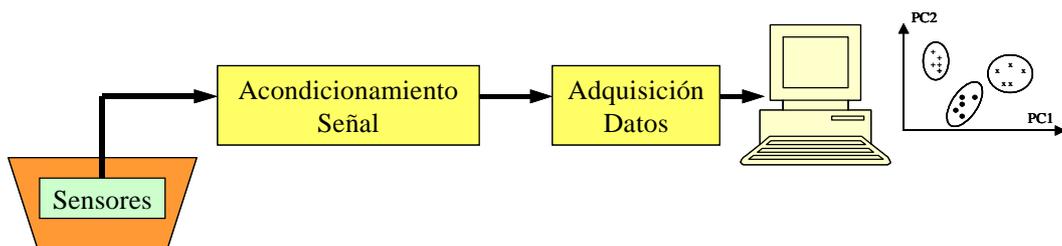
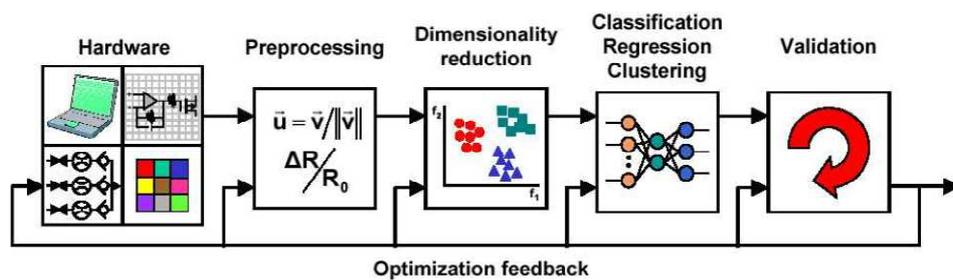


Capítulo 1

Introducción y Objetivos



1. Introducción.

Las investigaciones en el campo de los sensores químicos siempre han ido encaminadas a la búsqueda de dispositivos que sean lo más específicos posible hacia el parámetro que se desea medir, que logren la medida con la mayor precisión posible y que las medidas se realicen con la menor interferencia de otros parámetros no deseados. Bajo este planteamiento se desarrollaron electrodos específicos, como por ejemplo los electrodos selectivos de iones (ISE) [Camman, 1979]. Pero existen muchas otras aplicaciones en donde se busca una diferenciación cualitativa entre distintas muestras analizadas en lugar de la precisión de un parámetro químico. En este caso, se puede medir diversos parámetros y se busca la relación entre ellos, para así obtener una valoración cualitativa global de la muestra. Bajo estas premisas nacieron, hace algunas décadas, las denominadas narices electrónicas que aplican sensores de gases para la detección y discriminación de aromas [Gardner, 1992]. Reciben este nombre por su analogía con el sentido del olfato biológico [Pearce, 1998]. Más recientemente, y siguiendo un concepto similar a las narices electrónicas, se están desarrollando las llamadas lenguas electrónicas para su aplicación en medios líquidos, asemejando por ello al sentido del gusto biológico [Legin et al., 1997].

El punto de partida de las lenguas electrónicas fue una extensión de las aplicaciones de los sensores ion-selectivos potenciométricos. Este tipo de sensores suele tener una buena precisión en disoluciones simples, pero en disoluciones complejas surgen problemas con los elementos interferentes, que afecta a la fiabilidad de las medidas. Como alternativa, se planteó un conjunto de sensores inespecíficos con sensibilidad cruzada [Legin, 1996]. Es decir, sensores con sensibilidad a múltiples elementos interferentes, cada uno con diferente grado de sensibilidad pero logrando en conjunto una discriminación de muestras complejas. Por todo lo indicado, se puede definir una lengua electrónica como un sistema electrónico que, por medio de múltiples sensores no específicos, puede realizar un análisis cualitativo con muestras de naturaleza líquida [Winqvist et al., 2002]. Con el conjunto de electrodos se

puede obtener una “huella electrónica” de la muestra, que corresponde con la expresión de las medidas del conjunto de electrodos [Vlasov, 2005].

1.1. Etapas de los sistemas de lengua electrónica.

Un sistema de lengua electrónica está formado por un conjunto de bloques enlazados en forma de cadena, de tal forma que a partir de la información que captan los sensores se obtenga finalmente una clasificación de las muestras según las características que sean de interés (**Fig. 1-1**).

Los bloques que la constituyen suelen ser los siguientes [Gil, 2007]: sensores, sistema de acondicionamiento de la señal, sistema de adquisición de datos y análisis multivariante de los datos.

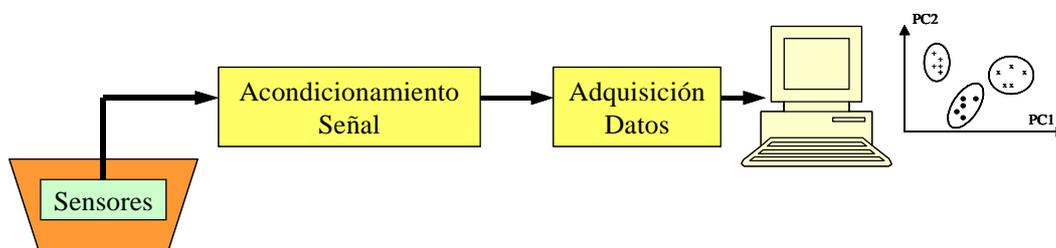


Figura 1-1. Esquema de bloques de una lengua electrónica

1.1.1. Sensores.

Un sensor es un dispositivo que obtiene una señal eléctrica proporcional al parámetro físico o químico que desea medir. Los sensores utilizados en lenguas electrónicas suelen ser de naturaleza electroquímica. Estos sensores se pueden clasificar según dos criterios: a) tipo de señal eléctrica utilizado y b) material activo.

1-a) **Técnicas de medida.** La señal eléctrica que se obtiene de un sensor electroquímico puede ser de distinto tipo: potencial eléctrico, corriente, resistencia, etc. De esta forma se obtienen distintas técnicas de medida. Las más utilizadas en lenguas electrónicas son las siguientes:

1-a-1 **Potenciometría.** Es la medida de un potencial que se genera entre un electrodo activo con respecto a otro de referencia estando ambos

sumergidos en un líquido común. El potencial generado depende de la actividad de una sustancia en disolución que genera un potencial eléctrico. Esta técnica ha sido aplicado ampliamente en sistemas de lenguas electrónicas [Gallardo et al., 2005], [Legin et al., 1997].

1-a-2 **Voltametría.** Esta técnica abarca un grupo de métodos electroanalíticos en los que la información sobre el analito se deduce de la medida de la intensidad de corriente en función del potencial aplicado en condiciones que favorezcan la polarización de un electrodo indicador, o de trabajo. Para ello, se aplica a la muestra una tensión variable y se mide la corriente generada. También se ha aplicado ampliamente a sistemas de lengua electrónicas [Apetrei et al., 2004], [Arrieta et al., 2004].

1-a-3. **Espectrometría de impedancias.** Consiste en aplicar una señal de entrada de frecuencia variable y determinar como afecta a la variación de la impedancia, tanto en su módulo como en fase. Se ha aplicado en menor medida a los sistemas de lenguas electrónicas [Pioggia et al., 2007]

1-b. **Materiales.** Los electrodos que se utilizan en las lenguas electrónicas poseen distintos tipos de materiales activos. Entre ellos destacan las membranas ion selectivas [Zampolli et al., 2004], biosensores [Dhawa et al., 2009], ISFET [Moreno et al., 2006], metales y materiales simples [Lvova et al., 2006]. En este último caso se logran dispositivos de fácil construcción, robustos y bajo mantenimiento.

1.1.2. Sistema de acondicionamiento de la señal.

En las medidas potenciométricas, las señales eléctricas suelen tener valores muy bajos y es posible que su impedancia de salida sea elevada. Por este motivo suelen ser muy sensibles a las interferencias y ruidos eléctricos. Por eso, las etapas del sistema de acondicionamiento de la señal están formadas por una etapa amplificación de muy alta impedancia de entrada y una segunda etapa de filtros para rechazar las señales de ruido.

1.1.3. Sistema de adquisición de datos.

Los sistemas de análisis y tratamiento de datos trabajan con señales digitales, por lo que debe existir una etapa de conversión analógica a digital con un número de bits adecuado para que la resolución sea la deseada. Este proceso suele realizarse en placa de adquisición de datos incluido en ordenador, que por medio de programas informáticos permiten la adquisición, monitorización y almacenamiento de las señales.

1.1.4. Análisis multivariante de los datos.

Para obtener una adecuada clasificación de las muestras analizadas u otra conclusión de los resultados, es necesario aplicar métodos de análisis multivariante con los datos de las medidas. Los análisis se realizan con programas informáticos que realizan cálculos matemáticos complejos de reconocimiento de patrones.

Dichas técnicas, se han utilizado en varias aplicaciones, incluyendo el control de las aguas residuales [*Sakai et al., 2000*] y el análisis de alimentos [*Gallardo et al., 2005*].

2. Procesado de Datos.

En el procesado de señal existen una serie de etapas, que se cumplen en la mayoría de los casos en las narices y lenguas electrónicas. Las etapas se pueden agrupar en un sistema de bloques como se refleja a continuación (**Fig. 1-2**) [*Gutierrez-Osuna, 2002*].

El punto de partida del sistema de bloques es el **hardware** del sistema. Bajo este nombre se engloba al conjunto de subbloques que forman: los sensores químicos, los sistemas neumáticos (narices) o hidráulicos (lenguas) que permite el movimiento de las muestras a analizar, los sistemas electrónicos para la transducción de la señal procedente de los sensores y el sistema informático para el tratamiento de los datos.

La etapa de **preprocesado**, como su nombre indica, contempla posibles modificaciones previas (p.e. normalizaciones) de los valores de las medidas

que posibiliten o faciliten las siguientes etapas de procesamiento de datos. El preprocesado suele ser necesario porque los sensores tienden a generar informaciones de magnitud muy variada, lo que dificulta su comparación, o las condiciones de las medidas no son constantes debido a derivas de los sensores.

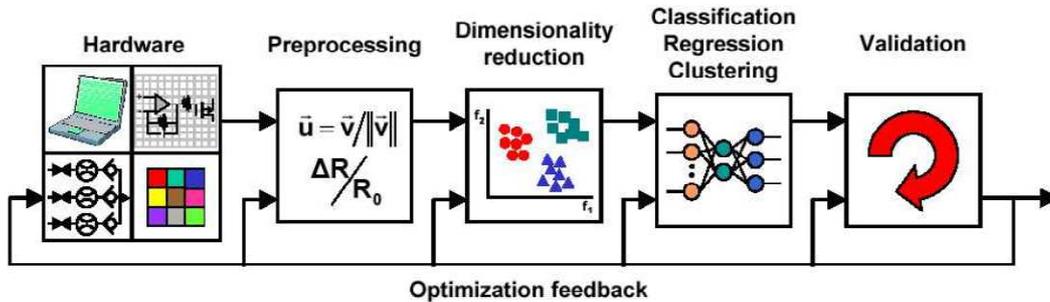


Figura 1-2. Diagrama de bloques de sistema de procesamiento de datos una nariz electrónica

Las técnicas de preprocesado de datos más utilizadas son:

- **Centrado respecto al valor medio.**
- **Estandarización.** Las medidas de cada sensor son divididas por el valor de la desviación estándar
- **Normalización del Sensor y de la Matriz.** Las medidas de cada sensor son divididas por el valor de la desviación estándar

El **procesado** propiamente dicho de los datos suele estar formado por una serie de técnicas que permiten obtener conclusiones de las medidas realizadas. Las tareas básicas de procesamiento de datos son la **clasificación** y la **regresión**.

Las técnicas de **clasificación** intentan lograr una distribución de los datos según parámetros de interés, mientras que las técnicas de **regresión** cuantifican la relación entre una matriz de entrada y otra de salida a través de unos coeficientes. La finalidad de las técnicas de regresión es realizar una predicción de una cualidad de la muestra a partir de los datos obtenidos en las medidas.

Con los resultados de las diferentes tareas del procesamiento de datos se puede realizar una **validación** de todo el sistema para determinar si cumple con las especificaciones para las que se había diseñado o no. En caso negativo se

modifican algunas de las etapas del sistema mediante un lazo de realimentación que permite el control y la optimización del sistema.

2.1. Selección de datos.

Normalmente, cada vez que se realiza una medida se obtiene un dato por cada sensor, la manera de como se obtiene el dato puede variar en función de la naturaleza del sensor o de la muestra. En general se busca un estado estacionario de la medida y de aquí se obtiene el dato final, pero hay ocasiones en que la información útil puede encontrarse durante el transitorio, o incluso en ambos momentos (transitorio y estacionario), de forma que se pueda obtener más de un dato útil por cada medida.

A las técnicas que permiten extraer la información útil de las medidas para así lograr los datos definitivos para realizar el análisis se les denomina técnicas de **Selección de Características**.

2.2. Matriz de datos.

Para realizar el procesado de datos se parte de una matriz bidimensional de datos, donde existen tantas columnas como sensores o parámetros se hayan establecido para realizar las medidas y tantas filas como medidas se hayan realizado para dichos sensores. Esta forma de trabajar hace que sea muy apropiado la aplicación de técnicas matemáticas matriciales, así como el uso de programas informáticos de cálculo matricial como el *MATLAB* [*MathWorks*].

$$Matriz\ de\ Datos = \begin{bmatrix} R_{11} & R_{12} & \dots & R_{1p} \\ & \dots & & \\ & & R_{ij} & \\ & & & \dots \\ R_{n1} & R_{n2} & \dots & R_{np} \end{bmatrix} \quad (1.1)$$

Siendo: n = número de sensores y p = número de medidas.

En el lenguaje de procesamiento de datos se denomina **característica** (*Feature*) a cada uno de los valores de matriz, y a cada fila de la matriz se le denomina **vector característico**.

3. Análisis de Datos.

El propósito fundamental del análisis de datos es la preparación de datos para transformar los datos crudos en otros donde el contenido de la información encerrada en el conjunto de datos pueda ser expuesta, o más fácilmente accesible [Pyle, 1999]. También se conoce como **reconocimiento de patrones** (*Pattern Recognition* - PARC).

3.1. Términos utilizados en el análisis de datos.

Los términos más importantes que caracterizan a las técnicas de reconocimiento de patrones son los siguientes:

- **Métodos paramétricos o no paramétricos.**

Algunos métodos suponen que los datos para realizar el análisis siguen una distribución según una función conocida de probabilidad, normalmente distribución Gaussiana. A estos métodos se le denominan **paramétricos**, en caso contrario son métodos **no paramétricos**.

- **Métodos supervisados o no supervisados.**

En los métodos **supervisados** se fijan previamente los grupos o clases que se desean obtener y se establece una regla para determinar cada medida a que grupo debe asociarse y en que grado se logra el éxito en la clasificación. A este proceso se le suele denominar entrenamiento.

En la clasificación de datos **no supervisada** no se establecen grupos previos de datos o cuantificación de las salidas, simplemente se comprueba el método, por si solo, es capaz de descubrir de forma natural afinidades o discrepancias entre los datos.

- **Métodos Lineales y No lineales**

En los métodos de reconocimiento de patrones se establecen unos valores de salida que están relacionado con los originales de entrada por medio de unos coeficientes. Estos coeficientes pueden ser lineales o no lineales.

4. Validación.

En el último punto de la cadena de procesos de datos, se plantean los criterios para la selección del modelo y la estimación del resultado.

Cuando se realiza una nueva aplicación se deben determinar, no solo cuál es el modelo apropiado entre una amplia variedad de algoritmos de procesado de datos, sino también establecer el modelo que logre el resultado óptimo. La validación de modelo se basa en la estimación de su capacidad para predecir nuevos resultados. El modelo de predicción debe tener la cualidad de aprovechar la fase de entrenamiento para extraer de los datos sólo su estructura fundamental y no fijarse en aspectos accidentales (por ejemplo el ruido) que den lugar a posteriores predicciones erróneas. Este hecho ocurre cuando el modelo es “sobre-entrenado” con los datos disponibles, con un número irrazonablemente grande de parámetros del modelo o por un exceso de iteraciones de entrenamiento. Encontrar un modelo de predicción que aproveche los datos de entrenamiento pero sin llegar a excederse es el objetivo de la validación.

Para evitar el sobreentrenamiento existen diversos métodos: el más inmediato es repartir los datos disponibles en datos para el entrenamiento y en datos para la validación. El conjunto de datos de entrenamiento se utiliza para enseñar a varios modelos con diferentes estructuras. El modelo de entrenamiento que logra la mejor validación de los datos es el seleccionado como modelo final. Esta técnica de validación sencilla es conocida como el método *holdout*. Aunque la técnica *holdout* funciona bien en muchas situaciones, posee dos inconvenientes. Primero, en problemas con número limitado de datos no es conveniente permitirse el lujo de desprenderse de un conjunto de datos para la validación. Segundo, siendo la técnica *holdout* un simple experimento de entre-

namiento y validación, la estimación del resultado puede ser errónea si se realiza un desafortunado reparto de datos. Los defectos de este método pueden solucionarse a expensas de una adicional computación, por medio de múltiples particiones del conjunto de datos.

La validación cruzada *K-fold* es una técnica que realiza K particiones de los datos de forma que cada dato es utilizado tanto para el entrenamiento como para la validación. En cada uno de los K repartos se utilizan N/K datos para la validación y el resto $N(K-1)/K$ se utilizan para el entrenamiento, donde N es el número total de datos. Por ejemplo, si se posee un total de 60 datos ($N=60$) y se realizan 5 particiones ($K=5$) cada partición posee 12 elementos, el entrenamiento se realiza con los elementos de 4 particiones (48 datos) y la validación con la partición restante (12 restantes). Posteriormente, se toma otra partición para la validación y las 4 restantes como entrenamiento y así sucesivamente hasta que todas las particiones hayan realizado la función de validación. En este caso se ha tenido una validación cruzada de orden 5. Cuando el número de grupos es igual al número de datos ($K=N$), el método es conocido como validación cruzada de orden 1 (*Leave-one-out-LOO*).

La validación final del método de procesado de datos consiste en determinar primeramente la tasa de éxito de cada partición de validación respecto a sus correspondientes particiones de entrenamiento y posteriormente realizar un promedio de todas las tasas de éxito, para así obtener una tasa de éxito promedio.

El parámetro importante a determinar es el número óptimo de particiones (K). En general para un conjunto de datos grande, un valor pequeño de K ($\cong 3$) será suficiente. Para conjuntos de datos escasos, es preferible el uso de la validación cruzada de orden uno (LOO) para así obtener el mayor número de ejemplos posibles. Los contrastes de los recursos computacionales también pueden tenerse en consideración ya que el número de ejecuciones del programa se incrementará de forma considerable con el aumento de K .

5. Redes neuronales artificiales (RNA).

5.1. Introducción.

La aparición de los ordenadores permitió que tareas lentas y tediosas se ejecutaran rápidamente y con mayor precisión. Los primeros programas computacionales resolvían importantes problemas de física, posteriormente aparecieron los procesadores de texto, programas gráficos, etc. Sin embargo, resulta irónico que computadoras capaces de realizar 100 millones de operaciones por segundo, no sean capaces de realizar tareas que resultan triviales y comunes, tales como:

- Aprendizaje experimental
- Reconocimiento de patrones complejos tales como, caracteres y voz
- Procesamiento de señales
- Clasificación de patrones, etc.

Las redes neuronales artificiales imitan el comportamiento de las redes neurológicas, por lo que, basándose en los procesos realizados por el cerebro humano e inspirados en su funcionamiento diversos investigadores han desarrollado la teoría, incluso antes de la aparición de los ordenadores.

Los elementos estructurales constituyentes del cerebro, las neuronas biológicas [*Ramón y Cajal, 1911*], son unos seis órdenes de magnitud más lentas que las puertas lógicas de silicio, ofreciendo tiempos de respuesta del orden del milisegundo frente a los vertiginosos tiempos de respuesta del orden del nanosegundo que tienen ciertos dispositivos digitales actuales.

La eficiencia energética del cerebro es aproximadamente de 10^{-16} julios por operación por segundo, mientras que los mejores ordenadores no superan los 10^{-6} julios por operación por segundo [*Faggin, 1991*].

Por lo tanto, informáticos, ingenieros y matemáticos trataron de hacer una abstracción intelectual que permitiera al ordenador trabajar de un modo similar al cerebro humano, lo que dio lugar a la aparición de la red neuronal artificial (RNA) que ha proporcionado una alternativa en algunos problemas en

los que los resultados obtenidos por la computación clásica no han sido satisfactorios.

En definitiva, una red neuronal es un intento de copiar (de forma simplificada) el funcionamiento del cerebro humano. Cuando decimos que es un intento, es, entre otras razones, porque la funcionalidad del cerebro no está en absoluto bien comprendida.

En los últimos años, el tema de las redes neuronales se ha convertido en algo cada vez más popular, tanto dentro del ambiente universitario de investigación como en el mundo comercial. Es fácil que se oigan referencias a este tema en relación a un amplio conjunto de áreas tales como ciencias de la computación, neurofisiología, robótica, control automático, comunicaciones, física, matemáticas, psicología, y otros numerosos campos donde se están investigando y encontrando nuevas aplicaciones.

Una RNA no es más que la existencia de muchos procesadores [*Haykin, 1994*], denominados unidades o nodos [*Russell & Norvig, 2003*], de una determinada sencillez e interconectados entre ellos, cuyo funcionamiento se basa en el de la neurona humana y que trata de emular el proceso de aprendizaje del cerebro humano, mediante un conjunto de patrones de entrenamiento. El término artificial indica, que su implementación se puede realizar, tanto por software, como por hardware.

También las redes neuronales [*Bhadeshia, 2008*] permiten el desarrollo de expresiones cuantitativas sin comprometer la complejidad conocida de un problema.

Independientemente de cómo se transmite la información (señales eléctricas y fenómenos químicos) o de cómo se consigue el aprendizaje, es sin duda esta filosofía de procesamiento la que predomina en el sistema nervioso humano.

Las ventajas de adoptar esta idea son genéricamente tres [*Rich & Knight, 1991*].

- 1) Procesamiento masivamente paralelo, con las ventajas que esto pueda suponer a la hora de rapidez de procesamiento.

- 2) Sistema altamente redundante, con la consiguiente tolerancia a fallos que esto pueda implicar.
- 3) Sistema con capacidad de aprendizaje fruto de la experiencia (ejemplos), a través de la actuación sobre los pesos en las conexiones.

Otras Definiciones Importantes

- *... una red neuronal es un sistema compuesto de muchos elementos de procesamiento simples, que operan en paralelo y cuya función es determinada por la estructura de la red, la fortaleza de las conexiones y el procesamiento en los elementos computacionales o nodos [DARPA, 1988]*
- *Una red neuronal es un sistema de procesamiento distribuido y masivamente paralelo, que tiene una propiedad natural para almacenar conocimiento por medio de la experiencia y lo hace disponible para su uso. Así podemos identificar dos elementos importantes: [Haykin, 1994]*
 1. *El conocimiento que es adquirido a través de un proceso de aprendizaje.*
 2. *Las fuerzas de conexión interneuronal, conocidas como pesos sinápticos, permiten almacenar el conocimiento.*
- *Una red neuronal es un circuito compuesto por un elevado número de elementos de procesamiento simples basados en neuronas. Cada elemento opera sólo con información local. [Nigrin, 1993]*
- *Las redes neuronales o sistemas neuronales artificiales son sistemas físico-celulares que pueden adquirir, almacenar y utilizar conocimiento basado en experimentos. [Zurada, 1992]*

5.2. Aplicaciones de las redes neuronales.

Una red neuronal puede ser utilizada para las siguientes aplicaciones:

- 1) Almacenar y obtener datos o patrones (por ejemplo esta es la aplicación de las memorias asociativas).
- 2) Clasificar patrones (esta es una de las aplicaciones fundamentales de los perceptrones multicapa).

- 3) Realizar mapeos generales desde patrones de entrada a patrones de salida. Esta aplicación en realidad podríamos considerar que engloba las dos anteriores junto con otras aplicaciones muy típicas, como son labores de traducción, predicción, compresión, etc. De hecho, los perceptrones multicapa realizan en general este tipo de aplicaciones.
- 4) Agrupar patrones similares: (*Clustering*). Este tipo de aplicación es la realizada por redes como los mapas autoorganizativos de Kohonen.
- 5) Encontrar soluciones a problemas de optimización con restricciones. Este es el tipo de aplicación realizada por redes como las máquinas de Boltzman.

Sobre este conjunto de aplicaciones se pueden hacer los siguientes comentarios:

- En realidad los tipos de aplicaciones se podrían reducir a los tres últimos, debido a que, los dos primeros son casos particulares del tercero.
- prácticamente cada tipo de red está especializado en un tipo de aplicación.

5.3. Historia de las Redes Neuronales.

Ya en los años 40 se empezaron a buscar alternativas al modelo de Von Neumann de computación secuencial, por diversos motivos ya expuestos: búsqueda de mecanismos robustos, tolerantes a fallos, sistemas adaptativos, no lineales, procesamiento masivo en paralelo con unidades baratas, imitación del cerebro humano.

Dentro de estos primeros intentos cabe destacar:

- **Mc Culloch y Pitts:** Primer modelo simple de neurona. Consideraban la neurona como una función lógica y por tanto las redes neuronales como una combinación de funciones lógicas. Introdujeron la idea del umbral. Su modelo estaba basado en neuronas simples, las cuales fueron creadas con circuitos electrónicos [*McCulloch & Pitts, 1943*].

- **Hebb:** Propuso una ley matemática bastante simple sobre el aprendizaje. Lo que se conoce como Regla de Hebb. Si dos neuronas se activan simultáneamente, su relación se refuerza; los caminos que llevaron a la respuesta correcta se potencian. Esto se modeliza variando los coeficientes de ponderación (pesos) que una neurona artificial aplica a sus entradas para producir una salida, reforzando las conexiones que mejor se acercan al resultado correcto, y así, los sistemas neuronales son capaces de "aprender" [Heb, 1949].

Este modelo fue implementado sobre una máquina en 1950 por Rochester. Esta máquina no funcionó.

Después de estos primeros intentos se dieron los primeros grandes resultados entre los que cabe destacar:

- **Rosenblatt:** Este psicólogo propuso el Perceptrón: teóricamente con un algoritmo de aprendizaje mejor que el de Hebb, mediante ajuste iterativo de los pesos. Utilizaba una función umbral, demostró la convergencia de su algoritmo de aprendizaje ante unos supuestos iniciales y describió varios tipos de perceptrones [Rosenblatt, 1958] [Rosenblatt, 1962].
- **Widrow-Hoff:** En este caso, el ingeniero eléctrico (Widrow) y su alumno (Hoff) propusieron el ADALINE (*Adaptive Linear System*), que utilizaba una regla de aprendizaje denominada "*delta rule*" que a la postre sería la precursora del algoritmo de retropropagación ("*backpropagation*"). Este modelo posteriormente se llegó a comercializar para la eliminación de ecos en las líneas de teléfono [Widrow & Hoff, 1960].
- **Grossberg:** propuso la red Avalanche [Grossberg, 1967].

Realmente en la década de los 60 el tema de las redes neuronales tomó una elevada importancia, parecida a la que existe actualmente, hasta que se llegó a un punto de formalización en el que se determinaron las limitaciones de las redes. En este sentido cabe destacar un artículo publicado en 1969 por **Minsky y Papert** [Minsky & Papert, 1969]. En general dos de las limitaciones más importantes señaladas y encontradas fueron la imposibilidad de solucionar el

problema de la XOR y el no dar con un algoritmo de aprendizaje para el entrenamiento de las redes multicapa.

Prácticamente desde entonces y fruto fundamentalmente del trabajo anterior, las redes neuronales entraron en un largo periodo en el cual el tema fue dejado de lado, y si alguien trabajaba en este campo no tenía ninguna repercusión notable.

La investigación en este campo en los EEUU se ve acallada por el libro "*Perceptrons*". Sin embargo se consolidan importantes pilares de las RNA bajo los nombres de proceso adaptativo de señales, reconocimiento de patrones y modelado biológico. Pertenecen a esta época Shun-ichi Amari, James Anderson, Kunihiko Fukushima, Stephen Grossberg, Harry Klopf, Teuvo Kohonen y David Willshaw. Esta época duró hasta 1981.

A partir de 1980 [*Chappelier & Grumbach, 1994*] y hasta nuestros días se ha vivido en el tema de las redes neuronales un importante resurgimiento, incluso un claro apogeo (1986 en adelante).

- **Principios de los 80.** Se produce el resurgimiento de la computación neuronal en EEUU, en parte financiado por la "Defense Advanced Research Projects Agency" (DARPA).

La reaparición de las redes neuronales en parte se debió a estudios realizados por el físico John Hopfield 1982, donde utilizó técnicas estadísticas para analizar el almacenamiento y la optimización de propiedades de las redes. Por otro lado Hopfield, se dedica entre 1983 y 1986 a relanzar la computación neuronal mediante publicaciones y conferencias.

- **1986** Publicación de "*PDP Books*" (*Parallel Distributed Processing, Vol. I and II*) editados por David Rumelhart y James Mc Clelland que supone un verdadero acontecimiento por la presentación del método de retropropagación.

- **1987** IEEE International Conference on Neural Networks con 1700 participantes (San Diego). Se crea la International Neural Networks Society (INNS)

- **1988** Publicación de la revista "*Neural Networks*" por el INNS.

- **1989** Publicación de la revista "*Neural Computation*"

- **1990** Publicación de la revista "*Transactions on Neural Networks*"

por el IEEE [Muñoz, 1996].

Personas que han contribuido a este resurgimiento son por ejemplo:

- **Hopfield.** Fundamental es la aportación de este físico galardonado con el Premio Nobel de Física. Aportó, entre otras cosas, unas redes de pesos fijos y activación adaptativa que llevan su nombre (redes de Hopfield) [Hopfield, 1982].
- **Rumelhart-Hinton-William:** Creadores del algoritmo de *Backpropagation*, que permitía aprender a las redes multicapa [Rumelhart & Hinton, 1986].
- **Kohonen.** Aporto los mapas autoorganizativos, que fueron utilizados para reconocimiento de la voz, composición musical, y para solucionar el famoso problema “Salesman Problem” [Kohonen, 1982], [Kohonen, 1984].
- **Grossberg,** Del que se computan hasta 146 artículos publicados desde 1967 hasta 1988, de los que cabe destacar sus estudios muy matemáticos y a su vez muy biológicos de sus modelos [Grossberg, 1967], [Grossberg, 1973], [Grossberg, 1988].
- **Carpenter.** Creadora del las redes ART1 y ART2 (Adaptative Resonance Teory). [Carpenter & Grossberg, 1987]

Las causas de este resurgimiento podríamos enunciarlos en los siguientes puntos:

- Nuevos algoritmos de aprendizaje.
- Nuevas topologías que reactivaron el tema.
- Experimentos demostrativos que confirmaban las expectativas. Es decir la posibilidad de realizar simulaciones.
- Desarrollo de ciertas tecnologías, fundamentalmente la de creación de circuitos VLSI, que posibilitaba la implementación hardware del software desarrollado, con el consiguiente incremento de las posibilidades de las redes neuronales. En este concepto también cobran importancia los descubrimientos en la tecnología óptica.
- Importante flujo de dinero hacia este tipo de investigaciones.

- Fracaso, por parte de los sistemas expertos, ante problemas complejos: imagen, voz, etc.

Actualmente el número de personas dedicadas a este tema es muy elevado. Es más, el tratamiento a este tema se está abordando desde diferentes puntos de vista que dificultan muchas veces el esclarecimiento del tema: punto de vista informático-software (filosofía de implementación de software), punto de vista fisiológico (acercamiento a la comprensión del funcionamiento del sistema nervioso), punto de vista psicológico (comprensión del comportamiento y aprendizaje humano), punto de vista informático-hardware (nuevas arquitecturas de computadores), punto de vista de los electrónicos (importante aplicación de los circuitos VLSI), etc.

En realidad muchas áreas de conocimiento están relacionadas con este tema y no es posible tratar las redes neuronales sin hacer obligada referencia a aspectos tales como los orígenes biológicos que implicaron el desarrollo de la idea de estas redes, o los diferentes modelos de redes (algoritmos) que determinan y propician las subsiguientes realizaciones hardware.

5.4. Funcionamiento de una neurona biológica.

Con objeto de entender mejor el funcionamiento de las redes neuronales artificiales examinemos brevemente el modelo biológico. Como puede apreciarse en la **Fig. 1-3**, las neuronas tienen tres componentes principales:

- las dendritas
- el cuerpo de la célula y
- el axón.

Las dendritas son conexiones de entrada, constituyen el árbol motor, que recogen estímulos de otras neuronas. El axón es una fibra larga, que lleva la conexión hacia otras neuronas, es por lo tanto, la conexión de salida, cuyas múltiples ramificaciones le permiten afectar a muchas otras neuronas y emitir señales de salida.

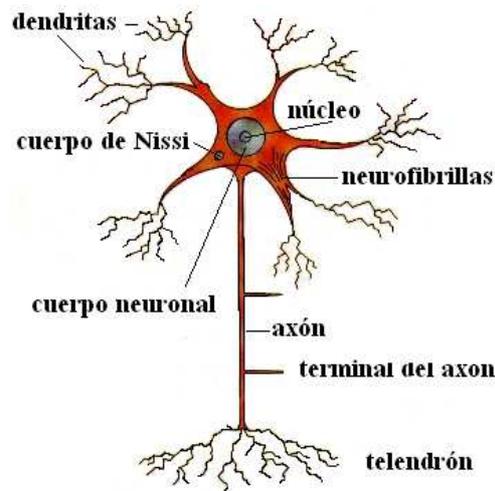


Figura 1-3. Neurona biológica.

El axón lleva estas señales a los terminales que se encargan de distribuir información a un nuevo conjunto de neuronas [Acosta & Zuloaga, 2010]. El axón posee estructuras especiales (vainas de mielina, nodos de Ranvier...) para ayudar a transmitir la información, pero esta información es del mismo tipo que la del resto de células. El terminal de los axones está vinculado con las dendritas de otras neuronas a través de una estructura llamadas **sinapsis** (Fig. 1-4).

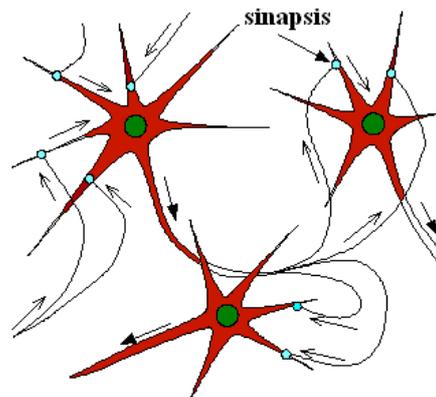


Figura 1-4. Conexión sináptica de una red neuronal.

La información recorre las neuronas en forma de impulsos eléctricos generados por fenómenos iónicos. Hasta tal punto es importante el proceso,

que una gran parte del diseño de la membrana celular parece destinado a ello en exclusiva.

Algunas neuronas son determinadas en el nacimiento en cambio otras, se desarrollan a través del aprendizaje, proceso en el que se crean nuevas conexiones neuronales perdiéndose otras por completo. El comportamiento ante diversos estímulos puede variar de forma muy diversa a lo largo del tiempo, mediante cambios en el modo de combinar los estímulos recibidos en cada neurona.

Estos cambios (aprendizaje) son objeto de intensos estudios, pues dictan el comportamiento de extensos conjuntos de neuronas. Esto tiene especial interés en la investigación de los sistemas neuronales artificiales.

