes, E.Martínez-Fàbregas, M.del Valle, *Amperometric biosensors based on bulk-modified epoxy-graphite biocomposites*, Sensors Mater., 8 (1996) 147-253.
181. S. Alegret, *Rigid carbon-polymer biocomposites for electrochemical sensing: A review*, Analyst, 121 (1996) 1751-1758.
182. Pingarrón JM, Sánchez P. *Química Electroanalítica. Fundamentos y Aplicaciones*. Ed. Síntesis, Madrid, 2003.
183. R. Bro, *Exploratory study of sugar production using fluorescent spectroscopy and multi-way analysis*, Chem. Intell. Lab. Systems, 46 (1999) 133-147.
184. D. Calvo, A. Durán, M. del Valle, *Use of sequential injection analysis to construct an electronic-tongue. Application to multidetermination employing the transient response of a potentiometric sensor array*, Anal. Chim. Acta, 600 (2007) 97–104.
185. D. Calvo, A. Durán and M. D. Valle, *Use of pulse transient response as input information for an automated SIA electronic tongue*, Sen. Actuators, B, Chem, 131 (2008) 77-84.
186. R. Cartas, A. Mimendia, A. Legin, M. del Valle, *Two analyte calibrations from the transient response of a single potentiometric sensor employed with the SIA technique*, Talanta, 80 (2010) 1428-1435.
187. X.Cetó, F. Céspedes, M.I. Pividori, J.M. Gutiérrez, M. del Valle, *Resolution of phenolic antioxidante mixtures employing a voltammetric bio-electronic tongue*, Analyst, 137 (2012) 349-356