5.5. Aspectos clave de las neuronas biológicas.

1. El elemento de procesamiento recibe muchas señales (x_i).
2. Las señales pueden ser modificadas por un peso (W_j) en las sinapsis receptoras.
3. La unidad de procesamiento suma las entradas ponderadas.
4. Bajo apropiadas circunstancias (suficiente entrada) la neurona transmite una salida.
5. La salida de una neurona particular puede ir a varias neuronas.
6. El procesamiento de información es local (aunque con la actuación de las hormonas esto no sea totalmente cierto).
7. La memoria es distribuida:
 - A largo plazo reside en los pesos de las sinapsis.
 - A corto plazo corresponde a las señales enviadas por las neuronas.
8. La fortaleza de la conexión sináptica puede ser modificada por la experiencia.
9. Los neurotransmisores pueden ser excitadores e inhibidores.
10. Los sistemas neuronales biológicos son tolerantes a fallos en dos aspectos:
 - Son capaces de reconocer muchas señales de entrada que son bastante diferentes de alguna señal procesada anteriormente.

- Existen del orden de 100 billones de neuronas, que lamentablemente decrecerán a lo largo de nuestra vida por cuanto no se regeneran. Sin embargo, nuestras facultades no se ven por ello mermadas, incluso en los casos de graves lesiones cerebrales, otras partes del cerebro han logrado asumir en buena parte las funciones que realizaban las áreas dañadas.

5.6. Características de una red neuronal artificial.

El modelo de una neurona artificial es una imitación del proceso de una neurona biológica (**Fig. 1-5**). Basado en los aspectos anteriores sugeridos por las referencias biológicas tenemos que en una Red Neuronal Artificial el papel de las neuronas lo cumplen elementos de proceso [Haykin, 1994] (PE's), o nodos, unidades compuestas por circuitos (analógicos, digitales o simulados) capaces de sumar ponderadamente sus entradas, y producir a través de una función no lineal una salida, representando su frecuencia de activación.

- **Las entradas** X_i representan las señales que provienen de otras neuronas y que son capturadas por las dendritas.
- **Los pesos** W_i son los coeficientes empleados para multiplicar las entradas, corresponde a la intensidad de la sinapsis que interconecta dos neuronas.
- θ es la función **umbral** que la neurona debe sobrepasar para activarse.
- f la función de **activación** o de transferencia.

El valor de activación se obtiene como suma de productos de las entradas por sus pesos.

$$\eta = \sum_{i=1}^n x_i W_i \quad (1.2)$$

El resultado η , denominado valor de activación, se compara con un cierto valor umbral, esto se realiza por **la función de activación o de transferencia**

[Haykin, 1994]. En general si el resultado es mayor que la función umbral, la neurona será activada, dando el valor Y a la salida.

En estos pesos es en donde se va a concretizar lo aprendido por una red neuronal, por lo tanto, como luego veremos, la mayoría de los métodos de entrenamiento de las redes neuronales consistirán en la modificación de estos pesos según ciertos patrones de aprendizaje [Andina, 2010].

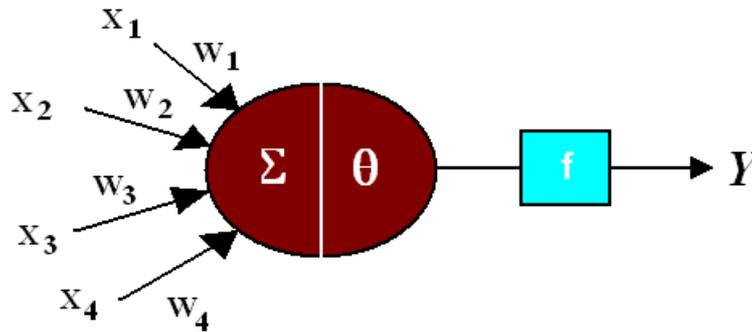


Figura 1-5. Neurona artificial.

De la observación detallada del proceso biológico, podemos decir que las redes neuronales artificiales han sido desarrolladas bajo los siguientes supuestos:

- El procesamiento de información ocurre en muchos elementos de procesado sencillo denominados neuronas o PE.
- Las señales pasan entre las neuronas a través de unas conexiones que tienen asociado un peso, el cual, en una red neuronal típica, multiplica la señal transmitida.
- Cada neurona aplica una función de activación (usualmente no lineal) a la suma de las entradas ponderadas para determinar la salida.

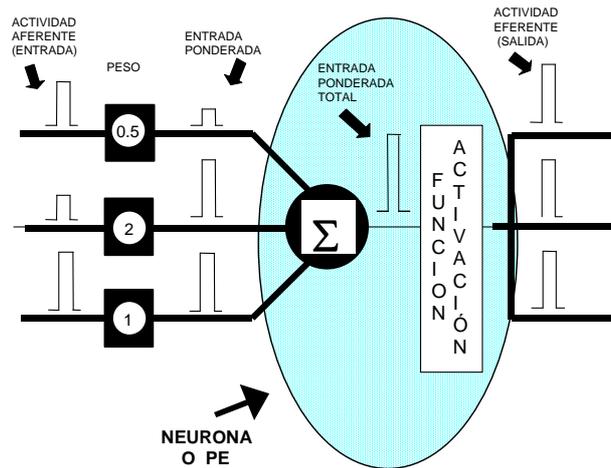


Figura 1-6. Diagrama funcional de una neurona artificial

Existen dos fases en toda aplicación de las redes neuronales. Estas son la *fase de aprendizaje* o entrenamiento y la *fase de prueba* o reconocimiento. En la fase de entrenamiento, se usa un conjunto de datos para determinar los pesos con los que hemos definido el modelo de Red Neuronal Artificial. Este modelo entrenado se usará posteriormente en la fase de prueba en la que se procesan patrones de prueba reales para dar los resultados definitivos.

- **Fase de prueba:** hay varios sistemas no lineales propuestos para los patrones almacenados. Los resultados pueden ser tanto calculados de una vez o adaptados iterativamente en función de las ecuaciones dinámicas de prueba. Los valores de la última neurona representan la salida deseada para ser comprobada con los valores objetivo.
- **Fase de aprendizaje:** una característica de las redes neuronales es su capacidad de aprender. Ellas aprenden por la actualización o cambio de los pesos sinápticos que caracterizan a las conexiones. Los pesos son adaptados de acuerdo a la información extraída de los patrones de entrenamiento nuevos que se van presentando. Normalmente los pesos óptimos se obtienen optimizando (minimizando o maximizando) alguna “función de energía”. Por ejemplo, un criterio muy extendido en el entrenamiento supervisado es minimizar el error cuadrático medio entre los valores del maestro (o valores objetivo) y los valores de las salidas actuales.

Las aplicaciones del mundo real deben de acometer dos tipos diferentes de requisitos en el procesado. Una requiere que la fase de prueba sea realizada en tiempo real, pero con una velocidad de entrenamiento “fuera de línea”. El otro tipo de requisito requiere que ambas fases sean realizadas en tiempo real. Estos dos requisitos diferentes implican evidentemente velocidades de proceso muy diferentes, que afectan a los algoritmos empleados y hardware usados.

Un tema interesante, dentro de estos conceptos básicos sobre las redes neuronales consiste en la caracterización de una red neuronal. Una red neuronal puede ser caracterizada básicamente por tres aspectos fundamentales:

- 1) El patrón de interconexiones entre neuronas: **arquitectura o estructura.**
- 2) El método de determinar los pesos en las conexiones: lo que denominamos **entrenamiento, aprendizaje o algoritmo.**
- 3) Sus funciones de **Base** y de **Activación.**

Estos aspectos van a ser concretados a continuación, explicando las diferentes posibilidades.

5.7. Arquitecturas típicas.

Los elementos de procesamiento de una red neuronal se encuentran agrupados por capas. Una capa es un conjunto de neuronas, las cuales normalmente se comportan de la misma manera. A menudo es conveniente visualizar las neuronas como organizadas en capas. Los factores fundamentales para determinar el comportamiento de una neurona es conocer que función de activación tiene y cuál es el patrón de conexiones con pesos a través de los cuales recibe y envía señales. Dentro de cada capa, las neuronas usualmente tienen las mismas propiedades con respecto a estos dos aspectos comentados anteriormente, es decir, en la mayoría de las redes neuronales, las neuronas pertenecientes a cada capa suelen tener el mismo patrón de interconexión y las mismas funciones de activación.

El conocimiento de la organización de las neuronas en capas y los patrones de interconexión entre capas suele ser conocido como arquitectura de la red.

Desde el punto de vista de organización por capas, las redes neuronales pueden clasificarse en:

- redes de una sola capa y
- redes multicapa

Dependiendo de la situación de la capa en la red neuronal, esta recibe diferentes nombres:

- **Capa de entrada:** Es la que recibe las señales de entrada, algunos autores no contabilizan las unidades de entrada como capa, ya que no realizan computación.
- **Capas ocultas:** Son las que no tienen contacto con el exterior. Los elementos que las componen pueden tener diferentes conexiones y determinan la topología de la red.
- **Capa de salida:** Es la que transmite la información al exterior en función de la información recibida de la capas ocultas.

Normalmente, para contabilizar el número de capas, suele observarse el número de interconexiones entre conjunto de neuronas. Este punto de vista es bastante interesante por cuanto realmente los pesos son los que contienen la información importante [Fuente & Calonge, 1999] [Perera, 2003].

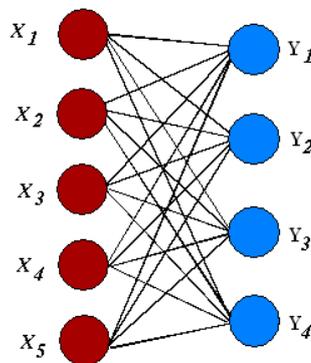


Figura 1-7. Red neuronal monocapa

A continuación se muestra una red neuronal monocapa (Fig. 1-7) y multicapa (Fig. 1-8).

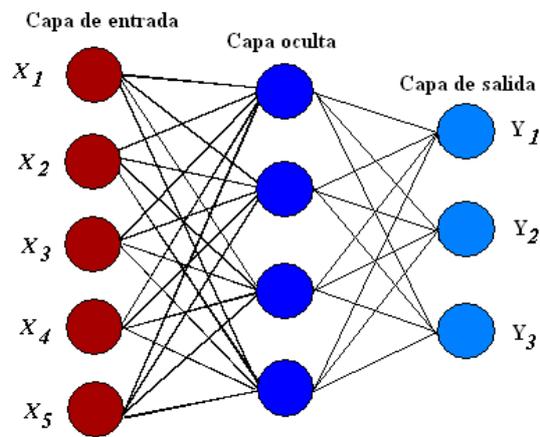


Figura 1-8. Red neuronal multicapa

Las dos redes ilustradas anteriormente son ejemplos de redes “*feedforward*”, es decir que las señales fluyen de las entradas hacia las salidas en una dirección hacia delante.

Existe otro tipo de red neuronal denominada red **recurrente**, en la cual cabe destacar interconexiones de cada neurona consigo misma (Fig. 1-9). Esta red es competitiva y totalmente interconectada.

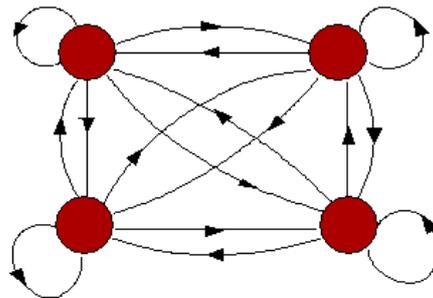


Figura 1-9. Red neuronal recurrente.

En general, y para clarificar las posibilidades desde el punto de vista de esquema de conexión, podemos concluir que existen los siguientes tipos: Hacia adelante (*feedforward*), hacia atrás (*feedback*), lateral y de retardo.

Conexiones hacia adelante: para todos los datos neuronales, los datos de las neuronas de una capa inferior son propagados hacia las neuronas de la capa superior por medio de las conexiones hacia adelante. Si bien esta es una definición bastante completa, habría que añadir que hablamos únicamente de la fase de prueba. Algunas redes neuronales típicamente *feedforward* como el perceptrón multicapa, tienen una fase de aprendizaje en la que el gradiente de error se propaga hacia atrás, de ahí el nombre de “*backpropagation*”.

1. *Conexiones hacia atrás:* estas conexiones llevan los datos de las neuronas de una capa superior a otras de la capa inferior. Este tipo de conexiones es muy típica de las redes recurrentes y algunas de las dinámicas como la red de Elman.
2. *Conexiones laterales:* un ejemplo típico de este tipo de conexión es el circuito “el ganador toma todo” (winner-takes-all), que cumple un papel importante en la elección del ganador.
3. *Conexiones con retardo:* los elementos de retardo se incorporan en las conexiones para implementar modelos dinámicos y temporales (redes dinámicas). Estos son apropiados para el reconocimiento de patrones temporales.

5.8. Aprendizaje.

El cerebro aprende basándose en la experiencia. El aprendizaje en las redes neuronales, a veces denominados algoritmos de aprendizaje de máquina, se refiere al proceso de adquirir un comportamiento deseado por medio del cambio de los pesos de las interconexiones. La fuerza de las conexiones entre las neuronas, se almacena como el valor del peso para una conexión específica. El sistema aprende un nuevo conocimiento ajustando los pesos en las conexiones.

Denominaremos aprendizaje o entrenamiento de la red al ajuste en los valores de los pesos, que viene reflejado por el cambio en las conexiones entre las neurona de la red.

El aprendizaje se lleva cabo mediante **entrenamiento**, el cual origina una matriz que mantiene los pesos entre las neuronas.

En la tabla 1.1, se muestra una posible clasificación de las redes neuronales en función de cómo se realiza la fase de aprendizaje.

Tabla 1.1. Clasificación de las redes neuronales según la forma de aprendizaje.	
Aprendizaje supervisado	
Feedforward	Perceptron [Roseblatt, 1958] Perceptron Multicapa [Minsky & Papert, 1969]. Adaline [Widrow & Hoff, 1960]
Feedback	Recurrent backpropagation [Pineda, 1987][Pineda, 1988] Maquina de Boltzman [Ackley et al., 1985] Fuzzy Artmap [Carpenter et al., 1992]
Aprendizaje no supervisado	
Feedforward	LVQ: Learning Vector Quantization [Kohonen, 1988]
Feedback	Hopfield Net [Hopfield, 1982] Kohonen Feature Map [Kohonen, 1982] ART1 [Carpenter & Grossberg, 1987a] ART2 [Carpenter & Grossberg, 1987b]

En la tabla anterior también se puede comprobar que existen dos grandes estrategias de aprendizaje:

- Aprendizaje supervisado.
- Aprendizaje no supervisado.

Vamos a ver las características de cada estrategia de entrenamiento y el tipo de problema adecuado para dicho entrenamiento.

- **Entrenamiento supervisado.**

Las redes neuronales de entrenamiento supervisado son las más populares. En este tipo de entrenamiento, es necesario disponer de unos patrones de aprendizaje consistentes en un conjunto de pares de elementos, que no son otros que los patrones de entrada y los correspondientes patrones de salida asociados. El patrón de entrada se envía a la red neuronal a través de la capa de entrada. Los pesos de la red son entonces

ajustados mediante un algoritmo de aprendizaje con la intención de que la respuesta de la red ante dichos patrones de entrada sea lo más parecida posible a los patrones de salida disponibles, que son considerados a lo largo del entrenamiento como valores deseados. Dependiendo de la diferencia entre el patrón de salida y la salida deseada, se calcula el error, el cual indica el esfuerzo de aprendizaje de la red, que puede ser controlado por un supervisor.

El conjunto de patrones de entrenamiento necesita ser lo suficientemente grande como para contener toda la información necesaria para que la red aprenda todas las relaciones importantes [Fuente & Calonge, 1999].

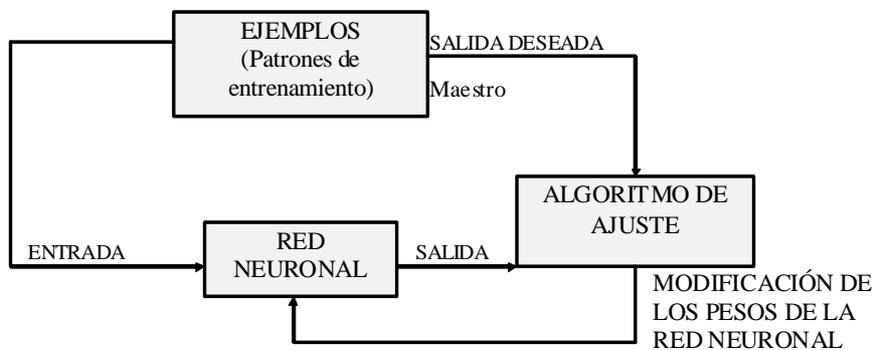


Figura 1-10. Entrenamiento supervisado.

Algunos algoritmos de aprendizaje supervisados incluyen:

- Aprendizaje por refuerzo.
- Aprendizaje estocástico.
- Aprendizaje por corrección de error.
- Least Mean Square error (LSM).

Las redes típicas que utilizan este tipo de entrenamiento supervisado suelen dividirse en dos grandes grupos dependiendo de la naturaleza de la información del maestro y la forma de usarla:

- 1) Redes basadas en la corrección de la **decisión**. Por ejemplo la red **perceptrón**
- 2) Redes basadas en la optimización de un **criterio de coste**. La red más importante es la aproximación del error cuadrático medio. Por ejemplo los

perceptrones multicapa y las redes **ADALINE/MADALINE**. [Widrow & Hoff, 1960], [Widrow & Winter, 1988].

- **Entrenamiento no supervisado.**

Las redes neuronales con entrenamiento no supervisado carecen de salidas esperadas, basándose solamente en la información de entrada. Son redes también denominadas autoorganizativas, las cuales agrupan juntos vectores de entrada, sin usar para ello datos de entrenamiento que especifiquen a que miembro típico de cada grupo se parece o a que grupo pertenece cada vector. Dicho de otro modo, la red es entrenada sin el beneficio de un supervisor. La red aprende a adaptarse basada en las experiencias recogidas de los patrones de entrenamiento anteriores. En la Fig. 1-11 se muestra un esquema típico de un sistema “No supervisado”.

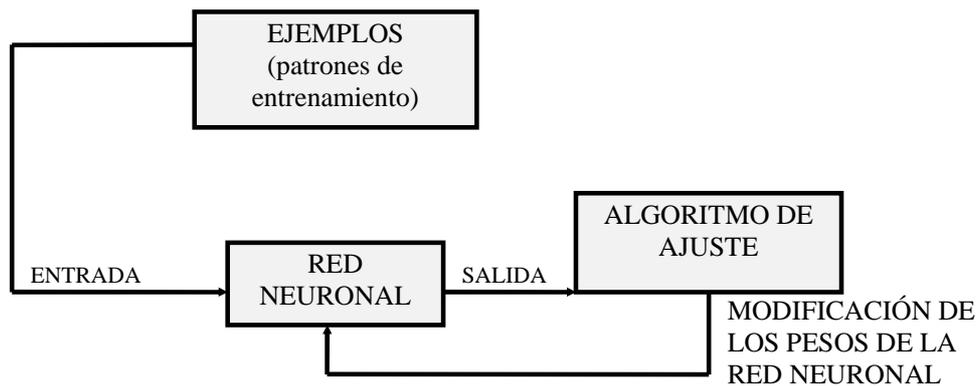


Figura 1-11. Entrenamiento no supervisado

5.9. Funciones de Base y Activación.

Un modelo básico de una red neuronal se puede caracterizar por la función de base y la función de activación. Cada nodo (PE o neurona), (Fig. 1-5), tiene una entrada de valor x_i y suministra un valor y_i a su salida. El valor de entrada se propaga a través de la red de conexiones unidireccionales hacia otras células de la red. Asociada a cada conexión hay un peso sináptico denominado w_{ij} que determina el efecto del nodo i -ésimo sobre el nodo j -ésimo. Las entradas al nodo j -ésimo que provienen de los otros nodos x_i junto

con los pesos y el umbral externo θ_j , nos van a permitir calcular el valor de la red η_j (red de conexión).

La salida final y_j , se obtiene al aplicar la función de **Activación** a la función de **Base**, por lo tanto, se puede expresar como una función de la entrada y de los pesos $y=f(x, W)$.

5.9.1. Función de Base (Función de Red).

Para un estudio analítico, las redes de conexión son matemáticamente representadas por la función de base $\eta(w, x)$, donde w es la matriz de pesos, y x el vector de entrada. La función de base tiene dos formas típicas:

- *Función de Base Lineal (LBF)* es una función de tipo hiperplano. Esto es, una función de primer orden. El valor de la red es una combinación lineal de las entradas.

$$\eta_j(w, x) = \sum_{i=1}^n \omega_{ij} x_i \quad (1.3)$$

- *Función de base Radial (RBF)* es una función de tipo hiperesférico. Esto implica una función de base de segundo orden no lineal. El valor de la red representa la distancia a un determinado patrón de referencia.

$$\eta_j(w, x) = \sqrt{\sum_{i=1}^n (x_i - \omega_{ij})^2} \quad (1.4)$$

La función de segundo orden se puede extender a otra más general llamada función de base elíptica.

5.9.2. Función de Activación (Función de neurona).

Para las unidades de entrada, la función utilizada es la función identidad (función lineal). Para las demás unidades, se utiliza normalmente funciones no lineales.

La función no lineal corresponde al carácter no lineal de la respuesta biológica, que puede ser modelizado con funciones lineales; escalones (hard limiters), cuya respuesta es más fiel a la biología, o sigmoides, cuyo comportamiento matemático es menos hostil y que podemos expresar mediante las ecuaciones denominadas:

- Sigmoide binaria (1.5)
- Sigmoide bipolar (1.6)
- Función gaussiana (1.7)

Mientras las dos primeras se asocian normalmente a la LBF, la tercera se asocia a la RBF.

$$f_1(\eta) = \frac{1}{1 + e^{-\sigma\eta}} \quad (1.5)$$

$$f_2(\eta) = \frac{1 - e^{-\sigma\eta}}{1 + e^{-\sigma\eta}} = 2f_1(\eta) - 1 \quad (1.6)$$

$$f(\eta_j) = ce^{-\eta^2 / \sigma^2} \quad (1.7)$$

En cualquier caso existe una gran variedad de funciones no-lineales que pueden ser utilizadas aunque la sigmoide es de las más importantes.

6. Red Neuronal Perceptrón.

El primer modelo de red neuronal artificial relativamente sencillo, pero capaz de aprender a reconocer patrones en entradas continuas o binarias fue llamado Perceptrón. Si a esta red se le incorpora umbrales y pesos variables, puede servir para clasificar patrones incluso en decisiones complejas. El Perceptrón es un tipo de red de aprendizaje supervisado, es decir necesita conocer los valores esperados para cada una de las entradas presentadas. La

red tipo Perceptrón fue inventada por el psicólogo Frank Rosenblatt en el año 1957 [Roseblatt, 1958], para emular las características principales de las neuronas biológicas. El modelo neuronal se divide en tres zonas o unidades:

- Sensorial
- Asociativa
- Respuesta

Las dos primeras zonas están interconectadas mediante pesos fijos y entre las dos últimas con pesos variables. Las unidades asociativas actúan como pequeños preprocesadores diseñados para extraer información a partir de los ejemplos presentados.

Su funcionamiento es el siguiente: Una serie de entradas multiplicadas por un peso, que representa la "fuerza" de la conexión sináptica se aplican al perceptrón, la suma de todas las entradas multiplicadas por su peso determinan su nivel de activación.

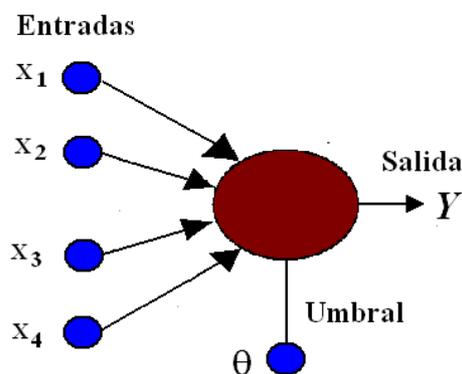


Figura 1-12. Perceptrón de Rosenblatt.

Si la entrada neta es mayor que un valor umbral entonces la salida de la red es 1, en caso contrario es 0.

- $y = 1$ si $salida \geq \theta$
- $y = 0$ de cualquier otra manera

De este modo, todos los vectores de entrada quedarán clasificados en dos grandes clases C1 y C2. En el caso de un vector de entrada de dos

componentes, estaremos hablando de una línea de decisión que separa cualquier (x_1, x_2) de las dos clases.

La elección y actualización de los pesos W_i , dependerá en particular de cada algoritmo de entrenamiento, pero es evidente que sin importar el algoritmo usado las soluciones finales han de ser parecidas.

Si los datos de entrada poseen tres componentes, tendremos que trazar un plano que diferencie zonas de R^3 , cuando tengan n será necesario trazar un **hiperplano** de $n-1$ dimensiones.

El perceptrón, con una única neurona, tiene una capacidad de representación bastante limitada, este modelo sólo es capaz de discriminar patrones muy sencillos, patrones linealmente separables, el caso más conocido es la imposibilidad del perceptrón de representar la función OR EXCLUSIVA.

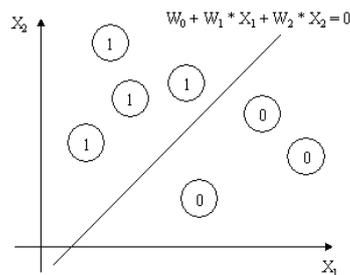


Figura 1-13. Particionado del conjunto de entrada.

En la Fig. 1-13 se muestra un perceptrón monocapa, su aplicación está limitada al uso de clasificación de patrones en dos clases o categorías. Para poder clasificar correctamente más de dos clases debemos expandir la capa de salida incluyendo más de una neurona, con la limitación demostrada en 1969 Marvin Minsky y Seymour Papert en su libro: "Perceptrons: An introduction to Computational Geometry", [Minsky & Papert, 1969] de que estas clases deben ser separables linealmente.

Se pueden aplicar diferentes funciones de activación (Figura 1-), tales como:

- escalón o *hardlim* con salidas 1, 0
- escalón o *hardlims* con salidas 1, -1
- *sigmoidea*

siendo esta última, la que más se utiliza porque, a diferencia de las hardlim, es continuamente diferenciable y eso es un requisito para poder aplicar el algoritmo de retro-propagación del gradiente del error o Backpropagation.

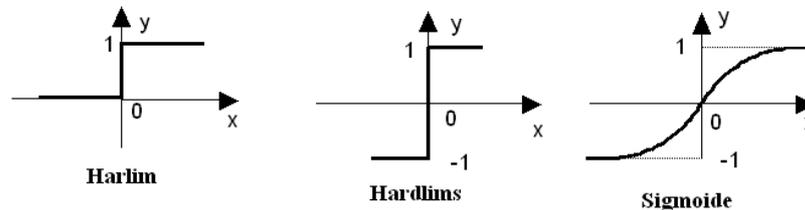


Figura 1-14. Funciones de activación de una neurona artificial.

6.1. El perceptrón multicapa.

Un perceptrón multicapa es una red compuesta de varias capas de neuronas ocultas entre la entrada y la salida de la misma. Esta red permite establecer regiones de decisión mucho más complejas que las de dos semiplanos, como lo hace el perceptrón de un solo nivel.

Un perceptrón multicapa es un tipo de red con propagación hacia delante (*“FeedForward”*), compuesta de varias capas de neuronas ocultas entre la entrada y la salida de la misma, esta red permite establecer regiones de decisión mucho más complejas que las de dos semiplanos, como lo hace el Perceptrón de un solo nivel.

El perceptrón multicapa (MLP, *MultiLayer Perceptron*) al igual que la mayoría de las redes neuronales artificiales, crean modelos a partir de multiplicadores, sumadores, funciones, etc., siendo el exponente más típico de las redes neuronales artificiales con aprendizaje supervisado.

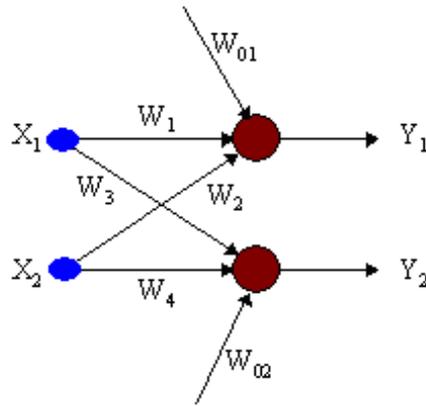


Figura 1-15. Asociación de dos perceptrones

La Fig. 1-15 muestra la asociación de dos perceptrones en paralelo. De esta forma, disponemos de dos salidas Y_1 e Y_2 , con las que se pueden diferenciar cuatro partes distintas dentro del espacio R^2 , tal como se ve en la Fig. 1-16.

Ahora las comparaciones se realizan sobre dos rectas distintas, lo que genera cuatro zonas de agrupamiento. Si dispusiéramos de n rectas tendríamos hasta un máximo de $2n$ posibles salidas.

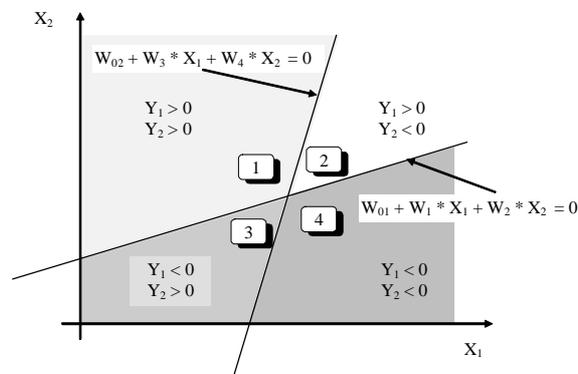


Figura 1-16. Particionado mediante dos perceptrones.

Hemos conseguido separar en subconjuntos independientes de R^2 todos los elementos del conjunto de entrada. Cuantas más neuronas añadamos en paralelo, más extensa será la capa codificadora, más rectas de decisión podremos trazar (si hay solo dos entradas) y más complejas serán las regiones delimitadas.

La estructura típica de un perceptrón multicapa se muestra en la Fig. 1-17, en donde, una neurona en una capa de la red está conectada a todas las neuronas de la capa anterior; en esta estructura identificamos dos tipos de señales.

- *Señal de Función.* Es la señal que se propaga de la entrada hacia la capa de salida.
- *Señal de Error.* Son las generadas por las neuronas de salida y que se retropropagarán en forma de ajuste de las conexiones sinápticas hacia la entrada con el fin de ajustar la salida obtenida.

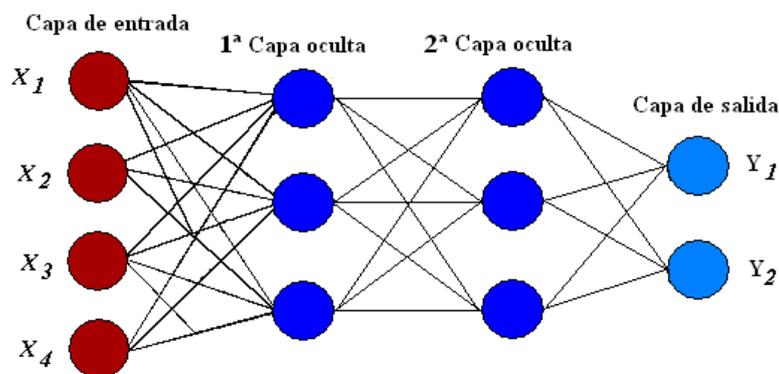


Figura 1-17. Perceptrón multicapa.

Con este tipo de estructura:

1. Tenemos una hipersuperficie de separación en lugar de un hiperplano (que es lo que podemos conseguir con un “*single layer*” como el perceptrón) con lo cual podemos resolver problemas que no sean linealmente separables (que son los únicos que puede resolver un “*single layer*”).
2. Con esta hipersuperficie de separación podemos dividir el espacio en regiones no conectadas y cada una de estas regiones asociarla a una clase en particular.
3. Si queremos regiones no conectadas cerradas podemos recurrir a dos capas escondidas o una capa oculta con un número de nodos en la capa oculta muy elevada.
4. Podemos resolver el problema de la OR exclusiva.

- En realidad tenemos una función discriminante lineal generalizada, capaz de convertir cualquier conjunto de datos no linealmente separable en un conjunto de datos linealmente separable mediante lo que podríamos llamar como una transformación entre espacios.

Para el entrenamiento de estas redes, se presentan sucesiva y reiteradamente pares de vectores en las capas de entrada y salida (vectores de entrada y salida deseada). El modelo de la red que se crea se basa en el ajuste de sus pesos en función de los vectores de entrenamiento. Para cada entrada se producirá un valor de salida muy similar a la salida esperada.

Cada neurona oculta o de salida de un perceptrón multicapa está pues diseñada para realizar dos cálculos: la propagación de la señal hacia adelante (*feedforward*) y el cálculo del vector necesario para la retropropagación.

6.1.1. Conexión entre neuronas.

Como se ha indicado anteriormente, existe una relación en la forma que se conectan las neuronas biológicas y las neuronas artificiales. Esta relación se expresa en la nomenclatura utilizada en las redes neuronales artificiales.

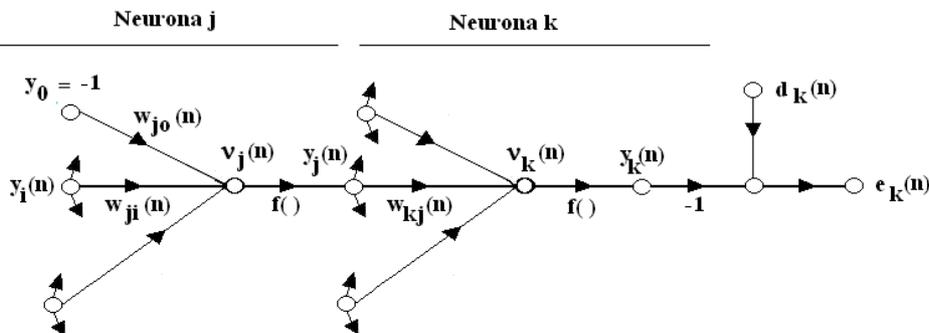


Figura 1-18. Señales de conexión entre una neurona oculta y una de salida.

En la Fig. 1-18, se muestra la conexión entre dos neuronas j y k , siendo la nomenclatura utilizada la siguiente:

- Los índices i, j, k denotan las neuronas ubicadas en las capas i, j, k , respectivamente en el sentido de la propagación.

2. La iteración n se refiere al n -ésimo patrón presentado.
3. El símbolo $E(n)$ se refiere a la suma cuadrática de los errores en la iteración n -ésima. La media sobre todas las muestras será E_{av} .
4. El símbolo $e_j(n)$ indica el error a la salida de la neurona j .
5. El símbolo $d_j(n)$ es la respuesta deseada del nodo j .
6. El símbolo $y_j(n)$ es la señal de salida de la neurona j .
7. El símbolo $w_{ji}(n)$ indica el peso de la conexión sináptica entre las neuronas j e i en la iteración n . La corrección aplicada a ese peso se denota por $\Delta w_{ji}(n)$.
8. El símbolo $x_i(n)$ se refiere al elemento i -ésimo del vector muestra.
9. El elemento k -ésimo de la salida total de la red se denota por $e_k(n)$.
10. El parámetro de aprendizaje será η y el momento α .

6.2. Algoritmo de retropropagación.

Al hablar de retropropagación hacemos referencia a un algoritmo de aprendizaje. El objetivo del aprendizaje consiste en encontrar la configuración de pesos que corresponda al mínimo global de la función de error, aunque en muchos casos es suficiente encontrar un mínimo local lo suficientemente bueno [Cauwenberghs, 1993].

Primero se aplica un dato a la entrada de la red para producir la salida de la misma. Esta salida se compara con la salida deseada y se calcula el error cometido en cada neurona de salida. Estos errores se transmiten desde la capa de salida hacia todas las neuronas de las capas intermedias [Fritsch, 1996]. El error en cada neurona es proporcional a su contribución sobre el error total de la red. Basándose en el error recibido, se ajustan los pesos sinápticos de cada neurona. Por lo tanto, habrá que calcular los errores para las unidades de las capas ocultas a partir de los errores de las unidades de la capa de salida que son propagados capa tras capa hacia la entrada.

Para aplicar el algoritmo de entrenamiento multicapa es necesario que la función de activación sea derivable, por lo tanto buscamos funciones similares al escalón del perceptrón monocapa que sean derivables.

- **Cálculo de errores**

Supongamos que la señal de salida de una neurona de la última capa es $y_k(n)$ y la salida deseada es $d_k(n)$, en donde corresponde a la presentación del *nésimo* patrón de entrenamiento. La señal de error viene definida como:

$$e_j(n) = d_j(n) - y_j(n) \quad (1.8)$$

Donde j es un nodo de salida. Además, la suma de los errores cuadráticos de la red se puede definir como:

$$E(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n) \quad (1.9)$$

donde el conjunto C incluye todos los nodos en la capa de salida de la red.

El Error Cuadrático Medio (E_{av}) se obtiene sumando $E(n)$ para todas las iteraciones y normalizando respecto N :

$$E_{av} = \frac{1}{N} \sum_{n=1}^N E(n) \quad (1.10)$$

en donde N es el número de patrones o ejemplos que forman el conjunto de entrenamiento y E_{av} , denominada *Función Coste*, representa una medida del aprendizaje de la RNA, cuyo objetivo es minimizar la función de error ajustando los parámetros libres.

- **Neurona j como nodo de salida de la red.**

El nivel de activación $v_j(n)$ en la entrada a la neurona j , es:

$$v_j(n) = \sum_{i=0}^p w_{ji}(n) y_i(n) \quad (1.11)$$

Siendo p el número total de entradas aplicadas a la neurona j , por lo tanto, en la iteración n , la salida de la neurona será:

$$y_j = f_j(v_j(n)) \quad (1.12)$$

Para expresar el gradiente de la suma cuadrática de los errores respecto de los pesos asociados podemos aplicar la regla de la cadena de la técnica de derivación parcial, con lo que:

$$\frac{\partial E(N)}{\partial w_{ji}(n)} = \frac{\partial E(N)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \quad (1.13)$$

Derivando las ecuaciones (1.8), (1.9), (1.10) y (1.11) y sustituyendo en la ecuación (1.13), resulta:

$$\frac{\partial E(N)}{\partial w_{ji}(n)} = -e_j(n) f_j'(v_j(n)) y_j(n) \quad (1.14)$$

La corrección $\Delta w_{ji}(n)$ aplicada al peso $w_{ji}(n)$ vendrá dado por:

$$\Delta w_{ji}(n) = -\eta \frac{\partial E(N)}{\partial w_{ji}(n)} \quad (1.15)$$

en donde la constante η determina la velocidad de aprendizaje.

Sustituyendo la Ec. (1.14) en la (1.15), resulta:

$$w_{ji}(n) = \eta \partial_j(n) y_j(n) = \eta e_j(n) f_j'(v_j(n)) y_j(n) \quad (1.16)$$

Siendo:

$$\partial_j(n) = e_j(n) f_j'(v_j(n)) \quad (1.17)$$

el gradiente local.

- **Neurona j como nodo oculto de la red.**

Podemos definir el gradiente local $\delta_j(n)$ para la neurona oculta j de la siguiente manera:

$$\partial_j(n) = \frac{\partial E(N)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} = -\frac{\partial E(N)}{\partial y_j(n)} f_j'(v_j(n)) \quad (1.18)$$

Para la neurona de salida la Ec. (1.9), la podemos escribir como:

$$E(n) = \frac{1}{2} \sum_{k \in C} e_k^2(n) \quad (1.19)$$

donde el subíndice k indica que se trata de una neurona de salida. Al diferenciar la ecuación anterior, obtenemos:

$$\frac{\partial E(n)}{\partial y_j(n)} = \sum_{k \in C} e_k(n) \frac{\partial e_k(n)}{\partial y_j(n)} = \sum_{k \in C} e_k(n) \frac{\partial e_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial y_j(n)} \quad (1.20)$$

La señal de error para la neurona de salida, será:

$$e_k(n) = d_k(n) - y_k(n) = d_k(n) - f_j(v_k(n)) \quad (1.21)$$

La derivada parcial:

$$\frac{\partial e_k(n)}{\partial v_k(n)} = -f'_j(v_k(n)) \quad (1.22)$$

El nivel de activación para la neurona k , vendrá dado por:

$$v_k(n) = \frac{1}{2} \sum_{j=0}^q w_{kj}(n) y_j(n) \quad (1.23)$$

Siendo q el número total de entradas aplicadas a la neurona k . De la expresión anterior, obtenemos:

$$\frac{\partial v_k(n)}{\partial y_j(n)} = w_{kj}(n) \quad (1.24)$$

Sustituyendo las Ecs. (1.24) y (1.22) en la (1.20)

$$\frac{\partial E(n)}{\partial y_j(n)} = - \sum_{k \in C} e_k(n) w_{kj}(n) f'_j(v_k(n)) \quad (1.25)$$

Con lo que el gradiente total la neurona oculta j será:

$$\partial_j(n) = \sum_{k \in C} \partial_k(n) w_{kj}(n) f'_j(v_k(n)) \quad (1.26)$$

6.3. Características del algoritmo de retropropagación.

Si observamos con detenimiento las ecuaciones anteriores observamos que la solución del aprendizaje se puede resolver mediante dos acciones fundamentales que son: la propagación desde la salida hasta la entrada de los

errores cometidos por los diversos niveles de células (δ), y la variación precisa de los pesos responsables (Δw). Estas acciones pueden ser realizadas sucesivamente en un proceso iterativo que ha dado lugar a un algoritmo sencillo para el entrenamiento de la red, más o menos rápido y con una cierta eficacia. En una generalización de los algoritmos de Minimización de Errores Cuadráticos, se puede conseguir una formulación recursiva adecuada del entrenamiento en múltiples pasos de cálculo.

También es evidente la necesidad de obtención de la derivada de la función de activación (f'), con lo cual nos obliga a que la función de activación sea continua, como lo es por ejemplo la función sigmoide.

Vamos a ver en el siguiente recuadro la descripción de cómo funciona el algoritmo:

1. **INICIALIZACIÓN:** En esta fase se cogen todos los pesos de las interconexiones de la red y se inicializan a un valor aleatorio.
2. **CÁLCULO ASCENDENTE (fase *forward*):** para cada una de los patrones o muestras de aprendizaje se calcula de forma ascendente (desde la entrada hasta la salida) las salidas de todas las neuronas hasta obtener las salidas de las neuronas de la última capa. Recordemos que esta obtención se realiza.
3.
$$y_j = f\left(\sum_i w_{ji} x_i\right)$$
4. **CÁLCULO DEL ERROR:** Una vez que tenemos las salidas obtenidas por la red para todos los patrones de entrenamiento, ya estamos en disposición de calcular el error (E).
5. **CÁLCULO DESCENDENTE (fase *backward*):** En esta fase del algoritmo, vamos a calcular las δ de todas las neuronas para cada uno de los patrones de aprendizaje. Luego cogeremos un patrón, veremos que valor se obtiene a las salidas de la red y haciendo uso de las ecuaciones que hemos visto, vamos obteniendo desde las salidas hacia la entrada, todos y cada uno de los valores δ de

cada una de las neuronas. El proceso lo repetimos para cada uno de los patrones

6. **ACTUALIZACION DE LOS PESOS:** Con los valores y de cada una de las neuronas para cada uno de los patrones, y de los valores δ para cada una de las neuronas para cada uno de los patrones, no existe ninguna dificultad para calcular las Δw de todas las interconexiones, para posteriormente actualizar todos los pesos con tan solo hacer la siguiente suma: $w_{ji} = w_{ji} + \Delta w_{ji}$.

Una vez realizadas estas fases principales del algoritmo, se vuelve de nuevo al punto 2 y se repite el proceso con los nuevos pesos calculados. Únicamente esta iteración principal se parará, y por tanto se considerará que el aprendizaje ha finalizado, cuando el E que se calcula en el punto 3 es menor a un valor ε , lo suficientemente pequeño para que nos garantice que la red responde ante los patrones de entrenamiento, de forma muy parecida a lo que las salidas requeridas exigían.

ALGORITMO DE RETROPROPAGACION

Se puede extraer fácilmente una versión de entrenamiento para una función de transferencia sigmoïdal:

1. Se inicializan los pesos a valores aleatorios cerca del cero.
2. Se presentan entradas (X_M) y las salidas requeridas para ellas (t_M).
3. Se hallan las respuestas de la red Y_M .
4. Se adaptan los pesos:

$$w_{ji}(t+1) = w_{ji}(t) + \eta * \delta_j * y_i;$$

$$\delta_j = y_j((1-y_j)(t_j-y_j)) \text{ para un nodo de salida}$$

$$\delta_j = y_j(1-y_j) * \sum_k \delta_k * w_{kj} \text{ para un nodo oculto};$$

Siendo K el número de nudos subsiguientes conectados a dicho nodo oculto

5. Repetir desde el paso 2.

Hay que hacer, después de visto el algoritmo, una serie de observaciones:

- a. Es importante observar como el funcionamiento de la red puede resumirse en tres pasos fundamentales: Cálculo ascendente de las actividades de las neuronas, cálculo descendente de las δ , y actualización de los pesos. Existe otro paso fundamental que es el cálculo del error, pero en realidad solo sirve para determinar cuando finaliza el proceso iterativo principal. Es importante identificar los pasos fundamentales del algoritmo, por cuanto en la implementación de la arquitectura que lo implemente también diferenciaremos estas tres acciones fundamentales.
- b. **¿Cuándo se modifican los pesos?** Después de haber pasado todo el conjunto de aprendizaje, con todos los patrones, lo que normalmente es conocido por **epoch**. La razón de plantearnos esta pregunta es porque existen principalmente dos versiones del algoritmo de **Retropropagación**. La aquí incluida y demostrada, que recibe el nombre de **off-line** y una versión heurística, que realiza la modificación de los pesos de la red después del paso de cada patrón de aprendizaje, con lo cual el funcionamiento del algoritmo es un tanto diferente, ya que para cada muestra del conjunto de aprendizaje, el algoritmo calcula las actividades de todas las neuronas (cálculo ascendente), calcula todas las δ de todas las neuronas de la red (cálculo descendente) y actualiza de inmediato todos los pesos de las interconexiones, con lo cual cuando se procesa la siguiente muestra de aprendizaje, todos sus cálculos son realizados con los pesos ya actualizados fruto de la muestra anterior. Este heurístico suele dar mejores resultados de convergencia que el anterior. Recibe el nombre de **on-line**.
- c. **¿Cuándo termina el método iterativo?:** El final del método iterativo, también reconocido como el problema de la **convergencia**, es un problema importante que tiene que ser analizado con propiedad. Evidentemente, y tal como

hemos enunciado el problema del aprendizaje, teníamos que encontrar unos pesos para la red de forma que ante el conjunto de aprendizaje se cumpliera una determinada condición. Todo ello ha dado lugar a la validación del algoritmo que tenemos: cuando $E \ll \epsilon$. Es un criterio bastante válido que quizá pueda ser sustituido por otros criterios de índole parecida como es el observar que prácticamente no se produce variación en los pesos de la red, es decir que los Δw son muy pequeños.

- d. Sin embargo no conviene olvidar la razón última del aprendizaje de una red neuronal, que no es otro que conseguir una red que ante entradas que no han participado en el aprendizaje, sea capaz de clasificarlas de forma correcta (si la aplicación es la clasificación por ejemplo). Dicho de otro modo: *el aprendizaje de la red tiene que ser un método que permita la generalización y extracción de ciertas normas generales que le permita enfrentarse a casos nuevos (mejor dicho no utilizados en el entrenamiento)*. El gran peligro del aprendizaje de una red neuronal es que la red memorice el conjunto de entrenamiento, pero sea incapaz de responder correctamente (asociando, prediciendo, asociando) ante patrones no incluidos en el aprendizaje.

Para evitar dicho problema, se han desarrollado diferentes métodos para determinar el final del proceso iterativo (convergencia) que suelen estar basados en la inclusión de un nuevo conjunto de pares de valores de entradas y respuesta conocida que recibe el nombre de **conjunto de test**. La existencia de dicho conjunto permite lo que es conocido como **VALIDACIÓN CRUZADA**, que consiste en ir entrenando con el conjunto de aprendizaje, y cada x número de iteraciones (**epochs**) es evaluada la red con el conjunto de test, que para nada ha intervenido en el entrenamiento. Mientras en estas evaluaciones, un valor como por ejemplo la tasa de aciertos vaya aumentando, o incluso el ERROR CUADRÁTICO MEDIO de la red para el conjunto de test vaya disminuyendo, consideraremos que la red sigue aprendiendo correctamente. En el momento que en estas evaluaciones aleatorias observemos lo contrario, será un síntoma claro de que la red está memorizando y ya no está

aprendiendo correctamente. El aprendizaje tiene que ser interrumpido y finalizado en ese punto. La red, con la estructura que tiene y con el conjunto de datos para el entrenamiento que dispone ya no es capaz de aprender más.

7. Redes Neuronales de Resonancia Adaptativa".

Las redes de tipo **Artmap** son una clase de redes neuronales que implementan un aprendizaje supervisado y una posterior clasificación de vectores multidimensionales de entrada en una serie de categorías de salida.

Los sistemas de redes neuronales tradicionales (tipo perceptrón) se han utilizado en múltiples ocasiones en los sistemas de narices y lenguas electrónicas. Pero también se han constatado que tienen ciertas limitaciones, entre la que destaca la siguiente:

Cuando se entrena una red neuronal perceptrón con unos determinados vectores de entrada, que representan ciertas características del entorno y posteriormente este entorno va evolucionando con el tiempo, la red irá perdiendo precisión porque los pesos ya están fijados, es decir esta red no tiene **plasticidad**. Una posible solución es reentrenar la red para que aprenda con los nuevos vectores, pero la consecuencia directa es la pérdida de la información de los vectores antiguos, es decir la red no tiene **estabilidad**. Por lo tanto, existe un dilema de la estabilidad de una red frente a su plasticidad. Para solucionar el dilema hay que diseñar un sistema de aprendizaje que sea plástico o adaptativo en la respuesta a los sucesos significativos y que siga siendo estable frente a los sucesos irrelevantes. Es decir, preservar los conocimientos previos de interés mientras se continúa aprendiendo conocimientos nuevos.

En respuesta al dilema expuesto se desarrolló la Teoría de la Resonancia Adaptativa (ART), que fue introducida por S. Grossberg [*Grossberg, 1980*] como una teoría que intentaba emular la manera de procesar la información por el cerebro humano. Desde entonces, y con la colaboración de G. Carpenter, esta teoría ha evolucionado hacia una serie de algoritmos neuronales para el aprendizaje **no supervisado**.

7.1. Teoría de la Resonancia Adaptativa (ART).

La teoría de la resonancia adaptativa (ART) fue introducida como una teoría que intenta emular la manera de procesar la información por el cerebro humano.

Esta teoría ha evolucionado hacia una serie de algoritmos neuronales para el aprendizaje no supervisado. Estos algoritmos son capaces de crear clases estables ante la presentación de secuencias de entradas arbitrarias con un ritmo de aprendizaje rápido o lento. Dentro de estos algoritmos se puede destacar ART1 y ART2.

ART1 [Carpenter & Grossberg, 1987a] es un algoritmo que solo trabaja con patrones de **entrada binarios**. ART2 es un algoritmo que trabaja con patrones de **entrada analógicos** [Carpenter & Grossberg, 1987b]. También se establecieron otros algoritmos como ART3 [Carpenter & Grossberg, 1987c].

7.2. Arquitectura de la Red ART.

La red ART está formada básicamente por dos subsistemas, denominados de **atención** y **orientador**. El primero posee un aprendizaje competitivo que se encarga de reaccionar ante nuevas entradas y de aprenderlas en caso necesario. El subsistema orientador se encarga de distinguir entre entradas relevantes e irrelevantes, avisa al sistema de atención de cuando hay que aprender. Además existe un término de control que consiste en un sistema que lleva el control de los módulos que deben actuar y gobierna las señales entre los distintos módulos.

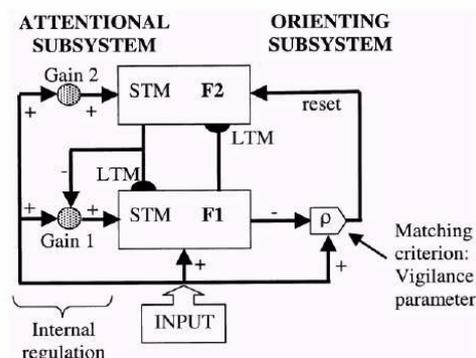


Figura 1-19. Arquitectura de una red ART

La arquitectura básica de la red ART1 se muestra en la Fig. 1-19. El sistema de **atención** está formado por dos capas de neuronas F1 y F2, formadas cada una de ellas por elementos de memoria de corto plazo (STM – *Short Term Memory*), que contiene los valores de activación de las neuronas. Entre ambas capas existen conexiones en un sentido (ascendente) como en otro (descendente). Estas conexiones se denominan filtros adaptativos y contienen las denominadas memorias de largo plazo (LTM – *Long Term Memory*), que determinan los pesos de las conexiones y estos suelen permanecer durante un largo tiempo. El subsistema de **orientación** genera una señal de reset a F2 cuando existe una diferencia grande entre los patrones ascendentes y descendentes en F1.

7.3. Fuzzy ART.

Fuzzy ART es una evolución del algoritmo ART1 en el que se utiliza la teoría de lógica difusa [Carpenter et al., 1991]. Para ello, se generaliza la función a vectores de entrada con valores analógicos comprendidos en el rango entre 0 y 1. Este cambio, con la ayuda de la codificación complementaria (“complement coding”), que preserva la información de amplitud a la vez que normaliza los vectores de entrada, permite implementar un algoritmo de clasificación no supervisada de gran rapidez de aprendizaje.

Su modo de funcionamiento es simple: cada vez que la red recibe un nuevo vector de entrada reacciona activando uno y solo uno de los nodos de salida. Cada uno de estos nodos representa una de las diferentes clases que se han creado con las entradas anteriores. En caso de que la medida no se parezca lo suficiente a ninguno de los nodos ya asignados se crea un nuevo nodo que representa una nueva clase cuyo primer miembro será este vector. El parámetro que determina si una nueva medida pertenece a una clase ya existente o hay que crear una clase nueva es el **parámetro de vigilancia** (ρ).

7.4. ARTMAP.

Las redes de tipo ARTMAP son una clase de redes neuronales formadas por estructuras de tipo ART que implementan un aprendizaje supervisado y una posterior clasificación de vectores multidimensionales de entrada en una serie de categorías de salida [*Carpenter et al., 1991b*].

7.5. FuzzyArtmap.

La red Fuzzy Artmap [*Carpenter et al., 1992*] procede de la red Artmap con las mismas transformaciones que permiten definir la red Fuzzy ART a partir de la red ART1. Es decir, la red Fuzzy Artmap es una generalización a vectores analógicos (con componentes comprendidas entre cero y uno) de la red binaria Artmap.

La red Fuzzy Artmap presenta múltiples ventajas que la hacen muy interesante para las aplicaciones con narices electrónicas [*Llobet e al., 1999b*]. De entre todas ellas cabe destacar las siguientes:

- **Aprendizaje rápido.** Las medidas que se presentan en entrenamiento necesitan muy poca carga computacional lo que permite programar el algoritmo en dispositivos programables de bajo coste y aplicar validaciones cruzadas de orden 1 y probar con diferentes combinaciones de parámetros.
- **Aprendizaje con un conjunto reducido de medidas de entrenamiento.** La red presenta una habilidad particular para aprender rápidamente eventos singulares, que aparecen muy pocas veces en el conjunto de entrenamiento. Por lo tanto, en dicho conjunto no es necesario que haya el mismo número de medidas de cada clase para que la red funcione correctamente.
- **Aprendizaje continuo** de nuevas características sin olvidar lo aprendido con anterioridad, algo muy útil para adaptarse a posibles derivas producidas por sensores.

En comparación con otros tipos de redes neuronales, la red Fuzzy Artmap determina automáticamente las neuronas de su capa oculta. Además,

maximiza el poder de generalización aprendiendo al 100% el conjunto de entrenamiento.

Una vez entrenada, es posible extraer reglas de clasificación a partir de los pesos obtenidos tras el periodo de entrenamiento, lo que puede dar luz sobre los procesos internos y cómo influyen en la categorización de resultados.

Sin embargo, la implementación práctica de la red Fuzzy Artmap presenta un problema que debe ser tratado con sumo cuidado. Tal y como se ha comentado con anterioridad, la red aprende el 100% de las medidas del conjunto de entrenamiento sacrificando lo mínimo posible la generalización. Sin embargo, la presencia de medidas erróneas o “*outlier*” en el conjunto de medidas para el entrenamiento puede requerir un incremento del valor de vigilancia excesivo, lo cual perjudicará seriamente la capacidad de generalizar de la red.

Resumiendo, la red Fuzzy Artmap es una red de clasificación con aprendizaje supervisado. En una fase de entrenamiento la red necesita que se le suministre un conjunto de medidas. Cada medida debe contener un vector de entrada, que detalla los parámetros medidos en cada experiencia, y un vector de salida que codifica la categoría que se le debe asignar. Posteriormente, en la fase de evaluación sólo se suministra el vector de entrada y la red clasifica dicha medida siguiendo los criterios que ha aprendido en la fase de entrenamiento.

Las propiedades de esta red han hecho que sea ampliamente utilizada en distintas aplicaciones en distintos campos de la investigación y de la industria.

Las aplicaciones más numerosas se encuentran en narices y lenguas electrónicas: narices relacionadas con alimentación [Brezmes et al., 2005] [Gil et al., 2010] [Vinaixa et al., 2005] [Amari et al., 2006], narices relacionadas con sistemas químicos [Cristhian et al., 2011] [Kim et al., 2007] [Llobet et al., 2001, 2002, 2004], lenguas relacionadas con alimentación [Martinez et al., 2005] [García et al., 2011] [Barat et al., 2008a] [Gil et al., 2006, 2008, 2011], lenguas relacionadas con agentes nerviosos [Campos et al., 2010a]. También encuentra

aplicación en biomedicina [Jeyabalan et al., 2008] [Bortolozzi et al., 1996][Downs et al., 1996], bioingeniería [Lerner & Vigdor, 2004] [S. Mohamed et al., 2006], comunicaciones [Raahemi et al., 2008] [Chralampidis et al., 2001], energía eléctrica e industria [Qasim et al., 2008] [Nafisi et al., 2009] [Demetgul et al., 2001] [Lopes et al., 2005], microelectronica [Serrano et al., 1998] [Sumithradevi et al., 2007], tratamiento de imagenes [Zhe, 2008] [Salah &Trinder, 2001],química [Yaffe et al., 2002] [Xu et al., 2009], materiales [Ramuhalli et al., 1999], seguridad [Vilakazi & Marwala, 2006], seguros [Nachev, 2007] o meteorología [Subhajini & Santhanam, 2011].

La aplicación de estas redes es complicada, por lo que diversos autores han presentado modelos adaptados de estas redes, como la ARTMAP-IC, Gaussian ARTMAP [Williamson, 1996], ART-EMAP y ART-EMAPQ [Carpenter & Ross, 1995], Fuzzy ARTVar [Dagher et al., 1998a], Ordered Fuzzy Artmap [Dagher et al., 1998b], PSFAM [Jervis et al., 1999] , IPSFAM [Jervis et al., 2004], arboART [Ishihara e al., 1995] , CFART[Hung et al.,1996], Fuzzy HART-S [Bartfai & White, 1997], LAPART [Han et al., 1994], MART [Fernandez & Barro 1990] [Andonie & Cataron, 2008] [Bartfai, 1995] [Lin & Soo, 1997] [Ramuhalli, et al., 2000].

En 1993, Kasuba [Kasuba, 1993] desarrolla una versión simplificada de la Fuzzy Artmap (SFAM). Dicha versión es la que se aplica en la presente tesis. En 2003, Rajasekaran [Rajasekaran & Vijayalakshmi, 2005] explica en su libro el algoritmo SFAM basado en trabajo de Kasuba.

8. Redes Fuzzy ART.

8.1. Introducción.

Fuzzy Art [Carpenter et al., 1991a] es una evolución del algoritmo ART1, que permite clasificar de forma estable entradas arbitrarias binarias. Las aplicaciones principales son el reconocimiento de letras y figuras con patrones en blanco y negro. Fuzzy Art, generaliza esta función a vectores de entrada analógicos con coordenadas comprendidas entre 0 y 1. Para ello substituye los operadores intersección y unión de ART1 por los operadores MIN (\wedge) y MAX

(V), respectivamente de la teoría de lógica difusa [Carpenter et al., 1992]. Este cambio, con la ayuda de la codificación complementaria (“complement coding”), que preserva la información de amplitud a la vez que normaliza los vectores de entrada, permite implementar un algoritmo de clasificación no supervisada de gran rapidez de aprendizaje. En la Fig. 1-20 se muestra un esquema del algoritmo.

Su modo de funcionamiento es simple: cada vez que la red recibe un nuevo patrón de entrada V se hace resonar con los prototipos de las categorías conocidas por la red, reacciona activando una de las clases de salida y el centro del cluster es desplazado ligeramente para adaptarse mejor al nuevo patrón que le ha sido asignado.

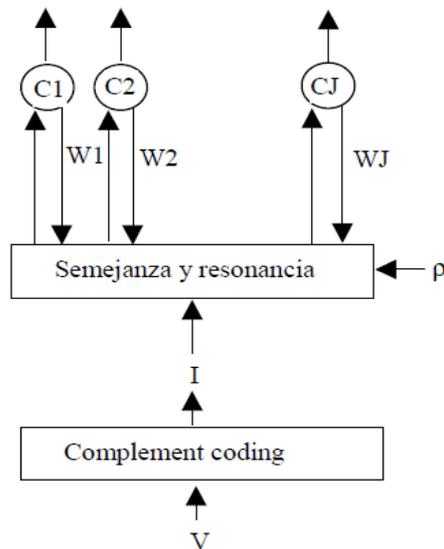


Figura 1-20. Esquema de la red Fuzzy Art

Las diferentes clases de salida han sido creadas a partir de las entradas anteriores. En caso de que la medida no se parezca lo suficiente a ninguno de las clases ya asignadas pueden suceder dos cosas:

- si la red posee una **capa de salida estática** entrará en saturación pues no puede crear una nueva clase para el patrón presentado pero tampoco puede asignarlo a una clase existente,
- si la red posee una **capa de salida dinámica** se creará una nueva clase para dicho patrón, esto no afectará a las clases ya existentes.

El algoritmo cuenta con dos parámetros que controlan su funcionamiento.

- El **parámetro de vigilancia** ρ : que determina lo riguroso que debe ser el algoritmo a la hora de agrupar medidas. Un parámetro de vigilancia muy cercano a la unidad implica una clasificación muy exigente, de manera que dos medidas deben ser muy parecidas para ser agrupadas en una misma clase. Por el contrario, un parámetro cercano a cero permite la agrupación de medidas poco parecidas, lo que, como resultado, genera una red con pocos vectores de salida, ya que el número de clases diferentes es reducido.
- El **parámetro** β : que controla el ritmo de aprendizaje, siendo su valor igual a la unidad para un aprendizaje rápido e igual a cero en caso de que no se deba aprender más.

8.2. Descripción del Algoritmo.

- Vector de entrada: Cada uno de los patrones de entrada V es un vector N -dimensional donde cada una de sus componentes tiene coordenadas incluidas en el intervalo $[0,1]$.
- Codificación complementaria: A partir del patrón de entrada V , se crea un nuevo vector normalizado I de dimensión $2N$ en el que la componente:

$$I_{J+N} = 1 - I \quad (1.27)$$

- Vector de pesos del nodo de salida j (categoría j): W . Inicialmente:

$$W_{j1} = W_{j2} = W_{j2M} = 1 \quad (1.28)$$

- Velocidad de aprendizaje: β entre $[0, 1]$. Aprendizaje rápido $\beta=1$; Aprendizaje lento $\beta \ll 1$; Sin aprendizaje $\beta=0$
- Parámetro de vigilancia: ρ entre $[0, 1]$, cercano a cero implica menos categorías al agrupar con criterios de semejanza poco exigentes, cercano a uno implica muchas clases, cada una con pocos miembros pero muy parecidos entre si.

- Parámetro de selección $\alpha > 0$. Debe ser muy cercano a cero. Sirve para deshacer igualdades. Un valor típico es 0,001.
- Selección de categoría: Para cada patrón de entrada V y cada categoría j se calcula la función de semejanza $T_j(V)$ como indica la ecuación siguiente:

$$T_j = \frac{|I \wedge W_j|}{\alpha + |W_j|} \quad (1.29)$$

Donde el operador AND (\wedge) en lógica difusa, se define como:

$$A \wedge B = \min(A, B) \quad (1.30)$$

Y la norma $| \cdot |$ se define como:

$$|I| = \sum_{i=1}^{2N} I_i \quad (1.31)$$

A partir de aquí inicialmente se escoge la categoría j para la que $T_j(V)$ es máximo.

- Resonancia o reset: Se dice que aparece resonancia si se cumple la desigualdad:

$$\rho \leq \frac{|I \wedge W_j|}{|I_j|} \quad (1.32)$$

En ese caso, se activa la categoría j como respuesta al valor de entrada V , lo que quiere decir que la red clasifica la entrada V como de clase j . Además, se ejecuta el proceso de actualización de los pesos de dicha categoría.

En el caso de que no se cumpla la desigualdad se produce un reset: El sistema desactiva temporalmente la categoría j y vuelve a escoger una categoría siguiendo el criterio de máxima semejanza (Ec. 1.29). Si ninguna categoría “resuena”, se crea un nuevo nodo para el patrón de entrada V .

- Aprendizaje: Una vez activada la categoría j debido a la entrada V , sus pesos son actualizados según la ecuación siguiente:

$$W_j^{NUEVO} = \beta(I \wedge W_j^{ANTERIOR}) + (1 - \beta)W_j^{ANTERIOR} \quad (1.33)$$

Si se quiere un aprendizaje rápido, se utiliza $\beta=1$, un aprendizaje nulo $\beta=0$. En general, para medidas ruidosas no interesa poner $\beta = 1$. Sin embargo, cuando el número de medidas es bajo y se requiere de un aprendizaje estable se puede demostrar que eso se consigue con $\beta=1$.

En la tabla 1.2, se muestra una comparación entre las redes ART1 y Fuzzy ART.

Tabla 1.2. Comparación entre las redes ART1 y Fuzzy ART.	
ART1 (Binaria)	Fuzzy ART (analógica)
Elección de categoría	
$T_j = \frac{ I \cap W_j }{\alpha + W_j }$	$T_j = \frac{ I \wedge W_j }{\alpha + W_j }$
Criterio de coincidencia	
$\rho \leq \frac{ I \cap W_j }{ I_j }$	$\rho \leq \frac{ I \wedge W_j }{ I_j }$
Aprendizaje rápido	
$W_j^{NUEVO} = I \cap W_j^{ANTERIOR}$	$W_j^{NUEVO} = I \wedge W_j^{ANTERIOR}$

9. Redes Fuzzy Artmap.

9.1. Introducción.

Las redes de tipo Artmap son redes neuronales de aprendizaje supervisado que permiten obtener una clasificación de vectores multidimensionales de entrada en una serie de categorías de salida en respuesta a una serie de vectores de entrada analógicos presentados en orden aleatorio.

La red Fuzzy Artmap [Carpenter et al., 1992] proviene de la red Artmap [Carpenter et al., 1991b] con las mismas transformaciones que permiten definir la red Fuzzy Art a partir de la red ART1. En definitiva, la red Fuzzy Artmap es

una generalización a vectores analógicos (con componentes comprendidas entre cero y uno) de la red binaria Artmap, que ha demostrado ser capaz de enfrentarse a muy distintos problemas de clasificación, por ejemplo: se ha utilizado para clasificar una base de datos muy dispersa de números manuscritos [Martínez, 1999]. Diversos autores que han profundizado en sus características [Georgiopoulos et al., 1996], propuesto mejoras de tipo estadístico [Lim et al., 1997], o de programación distribuida [Carpenter, 1997].

De las múltiples ventajas que presenta la red Fuzzy Artmap, destacaremos las más interesantes para las aplicaciones con narices electrónicas [Gardner et al., 1996],[Llobet et al., 1999].

- Aprendizaje rápido a partir de las medidas que se presentan en entrenamiento, lo que permite una fácil programación del algoritmo en dispositivos programables de bajo coste, a la vez que probar con diferentes combinaciones de parámetros.
- El aprendizaje se puede realizar con un conjunto reducido de medidas de entrenamiento, lo que resulta muy interesante en cualquier aplicación experimental en la que sea costoso la obtención de conjuntos de medida extensos. La red aprende rápidamente eventos singulares que aparecen muy pocas veces en el conjunto de entrenamiento. Por lo tanto, en dicho conjunto no es necesario que haya el mismo número de medidas de cada clase para que funcione correctamente.
- Aprendizaje continuo de nuevas características sin olvidar lo aprendido con anterioridad, algo muy útil para adaptarse a derivas producidas en los sensores.
- En comparación con otros tipos de redes neuronales, Fuzzy Artmap determina automáticamente las neuronas de su capa oculta. Además maximiza el poder de generalización aprendiendo al 100% el conjunto de entrenamiento.
- Una vez entrenada, es posible extraer reglas de clasificación a partir de los pesos obtenidos tras el periodo de entrenamiento, lo que puede dar luz sobre los procesos internos y cómo influyen en la categorización de

resultados.

Resumiendo, la red Fuzzy Artmap es una red de clasificación con aprendizaje supervisado. En una fase de entrenamiento la red necesita que se le suministre un conjunto de medidas. Cada medida debe contener un vector de entrada, que detalla los parámetros medidos en cada experiencia, y un vector de salida que codifica la categoría que se le debe asignar. Posteriormente, en la fase de evaluación solo se suministra el vector de entrada y la red clasifica dicha medida siguiendo los criterios que ha aprendido en la fase de entrenamiento.

9.2. Algoritmo de la red Fuzzy Artmap.

Básicamente, una red Fuzzy Artmap está formada por dos redes Fuzzy ART conectadas entre sí por un vector de relaciones denominado "mapfield". A una de las dos redes (la que denominaremos ARTa) se encarga de procesar los vectores de entrada (V). A la red ARTb le llegan, en la fase de entrenamiento, los vectores que codifican la categoría correcta de cada medida del conjunto de entrenamiento (C). Si las clases de salida son conocidas, la red Fuzzy ARTb se reduce a mantener un mapa de las clases posibles. Es capaz de rechazar la categoría propuesta por el módulo Fuzzy ARTa en caso de que no coincidan. En la **Fig. 1-21** se muestra un esquema de la red Fuzzy Artmap.

Inicialmente, en la red ARTa el vector de vigilancia es cero. En la red ARTb se suele dar un valor igual a la unidad, ya que medidas que deban ser clasificadas conjuntamente enviarán a la red ARTb codificaciones idénticas. Además, cualquier vector de codificación diferente, por parecido que sea al resto, debe ser detectado y debe activar una neurona de salida diferente en la red ARTb.

Cada vez que se suministra una medida de entrenamiento, la red ARTa activa un nodo y la red ARTb activa otro. El mapa que las une aprende a relacionar nodos activados. De esta forma, a cada nuevo nodo que se activa en la red ARTa se le asocia un nodo en ARTb. Cabe destacar que los nodos ARTb

normalmente serán imagen de varios nodos ARTa (cada categoría contiene varias medidas), mientras que cada una de las categorías creadas en ARTa sólo tendrá una imagen en ARTb (cada medida solo puede pertenecer a una categoría).

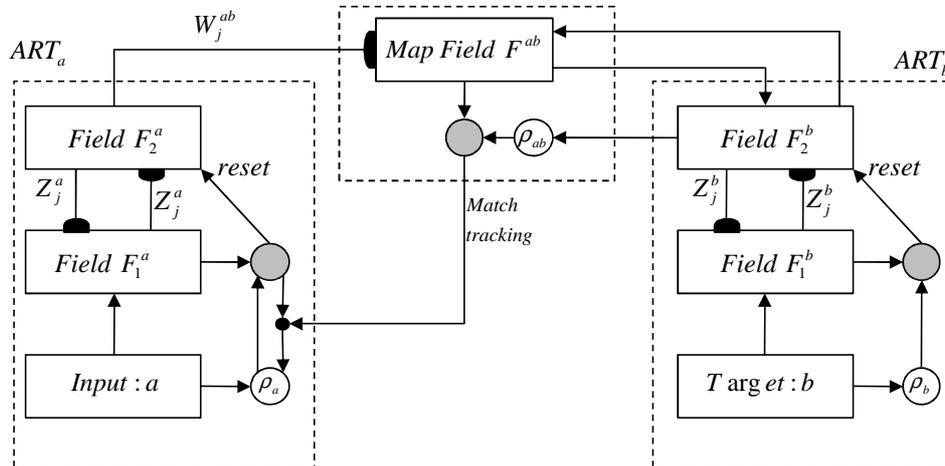


Figura 1-31. Esquema general de una red Fuzzy Artmap.

Cuando una nueva medida activa un nodo en ARTa ya existente, se comprueba si la imagen de ese nodo asignada por el mapfield coincide con el nodo que se ha activado en ARTb paralelamente. En el caso de que no coincidan, se incrementa el valor del parámetro de vigilancia hasta que la neurona que se active en ARTa tenga por imagen la neurona activada en ARTb. Si no se encuentra ninguna se creará una nueva y el mapa le asignará como imagen el nodo ARTb activado.

En definitiva, el valor de vigilancia en ARTa solo se incrementa lo estrictamente necesario para que la red ARTb separe en nodos diferentes las medidas que deben estar clasificadas en diferentes categorías. Suponiendo un parámetro de aprendizaje igual a la unidad para ambas redes se puede demostrar que este algoritmo aprende a clasificar correctamente el 100% de los vectores de entrenamiento. Además, ese aprendizaje es rápido y estable. A continuación se detalla el algoritmo de forma esquemática:

- ARTa: Red Fuzzy Art a la que llegan los vectores de entrada.

- ARTb: Red Fuzzy Art a la que llegan los vectores de salida..
- Mapfield: Módulo del mapa que relaciona los nodos de salida de ARTa con los nodos de salida de ARTb.
- Entradas: Supondremos que el vector de entrada de cada medida es V y el vector que codifica su clasificación correcta C .
- Normalización: Los vectores V y C se normalizan con codificación complementaria, pasando a ser los vectores I y D respectivamente.
- Pesos: El vector de pesos de la categoría k de la red ARTa se denominará W_{ak} . El vector de pesos de la categoría j de la red ARTb se denominará W_{bj} . El mapfield tiene un solo vector de longitud igual al número de nodos activados en ARTa. La componente k-ésima del vector indica el nodo imagen en ARTb del nodo k de la red ARTa
- Match tracking: Originalmente, el parámetro de vigilancia de ARTa es un valor base. Si una medida a activa un nodo J en ARTa cuya imagen a través del mapfield no coincide con la activación producida en ARTb, entonces el valor de vigilancia en ARTa se incrementa según la Ec. 1.33, lo que forzará la activación de un nodo diferente en ARTa.

$$\rho_a = \frac{|I \wedge W_j^E|}{|I_j|} \quad (1.34)$$

- Modo de evaluación: La red ARTb se desactiva. La red ARTa recibe un vector que hace que se active su nodo k . La salida de la red es el valor de la componente k-ésima del vector de mapfield.

9.3. Fuzzy Artmap Simplificada (“*Simplified Fuzzy ArtMap*” SFAM).

A pesar de las numerosas aplicaciones de la red Fuzzy Artmap su algoritmo puede resultar complejo. También puede presentar dificultades en aplicaciones donde exista una restricción de memoria. En la mayoría de las aplicaciones citadas el algoritmo se implementa en un PC. La memoria utilizada en un PC suele ser lo suficientemente grande para que el algoritmo trabaje correctamente. El problema surge cuando se quiere utilizar la Fuzzy Artmap en

sistemas portátiles donde se suelen emplear microcontroladores de bajo coste dotados de una memoria limitada. En este tipo de sistema se buscan algoritmos que ocupen el menor espacio de memoria posible.

En 2003, Aaron Garrett, [Garret, 2003] (Jacksonville State University) desarrolla un toolbox para MATLAB basado en dicha SFAM. Este toolbox es la base de la GUI desarrollada en la presente tesis.

La red SFAM va un paso por delante de Fuzzy Artmap en la reducción de la sobrecarga computacional y en la redundancia de la arquitectura de Fuzzy. El modelo emplea ecuaciones simples de aprendizaje con un único parámetro seleccionable por el usuario y puede aprender cada patrón de entrenamiento individual con un pequeño número de iteraciones de entrenamiento. SFAM es más rápido que Fuzzy Artmap y más fácil de programar. La Fig. 1-22 muestra la arquitectura de la red SFAM.

Los vectores de entrada a , con un tamaño de d rasgos, son introducidos en el “*Complement Code*” donde el vector de entrada es ampliado al doble de su tamaño por la adición de sus complementos. El vector de entrada complementado, llamado la y con un tamaño $2d$, se introduce en la siguiente capa, llamada “*Input Layer*”. Los pesos W_j de cada una de los nodos o neuronas de las categorías de salida O_N o subclases se relacionan con los vectores de la “*Input Layer*”. Por esta circunstancia se les denomina “*Top-Down Weights*.” Aaron Garret denomina a la “*Output Category Layer*” como “*mapfield*” por su similar función al “*mapfield*” de la FAM. La llamada Category Layer CM contiene los nombres o etiquetas de las M categorías o clases que la red tiene que aprender para cada uno de los vectores de entrada. “*Vigilance parameter*”, “*Match tracking*” y “*Reset*” son mecanismos de la red que se emplean durante el entrenamiento.

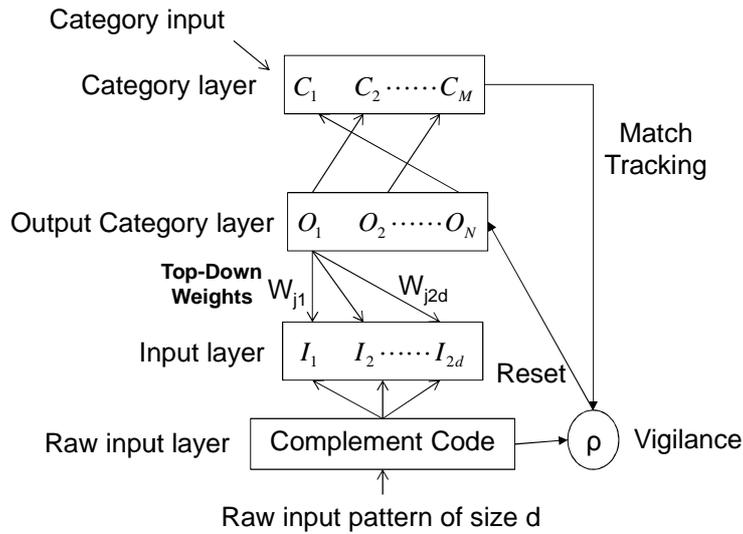


Figura 1-22. Arquitectura de la red neuronal Fuzzy Artmap Simplificada (SFAM).

La red SFAM es muy sensible a las magnitudes absolutas y fluctuaciones en las entradas pudiendo provocar un malfuncionamiento de la red. Por ello es necesario normalizar las entradas dentro de un mismo rango de valores. Por otra parte todos los valores de entrada a deben estar en el rango $[0,1]$. La complementación de la señal es una normalización que preserva la amplitud de la información, es decir representa la presencia de un rasgo particular en el vector de entrada y su ausencia. Si es vector de entrada a está formado por d rasgos, el vector complemento \bar{a} representa la ausencia de cada rasgo. Así el vector de entrada I^a será la entrada y el complemento (Ec. 1.35) de la entrada.

$$I^a = (a, a^c) = (a_1, \dots, a_d, a_1^c, \dots, a_d^c) \quad (1.35)$$

Donde $a_i^c = (1 - a_i)$

Un efecto interesante de la codificación del complemento es la normalización rápida del vector de entrada.(Ec. 1.35)

$$\|I^a\| = \|a, a^c\| = \sum_{i=1}^d a_i + \left(d - \sum_{i=1}^d a_i \right) = d \quad (1.36)$$

donde la norma $\|\cdot\|$ se define mediante la ecuación.

$$|p| \equiv \sum_{i=1}^M |p_i| \quad (1.37)$$

La red emplea un mecanismo para activar la capa de salida en respuesta a una entrada a la red. Cuando se presenta a SFAM un patrón de entrada cuyo complemento codificado se representa por I^a , todos los nodos de salida se activan en algún grado. Esta activación de salida se denota como $T_j(I^a)$ del nodo de salida j y W_j sus pesos. La función para producir esta activación se define por la Ec.(1.38).

$$T_j(I^a) = \frac{|I_j^a W_j|}{\alpha + |W_j|} \quad (1.38)$$

Donde $w_j=(W_{j1}, W_{j2}, \dots, W_{jd})$, α es llamado el parámetro de polarización (α se mantiene como un valor pequeño, cercano a 0). Al aumentar el valor de α aumentará el número de subclases. Y donde la norma $|\cdot|$ se define por la Ec.(1.37)

$$(p \wedge q)_i \equiv \min(p_i, q_i) \quad (1.39)$$

El nodo de salida de la categoría q que gana es el nodo con la mayor activación de la función de T_j . Si hay más de un T_j máximo, se elige arbitrariamente el nodo de salida j con el menor índice. La elección de la categoría está indexada por J , Ec.(1.40).

$$(1.40)$$

La función de coincidencia se utiliza para comparar las características del vector de entrada y los pesos de un nodo de salida particular para ayudar a determinar si el aprendizaje debe ocurrir. La función de coincidencia se define por la Ec.(1.41).

$$(1.41)$$

Y teniendo en cuenta la Ec. (1.36), la función de coincidencia también se define mediante la Ec. (1.42)

(1.42)

Cuando se utiliza junto con el parámetro de vigilancia ρ , el valor de la función de coincidencia determina si la corriente de entrada es suficiente para igualar a un nodo de salida en particular (O_j), para ser codificada por dicho nodo o si debe formarse un nuevo nodo de salida para codificar el patrón de entrada. Si el valor de la función de coincidencia es mayor que el parámetro de vigilancia, la red se dice que está en un estado de resonancia (Ec. 1.43).

(1.43)

Si el nodo de salida de la categoría ganadora coincide con la categoría de entrada ($O_j=C_i$), se debe actualizar vector de pesos W_j de acuerdo con la Ec. (1.44).

$$W_j^{new} = \beta (I \wedge W_j^{new}) + (1 - \beta)W_j^{old} \quad (1.44)$$

donde el parámetro de aprendizaje β (entre 0 y 1) determina la velocidad de aprendizaje de la red, los valores más elevados de β dan lugar a una velocidad de aprendizaje alta (modo rápido de aprendizaje), mientras que los valores bajos hacen que la velocidad de aprendizaje sea baja (modo lento de aprendizaje). Además, contribuye a la robustez del algoritmo de clasificación, especialmente cuando se trata de clasificar datos que pueden tener algo de ruido en sus valores (mediante el modo lento de aprendizaje).

Si el valor del parámetro de vigilancia es mayor que la función de coincidencia la red se encuentra en el estado llamado "*Mismatch Reset*". Este estado indica que el nodo de categoría de salida no se ajusta lo suficiente a la categoría de entrada ($O_j \neq C_i$).

Se debe aumentar temporalmente el parámetro de vigilancia ρ , de modo que se viole el estado de la Ec.(1.44). Establecer ρ por la Ec.(1.45).

(1.45)

Donde ε es un número positivo pequeño, es decir $\varepsilon \approx 0,001$. Un valor grande de ε puede disparar una alarma de falta de coincidencia de datos falsos, haciendo ir más allá el valor de ρ . En los casos en que existen algunos desajustes reales, un mayor valor de ε aumentará el error.

El algoritmo SFAM para el entrenamiento (fase supervisada) puede ser definido por el flujograma de la **Fig. 1-23**.

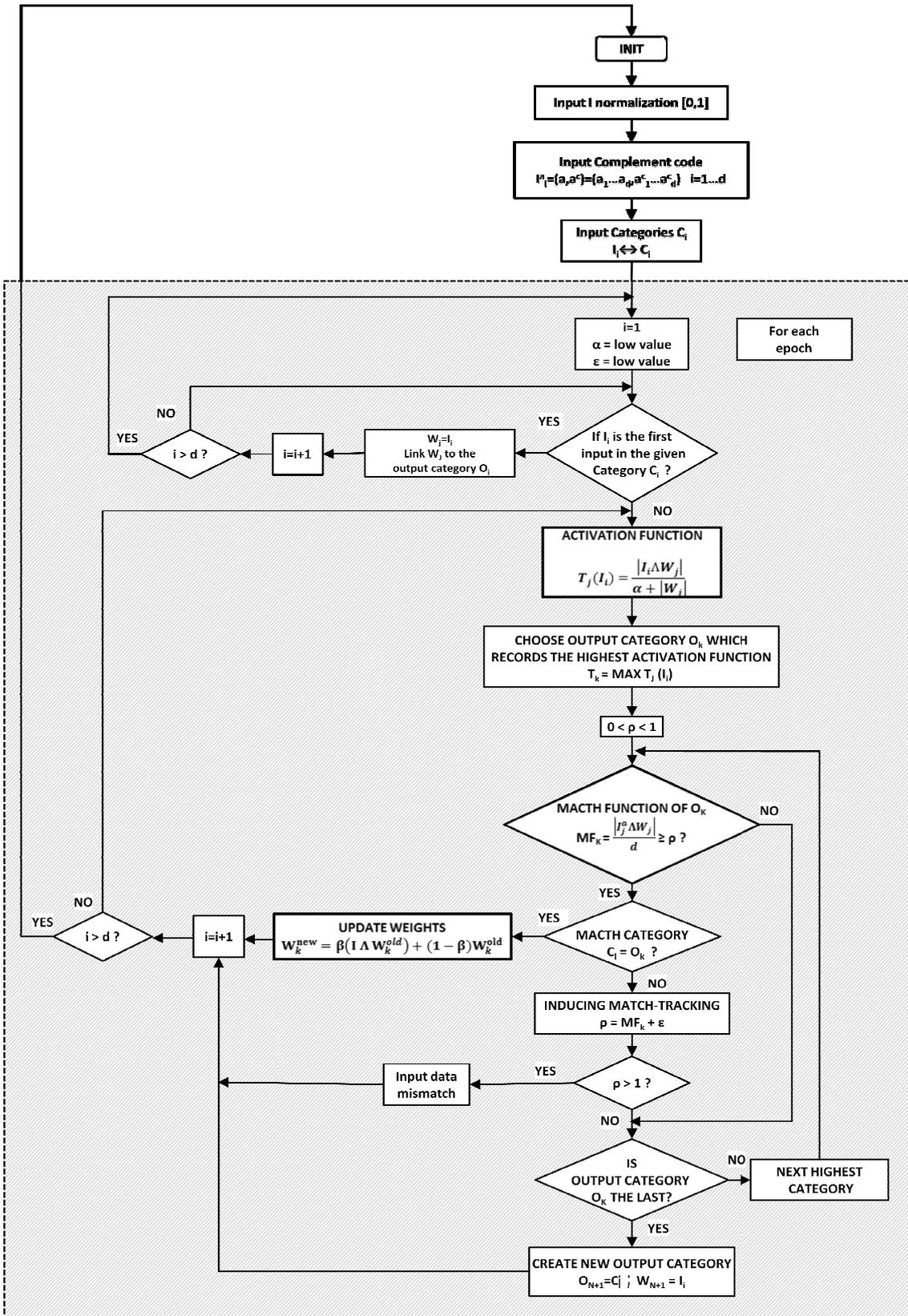


Figura 1-23. Algoritmo SFAM para el entrenamiento

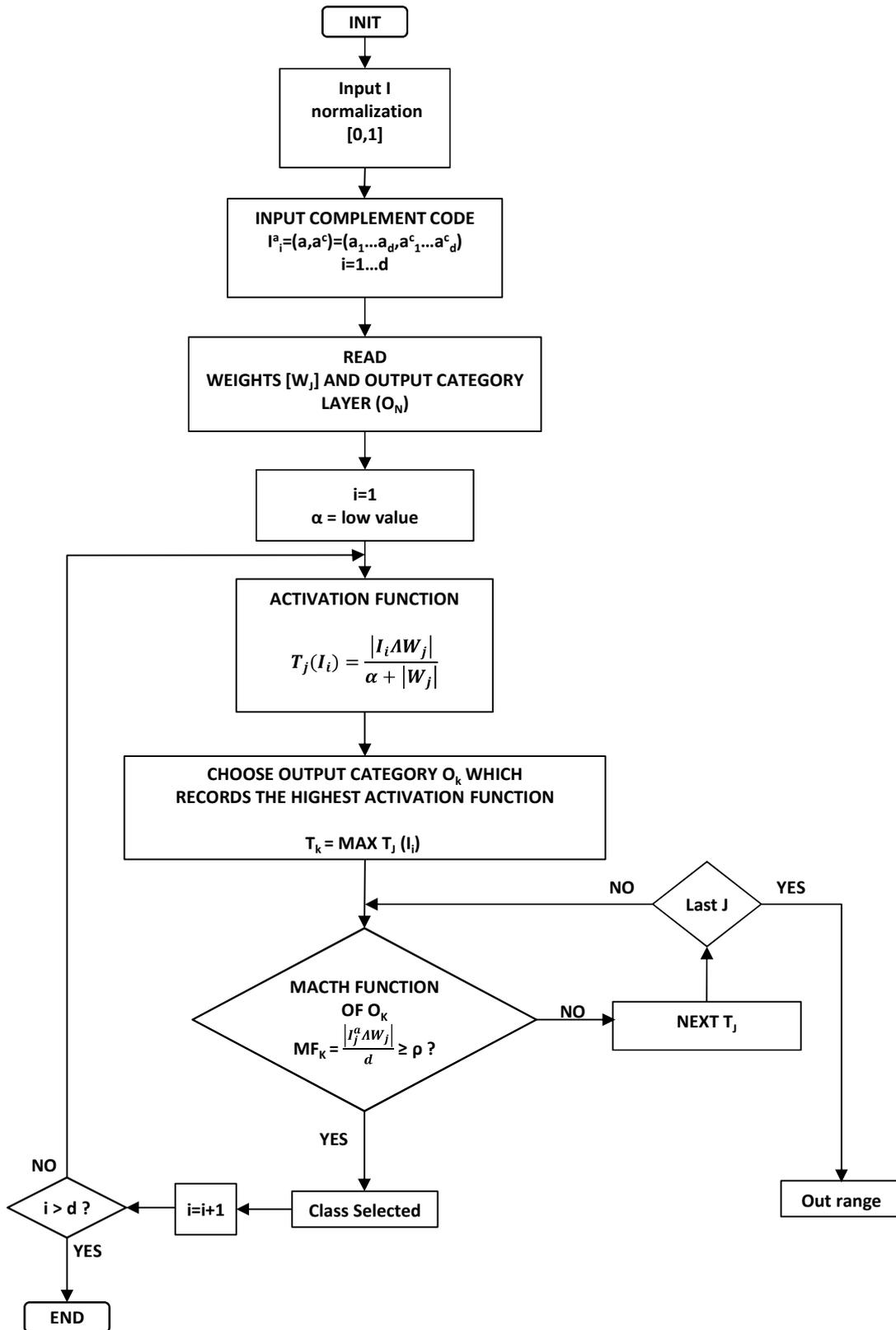


Figura 1-24. Algoritmo de la red SFAM para la fase no supervisada.

El algoritmo para la fase no supervisada (**Fig. 1-24**) es más sencillo puesto que se limita a determinar la clase del vector de entrada.

La elección de la categoría está indexada por k . Este índice k apunta a la subclase en la capa de salida o categoría de mapfield. Si la función de concordancia es menor que el criterio de vigilancia se selecciona una función menor y la resonancia se comprueba de nuevo. Finalmente, si no es posible elegir una clase cuya función de coincidencia sea mayor que el criterio de vigilancia, el vector de entrada se clasifica como fuera del intervalo. Si hay resonancia entonces el vector de entrada está clasificado.

En 2003, Vakil-Baghmisheh et al. [*Vakil & Pavešić, 2003*] publican un modelo de SFAM distinto del modelo de Kasuba, señalando algunas críticas al modelo de Kasuba. A partir de la SFAM original se han presentado distintas versiones, por ejemplo la Probabilistic Simplified Fuzzy Artmap (PSFAM) [*Jervis et al., 1999*]. Otra versión mejorada es la presentada por Vuskovic [*Vuskovic & Sijiang, 2002*] en 2002 con la llamada SFAM basada en la distancia de Mahalanobis.

Se pueden encontrar aplicaciones de la red SFAM en procesado e imágenes [*Zhe, 2008*], [*Salah & Trinder, 2001*], reconocimiento facial [*Xu et al., 2009*] [*Yaffe et al., 2002*], [*Ramuhalli et al., 1999*], energía [*Nachev, 2007*], [*Vilakazi & Marwala, 2006*], biomedicina [*Subhajini & Santhanam, 2011*], [*Rajasekaran & Vijayalakshmi, 2000a*], [*Rajasekaran & Vijayalakshmi, 2000b*] y lenguas electrónicas [*Vinaixa et al., 2005*].

10. Análisis Característica Operativa del Receptor (ROC).

La aplicación más general de las redes neuronales artificiales es la clasificación de muestras según diferentes categorías. Para evaluar la capacidad de clasificación de una red neuronal existen diversas técnicas matemáticas, todas ellas se basan en determinar si cada muestra ha sido clasificada apropiadamente o no.

La característica de funcionamiento del receptor es una medida utilizada para comprobar la calidad de los clasificadores. Para cada clase de un

clasificador, ROC aplica valores de umbral en todo el intervalo [0,1] de las salidas. Para cada umbral, se calculan dos valores, la relación de **verdaderos positivos** (el número de salidas mayor o igual que el umbral, dividido por el número de objetivos de uno), y el ratio de **falsos positivos** (el número de salidas menores que el umbral, dividido por el número de objetivos de cero).

El **análisis ROC** (*“Receiver Operating Characteristic”*, o Característica Operativa del Receptor) provee herramientas que permiten seleccionar el subconjunto de clasificadores que tienen un comportamiento óptimo en general.

Las **curvas ROC** son una representación gráfica de la sensibilidad (VPR = Razón de Verdaderos Positivos) frente a (1 – especificidad) (FPR = Razón de Falsos Positivos) para un sistema de clasificación según se varía el umbral de discriminación, valor a partir del cual decidimos que un caso es un positivo.

Un análisis ROC es una técnica para visualizar, organizar y seleccionar clasificadores en base a su rendimiento. La curva ROC se desarrolló por ingenieros eléctricos para medir la eficacia en la detección de objetos enemigos en campos de batalla mediante pantallas de radar, a partir de lo cual se desarrolla la Teoría de Detección de Señales (TDS).

Los gráficos ROC han sido utilizados en la teoría de detección de señales (TDS) para representarel equilibrio entre las tasas de éxito y las tasas de falsas alarmas en la clasificación [Egan, 1975], [Swets et al., 2000].

El análisis ROC ha sido extendido para su uso en la visualización del comportamiento de los sistemas de diagnóstico [Swets, 1988]. En la toma de decisiones, la comunidad médica tiene una extensa literatura sobre el uso de gráficos ROC para las pruebas de diagnóstico [Swets et al., 2000], [Zou, 2002].

Son de uso común en las decisiones médicas y en los últimos años se han adoptado cada vez más en otras áreas [Concejero, 2009], tales como:

- El análisis ROC y la detección temprana en Psicología Clínica.
- El análisis ROC en Psicología Social y Forense.
- Aplicaciones del análisis ROC en tecnologías de la información.
- El análisis ROC en la investigación de mercados.

10.1. El análisis ROC en las redes neuronales artificiales.

Aparentemente las curvas ROC son simples, pero hay algunos malentendidos comunes y dificultades cuando se utilizan en la práctica que conviene aclarar.

El esquema básico de partida del análisis de las curvas ROC es el mismo que en TDS: se trata de una tabla de doble entrada en la que se representa por un lado el resultado de una clasificación, y por otro el estado salida conocido. Por lo tanto, tenemos cuatro posibles resultados. Utilizaremos la notación y el ejemplo de Swets [Swets, 1986], [Swets, 1988], con la ventaja de que esta notación también es la utilizada por el programa MATLAB®.

Consideremos primeramente los problemas de clasificación con sólo dos clases. A cada entrada se le asigna uno de los elementos del conjunto {p,n}. Un modelo de clasificación es una aplicación de las entradas de clases previstas. Para distinguir entre la clase real y la clase prevista utilizamos las etiquetas {P,N} de las predicciones de clase producidas por el modelo.

Dado un sistema de clasificación y una entrada, hay cuatro posibles resultados. Si la entrada es positiva y se clasifica como algo positivo, se cuenta como un verdadero positivo, si se clasifica como algo negativo, se cuenta como un falso negativo. Si la entrada es negativa y se clasifica como algo negativo, se cuenta como un verdadero negativo, si se clasifica como algo positivo, se cuenta como un falso positivo. Dado un sistema de clasificación y un conjunto de casos de prueba, se puede construir una matriz de 2x2, denominada **matriz de confusión**, que representa las disposiciones del conjunto de entrada.

Tabla 1.3. Matriz de confusión de un sistema de clasificación.				
		Clase real		Suma
		Positivo p	Negativo n	
Clase hipotética	Positivo P	Verdaderos Positivos (VP)	Falsos Positivos (FP)	VP+FP
	Negativo N	Falsos Negativos (FN)	Verdaderos Negativos (VN)	FN+VN
Suma		VP+FN	FP+VN	VP+FP+FN+VN

La tabla 1.3 muestra la matriz de confusión y las ecuaciones de varios indicadores comunes que pueden calcularse a partir de ella. Los números a lo largo de la diagonal principal representan las decisiones correctas, y los números fuera de la diagonal representan los errores (la confusión) entre las distintas clases.

La verdadera tasa positiva (también llamada tasa de éxito) de un sistema de clasificación se calcula como:

$$tpr = \text{positivos clasificados correctamente} / \text{positivos totales}$$

La falsa tasa positiva se calcula como:

$$fpr = \text{negativos clasificados incorrectamente} / \text{negativos totales}$$

La tabla 1.3 describe completamente el comportamiento de un sistema de clasificación con dos únicos posibles resultados (positivo o negativo). Dado que, si calculamos las proporciones por columna, éstas suman 1, hay dos en vez de cuatro entradas independientes en la matriz. Por la misma razón, se deben especificar dos cantidades para obtener una representación completa de la clasificación. Estas dos medidas son la sensibilidad y la especificidad:

- **La sensibilidad** es la proporción de salidas afirmativas, a partir del criterio o regla de decisión establecida, en los que se comprueba que efectivamente sucede el estado que se pretende detectar.
- **La especificidad** es la proporción de salidas negativas, a partir de la regla de decisión establecida, en los que se comprueba que efectivamente no sucede el estado que se pretende detectar.

La sensibilidad y la especificidad están relacionadas entre sí y dependen del umbral o punto de corte, de manera que, si la regla de decisión impuesta establece un umbral muy bajo, tendremos alta especificidad pero baja sensibilidad, y si, por el contrario establecemos un umbral alto tendremos alta sensibilidad pero baja especificidad.

La **sensibilidad** viene dada por:

$$\text{Sensibilidad} = \frac{VP}{VP + FN} = TPR \quad (1.46)$$

Y la **especificad** por:

$$\text{Especificidad} = \frac{VN}{VN + FP} = 1 - FPR \quad (1.47)$$

10.2. Espacio ROC.

Los gráficos ROC son gráficos bidimensionales en el que el eje de ordenadas representa la tasa de verdaderos positivos (TPR) y el de abscisas la tasa de falsos positivos (FPR). Un gráfico ROC representa la relación de compensaciones entre los beneficios (verdaderos positivos) y los costes (falsos positivos).

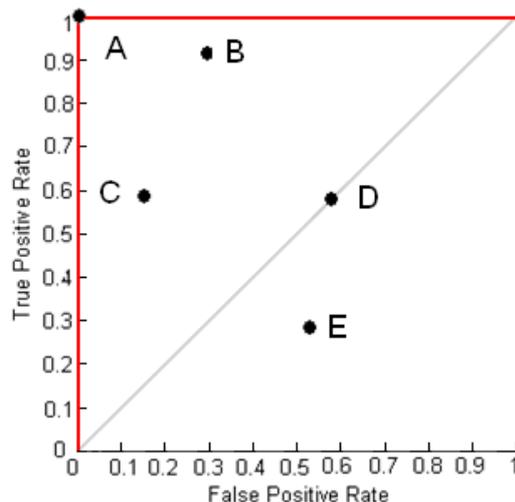


Figura 1-25. Gráfico ROC discreto con cinco clasificaciones

En la **Fig. 1-25** se muestra un gráfico ROC con cinco clasificaciones discretas numeradas de la A a la E. Un sistema de clasificación discreto es aquel que sólo emite una etiqueta de salida perteneciente a una clase, lo que produce un par discreto (FPR, TPR), que corresponde a un solo punto en el espacio ROC [Fawcett, 2003].

Hay varios puntos importantes en el espacio ROC a tener en cuenta. El punto inferior izquierdo (0, 0) indica que un sistema de clasificación no comete errores de falsos positivos, en cambio el punto superior derecho (1,1),

representa la estrategia opuesta, la emisión de clasificaciones positivas. Tradicionalmente se dibuja también la diagonal positiva (punto D) para indicar el punto de nula capacidad predictiva. Efectivamente, si la sensibilidad es igual a 1-especificidad a lo largo de todos los puntos de corte es que nuestra prueba simplemente selecciona unos casos de otros por puro azar.

El punto (0, 1) representa la clasificación perfecta, punto A.

En el punto C se hacen clasificaciones positivas sólo con una fuerte evidencia de algunos falsos errores positivos, mientras que en el punto B, se hacen clasificaciones positivas con pruebas débiles, de manera que clasifica casi todos los positivos correctamente, pero a menudo tienen altos índices de falsos positivos. Finalmente el punto E es un mal clasificador (bajo TPR y alto FPR).

Informalmente, un punto en el espacio ROC es mejor que otro cuanto más cerca esté de la esquina superior izquierda, (TPR es más alto, FPR es más bajo, o ambas cosas).

10.3. Área bajo la curva ROC (AUC).

Una curva ROC es una representación en dos dimensiones del rendimiento del clasificador. Para comparar varios clasificadores, se puede reducir el rendimiento de la curva ROC a un solo valor escalar que representa el rendimiento esperado. Un método común es calcular el área bajo la curva ROC, abreviado AUC [Bradley, 1997], [Hanley&McNeil, 1982]. Dado que la AUC es una parte de la zona unidad, su valor siempre estará entre 0 y 1. Sin embargo, debido al adivinar al azar se produce la línea diagonal entre (0, 0) y (1, 1), que tiene una superficie de 0,5, un mal clasificador debe tener un AUC menor de 0,5.

El AUC tiene una propiedad estadística importante: el AUC de un clasificador es equivalente a la probabilidad de que el clasificador ubicará un ejemplo elegido al azar positivo más alto que un ejemplo elegido al azar negativo. Es equivalente a la prueba de Wilcoxon de rangos [Hanley&McNeil, 1982].

Las AUC también está estrechamente relacionadas con el índice de Gini [Breiman et al.,1984] (que es el doble del área entre la diagonal y la curva ROC. Hand and Till [Hand & Till, 2001] señalan que: $Gini + 1 = 2 \times AUC$.

El área bajo la curva ROC es una medida global de la exactitud de una prueba diagnóstica. Se define como la probabilidad de clasificar correctamente un par de individuos sano y enfermo, seleccionados al azar de la población, mediante los resultados obtenidos al aplicarles la prueba diagnóstica. Cada punto de la curva representa un par $S/1-E$ correspondiente a un nivel de decisión determinado [Burgueño et al., 1995].

La Fig. 1-26 muestra el área bajo la curva de un clasificador binario.

Los diferentes puntos de la curva ROC pueden provenir de varios clasificadores o de un mismo clasificador (variando el umbral).

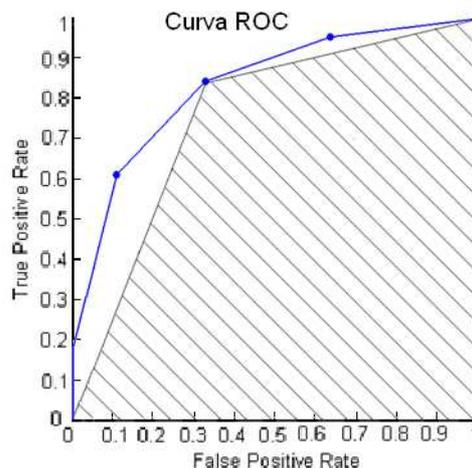


Figura 1-26. Área ROC para un clasificador binario.

Si cada punto de la curva ROC corresponde a un umbral con el que se toma la decisión hay que seleccionar el clasificador con mayor área bajo la curva (AUC) [Abdelmalik et al., 2010].

11. Implementaciones hardware de redes neuronales.

Hasta ahora, los científicos y los ingenieros han dependido de las simulaciones por ordenador, a veces ejecutadas sobre computadoras especializadas, para realizar su labor de investigación y desarrollo. Principalmente se han

empleado microprocesadores gracias a su pequeño tamaño, bajo precio, alto nivel de prestaciones y bajo consumo. El inconveniente fundamental de tales simuladores es que el paralelismo espacio temporal en el procesamiento de la información, inherente a las redes neuronales, es completa o parcialmente perdido de modo que el tiempo de computación de la red simulada, especialmente para una gran asociación de neuronas, crece hasta unos órdenes de magnitud que hacen imposible o al menos retrasan en gran medida la obtención de resultados.

Sin embargo, se ha iniciado el camino hacia la integración hardware de las redes neuronales, prueba de ello es el número de chips bajo desarrollo, así como los que ya están en el mercado. Mediante el diseño especializado de hardware neuronal se puede conseguir una apreciable reducción del tiempo de computación, haciendo incluso posible aplicaciones de tiempo real.

Tanto en Estados Unidos, como en Japón y Europa, se están realizando gran cantidad de estudios que van emergiendo continuamente. En muchos de estos casos estos estudios conciernen a neurocomputadoras que nunca serán construidos completamente.

Una neurocomputadora es básicamente un conjunto de procesadores conectados de forma regular y que operan concurrentemente. En la actualidad ya existen una serie de neurocomputadoras comerciales destinados a la realización de redes neuronales.

La construcción de neurocomputadoras es cara en términos de tiempo de desarrollo y recursos, y además se desconoce si tendrán perspectivas reales de uso comercial.

La mayoría de los productos disponibles comercialmente en el terreno de las neurocomputadoras son implementaciones dedicadas a los tipos de redes neuronales más conocidos y exitosos, como es el caso del perceptrón multicapa [*Rumelhart, 1986*], la red de Hopfield [*Hopfield, 1982*] o los modelos de Kohonen [*Kohonen, 1984*].

Por otro lado, la forma más simple, rápida y económica de realización de una red neuronal es simular la red en un PC utilizando software específico.

Otro método es la realización de la red neuronal mediante circuitos integrados específicos o neurochips analógicos o digitales.

Entre los neurochips analógicos cabe destacar, el circuito integrado comercializado por Intel denominado 80170NX, también conocido como *Electrically Trainable Analog Neural Network* [Holler et al., 1989a], [Holler et al., 1989b], [Intel, 1992].

Las principales ventajas que presenta la implementación hardware de redes neuronales son:

- Velocidad - la mayoría de las aplicaciones puede acelerarse mediante el hardware especializado. El procesamiento paralelo con múltiples elementos simples de procesamiento puede proporcionar aceleraciones significativas.
- Fiabilidad - es conocido el hecho de que una implementación de hardware puede ofrecer una mayor fiabilidad en el funcionamiento
- Las condiciones de servicio especiales - para aquellas aplicaciones que requieren limitación de tamaño o peso.
- Secreto - una mejor protección ante la ingeniería inversa.

Objetivos de la Tesis Doctoral

1. Objetivos de la Tesis Doctoral

1.1. Introducción

A lo largo de la primera parte de este capítulo de la memoria se ha realizado una amplia revisión de las redes neuronales artificiales, indicando los tipos de redes más importantes, sus arquitecturas y los algoritmos que rigen su funcionamiento. La finalidad de la tesis es lograr implementar redes neuronales en dispositivos de microcontroladores, por lo tanto, es imprescindible conocer el algoritmo que regula el comportamiento de cada red. De este modo, el primer capítulo ha estado centrado en el estudio de los dos tipos de redes neuronales que se han utilizado en la tesis doctoral: la perceptrón multicapa y la Fuzzy Artmap.

1.2. Precedentes.

Durante los últimos años, el Grupo de Microelectrónica Híbrida del Instituto de Reconocimiento Molecular y Desarrollo Tecnológico (IDM) de la Universidad Politécnica de Valencia, ha desarrollado diversos sistemas electrónicos para la adquisición de datos basados fundamentalmente en técnicas electroquímicas, como la potenciometría, voltamperimetría y espectroscopía de impedancia, pero también con sensores ópticos utilizando fotodiodos. En este caso se obtiene lo que se puede denominar como ojo electrónico.

Los datos obtenidos con los sistemas de medida se suelen analizar posteriormente en un ordenador personal (PC), aplicando técnicas estadísticas de análisis multivariante no supervisadas, tal como el análisis de componentes principales (PCA). Pero en muchas ocasiones, es conveniente realizar análisis supervisado, donde, las categorías de las muestras están predefinidas y la finalidad es comprobar si es posible conseguir un sistema que sea capaz de clasificar adecuadamente muestras nuevas que entran en el sistema de medida. En este caso es cuando se utilizan las redes neuronales artificiales.

Una de las líneas de mejora de los sistemas de lenguas electrónicas es el desarrollo de sistemas electrónicos que sean capaces de realizar un análisis

“in situ” de las muestras. Para conseguir este fin, hay que lograr la incorporación de la red neuronal en un sistema electrónico digital autónomo, la forma más inmediata de lograr un sistema electrónico digital de estas características es mediante el uso de dispositivos programables, como el microcontrolador, porque son económicos, relativamente sencillos de programar, con abundante información técnica, fácil de manipular y con bajo consumo energético.

Pero este tipo de dispositivos tienen como característica principal, que les diferencia de los sistemas tipo PC, la limitación de la memoria, por lo que a la hora de abordar la tarea de intentar la inclusión de una red neuronal en un microprocesador hay que lograr la minimización de la memoria necesaria.

Pero la minimización de la memoria de los sistemas microcontroladores está limitado por el hecho de que la red neuronal obtenga los mejores resultados posibles en la clasificación de las muestras a analizar. Por ello se realiza una variación de los parámetros que determinan el rigor en la clasificación de cada una de las muestras en las categorías establecidas.

Del amplio abanico de redes neuronales disponibles se utilizan dos de las más fácilmente integrables en sistemas programables y que han sido ampliamente utilizadas y verificadas: Las redes perceptrón multicapa y la red Fuzzy Artmap. En este último caso, se utiliza la versión simplificada dada su adecuada estructura para la integración en microcontroladores.

1.3. Definición de Objetivos

Por todo lo precedentes anteriormente mencionales, los objetivos de la presente tesis son los siguientes:

- Utilizar e integrar los algoritmos de las redes neuronales artificiales de tipo Perceptrón multicapa y la versión simplificada de la red Fuzzy Artmap desarrollada por Kasuba en 1993 en un sistema microcontrolador. Elegir el tipo de dispositivo más adecuado para implementar las redes neuronales.
- Desarrollar programas de optimización de las redes neuronales con el objetivo de mejorar los parámetros de integración de dichas redes en

sistemas programables. En concreto, el desarrollo de programas para la red FuzzyArtmap Simplificada mediante el software MATLAB de un entorno gráfico o Graphical User Interface (GUI), que permita modificar las características de la red durante el entrenamiento de la misma con objeto de obtener los pesos, valores máximos y mínimos, etc. necesarios para su implementación en el microcontrolador. Estos programas deben incluir una selección de variables de forma que permitan elegir el conjunto de variables mínimo que mantengan los valores óptimos de la red con la mínima cantidad de memoria. Finalmente para evitar el sobreentrenamiento de la red se debe poder realizar una validación cruzada.

- Aplicar el programa anterior, que obtiene los parámetros de las redes neuronales a diversas aplicaciones para clasificación y predicción, a saber, análisis de alimentos, análisis de aguas y análisis químicos.

1.4. Aplicaciones

Una vez obtenidas las herramientas necesarias para la implementación de algoritmos de las redes neuronales en un sistema de microcontrolador, se van a utilizar en algunas aplicaciones de interés, tanto científico como industrial.

En los siguientes capítulos de la presente memoria se desarrollan las aplicaciones seleccionadas. La elección de las aplicaciones viene motivada por las líneas de trabajo e investigación que desarrolla o colabora el Instituto de Investigación. Estas líneas están centradas en la monitorización de parámetros físicos y químicos de interés en análisis químicos, medio ambiente, control calidad de alimentos. Es de destacar que dichas aplicaciones son muy diversas entre si, tanto por la naturaleza de la aplicación, como por el tipo de sensor utilizado, la técnica de medida aplicada y el sistema electrónico de medida utilizado.

Es necesario matizar que los sistemas medida que se han utilizado en cada una de las aplicaciones han sido desarrollados previamente por otros investigadores del Instituto IDM, así como la preparación y mediciones con las muestras analizadas. De esta forma, la tarea fundamental de esta tesis ha sido

la recogida de resultados y la implementación de redes neuronales en sistemas microcontroladores

Las aplicaciones desarrolladas en esta tesis son las siguientes:

- Cálculo de la turbidez del agua conforme aumenta la profundidad en aguas estancas. Este parámetro se conoce como profundidad de Secchi. Se utilizaron sensores ópticos.
- Sensor de humedad capacitivo en etiquetas flexibles de RFID. Se utilizaron sensores capacitivos.
- Clasificación de aguas embotelladas de diversas marcas comerciales. Se utilizaron electrodos en tecnología Thick-Film aplicando técnicas potenciométricas.
- Clasificación de mieles de distinto origen floral y diferente tratamiento térmico. Se utilizaron electrodos metálicos aplicando técnicas potenciométricas.
- Determinación de la presencia y concentración de glifosfato en disolución. Este producto es utilizado como herbicida. Se utilizaron electrodos metálicos aplicando técnicas voltamétricas.

El formato de cada uno de los siguientes capítulos es el siguiente:

- Introducción a las características de cada aplicación y a los parámetros a medir.
- Tipo y cantidad de muestras a medir
- Descripción de los sensores y del sistema y método de medida.
- Presentación de los datos obtenidos de las medidas.
- Análisis de datos con redes neuronales. Entrenamiento y verificación. Se utilizarán redes de tipo Perceptron Multicapa o Fuzzy Artmap según el tipo de análisis a realizar. En caso de que solo necesitar una clasificación de la muestras se utilizará la red Fuzzy Artmap Simplificada. En caso de realizar una predicción de un parámetro se utilizará la red Perceptron Multicapa.
- Implementación de la red óptima en un dispositivo microcontrolador.

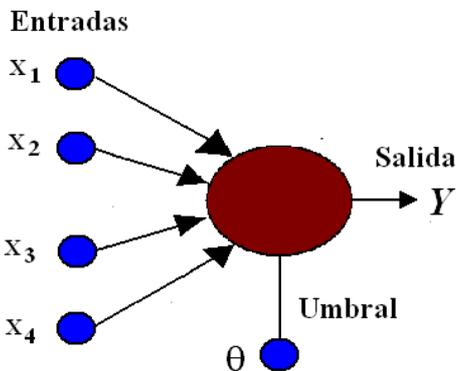
- Resultados de la simulación y discusión.
- Conclusiones.

En los dos primeros capítulos de aplicaciones solamente se realiza una predicción de un parámetro físico (profundidad y humedad) por lo que se han utilizado solo redes neuronales de tipo perceptrón, los dos siguientes (agua embotelladas y miel) son aplicaciones de clasificación de muestras por lo que se han utilizado redes de tipo Fuzzy Artmap y en el último capítulo (glifosato) se realiza una aplicación mixta (clasificación y predicción) por lo que se han aplicado los dos tipos de redes neuronales.

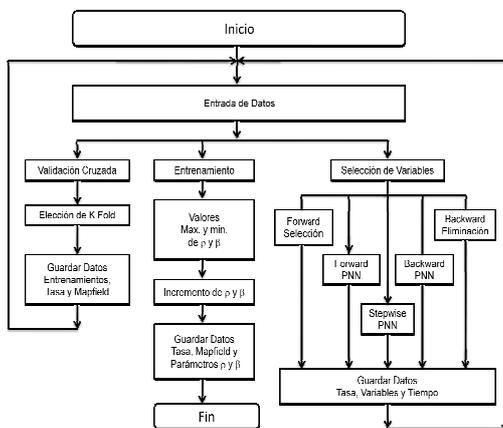
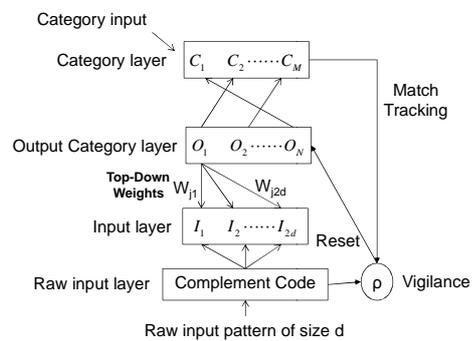
Capítulo 2

Algoritmos de Redes Neuronales Artificiales para Implementarlos en Microcontroladores de 8 Bits

Perceptrón



Fuzzy Artmap Simplificada



1. Introducción.

El objetivo principal de la presente tesis doctoral es la implementación de redes neuronales artificiales (RNA) en dispositivos electrónicos digitales programables. La principal ventaja de este objetivo radica en poder desarrollar equipos portátiles con tecnologías de bajo coste, con un rápido tiempo de respuesta, una vez aprendido el funcionamiento del sistema. También hay que tener presentes algunas desventajas como que, una vez entrenada la RNA e implementada en el microcontrolador no se pueden añadir nuevas variables. Si se desea realizar una ampliación, habrá que entrenar nuevamente la RNA.

Para lograr la implementación de una red neuronal artificial en un sistema microprocesador es necesario realizar dos tareas: Entrenamiento y Validación

Para realizar las tareas de entrenamiento de una red neuronal se utilizarán programas informáticos que permitan el análisis de los datos de entrenamiento y obtener los parámetros de la red neuronal para su posterior implementación en el microcontrolador.

Una vez obtenidos los parámetros óptimos de la red neuronal, la segunda parte del proceso consiste en desarrollar el programa que implementará la red neuronal en un microprocesador y poder realizar las tareas de validación de la red. Para ello se utilizará el lenguaje de alto nivel C que servirá para obtener el programa en lenguaje máquina que se almacenará en la memoria de programa del microcontrolador.

1.1. Entrenamiento de Redes Neuronales.

Para el realizar el entrenamiento de las redes neuronales se va a utilizar el programa de cálculo matricial MATLAB®. La forma de trabajar será mediante el uso de programas de Interfaz Gráfica de Usuario (GUI). Este tipo de programas actúan de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz. Su principal uso, consiste en proporcionar un entorno visual sencillo para permitir la comunicación con el sistema operativo de un ordenador.

El paquete informático MATLAB viene incorporado con un paquete de programas para el desarrollo de redes neuronales artificiales denominado

Neural Network Toolbox. En las últimas versiones de MATLAB esta herramienta posee algunos programas de tipo gráfico (GUI) para trabajar con los tipos de redes más utilizados como son la red perceptrón multicapa, como son: *Neural Pattern Recognition* para obtener una red perceptrón para aplicaciones de clasificación de datos de entrada y el programa *Neural Fitting Toolbox* para aplicaciones de predicción.

La herramienta de MATLAB *Neural Network Toolbox* no posee aplicaciones para redes neuronales de tipo Fuzzy Artmap, por ello ha sido necesario desarrollar un programa para obtener las principales características de una red de este tipo, como son: *mapfield*, valores máximos y mínimos, pesos, etc. La descripción de este programa es la parte fundamental del presente capítulo de la tesis doctoral.

1.2. Implementación de Redes Neuronales en Microprocesador

La segunda parte de la tarea consiste en escribir el algoritmo que implementa la red neuronal obtenida en la fase de entranamiento, en un lenguaje de alto nivel, como es el C, para implementarlo en la memoria de programa de un dispositivo microcontrolador. De esta forma estamos en disposición de poder validar la red neuronal.

2. GUI_Fuzzy_Artmap.

En este punto se describe el programa GUI desarrollado en MATLAB para obtener los parámetros característicos de la red Fuzzy Artmap Simplificada (SFAM).

En sistemas de bajo coste los microcontroladores empleados suelen tener un tamaño limitado de memoria. En estos casos es necesario optimizar los algoritmos de tratamientos de datos. Es aquí donde la red SFAM aporta una serie de ventajas: reducida de la ocupación de memoria, rapidez y facilidad de programación.

Los datos que deben ser programados en la memoria del microcontrolador son la matriz de pesos, el *mapfield* (*output category layer*) y los valores máximos y mínimos de los valores de entrada para la normalización de los vectores de entrada.

Tabla 2.1. Evolución del tamaño de la memoria y de la matriz de pesos en función del número de muestras.			
Muestras de entrenamiento	Matriz de pesos	Memoria RAM (bytes)	Memoria de Programa (bytes)
8	12x16	1635	12726
10	12x17	1700	12812
12	12x20	1895	13064
14	12x22	2025	13224
16	12x23	2090	13304
18	12x26	2285	13356

Pero uno de los problemas que plantea, tanto la Fuzzy Artmap como la SFAM, es el tamaño del *mapfield* y los pesos que dependen de los valores elegidos para los parámetros β y ρ . El tamaño de la matriz de pesos y del *mapfield* aumenta al incrementarse el número de datos de entrenamiento. La tabla 2.1 muestra la variación del tamaño de la matriz de pesos y el tamaño de memoria de programa y RAM utilizada según el número de muestras de entrenamiento (por ejemplo entre 8 y 18 muestras). Se puede observar que la memoria de programa apenas sufre variación, no así la memoria RAM que de 8 a 18 muestras sufre un 40% de aumento en su tamaño. Esto en muchos casos supondría tener que cambiar de microcontrolador al superar el límite de memoria.

Normalmente no hay un criterio inicial para establecer los valores de estos parámetros y se deben realizar distintos entrenamientos variando su valor hasta encontrar un porcentaje de clasificación óptimo. Cuando se encuentra un porcentaje de clasificación adecuado no suele tenerse en cuenta el tamaño de la “*Output category layer*” (“*mapfield*”) y la matriz de pesos ya que en sistemas sobre PC, el tamaño de la memoria de trabajo no es significativo. Por el contrario en sistemas basados en microcontrolador es necesario comprobar el tamaño de estos dos datos y procurar que sea lo mínimo posible. Esto solo se puede hacer, realizando distintos entrenamientos de la red con distintos valores de β y ρ , buscar los mejores porcentajes de clasificación y ver en que casos el tamaño del “*mapfield*” y la matriz de pesos es menor.

Con el fin de obtener la máxima tasa de aciertos en la clasificación para el valor mínimo de matriz de pesos y *mapfield* se ha desarrollado una Interface Gráfica de Usuario (*Graphical User Interface*, GUI) en MATLAB basada en el “*toolbox*” desarrollado por Aaron Garret. Mediante esta GUI se puede realizar un entrenamiento con el barrido de todos los valores posibles de β y ρ , determinar las mejores clasificaciones y buscar, entre ellas, la que posee un menor tamaño de “*mapfield*” y de matriz de datos para la mejor tasa de aciertos con el objeto de implementar la red en el microcontrolador. Además de este objetivo básico se ha incorporado a la GUI la posibilidad de realizar una validación cruzada y una selección de variables. La Fig. 2-1 muestra el diagrama de bloques de la GUI desarrollada.

La Interface Gráfica de Usuario (GUI_Fuzzy_ARTMAP) desarrollada para la presente tesis se ha dividido en tres apartados:

- Entrenamiento
- Validación Cruzada
- Selección de variables

Cada uno de los cuales consta, a su vez, de varias pantallas graficas que facilitan la elección de los parámetros variables. Cada pantalla consta a su vez de dos ficheros con el mismo nombre y distinta extensión (.m y .fig).

Los ficheros con extensión .m, contienen el código, mientras que los .fig permiten al usuario la introducción de datos y variables necesarias para el estudio de la red neuronal Fuzzy Artmap Simplificada. El entrenamiento y la validación cruzada permiten obtener la mejor clasificación de los datos con una matriz de pesos óptima mientras que la selección de variables reduce el conjunto de datos que se dispone, mejorando la matriz de pesos y la tasa de aciertos.

Para el desarrollo de los programas Fuzzy ARTMAP de Entrenamiento y Validación Cruzada se ha modificado el “*toolbox*” de MATLAB® diseñado por Aaron Garret (Jacksonville State University) en MATLAB® para adaptarlo a nuestras necesidades, mientras que para la Selección de Variables se han utilizado parte de los programas desarrollados por Lapiedra [Lapiedra, 2004], en MATLAB®.

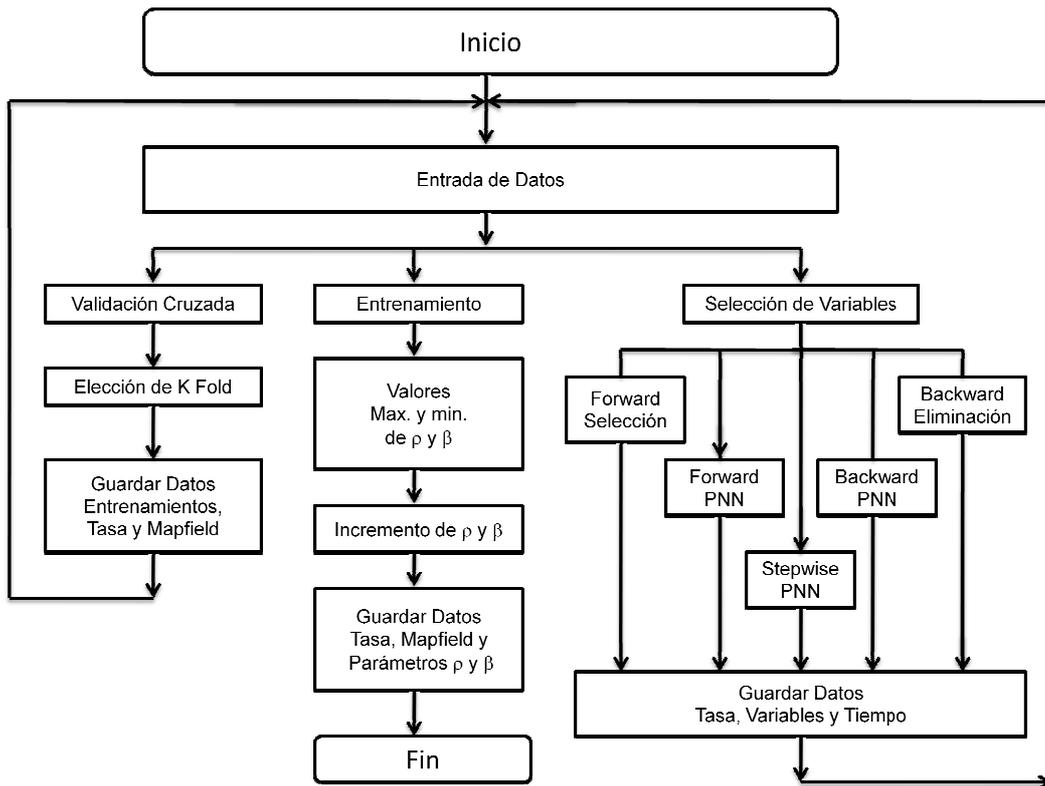


Figura 2-1. Organigrama de la Interface Gráfica de usuario (GUI).

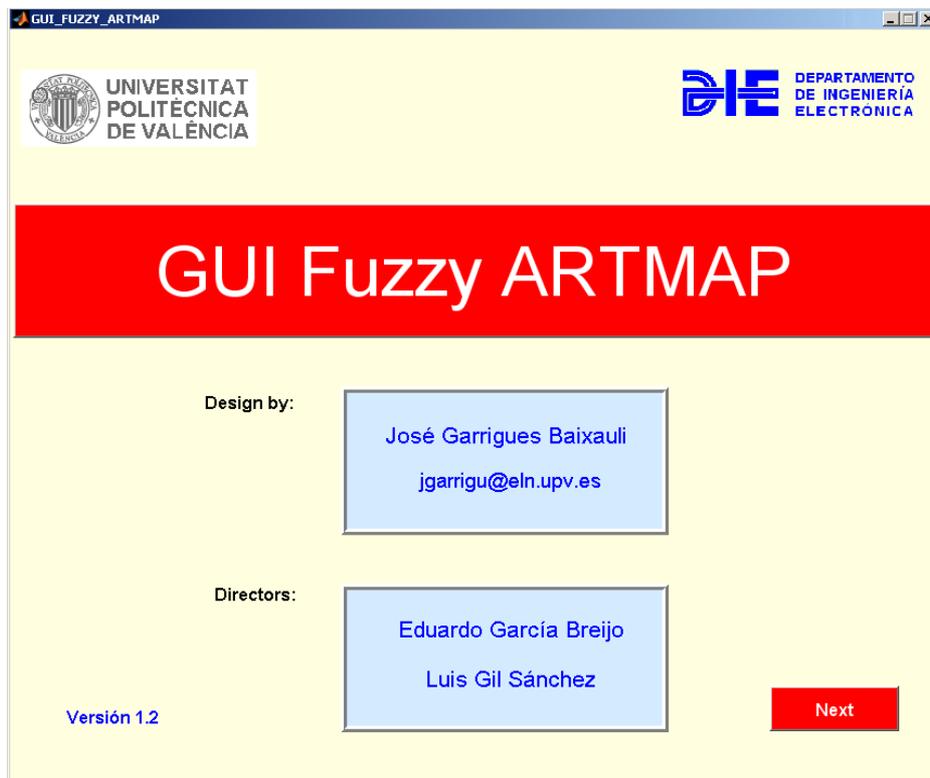


Figura 2-2. Pantalla de presentación de la GUI.

Tras la pantalla de presentación inicial (Fig. 2-2) se pasa a la pantalla de entrada de datos (Fig. 2-3) mediante un programa denominado **ARTMAP_DATOS**, que contiene:

- entrada de datos por filas o columnas
- objetivos de entrenamiento y validación
- categorías de salida
- % de datos para el entrenamiento y validación

Los datos de entrada deben de estar en formato .txt. En primer lugar se colocarán los datos para el entrenamiento y a continuación los datos para la validación. Estos datos constituyen una matriz de orden $n \times m$, en donde:

- $n-1$ es el vector de entrada correspondiente a cada muestra.
- m es el número total de datos muestreados, tanto para el entrenamiento como para la validación.
- la última fila contiene las salidas esperadas para cada uno de los vectores de entrada.

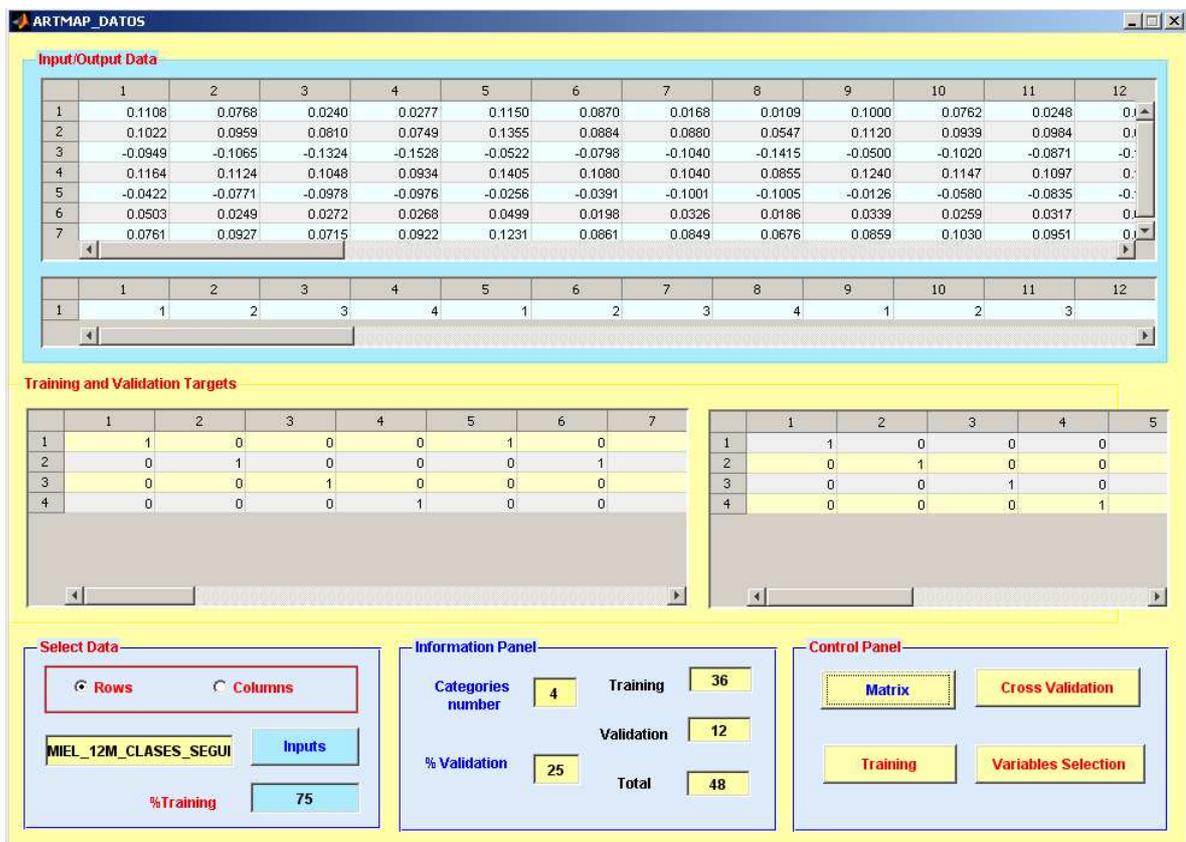


Figura 2-3. Entrada de datos de la GUI desarrollada en MATLAB®.

El usuario elige la entrada de datos por filas o columnas. El fichero debe contener en formato texto, los datos de entrada, en filas (columnas) estarán las diferentes variables y en la última fila (columna) las salidas esperadas. Finalmente, se introduce el % de datos para el entrenamiento.

El programa calcula:

- el número de categorías o clases.
- el % de datos para la validación.
- el número de datos de entrenamiento, validación y total.
- separa el vector de entrada de las salidas esperadas.

Por último, se pulsa el botón “**Matrix**” para calcular las matrices u objetivos de entrenamiento y validación que se necesitan para generar el Plotconfusion y el PlotROC y visualizar los botones de:

- Entrenamiento
- Validación Cruzada
- Selección de Variables

2.1. Entrenamiento.

Para poder implementar la red Fuzzy Artmap en un microcontrolador se necesita un mapfield reducido, lo que disminuye la cantidad de memoria del microcontrolador y una tasa de aciertos óptima. Por lo tanto, el objetivo del entrenamiento será obtener los valores de ρ y β que satisfagan las condiciones anteriores:

- mapfield mínimo
- tasa máxima.

Para cumplir con dichas condiciones se han desarrollado los programas:

- ARTMAP_PARAMETROS
- ARTMAP_FICHEROS
- ARTMAP_DOBLE_ROYBETA

2.1.1. ARTMAP_PARAMETROS.

Este programa permite fijar los valores iniciales, finales y el incremento de los parámetros ρ y β para realizar el estudio de la red neuronal Fuzzy Artmap. La Fig. 2-4 muestra la pantalla correspondiente a la entrada de los parámetros ρ y β .

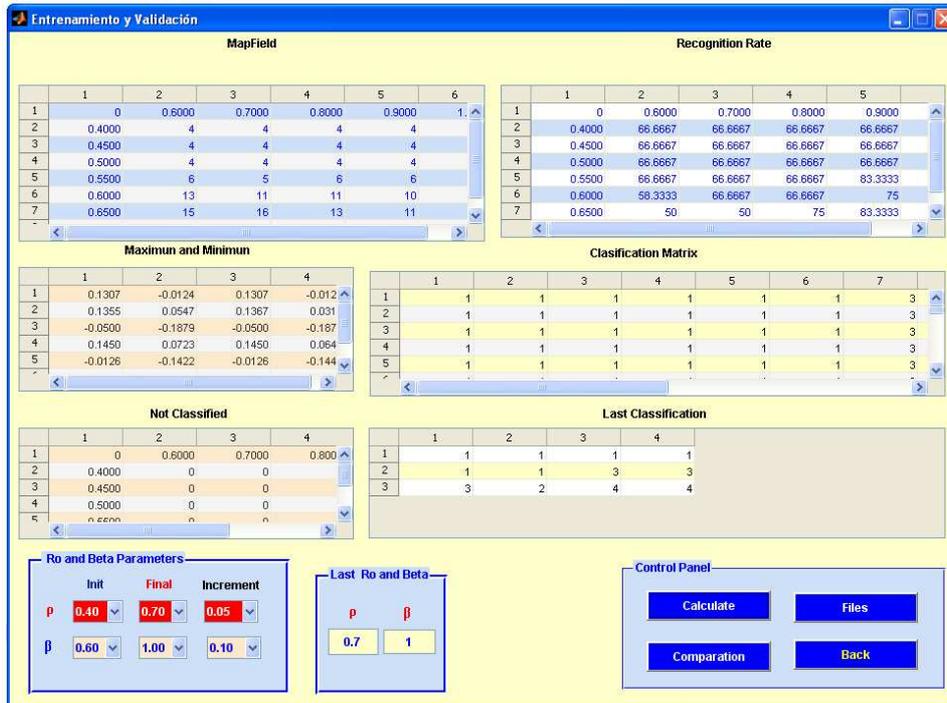


Figura 2-4. Parámetros ρ y β para el entrenamiento de la Fuzzy Artmap.

Parámetros ρ y β ("Ro and Beta Parameters"). Menú desplegable que nos permite seleccionar por columnas, los valores iniciales, finales y el incremento de cada uno de los parámetros (filas).

Botones del Panel de Control ("Control Panel").

- **Calcular ("Calculate").** Permite arrancar el programa y pasarle los parámetros ρ y β al "toolbox", diseñado por Aaron Garret al que previamente se le han hecho las modificaciones necesarias.
- **Ficheros (Files").** Permite pasar a la siguiente pantalla para guardar los datos en Excel que nos interesen.
- **Comparación ("Compare").** Permite pasar a la pantalla ARTMAP_DOBLE_ROYBETA, para realizar la comparación de áreas

bajo la curva (*Area Under the Curve* AUC) en el caso de que coincidan varias tasas para diferentes valores de ρ y β .

- **Volver ("Back").** Vuelve a la pantalla anterior.

Tablas. Permiten visualizar previamente los resultados obtenidos en función de ρ y β antes de guardar los datos. De izquierda a derecha y de arriba abajo, son las siguientes:

- **Mapfield.** Nos da el tamaño del mapfield en función de ρ (columnas) y β (filas).
- **Tasa de aciertos ("Recognition Rate").** Nos informa de la tasa de aciertos en función de ρ y β .
- **Matriz de clasificaciones ("Calification Matrix").** Nos informa en cada fila de la clasificación correspondiente a cada vector de entrada introducido en la fase de validación en función de ρ y β .
- **Máximos y Mínimos ("Maximun and Minimum").** Nos da los valores máximos y mínimos de los vectores de entrada en el entrenamiento (las dos primeras columnas) y en la validación (las dos últimas columnas).
- **No clasificados ("Not Classified").** Nos informa de la tasa de aciertos en función de ρ y β . La mayoría de los datos se clasifican o bien en la categoría a la que pertenecen o bien en cualquier otra. Pero para valores altos de ρ , algún vector de entrada puede no clasificarse en ninguna de las categorías. El caso extremo sería para $\rho=1$, en donde no se clasificaría ningún dato.
- **Última Clasificación ("Las Clasfication").** Informa de la clasificación de cada uno de los vectores correspondientes a la validación. Ej. el primer valor se ha clasificado correctamente en la clase 1.

Último ρ y β ("Last Ro and Beta"). Nos informa en cada momento del valor de dichos parámetros durante la ejecución del programa. Debe coincidir al terminar el estudio con los valores finales seleccionados previamente.

2.1.2. ARTMAP_FICHEROS.

Este programa permite seleccionar la información a guardar en ficheros Excel. Inicialmente todos los ficheros están marcados para guardar. El usuario puede desmarcar los que no le interese guardar.

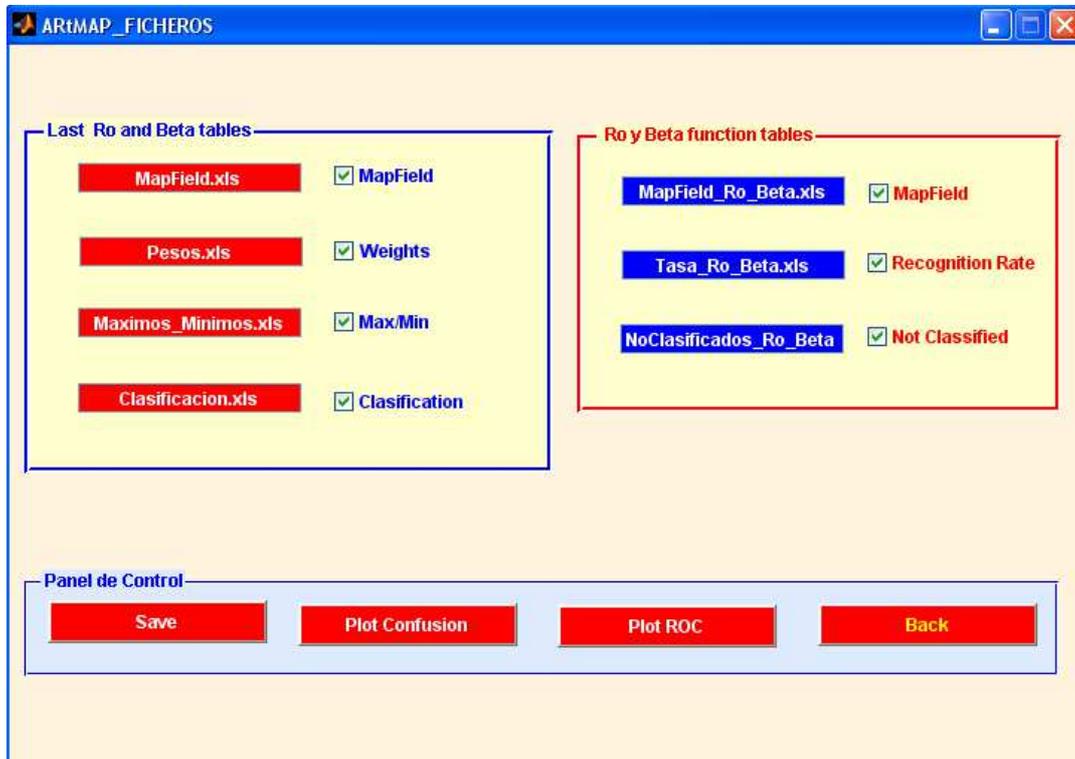


Figura 2-5. Pantalla correspondiente a ARTMAP_FICHEROS.

Por defecto se guardan con el nombre que aparece en la Fig. 2-5, pero el usuario puede asignarles el nombre que considere más adecuado, simplemente colocando el cursor sobre el texto, borrando y tecleando el nuevo nombre. No olvide que obligatoriamente la extensión debe ser **.xls**.

2.1.3. ARTMAP_DOBLE_ROYBETA.

Calcula el AUC de las diferentes clasificaciones para dos valores de ρ y β distintos que tengan la misma tasa de aciertos, lo que permite averiguar, de forma fácil y simple, que valores de ρ y β son óptimos.

En la Fig. 2-6. se muestra un ejemplo para cuatro clasificaciones, con dos valores de ρ y β (0,55/0,9, y 0,65/0.9).

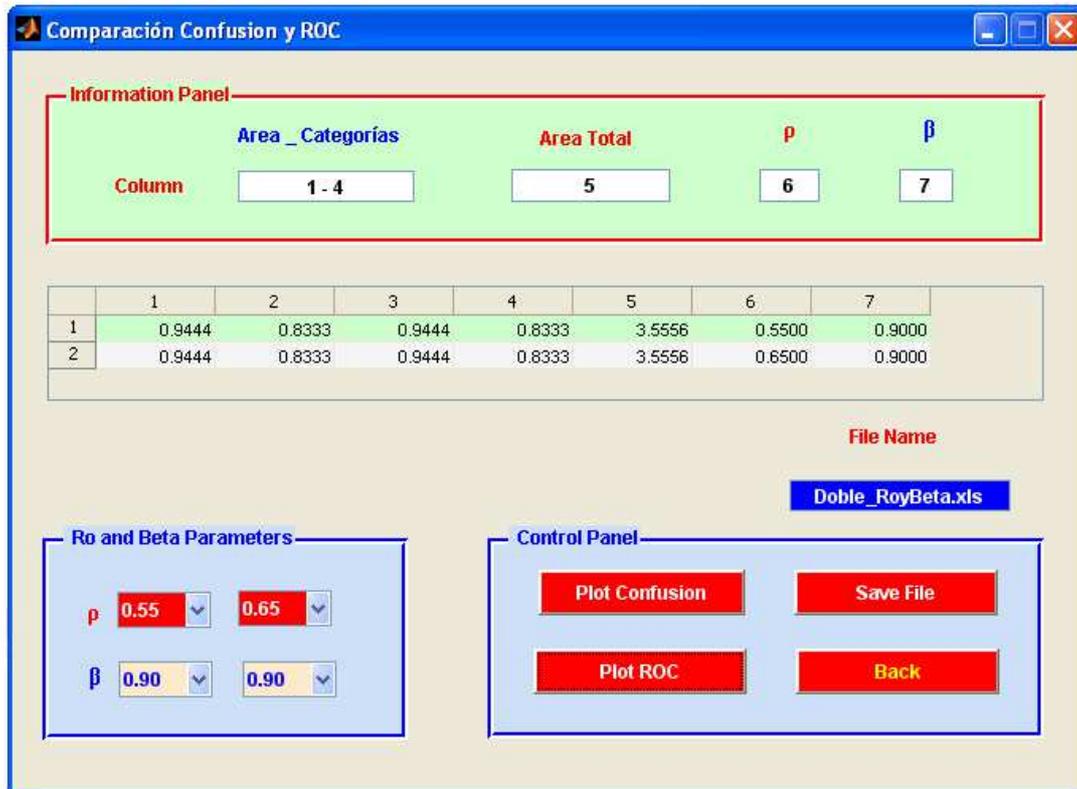


Figura 2-6. Pantalla correspondiente a ARTMAP_DOBLE_ROYBETA.

Los diferentes valores de AUC para cada una de las clasificaciones se muestran en dos filas (columnas 1 a 4), de donde se deduce que para los valores de $\rho=0,55$ y $\beta=0,9$ la sensibilidad y la especificidad es la misma que para $\rho=0,65$ y $\beta=0,9$, debido a que el AUC es el mismo en ambos casos. Por lo tanto, las dos matrices de confusión (Fig. 2-7) y las dos gráficas ROC (Fig. 2-8) serán iguales.

La columna 5 muestra el área total (suma de los cuatro valores de AUC), mientras que las columnas 6 y 7 muestran los valores de ρ y β respectivamente.

Botones del Panel de Control.

El botón **Plot Confusion** permite obtener las dos matrices de confusión (Fig. 2-7), mientras que el botón **Plot ROC**, permite obtener ambas gráficas ROC (Fig. 2-8).

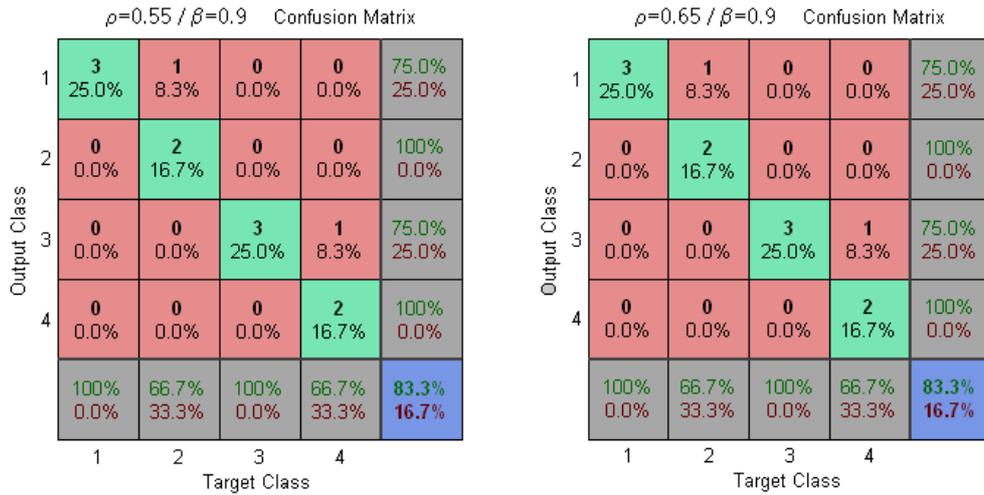


Figura 2-7. Matrices de confusión con misma área AUC.

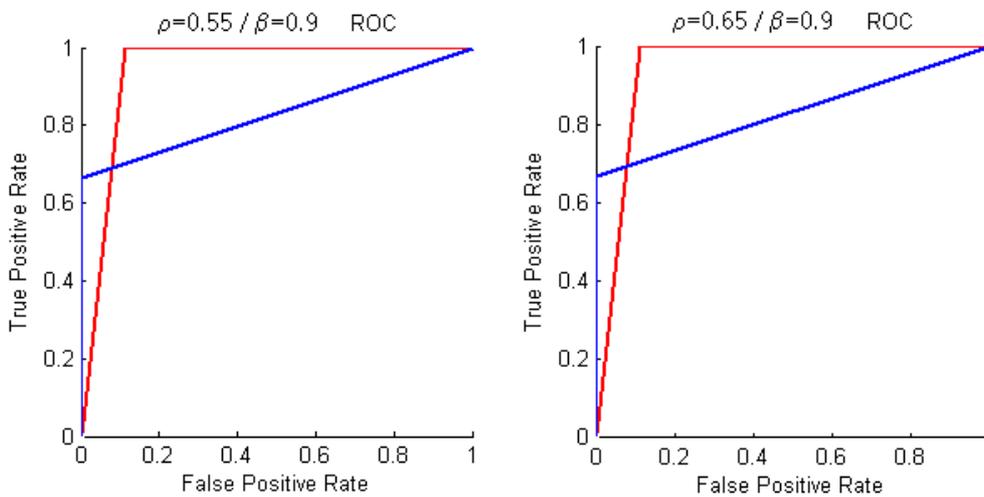


Figura 2-8. Gráficas ROC con la misma área AUC.

Finalmente se puede guardar la tabla mediante el botón **Guardar ("Save File")**. Por defecto se guardan con el nombre que aparece en la Fig. 2-5, pero el usuario puede asignarles el nombre que considere más adecuado, simplemente colocando el cursor sobre el texto, borrando y tecleando el nuevo nombre. No olvide que obligatoriamente la extensión debe ser **.xls**.

En la tabla 2.2 se muestran los diferentes valores de las áreas AUC para cada una de las clasificaciones (columnas 1 a 4) el área total y los valores de ρ y β correspondientes, de donde se deduce que para, los valores de $\rho=0,8$ y $\beta=0,7$ la sensibilidad y la especificidad es óptima, debido a que el AUC mínimo (Área 2 y 4) es mayor que para $\rho=0,75$ y $\beta=0,6$, así como el área total.

Las matrices de confusión y las gráficas ROC correspondientes se muestran a continuación.

Tabla 2.2. Áreas AUC para dos valores de ρ y β						
Área 1	Área 2	Área 3	Área 4	Total	ρ	β
0.889	0.667	1.000	1.000	3.556	0.750	0.600
0.944	0.833	1.000	0.833	3.611	0.800	0.700

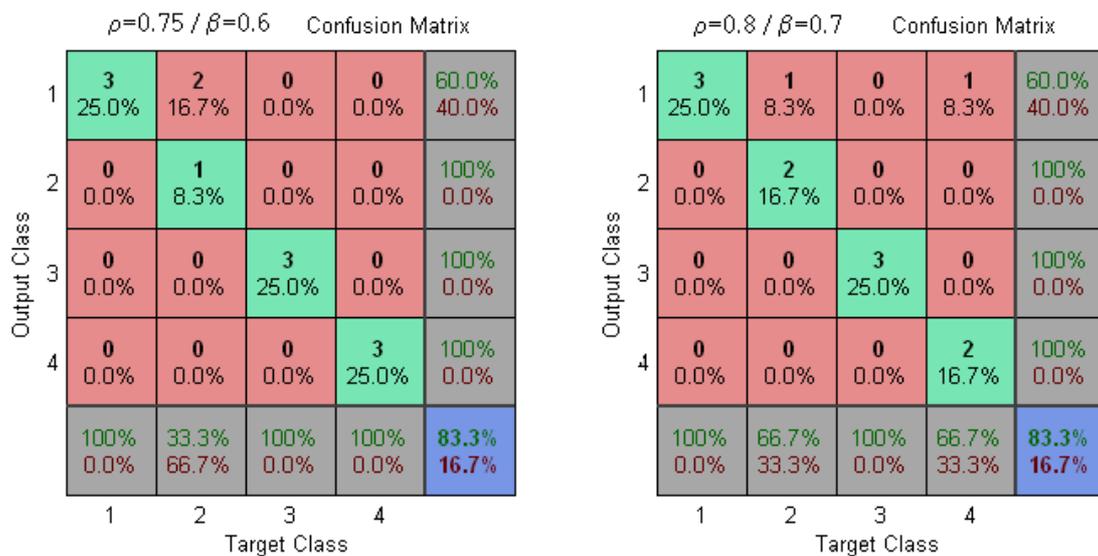


Figura 2-9. Matrices de confusión con distinta área AUC.

El MATLAB permite dibujar las gráficas ROC para cada clase, pero no indica la clase a la que pertenece cada recta, por lo tanto habrá que hacer el análisis a “*posteriori*”.

La matriz de confusión y el área de cada clase nos permite identificar la clase correspondiente a cada recta de la gráfica ROC.

En la matriz de confusión, para $\rho=0,75$ y $\beta=0.6$, se observa que la clase 1 presenta 2 falsos positivos, de un total de 10 ($\text{fpr} = 0,2$, curva azul). En la clase 2, ha habido 1 acierto de un total de 3 ($\text{tpr} = 0,33$, curva verde). Finalmente, las clases 3 y 4 están predichas correctamente, lo que se traduce en un área unidad (tabla 2.2) con una curva ROC situada en la esquina superior izquierda (curva en rojo).

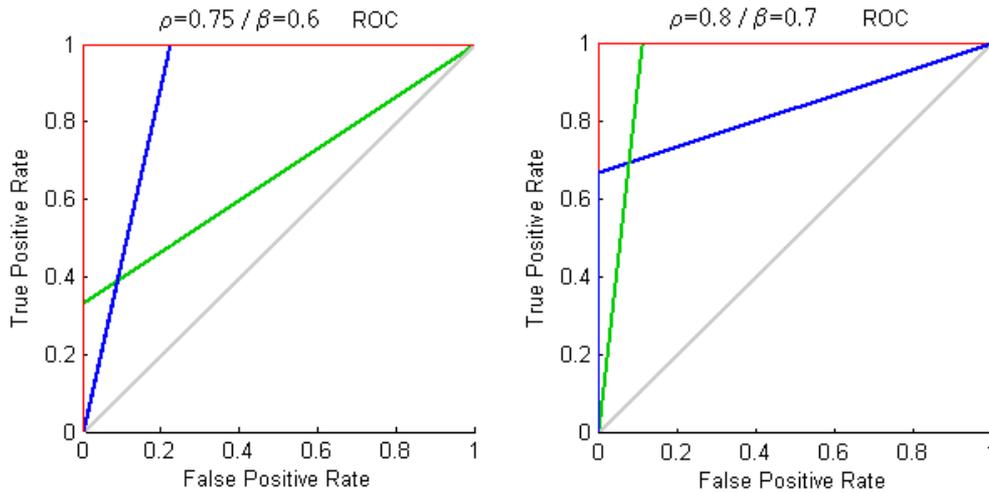


Figura 2-10. Gráficas ROC con distinta área AUC.

En la matriz de confusión, para $\rho=0,8$ y $\beta=0,7$, se observa que en la clase 1, se han predicho correctamente los 3 datos, pero hay dos falsos positivos pertenecientes a las clase 2 y 4 ($\text{fpr} = 0,1$, curva verde). Las clases 2 y 4, en ambas, ha habido 2 aciertos de 3 ($\text{tpr} = 0,66$, curva azul). Finalmente, la clase 3 está predicha correctamente, lo que se traduce en un área unidad (tabla 2.4) con una curva ROC situada en la esquina superior izquierda (curva en rojo).

2.2. Validación Cruzada.

En la validación cruzada de K iteraciones o *K-fold cross-validation* los datos de muestra se dividen en K subconjuntos. Uno de los subconjuntos se utiliza como datos de prueba y el resto ($K-1$) como datos de entrenamiento. El proceso de validación cruzada es repetido durante k iteraciones, con cada uno de los posibles subconjuntos de datos de prueba. Finalmente se realiza la media aritmética de los resultados de cada iteración para obtener un único resultado.

Resulta evidente que si variamos los datos del entrenamiento y validación, la tasa máxima y el mapfield van a ser diferentes en cada entrenamiento. En nuestro caso necesitamos un conjunto de datos concreto que nos dé un mapfield mínimo determinado para implementarlo en el microcontrolador. La validación cruzada nos permite ir variando el conjunto de datos que constituyen el entrenamiento y la validación. De esta forma, obtenemos para unos datos de

entrada concretos, el conjunto que nos da una tasa de aciertos máxima. Con este objetivo se han desarrollado los programas:

- ARTMAP_CRUZADA
- ARTMAP_VALIDACION
- ARTMAP_MUESTRAS

2.2.1. ARTMAP_CRUZADA

Nos permite introducir los valores iniciales, finales y el incremento de los parámetros ρ y β a realizar en el estudio de la red neuronal Fuzzy ARTMAP. La Fig. 2-11 muestra la pantalla correspondiente a la entrada de los parámetros ρ y β . Los datos de entrada son los iniciales, correspondientes al programa denominado ARTMAP_DATOS (Fig. 2-3).

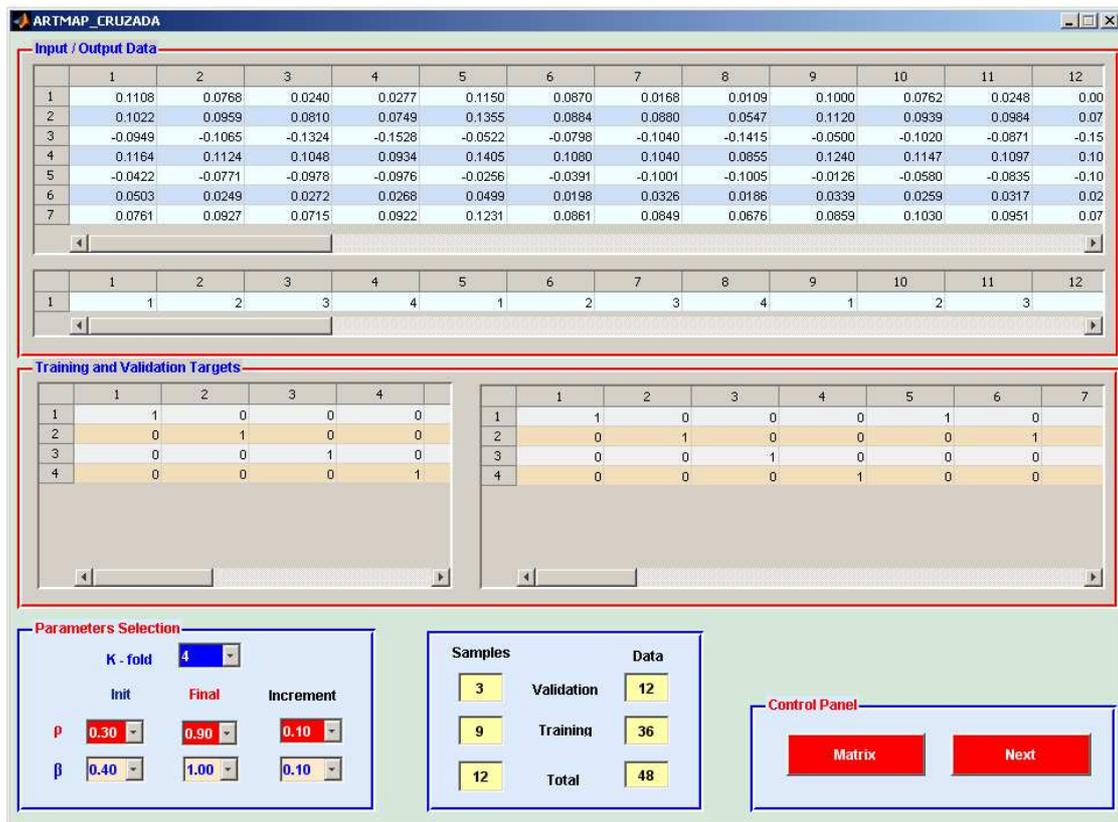


Figura 2-11. Datos y parámetros de entrada para la Validación Cruzada.

Los d datos de entrada se dividen en m muestras, de forma que cada muestra contenga un dato de cada clase. Si c es el número de clases, el número de muestras será:

$$m \text{ (muestras)} = d \text{ (datos)} / c \text{ (clases)}$$

Evidentemente para que el programa funcione correctamente el número de muestras m debe ser un número **entero**.

La tabla 2.3 Indica el formato de los datos de entrada.

Tabla 2.3. Formato de los datos de entrada.									
	Muestra n				Muestra n+1				
Dato	$4n+1$	$4n+2$	$4n+3$	$4n+4$	$4(n+1)+1$	$4(n+1)+2$	$4(n+1)+3$	$4(n+1)+4$...
Clase	1	2	3	4	1	2	3	4	...

Selección de Parámetros.

- **K-Fold.** El programa calcula los divisores del número de muestras y nos lo presenta en un menú desplegable para que el usuario elija las K iteraciones o *K-fold cross-validation*. Como se ha comentado anteriormente, los datos de muestra se dividen en K subconjuntos. Uno de los subconjuntos se utiliza como datos de prueba y el resto ($K-1$) como datos de entrenamiento. El proceso se repite durante k iteraciones, con cada uno de los posibles subconjuntos de los datos de prueba. El programa calcula el número de muestras y datos para el entrenamiento y la validación, así como el total y los presenta en pantalla.
- **Parámetros ρ y β .** Menú desplegable que nos permite seleccionar por filas, los valores iniciales, finales y el incremento de cada uno de los parámetros (columnas).

Botones del Panel de Control.

Por último, click en el botón "**Matrix**" para calcular las matrices u objetivos de entrenamiento y validación que se necesitan para generar el Plotconfusion y el PlotROC y visualizar el botón "**Next**" que nos permite pasar a la pantalla siguiente.

2.2.2. ARTMAP_VALIDACION.

La Fig. 2-12 muestra la pantalla correspondiente. Click en el botón “Cross Validation” para empezar la validación cruzada y realizar las iteraciones seleccionadas en la pantalla anterior en función de los parámetros ρ y β .

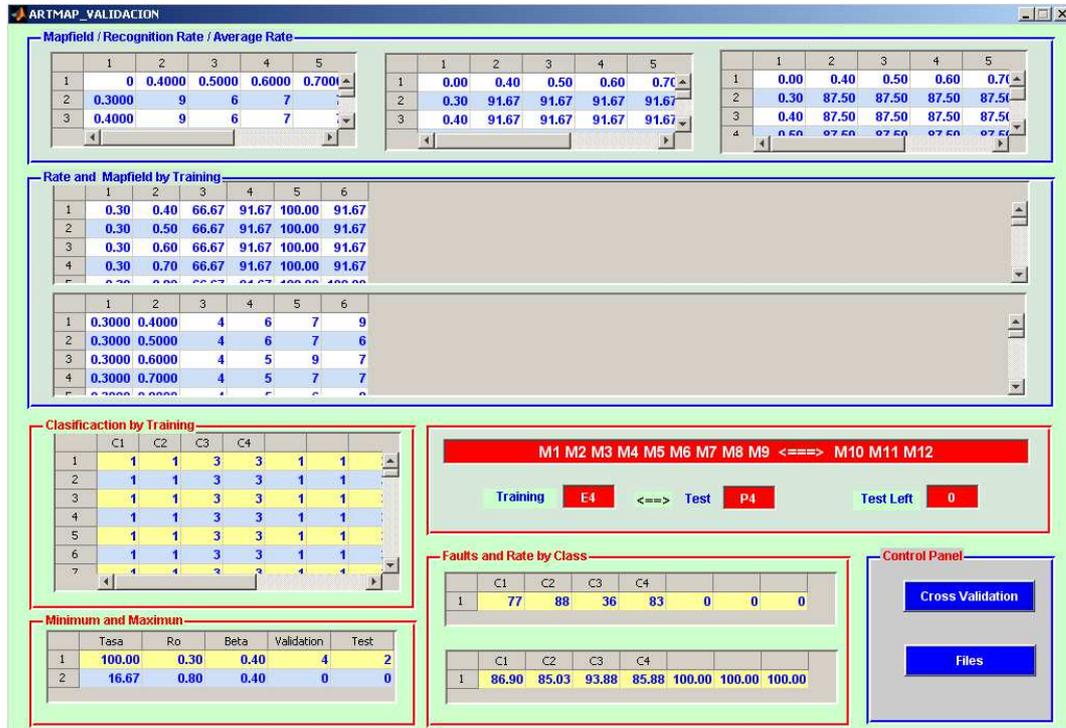


Figura 2-12. Pantalla correspondiente a ARTMAP_VALIDACION.

Tablas. Permiten visualizar previamente los resultados obtenidos en función de ρ y β antes de guardar los datos. De izquierda a derecha y de arriba abajo, son las siguientes:

- **Mapfield.** Nos da el tamaño del mapfield en función de ρ (columnas) y β (filas).
- **Tasa de aciertos (“Recognition Rate”).** Nos informa de la tasa de aciertos en función de ρ y β .
- **Tasa media (“Average Rate”).** Nos informa de la tasa de aciertos media para cada valor de ρ y β .
- **Tasa por entrenamiento (“Rate by training”).** Nos informa de la tasa de aciertos en función de ρ (columna 1) y β (columna 2) para cada una de las iteraciones o entrenamientos.

- **Mapfield por entrenamiento (“Mapfield by training”).** Nos informa del mapfield en función de ρ (columna 1) y β (columna 2) para cada una de las iteraciones o entrenamientos.

Botón **“Files”** permite pasar a la pantalla siguiente (**Fig. 2-13**) para guardar los datos obtenidos.

2.2.3. ARTMAP_MUESTRAS.

Permite guardar los ficheros obtenidos en formato excell y/o texto. Por defecto se guardan con el nombre que aparece en la **Fig. 2-13**, pero el usuario puede asignarles el nombre que considere más adecuado, simplemente colocando el cursor sobre el texto, borrando y tecleando el nuevo nombre. En este caso no hay que incluir la extensión, ya que esta se coloca al elegir el formato en las casillas de verificación **“Type Selection”**.

“Training Files to Save” indican las muestras de la validación en cada uno de los entrenamientos. El fichero que se guarda contiene todos los datos iniciales. En primer lugar los datos de entrenamiento y a continuación los datos de validación.

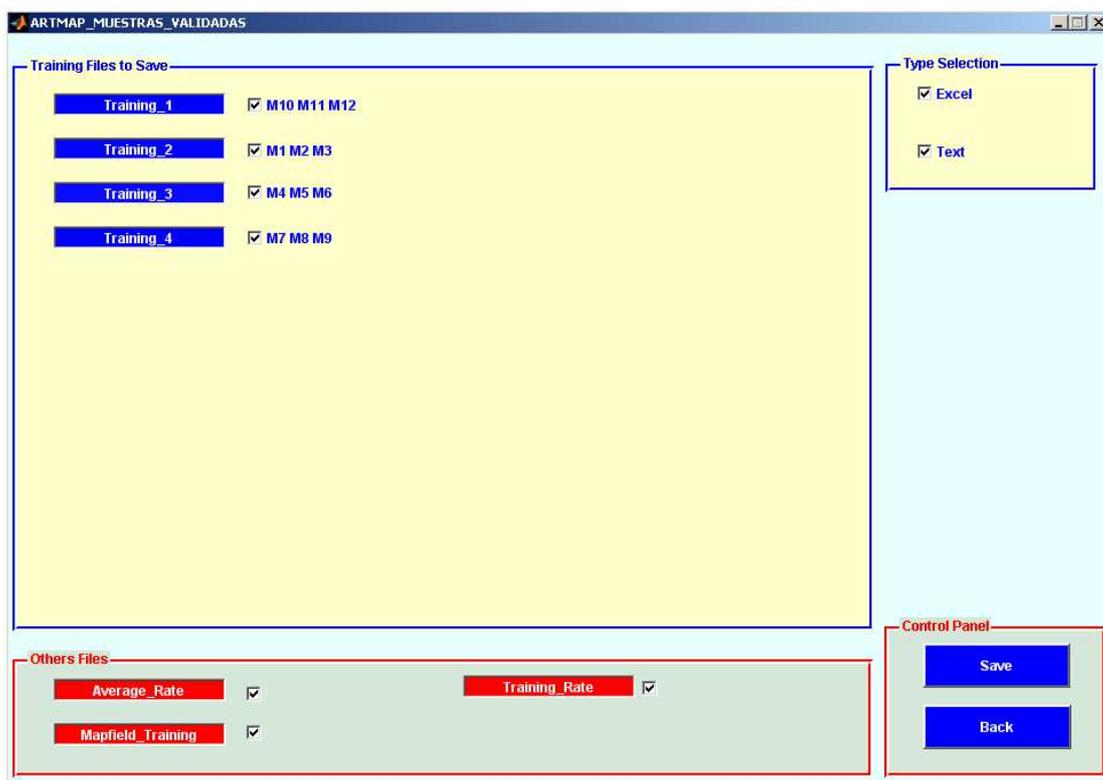


Figura 2-13. Pantalla correspondiente a ARTMAP_MUESTRAS.

Una vez obtenido el conjunto que mejores resultados ha dado, mapfield mínimo y tasa máxima, podemos volver a la entrada de datos, y cargar el conjunto de entrenamiento para realizar un nuevo entrenamiento con un ajuste más fino de los parámetros ρ y β .

3. Selección de Variables.

El objetivo de todo problema de selección de variables es encontrar un subconjunto de ellas que mejor expliquen la clase a la que pertenece cada patrón. Se han desarrollado diversos métodos para tratar de resolver el paradigma de la selección de variables. Como se ha comentado anteriormente, en la presente GUI se ha partido de los métodos desarrollados por Lapedra [Lapedra, 2004] en MATLAB®, por ser fácilmente adaptables y con buenos resultados a la vez que son sencillos y rápidos.

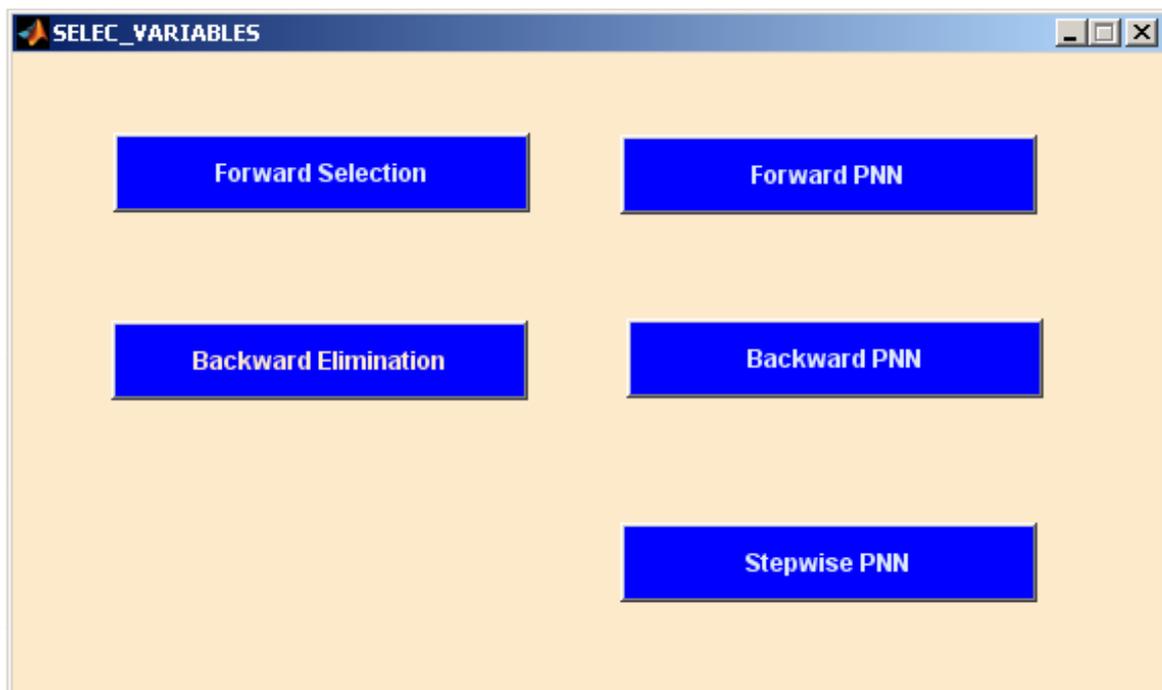


Figura 2-14. Pantalla de Selección de Variables.

En concreto los métodos implementados han sido:

- Forward Selección
- Backward Eliminación
- Forward PNN

- Backward PNN
- Stepwise PNN

Los cuales, se presentan en la **Fig. 2-14**.

Cada uno de los botones de la figura nos lleva a otra pantalla, en la que inicialmente tendremos los datos de entrada, pero las variables están ahora ordenadas por columnas, y los datos de salida, variables seleccionadas por filas. Los programas desarrollados han sido:

- SV_FORWARD
- SV_BACKWARD
- SV_FORWARD_PNN
- SV_BACKWARD_PNN
- SV_STEPWISE_PNN

Estos programas recogen los datos de entrada y pasan los parámetros necesarios y con el formato adecuado a los programas desarrollados por Lapiedra [Lapiedra, 2004] y que previamente han sido modificados para su correcto funcionamiento.

3.1. SV_FORWARD.

En el método “*Forward Selection*” el usuario fija mediante un menú desplegable el número de variables de salida seleccionadas, de 1 a n variables de entrada.

El funcionamiento es sencillo, inicialmente se parte de un modelo sin variables, y se van eligiendo las variables una a una que son más significativas estadísticamente, hasta llegar a un máximo predeterminado o bien hasta que se cumpla cierta condición.

Las principales ventajas son:

- implementación muy simple
- tiempo de cálculo reducido

Los inconvenientes son:

- el número de variables a elegir es arbitrario
- una vez seleccionada una variable no se puede eliminar.

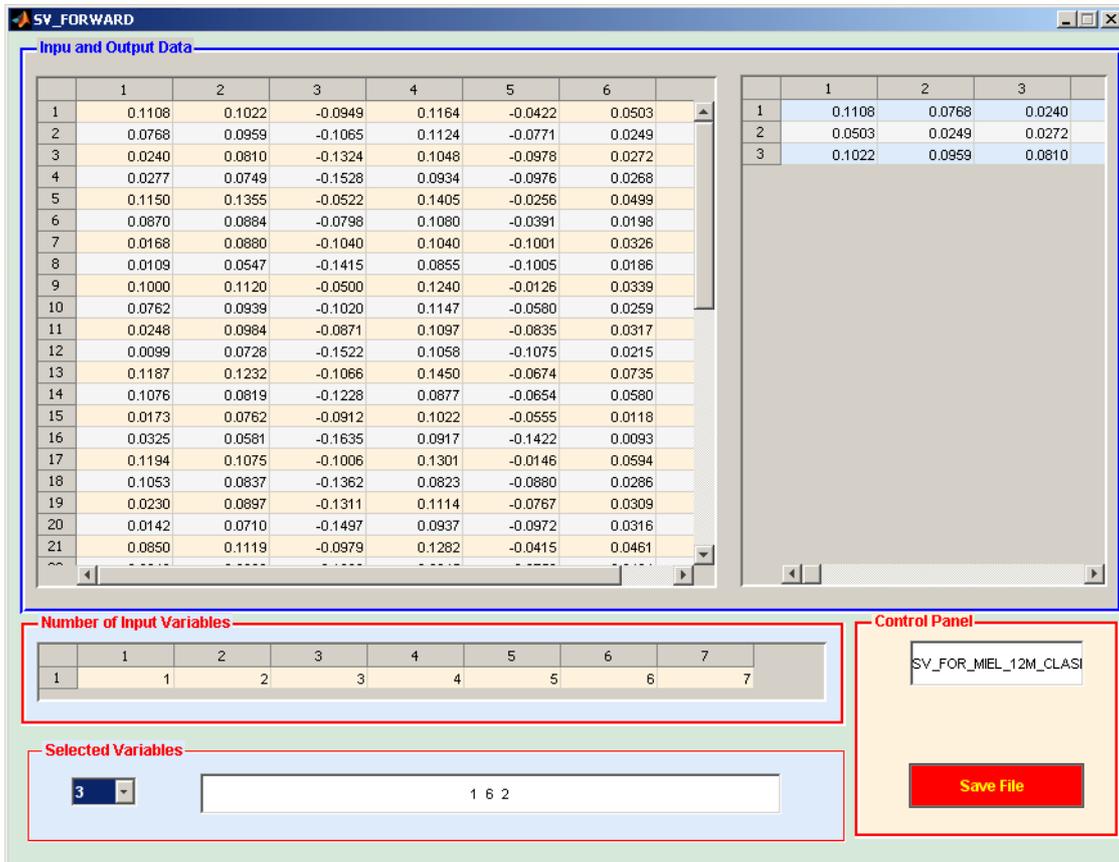


Figura 2-15. Pantalla correspondiente al método Forward Selección.

El nombre del fichero de salida es el mismo que el de entrada, pero añadiendo el método delante, con la extensión .txt:

SV_FOR_NOMBREINICIAL.txt

3.2. SV_BACKWARD.

En el método "*Backward Elimination*" el usuario fija mediante un menú desplegable el número de variables de salida seleccionadas, de 1 a n variables de entrada.

En este método se parte de todas las variables, y se van eliminando las que no son significativas estadísticamente. Al igual que en el método anterior, su implementación es simple, tiempo de cálculo reducido, el usuario elige el número de variables y una vez eliminada una variable no se puede volver a incluir.

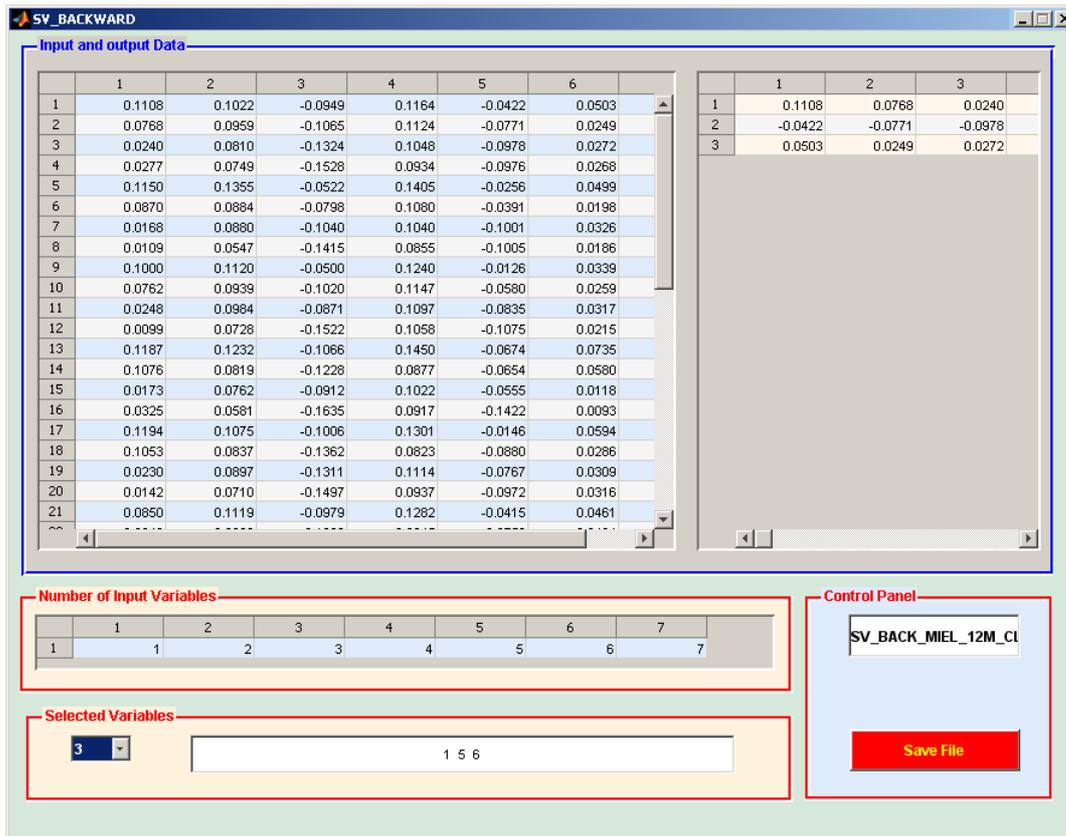


Figura 2-16. Pantalla correspondiente al método “Backward” Eliminación.

El nombre del fichero de salida es el mismo que el de entrada, pero añadiendo el método delante, con la extensión .txt:

SV_BACK_NOMBREINICIAL.txt

3.3. SV_FORWARD_PNN.

Este método de selección es del tipo *forward* en el que se ha utilizado para la validación una red neuronal del tipo Probabilística (“*Probabilistic Neural Network*”).

El nombre del fichero de salida es el mismo que el de entrada, pero añadiendo el método delante, con la extensión .txt:

SV_FOR_PNN_NOMBREINICIAL.txt

En este método las variables de salida se deciden en función de la tasa de aciertos (98%). Se ha incluido en la pantalla el tiempo ejecución (39 s), aunque en este caso, debido a que tenemos pocas variables no es crítico.

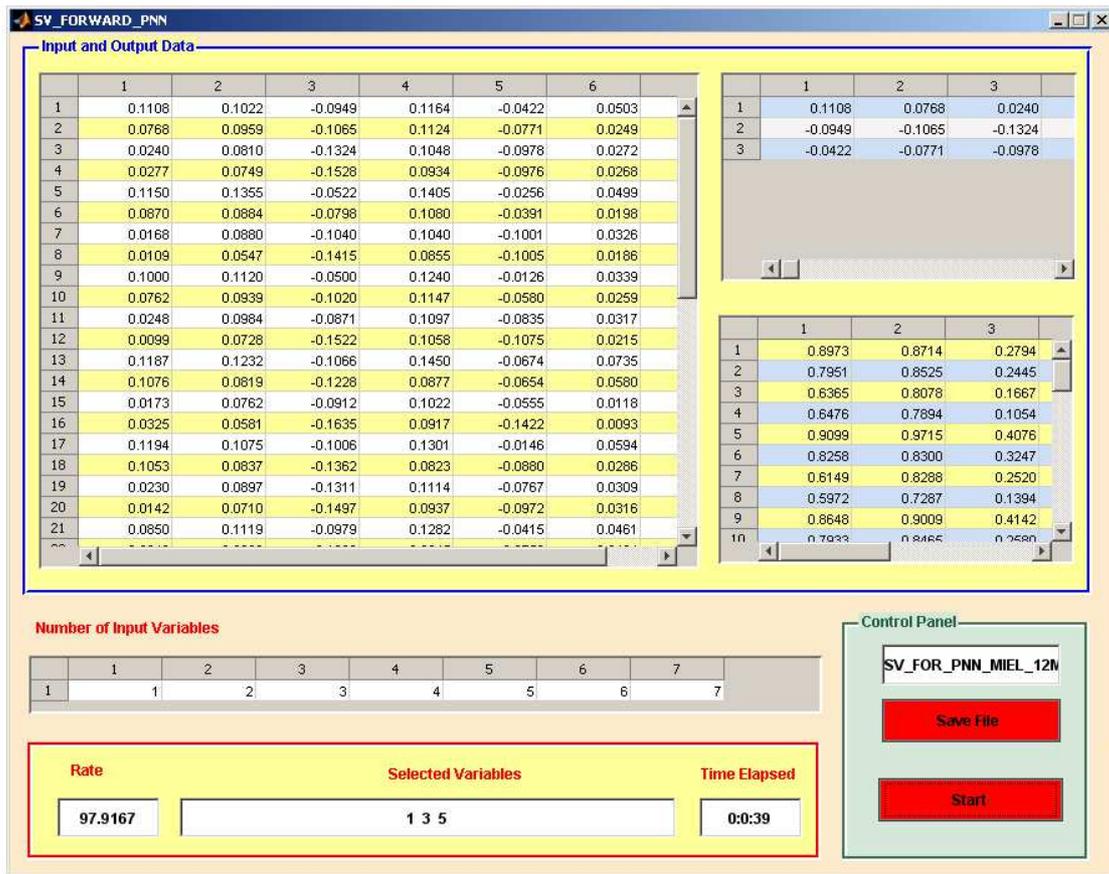


Figura 2-17. Pantalla correspondiente al método *Forward* PNN.

3.4. SV_BACKWARD_PNN.

Este método de selección es del tipo *forward* en el que se ha utilizado para la validación una red neuronal del tipo Probabilística. El nombre del fichero de salida es el mismo que el de entrada, pero añadiendo el método delante, con la extensión .txt:

SV_BACK_PNN_NOMBREINICIAL.txt

En este método las variables de salida se deciden en función de la tasa de aciertos (98%). Se ha incluido en la pantalla el tiempo ejecución, aunque en este caso, debido a que tenemos pocas variables no es crítico.

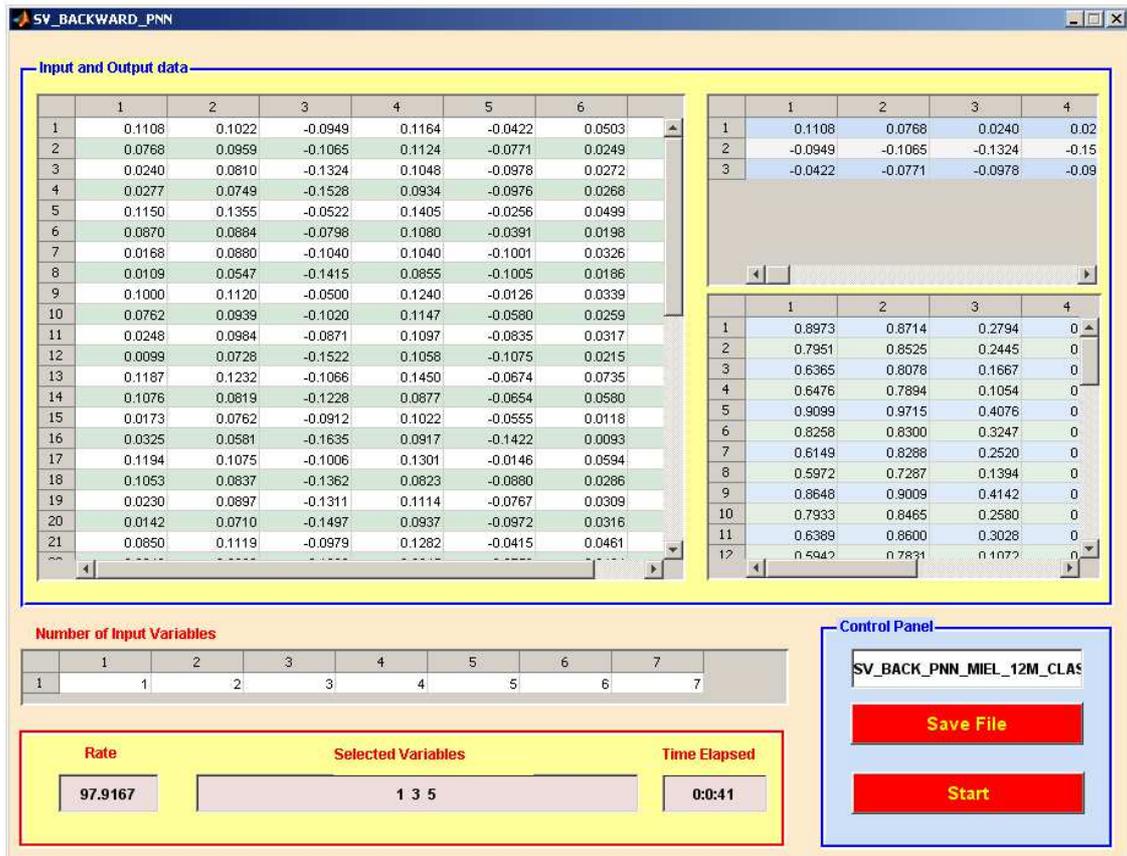


Figura 2-18. Pantalla correspondiente al método *Backward* PNN.

3.5. SV_STEPWISE_PNN.

El método “*Stepwise selection*” es una mezcla de los dos anteriores, primeramente se ejecuta uno (*forward selection*), lo que nos permite seleccionar variables y luego el otro (*backward elimination*), lo que nos permite eliminar variables. El método se ejecuta tantas veces como el usuario quiera, y se valida mediante red neuronal probabilística. Por lo tanto el usuario sólo debe introducir por teclado el número de iteraciones que quiera realizar.

El nombre del fichero de salida es el mismo que el de entrada, pero añadiendo el método delante, con la extensión .txt:

SV_STEPWISE_PNN_NOMBREINICIAL.txt

En este método las variables de salida se deciden en función de la tasa de aciertos. Se ha incluido en la pantalla el tiempo ejecución, aunque en este caso, debido a que tenemos pocas variables no es crítico

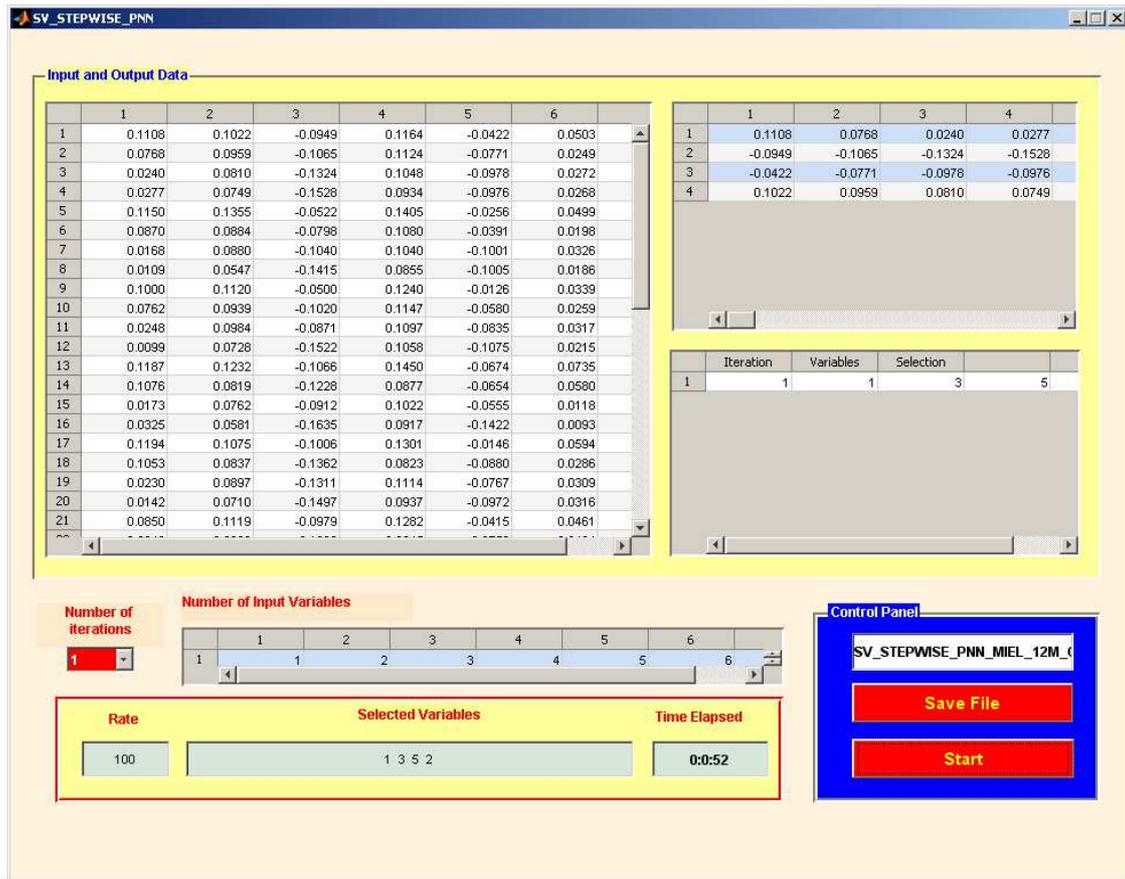


Figura 2-19. Pantalla correspondiente al método *Stepwise* PNN.

4. Implementación de una Red Perceptrón Multicapa en Microcontrolador.

El primer tipo de red neuronal a implementar en un microcontrolador es el Perceptrón multicapa. Esta red puede tener dos tipos de aplicaciones: predicción y clasificación de patrones. Se va a presentar el diagrama de flujo para obtener el algoritmo en cada una de las dos aplicaciones. El algoritmo se desarrolla en el lenguaje de alto nivel C.

4.1. Predicción con red Perceptrón Multicapa.

Para lograr la predicción de un determinado parámetro de salida a partir de los datos de entrada se tienen que realizar las siguientes tareas:

- Rutinas de iniciación del microcontrolador
- Lectura de datos de entrada
- Definición de las funciones a implementar

- Mostrar los resultados en un display

Todo el proceso se puede mostrar en un diagrama de flujo como se indica en la Fig. 2-20.

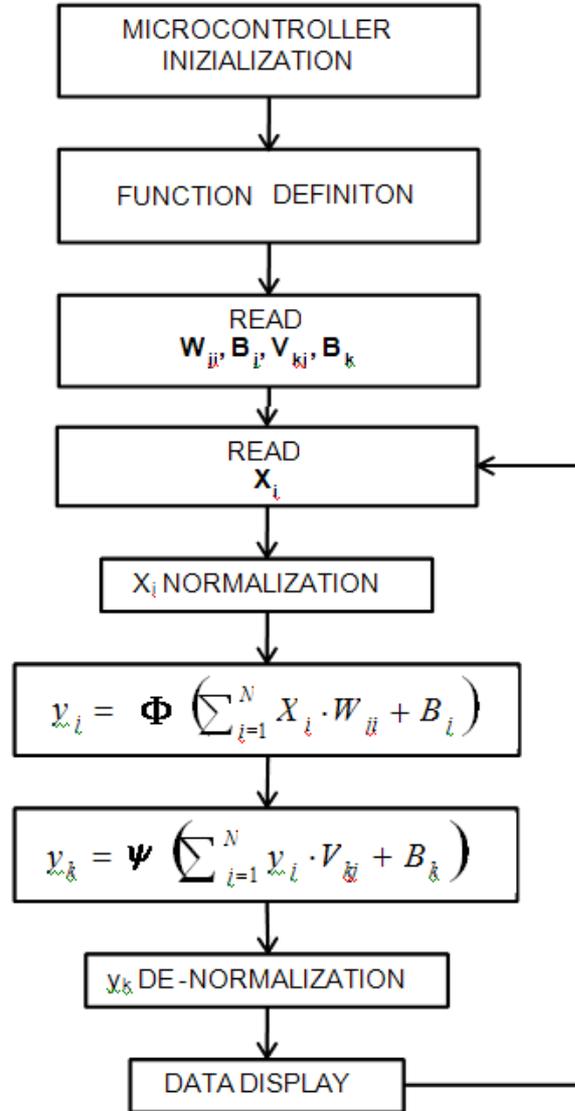


Figura 2-20. Diagrama de flujo en C de la aplicación de predicción de la FF-MLP.

Para el funcionamiento adecuado de la red neuronal los datos de entrada se deben de normalizar entre valores de 0 a 1. Los resultados se deben posteriormente de desnormalizar. La normalización de los datos para el cálculo de los coeficientes se realiza con la misma función de normalización que utiliza MATLAB[®] (Ec. 2.1).

$$y = \frac{(y_{\max} - y_{\min}) \cdot (x - x_{\min})}{(x_{\max} - x_{\min})} + y_{\min} \quad (2.1)$$

Donde x es el valor de entrada, y es el valor de entrada normalizado, Y_{max} e Y_{min} , son los valores máximos y mínimos, respectivamente, del intervalo $[0, 1]$, x_{max} e x_{min} son los valores de entrada máximo y mínimo obtenidos durante el entrenamiento. Los valores x_{max} y x_{min} se pueden cambiar por el algoritmo del microcontrolador en función de las entradas nuevas

Los pesos (W_{ji}) y sesgos (B_j), de entrenamiento de la RNA se obtienen a partir del programa GUI de MATLAB®. Utilizando los coeficientes (X_i) como datos en los registros, así como los pesos y sesgos, el microcontrolador calcula la salida para cada uno de los n nodos ocultos utilizando la siguiente expresión (Ec. 2.2).

$$y_j = \Phi \left(\sum_{i=1}^N X_i \cdot W_{ji} + B_j \right) \quad (2.2)$$

donde:

- Φ es cualquier función de activación.
- i es el nodo de entrada (i : 1 a n).
- j es el nodo oculto (j : 1 a n):

Mediante el uso de los valores de datos y_j , pesos (V_{kj}) y sesgos (B_k), los valores de nodo de salida se obtiene mediante la siguiente expresión (Ec.2.3).

$$y_k = \Psi \left(\sum_{j=1}^N y_j \cdot V_{kj} + B_k \right) \quad (2.3)$$

donde:

- Ψ es cualquier función de activación.
- k es el nodo de salida (k , 1).
- j es el nodo oculto (j : 1 a n):

Los datos de salida se desnormalizan y, finalmente, se muestran en el panel LCD del equipo de medida.

El código de esta rutina se escribe en lenguaje C y se convierte en código hexadecimal utilizando un compilador cruzado. El archivo HEX se descarga en la memoria flash de programa del microcontrolador.

4.2. Clasificación con red Perceptrón Multicapa.

En las aplicaciones de clasificación se desea que la RNA clasifique los datos en categorías preestablecidas. La implementación de la red perceptrón multicapa para aplicaciones de clasificación es muy similar a la implementación para las aplicaciones de predicción y se puede resumir en el siguiente diagrama de flujo (Fig. 2-21).

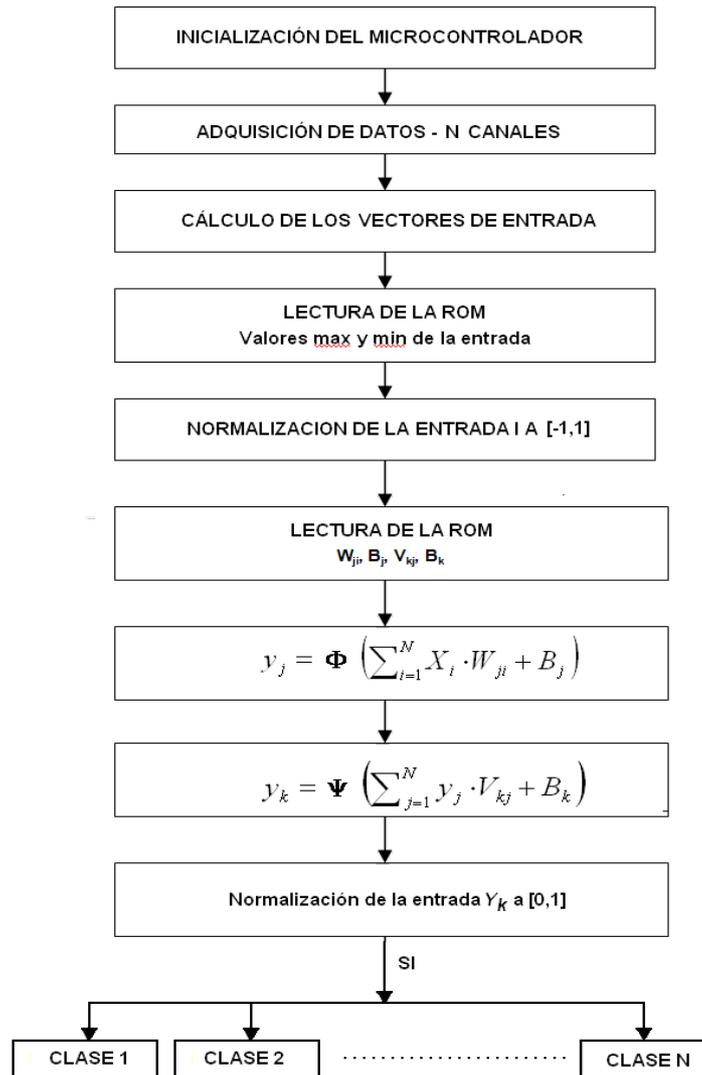


Figura 2-21. Diagrama de flujo en C de la aplicación de clasificación de la FF-MLP.

Las diferencias más importantes con respecto a las aplicaciones de predicción radican en la capa de salida y son:

- el número de neuronas de salida es igual al número de clases
- la función de activación de salida es la misma que a la entrada.

5. Implementación de una red Fuzzy Artmap Simplificada (SFAM) en un microcontrolador.

Para lograr el programa que implemente una red neuronal de tipos SFAM en un microcontrolador se va a seguir las ecuaciones que rigen dichas redes y que fueron resumidas en los diagramas de flujo de las figuras 23 y 24 del capítulo 1. En ellas se indican, respectivamente, los pasos a realizar para el entrenamiento de la red y para la fase no supervisada.

En la rutina de entrenamiento de la red, los n vectores de entrada I , se calculan a partir del sistema de adquisición de datos. Dichos datos son normalizados para establecer su rango de $[0,1]$, con la misma función de normalización de MATLAB®, (Ec. 2.4)

$$I_N = \frac{(Y_{max} - Y_{min})(I - I_{min})}{I_{max} - I_{min}} + Y_{min} \quad (2.4)$$

Donde I es el valor de entrada, I_N es el valor de entrada normalizado, Y_{max} e Y_{min} , son los valores máximos y mínimos, respectivamente, del intervalo $[0,1]$, I_{max} e I_{min} son los valores de entrada máximo y mínimo obtenidos durante el entrenamiento. Los valores I_{max} e I_{min} se pueden cambiar por el algoritmo del microcontrolador en función de las entradas nuevas.

Con el fin de preservar la amplitud de la información el dato de entrada se complementa mediante la Ec. (2.5).

$$I_N = (a, a^c) = (a_1, \dots, a_M, a_1^c, \dots, a_M^c) \quad (2.5)$$

Donde:

$$a_i^c = 1 - a_i \quad (2.6)$$

En un sistema Fuzzy Artmap el proceso de búsqueda ocurre de la siguiente manera:

- 1) Para cada entrada I , se elige la categoría J para la que T_J es máxima

$$\text{—————} \quad (2.7)$$

Siendo W_j , los pesos obtenidos durante el entrenamiento.

- 2) Si la categoría elegida cumple el criterio de vigilancia (Ec. 2.8), se produce la resonancia, y la entrada I , se clasifica en la clase J .

$$\frac{|I\Delta W_j|}{d} \geq \rho \quad (2.8)$$

3) Si la categoría elegida no cumple el criterio de vigilancia (E. 2.9), la clase J se desactiva y se repiten los pasos 1) y 2). Es decir, se elige la siguiente entrada que maximice T_J (Ec. 2.8), y se comprueba nuevamente la resonancia.

4) Por último, si no hay una función de concordancia T_J que verifique el criterio de la vigilancia, el vector de entrada queda clasificado como fuera de rango.

Si hay resonancia entonces el vector de entrada queda clasificado. La elección de la categoría está indexada por el nodo J . Este índice J son los puntos máximo y mínimo de la clase en el "mapfield". La clase se muestra en la pantalla LCD.

La función de clasificación se implementa en el microcontrolador. Esta rutina es escrita en el lenguaje C y convertida en código hexadecimal usando un compilador cruzado. El archivo HEX se descarga en la memoria flash del microcontrolador.

6. Implementación en el microcontrolador.

Para materializar los algoritmos de las redes neuronales artificiales se ha procurado utilizar dispositivos de microcontroladores comunes para que así sea fácil y económico su desarrollo.

En la aplicación más sencilla y que además fue la primera en realizarse (medida humedad con sensor capacitivo) se ha utilizado el PIC16F877 que es un microcontrolador de la familia PIC18/8-bit y tiene 368 bytes de RAM y 8 KB de memoria flash reprogramable. Las características principales quedan reflejadas en la Tabla 2.5.

Tabla 2.5. Características del μ C PIC16F877.	
Parámetro	Valor
Program Memory Type	Flash
Program Memory (KB)	8
CPU Speed (MHz)	20
Data Memory (bytes)	368
Data EEPROM (bytes)	256
I/O Ports	Ports A,B,C,D,E
Capture/Compare/PWM Peripherals	2
Timers	3 x 16-bit
ADC	8 ch, 10-bit
Serial Communications	MSSP, USART
Parallel Communications	PSP

EL μ C Pic16F877 pertenece a la gama media de Microcontroladores con un total de 35 instrucciones, de 14 bits cada una (Todas las instrucciones son de un solo ciclo con excepción de las ramificaciones del programa que son de dos ciclos). Disponen de 14 fuentes posibles de interrupciones y una pila de 8 niveles que permite el anidamiento de subrutinas. Cuenta con una arquitectura hardware en la que son independientes la memoria de instrucciones y la memoria de datos y cada una dispone de su propio sistema de buses para el acceso.

El μ C PIC18F877 dispone de tres tipos de memorias:

- Memoria de programa: memoria EEPROM en donde almacenaremos el programa, las constantes y los datos. Esta memoria solo podrá ser leída por el pic.
- Memoria RAM de datos: en cuyos registros se irán almacenando los valores de las variables de forma temporal durante la ejecución del programa.
- Memoria EEPROM de datos, es un espacio de memoria EEPROM en la que se pueden guardar variables que queremos conservar aunque se apague el dispositivo.

Para el resto de los capítulos se ha escogido un dispositivo de mayores prestaciones, fundamentalmente por el tamaño de la memoria tanto de datos como de programa. Concretamente se ha elegido un sistema integrado basado en un microcontrolador **PIC18F4550**. Este dispositivo es de la familia PIC18/8-

bit y tiene 2 KB de RAM y 32 KB de memoria flash reprogramable. Las características principales se describen en la Tabla 2.6.

Tabla 2.6. Características del μ C PIC18F4550.	
Parámetro	Valor
Program Memory Type	Flash
Program Memory (kB)	32
CPU Speed (MIPS)	12
RAM (Bytes)	2048
Data EEPROM (Bytes)	256
Digital Communication Peripherals	1-A/E/USART, 1-MSSP(SPI/I2C)
Capture/Compare/PWM Peripherals	1 CCP, 1 ECCP
Timers	1 x 8-bit, 3 x 16-bit
ADC	13 ch, 10-bit
Comparators	2
USB (ch, speed, compliance)	1, Full Speed, USB 2.0

El μ C PIC18F4550 dispone de las siguientes memorias:

- Memoria de programa: memoria flash interna de 32.768 bytes, en la que se almacenan las instrucciones, las constantes y los datos.
- Memoria RAM de datos: memoria SRAM interna de 2048 bytes en la que están incluidos los registros de función especial y en donde se almacenan datos de forma temporal durante la ejecución del programa. Puede ser escrita/leída en tiempo de ejecución mediante diversas instrucciones
- Memoria EEPROM de datos: memoria no volátil de 256 bytes que almacena datos que se deben conservar aun en ausencia de tensión de alimentación. Puede ser escrita/leída en tiempo de ejecución a través de registros
- Dispone de una Pila formada por un bloque de 31 palabras de 21 bits en donde se almacena la dirección de la instrucción que debe ser ejecutada después de una interrupción o subrutina
- Memoria de configuración: memoria en la que se incluyen los bits de configuración (12 bytes de memoria flash) y los registros de identificación (2 bytes de memoria de sólo lectura).

El μ C PIC18F4550 dispone de buses diferentes para el acceso a memoria de programa y memoria de datos (arquitectura Hardware). Esto permite acceder simultáneamente a la memoria de programa y a la memoria de datos.

Por otro lado, es de destacar que algunos de los sistemas de medida para las aplicaciones desarrolladas en la tesis doctoral también han utilizado sistemas de microprocesadores que han sido desarrollados previamente por otros investigadores del instituto IDM. Por ello, el desarrollo de cada equipo de medida y los programas utilizados quedan fuera del alcance de la presenta tesis doctoral.

7. Producción científica.

Capítulo de Libro

Eduardo Garcia-Breijo, Jose Garrigues and Luis Gil-Sanchez

Book Title: *Pattern Recognition: Practices, Perspectives and Challenges*

Chapter Title: Optimization of an Embedded Simplified Fuzzy ARTMAP Implemented on a Microcontroller using MATLAB GUI Environment

NOVA SCIENCE PUBLISHERS, INC. 2013. **ISBN: 978-1-62618-196-0**

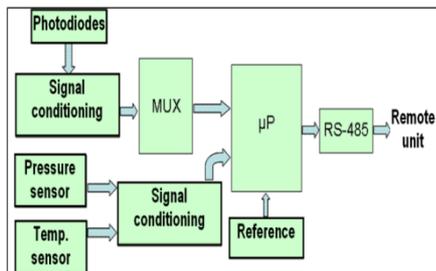
Congreso.

J. Garrigues Baixauli, E. García Breijo, L. Gil Sánchez. *Interface Gráfica de Usuario (GUI) para desarrollo de redes neuronales artificiales de tipo Fuzzy Artmap*. VI Workshop on Sensors and Molecular Recognition. Valencia – España. Julio 2012.

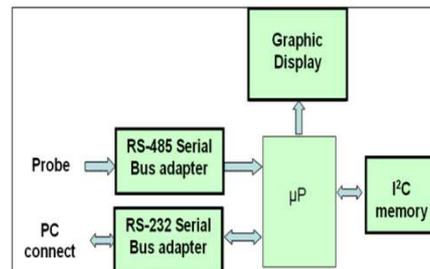
Capítulo 3

Aplicación de la Red FF-MLP en Microcontroladores de 8 Bits para Calcular la Profundidad de Secchi

Sonda sumergible



Registrador de datos y unidad de control



En este capítulo se describe una aplicación predictiva de una red neuronal artificial implementada en un microcontrolador de 8 bits que permite calcular la denominada profundidad del disco Secchi, tanto en aguas profundas como en aguas poco profundas a partir de los datos procedentes de un sistema de medida diseñado y fabricado en el Instituto IDM denominado LUZEX.

Como el tipo de aplicación que se va a realizar es de carácter predictivo se va utilizar una red Perceptrón Multicapa con propagación hacia delante (FF-MLP, *Feed Forward MultiLayer Perceptron*),

1. Introducción.

La turbidez es uno de los parámetros más importantes a ser monitorizados en el agua, tanto en aguas continentales como en agua marina. Esto se debe al hecho de que la turbidez influye directamente en la presencia de las comunidades de algas en el agua. La turbidez en un cuerpo de agua depende de la concentración de sólidos en suspensión, de la materia disuelta, del fitoplancton y de las sustancias amarillas presentes en el agua.

Para medir la turbidez del agua se suele utilizar de modo tradicional el denominado disco Secchi que se compone de un círculo con un diámetro de 20 cm, dividido en cuadrantes que alternativamente se pintan en blanco y negro, y una cuerda graduada para medir la profundidad (**Fig. 3-1**). El disco se sumerge en el lado sombreado del barco poco a poco aumentando su profundidad, mientras pueda ser visto por el operador. Cuando no se puede ver, se mide la profundidad y se repite la operación inversa: el disco Secchi mientras sigue siendo invisible en el agua, se sube lentamente hasta que se hace de nuevo visible, esta segunda profundidad también se mide. Entonces, el promedio de estas dos medidas (desaparición y aparición del disco Secchi) se utiliza para obtener un valor relacionado con la claridad del agua que se denomina "profundidad de Secchi "(DS).

Este sistema de medida tiene la ventaja de que es una manera fácil, rápida e "in situ" de determinar la turbidez en el agua y se utiliza a menudo por la comunidad científica para determinar la turbidez en las masas de aguas costeras y continentales. Pero su uso es a veces limitado y hay algunos casos particulares que la hacen difícil de usar o ineficiente.



Figura 3-1. Determinación de la profundidad de Secchi

Existen varios inconvenientes al utilizar el disco Secchi. El más importante es la subjetividad del operador, debido a que la percepción del disco a una profundidad determinada depende en gran medida de la agudeza visual de quien realiza las mediciones. Algunos autores estiman que esta subjetividad es en el 5% -10% [Preisendorfer, 1986] [Megard & Bergman, 1989]. Pero este no es el único inconveniente, además, la visibilidad del disco Secchi depende también de factores ambientales tales como el estado de las aguas superficiales, la inclinación solar o el momento en que se efectúa la medición.

Otro gran inconveniente, sobre todo cuando se pretende relacionar la calidad del agua con la profundidad de Secchi, es la imposibilidad de medir la profundidad en aguas poco profundas (ríos, playas, etc.). En estos casos específicos, el disco Secchi alcanza el fondo, pero todavía puede ser visto por el operador. Entonces, la profundidad de Secchi se hace imposible determinar por este método.

Con el fin de minimizar los inconvenientes mencionados, el Instituto de Reconocimiento Molecular y Desarrollo Tecnológico (IDM) de la UPV ha desarrollado, construido y patentado (Ref patente nº P200803488) un procedimiento alternativo para determinar la profundidad de Secchi sin utilizar el disco. La idea es utilizar un nuevo dispositivo electrónico para calcular la extinción de la luz del sol y la transparencia en los cuerpos de agua. Este dispositivo calcula los coeficientes de extinción de la luz solar en las columnas de agua con cinco longitudes de onda diferentes (tres para el rango visible, los rayos UV e IR). A partir de estas medidas podemos obtener las concentraciones relativas de los

componentes de agua más comunes que se relacionan con la extinción de la luz del sol:

- el fitoplancton,
- las sustancias amarillas
- sólidos en suspensión.

Este dispositivo es muy útil para la determinación de turbiedad en el agua, ya que elimina dos de las cuestiones planteadas por el disco Secchi: por un lado, se elimina la subjetividad de medición, ya que el valor final ya no depende de la observación del operador que gestiona el disco y, por otra parte, se puede utilizar en aguas poco profundas, donde el disco Secchi alcanza el fondo sin que se quede fuera de la vista del operador [Ibáñez *et al.*, 2010].

Con los datos obtenidos con el sistema de medida descrito se va a utilizar una red neuronal artificial (RNA) del tipo FF-MLP para predecir la profundidad de Secchi en función de los coeficientes de extinción de la luz solar en las columnas de agua.

La RNA se implementa como una rutina de software adicional, entrenándose en un PC por medio de MATLAB® y utilizando datos reales de la unidad foto-sensora. Una vez entrenada, la rutina de software que implementa el procedimiento de linealización, se programa en el PC [Medrano & Martin, 2005].

2. Descripción del sistema de medida óptico (LUZEX).

El coeficiente de absorción de agua se considera una propiedad inherente óptica. Es difícil de medirla debido a que requiere la eliminación de los efectos de interferencia del sistema de medición en sí y no es fácil de obtener muestras de agua de pureza adecuada. Es más fácil medir las propiedades ópticas aparentes, tales como la atenuación difusa cuya relación de desmultiplicación determina la energía disponible para la fotosíntesis con la profundidad [Tetsuhiro & Naonobu, 1999] [Piazena *et al.*, 2002].

Durante el desarrollo del sistema de medida LUZEX, [Ibáñez, 2010] la idea de los autores fue obtener un coeficiente de absorción de la luz en el agua

(aw) por medio de fotodiodos comerciales. El modelo se basa en el peso del espectro de absorción del agua propuesto en la literatura, el espectro de la radiación solar y la sensibilidad de los fotodiodos en el rango de longitud de onda de medición.

Para lograr un amplio barrido del espectro de las las emisiones solares, se han usando cinco fotodiodos sensibles a cinco diferentes bandas: ultravioleta (AUV), azul (AB,), verde (AG), rojo (AR) y el infrarrojo cercano (AIR). No se obtiene el valor de $w(\lambda)$ para todas las longitudes de onda, pero sí se obtienen los coeficientes de absorción de luz por cualquier cuerpo de agua en la banda de sensibilidad de cada fotodiodo: ultravioleta (AUV), azul (AB), verde (AG), rojo (AR) y el infrarrojo cercano (AIR).

El instrumento consta de dos unidades físicas:

- La sonda: sumergible en el agua y que incluye los fotodiodos, transductor de temperatura y un sensor de presión y
- El registrador de datos: que los recibe de la sonda sumergida y guarda los datos recopilados.

2.1. La sonda

La sonda (Fig. 3-2) utiliza fotodiodos comerciales cuya sensibilidad espectral, aunque limitada, no es monocromática. Comprende ultravioleta infrarrojo, y tres receptores de la radiación visible (longitudes de onda roja, verde y azul), también están incluidos un transductor de temperatura y un sensor de presión. Todos los sensores son controlados por un microcontrolador Microchip PIC16F876.

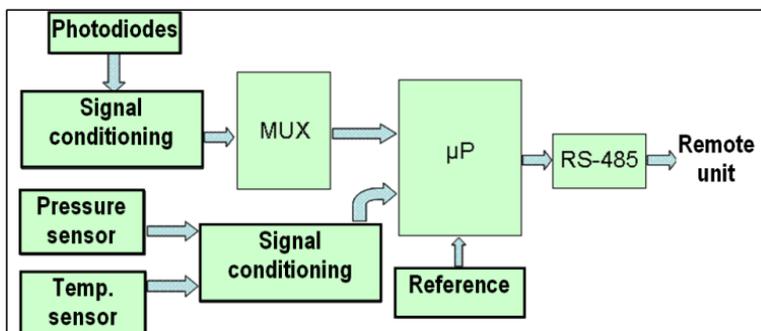


Figura 3-2. Diagrama de bloques de la sonda y Sonda sumergible.

2.2. Registrador de datos y unidad de control

La información recibida se almacena en la unidad de registro de datos. Esta unidad también incluye la presentación de datos que muestra la cantidad de luz recibida por cada fotodiodo, la profundidad y la temperatura (Fig. 3-3). La caja también contiene la fuente de alimentación y los interruptores de navegación para los menús. Esta unidad está controlada por un microcontrolador PIC16F4550 de Microchip. Este microcontrolador calcula los coeficientes de extinción de los fotodiodos, que son los datos de entrada de la red neuronal implementada.

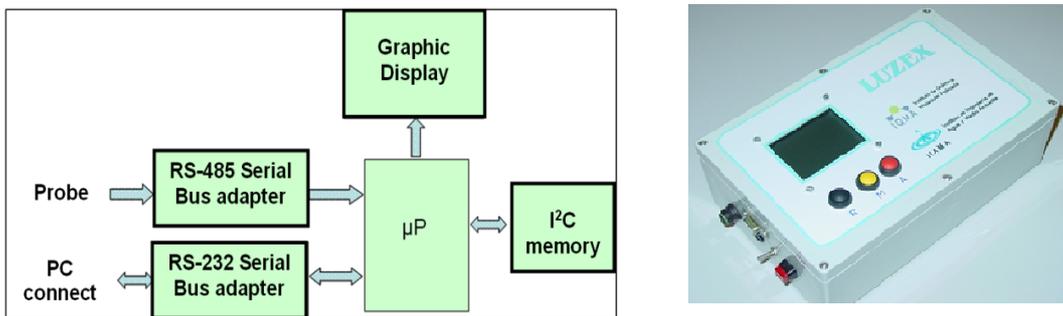


Figura 3-3. Diagrama de bloques para la unidad remota y Registrador de datos y visualización

Los fotodiodos no son estrictamente selectivos a una determinada longitud de onda sino que cubren un espectro más o menos amplio de longitudes de onda. Por lo tanto, es necesario calcular un coeficiente de extinción para el ancho de banda de cada fotodiodo y que éste debe ser equivalente al proporcionado por el proveedor, que varía con la longitud de onda.

Así, el coeficiente de absorción específico para cada fotodiodo, (λ) , es la cantidad de área cubierta por el producto de la sensibilidad del sensor normalizada ($S(\lambda)$) y el coeficiente de absorción de agua para cualquier ($w(\lambda)$) longitud de onda en el rango de longitudes de onda sensible, de acuerdo con la siguiente ecuación:

$$a_{\lambda} = \frac{1}{\lambda_{m\acute{a}x} - \lambda_{m\acute{i}n}} \int_{\lambda_{m\acute{i}n}}^{\lambda_{m\acute{a}x}} S'(\lambda) \cdot a_w(\lambda) \cdot d\lambda \quad (3.2)$$

En esta expresión, λ_{max} y λ_{min} son las longitudes de onda en la que la sensibilidad de cada fotodiodo es el 10% de la máxima sensibilidad.

Para la determinación en tiempo real del coeficiente de extinción de la luz solar en el agua, se utiliza comúnmente el disco Secchi. De hecho, algunos investigadores proponen la expresión (3.1) para calcular a partir de datos del disco Secchi [Jamu & Piedrahita, 1999].

$$K_d = \frac{K_e}{SD} \quad (3.1)$$

En donde, K_e es una constante cuyo valor cambia entre 1.1 y 2.0 [Jamu & Piedrahita, 1999] y SD es la profundidad de Secchi.

En nuestro caso usamos la relación inversa, ya que queremos determinar la profundidad de Secchi a partir de los coeficientes de extinción. En primer lugar, debemos entrenar la red neuronal con los datos suministrados (un coeficiente de extinción para cada una de las cinco bandas de longitud de onda del espectro). En el ancho de banda de cada fotodiodo, el coeficiente de extinción obtenido "in situ" para la luz del sol, se puede comparar con el valor calculado a partir de las especificaciones de cada fotodiodo comercial usado.

2.3. Medidas.

El objetivo de las experiencias es comprobar el dispositivo en condiciones de agua diferentes. De esta manera, se seleccionaron tres puntos de muestreo: cerca de la costa del mar, la laguna costera y una zona media-baja de río. Estos tres sistemas acuáticos están lo suficientemente cerca de Valencia) para garantizar un transporte fácil y campañas de muestreo económicas. Se han seleccionado tres masas diferentes de agua para desarrollar este estudio: mar Mediterráneo, río Turia y laguna costera (Albufera).

Se han utilizado un total de 25 muestras para generar los pesos y sesgos (Tabla 3.1).

Tabla 3.1 Coeficientes de extinción en función del tipo de agua.							
	Nombre	Kuv	Kr	Kg	Kb	Kir	Profundidad Secchi [cm]
MAR	PAL1	0.055	0.022	0.019	0.032	0.067	90
	PAL2	0.015	0.009	0.007	0.011	0.041	110
	PAL3	0.015	0.011	0.008	0.01	0.053	160
	PAL4	0.009	0.006	0.004	0.006	0.035	130
	PAL5	0.008	0.007	0.004	0.004	0.041	190
	PAL6	0.006	0.005	0.003	0.003	0.037	200
	PAL7	0.006	0.005	0.003	0.004	0.041	250
RIO	TU1	0.029	0.021	0.018	0.02	0.052	70
	TU2	0.038	0.027	0.02	0.022	0.076	65
LAGUNA COSTERA	AL1	0.23	0.107	0.117	0.144	0.115	15
	AL2	0.202	0.109	0.119	0.147	0.117	15
	AL3	0.175	0.045	0.051	0.082	0.082	27
	AL4	0.139	0.073	0.073	0.123	0.074	17
	AL5	0.127	0.058	0.054	0.078	0.083	22
	AL6	0.106	0.07	0.068	0.109	0.072	19
	AL7	0.094	0.068	0.067	0.1	0.071	18
	AL8	0.098	0.049	0.052	0.078	0.08	23
	AL9	0.072	0.038	0.041	0.061	0.07	23
	AL10	0.133	0.08	0.076	0.11	0.08	20
	AL11	0.056	0.009	0.01	0.016	0.026	27
	AL12	0.112	0.024	0.025	0.038	0.052	32
	AL13	0.15	0.043	0.04	0.081	0.08	13
	AL14	0.155	0.048	0.044	0.167	0.065	13
	AL15	0.089	0.036	0.036	0.004	0.073	15
	AL16	0.096	0.028	0.028	0.044	0.055	25

3. Análisis de datos.

La etapa de entrenamiento se realiza con una parte de las medidas disponibles. En esta etapa se establece la salida de la red (en nuestro caso la profundidad de Secchi). Como vector de entrada se aplican los datos de los cinco coeficientes de extinción para cada medición. Con estos datos se calculan los coeficientes del algoritmo que configuran la red. En la etapa de

verificación, los datos de las nuevas medidas se aplican a las entradas, comprobando la salida.

Para el entrenamiento de las redes se ha utilizado el programa MATLAB 2010b®, que se ejecuta en un ordenador PC. El uso de ordenador está determinado por su potencia de cálculo y facilidad de aplicación de los algoritmos de redes neuronales. En cambio, la etapa de verificación se realiza íntegramente en microcontroladores.

Mediante esta forma de trabajar, una vez que la etapa de entrenamiento ha concluido, el sistema desarrollado puede trabajar independientemente de un PC. Esta es una de las principales características de los equipos presentados en esta tesis.

Para realizar la verificación se utilizan los resultados obtenidos en la etapa de entrenamiento, que corresponden a los coeficientes de los algoritmos de la red neuronal. Estos coeficientes se incorporan en el programa del microcontrolador y con ellos se realiza la verificación.

3.1. Entrenamiento de la red FF-MLP en MATLAB

Con el fin de obtener la mejor arquitectura de red neuronal se han realizado varios entrenamientos de la red en MATLAB. En este proceso se ha cambiado el número de capas, el número de nodos y las funciones de activación.

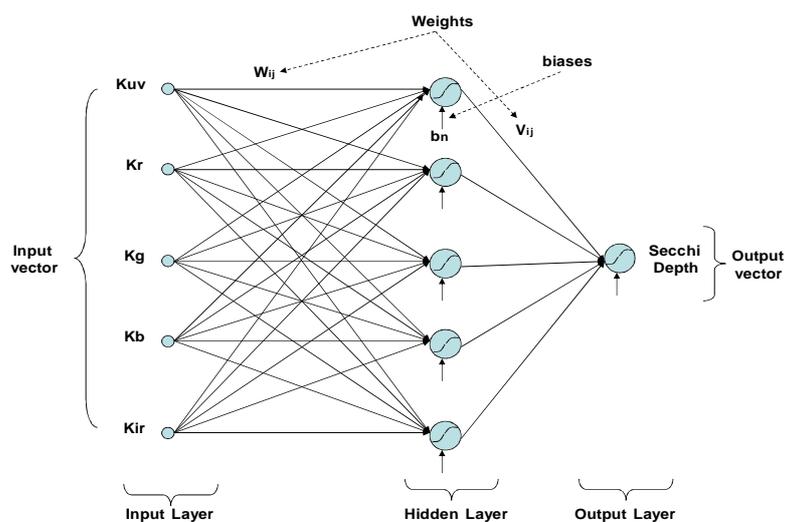


Figura. 3-4. Red neuronal artificial multicapa con alimentación hacia adelante (FF-MLP).

El mejor resultado se ha obtenido mediante la red FF-MLP con dos capas, cinco de entrada correspondientes a cada coeficiente de extinción, cinco nodos ocultos y un nodo de salida (**Fig. 3-4**).

Se ha utilizado la función de activación Tansig (Función de transferencia sigmoide: tangente hiperbólica) para las neuronas en la capa oculta y en la de salida. Esta función se muestra en la **Fig. 3-5**, y corresponde a la Ec. (3.3).

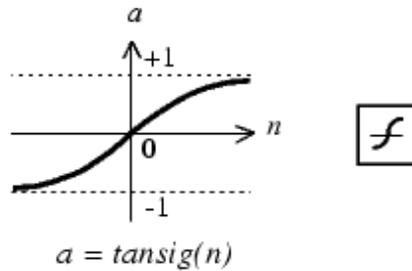


Figura 3-5. Tangente Hiperbólica. Función de transferencia sigmoideal.

La función *tansig* se puede definir mediante la siguiente expresión:

$$y = \frac{e^n - e^{-n}}{e^n + e^{-n}} \quad (3.3)$$

La función *tansig* exige que los datos estén normalizados entre [-1,1]. Los pesos y sesgos de la red neuronal se obtienen en el PC. , El algoritmo de retropropagación ha sido implementado en MATLAB® utilizando para entrenar a esta red la técnica de “*Leave-one-out cross validation*” (LOOCV) descrita en el capítulo 1.

Los pesos (weights) y sesgos (biases) se obtienen para el entrenamiento de mayor coeficiente de correlación y se envían al microcontrolador por el protocolo RS-232 para programar estos datos en la memoria del microcontrolador (**Fig. 3-6**).



Figura 3-6. Interfase serie entre MATLAB® y PIC

El desarrollo y entrenamiento de la red neuronal se explican en el organigrama de la **Fig. 3-7**.

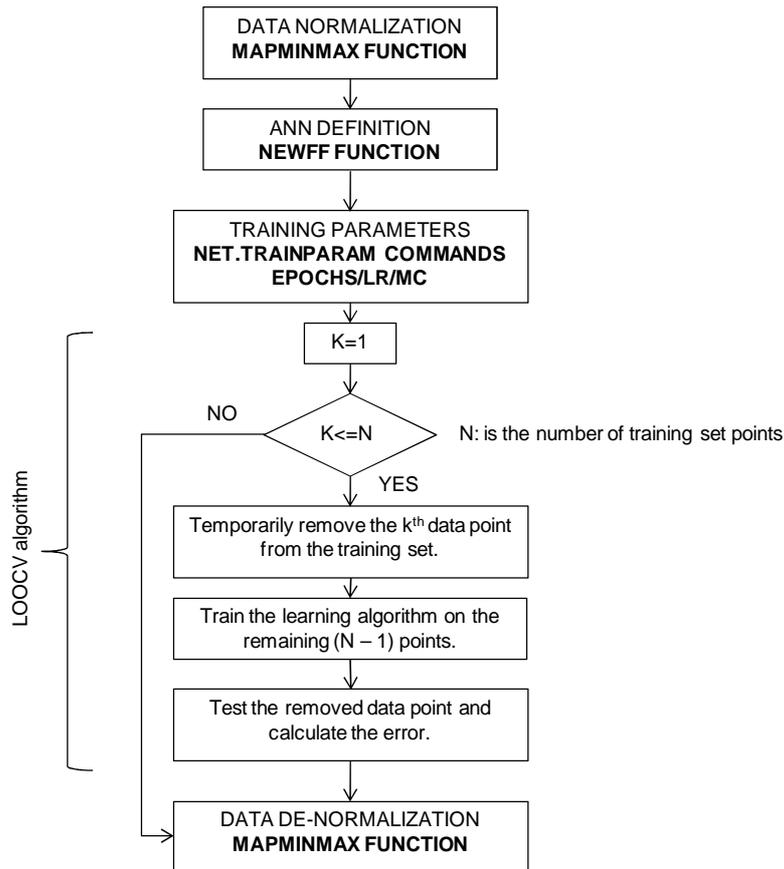


Figura. 3-7. Diagrama de flujo de la aplicación y entrenamiento en MATLAB ®

Inicialmente se normalizan los datos mediante la función *mapminmax* de MATLAB que procesa matrices para la normalización de los valores mínimo y máximo de cada fila [Ymin, Ymax]. Se define la red Perceptrón Multicapa Feed-Forward mediante la función *newff*. A continuación se indican los parámetros de entrenamiento de la red con *net.trainparam*.

Los datos se separan de forma que para cada iteración tengamos una sola muestra K (inicialmente K=1) para los datos de prueba y todo el resto constituyen los datos de entrenamiento.

Se incrementa K y se repite la iteración tantas veces como muestras N tengamos.

Una vez analizados los N datos tanto de entrenamiento como de prueba, se desnormalizan los datos mediante la función *mapminmax*.

Se calcula el coeficiente de correlación c y se repite el entrenamiento mientras c sea inferior a 0,9.

Los datos necesarios para programar el microcontrolador son los pesos y sesgos (Tabla 3.2) que se pueden obtener utilizando los comandos NET.

Tabla 3.2. Pesos y sesgos de la capa oculta y de salida.							
Capa oculta					Capa de salida		
Pesos					Sesgos	Pesos	Sesgos
2.9752	2.5274	2,0234	-0.7141	3.2312	-2.3959	-6.1144	1.7600
-5.2723	-7.2018	4.6651	-4,1198	7.5106	-3.1192	-10.0072	
-6.0539	-3.9551	-1.4549	3.8072	8.7158	1.2433	17.7016	
-4.8230	-7.0756	-3.7610	1.6904	-0.4652	1.6428	4.1322	
0.9321	-1.6017	-2.1396	-3.7033	-0.2278	-2.8425	5.9502	

Para los pesos y sesgos de la capa oculta, $net.iw\{1,1\}$ y $net.b\{1\}$, respectivamente. Para los pesos y sesgos de la capa de salida, $net.iw\{2,1\}$ y $net.b\{2\}$, respectivamente.

3.2. Implementación en el microcontrolador.

Para implementar la red neuronal artificial se utilizó el microcontrolador PIC18F4550 cuyas características principales se expusieron en capítulo 2. El software fue codificado en lenguaje C y consta de dos rutinas:

a) Control general del sistema.

Que realiza:

- la comunicación serie entre la sonda y el PC,
- el cálculo de los coeficientes de extinción y
- la presentación, gestión y otras tareas

b) Aplicación de la red neuronal.

Los datos de cálculo de los coeficientes se normalizan, se obtienen los valores de salida y se desnormalizan tal como se ha visto anteriormente (Cap. 2). El código de esta rutina es en lenguaje C y se convierte en código hexadecimal utilizando el compilador cruzado. El archivo HEX se descarga en la memoria flash del microcontrolador. La RNA se ha programado en 5908 bytes de memoria de programa y 221 bytes de memoria de datos y se ejecuta en menos de 1 milisegundo. La Fig. 3-8 muestra el organigrama correspondiente, donde se comprueba que tras las instrucciones de inicialización del microcontrolador y la

definición de las funciones de la red neuronal con función de transferencia Tansig, se produce una lectura de los datos de entrada. Estos datos son normalizados en el rango entre 1 y 0 y, con ellos, se aplican las ecuaciones de salida. Por último, se produce una desnormalización de los datos de salida, antes de mostrar los resultados en un display.

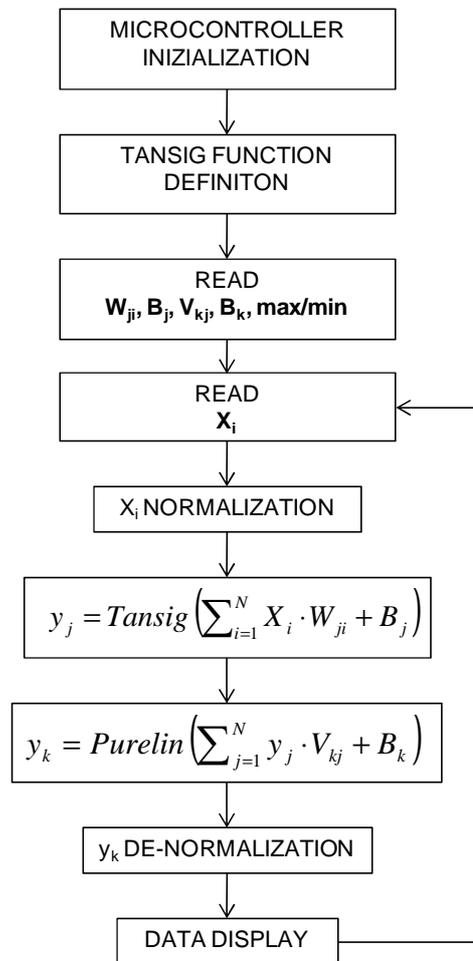


Figura 3-8. Diagrama de flujo de la implementación de la FF-MLP en el microcontrolador.

4. Resultados y discusión.

Los valores del error cuadrático medio MSE (Mean Squared Error) y el valor de regresión R para el entrenamiento, validación y prueba se muestran en la Tabla 3.3. La Fig. 3-9 muestra el valor de R para todas las muestras (entrenamiento, validación y prueba), así como para todo el grupo de muestras. La Tabla 3.4 muestra la profundidad medida y simulada de Secchi y los valores de error.

Tabla 3.3. Valores del MSE y R.			
	Muestras	MSE	R
Entrenamiento	16	0.8059	0.999
Validación	4	12.4811	0.999
Prueba	4	98.7911	0.997

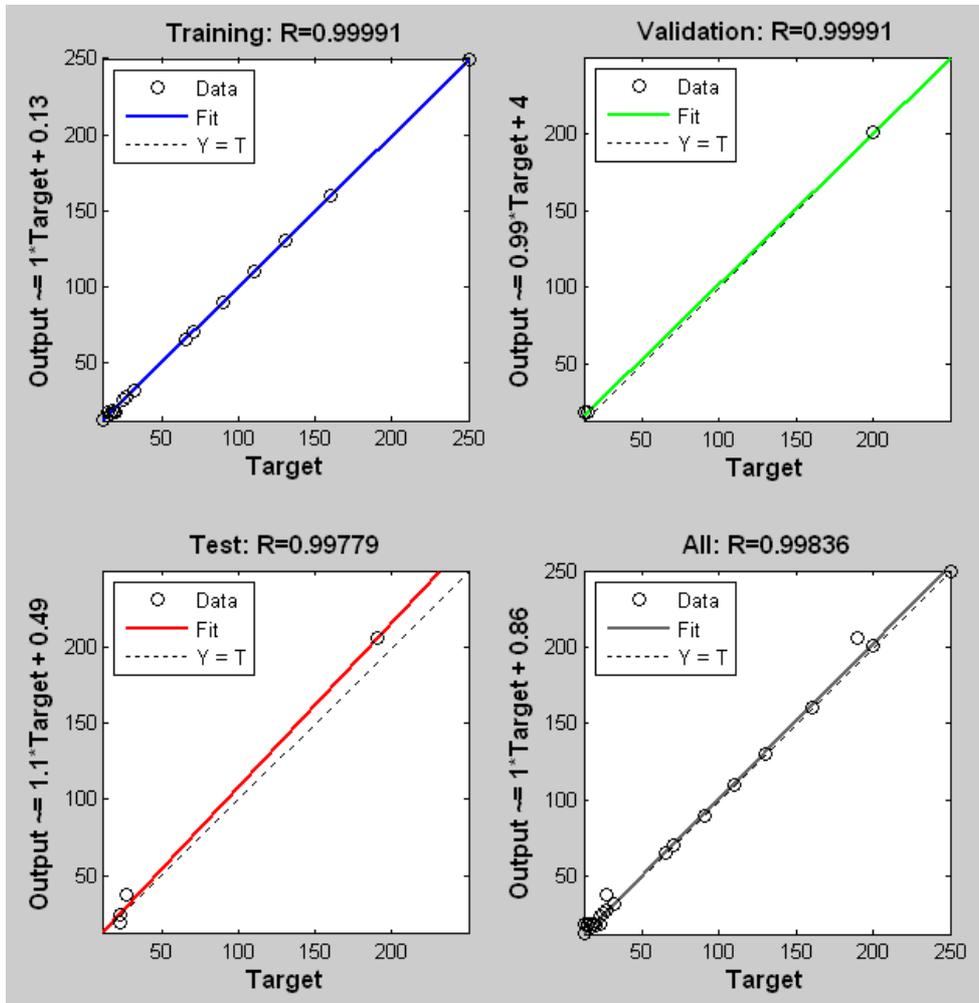


Figura 3-9. Valores de ajuste y de R para las muestras estudiadas.

La (Fig. 3-10) muestra el histograma de error, en ella se comprueba que hay dos muestras con un error notable (PAL5 y AL3 en la tabla 3.4).

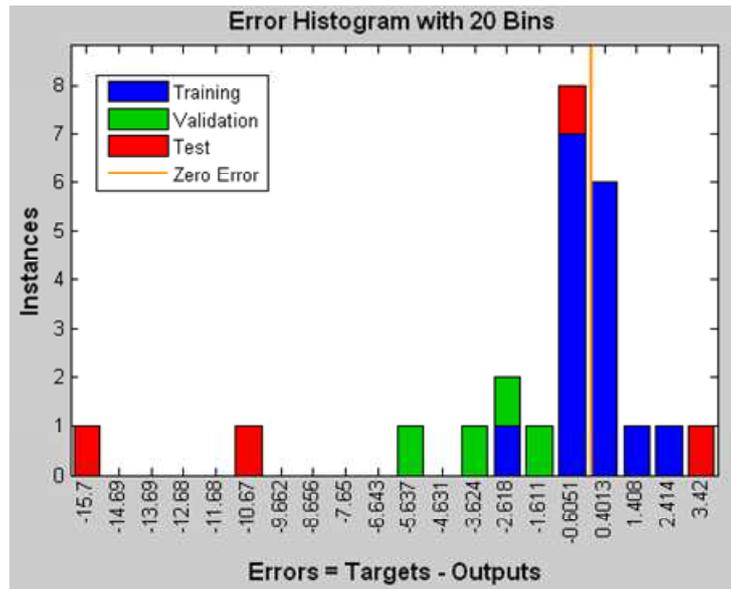


Figura 3-10. Histograma de error para las muestras estudiadas.

Tabla 3.4. Profundidad medida y simulada de Secchi.					
	Lugar	Profundidad de Secchi [cm] Medida	Profundidad de Secchi [cm] Simulada	error	
MAR	PAL1	90	89.40	0.60	Entrenamiento
	PAL2	110	110.08	-0.08	Entrenamiento
	PAL3	160	160.10	-0.10	Entrenamiento
	PAL4	130	130.24	-0.24	Entrenamiento
	PAL5	190	206.20	-16.20	Prueba
	PAL6	200	201.15	-1.15	Validación
	PAL7	250	249.91	0.09	Entrenamiento
RIO	TU1	70	69.99	0.01	Entrenamiento
	TU2	65	65.28	-0.28	Entrenamiento
LAGUNA COSTERA	AL1	15	17.68	-2.68	Validación
	AL2	15	17.62	-2.62	Entrenamiento
	AL3	27	37.79	-10.79	Prueba
	AL4	17	16.79	0.21	Entrenamiento
	AL5	19	18.10	0.90	Entrenamiento
	AL6	18	18.49	-0.49	Entrenamiento
	AL7	23	19.08	3.92	Prueba
	AL8	23	23.90	-0.90	Prueba
	AL9	20	18.01	1.99	Entrenamiento
	AL10	27	27.56	-0.56	Entrenamiento
	AL11	32	32.19	-0.19	Entrenamiento
	AL12	12	12.07	-0.07	Entrenamiento
	AL13	13	18.58	-5.58	Validación
	AL14	15	18.20	-3.20	Validación
	AL15	25	25.31	-0.31	Entrenamiento

La relación entre los valores de profundidad estimada de Secchi y los medidos se muestran en la Fig. 3-11 y Fig. 3-12. También se presenta la línea

de ajuste y el valor de R. La relación entre los valores de salida calculados por el algoritmo LOOCV y los medidos se muestran en la Fig. 3-11, el valor de R es de 0,942 y MSE es 0,0368. La relación entre los valores de profundidad de Secchi simulados y los medidos se muestran en la Fig. 3-12, el valor de R es 0.99.

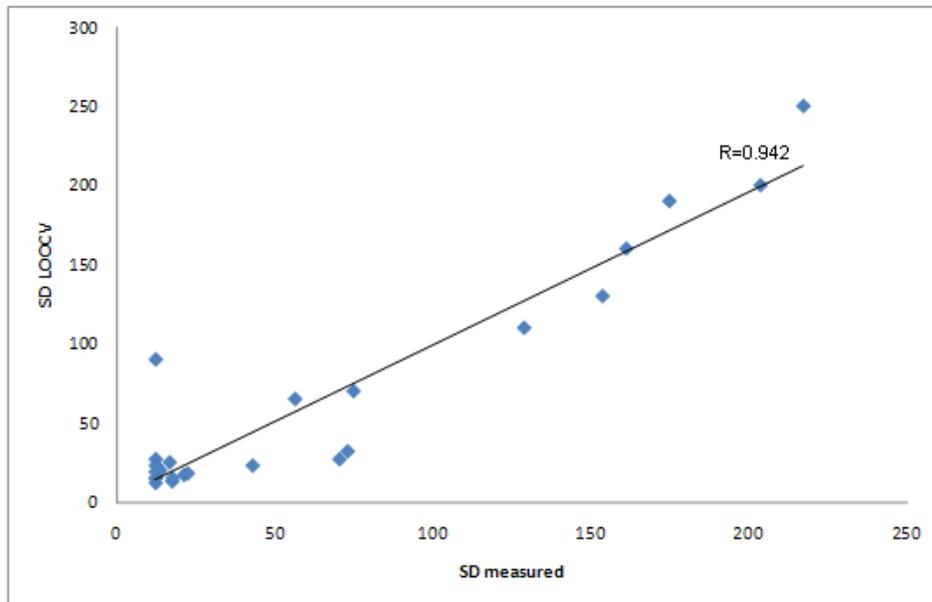


Figura 3-11. Relación entre la medida estimada de la profundidad de Secchi y los medidos con (LOOCV) el coeficiente de extinción.

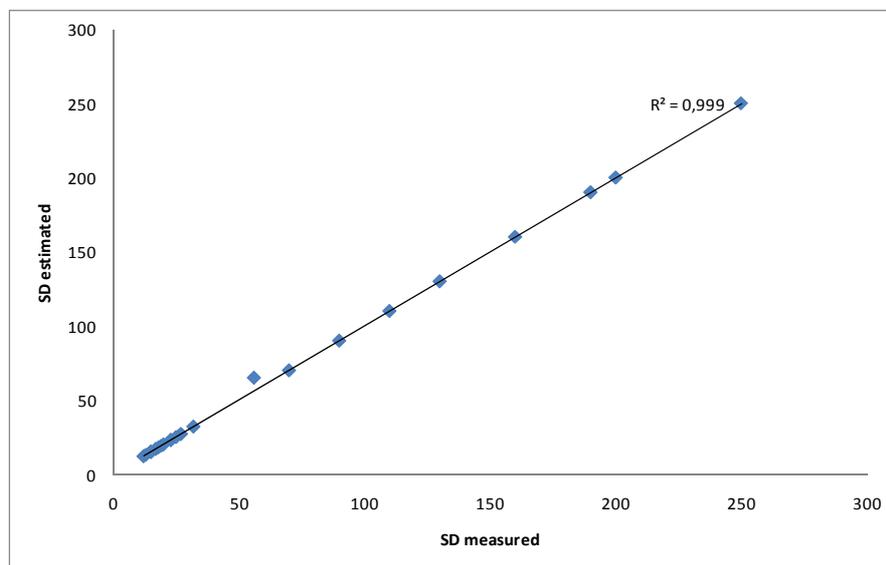


Figura 3-12. Relación entre la medida estimada de la profundidad de Secchi y el coeficiente de extinción.

Como se muestra en las figuras, el ajuste es muy bueno, a pesar del error debido a la percepción del ojo humano en la medida de la profundidad de Secchi y al hecho de que la base de datos no es demasiado amplia. Por lo tanto, se espera que la futura ampliación de esta base de datos aumentará la relación R.

5. Conclusión.

En este capítulo se ha mostrada la aplicación de una red neuronal artificial para el caso específico de cálculo de turbidez en los cuerpos de agua. Los valores de turbidez se obtuvieron por medio de un disco Secchi y un nuevo dispositivo patentado (LUZEX) para medidas de extinción la luz del sol en el agua. Este nuevo dispositivo utiliza cinco fotodiodos comerciales para obtener los coeficientes de extinción de la luz del sol para cada rango de la longitud de onda del fotodiodo (R, G, B, UV e IR).

Con el fin de tener una amplia gama de valores de turbidez y diferentes condiciones de agua, se seleccionaron tres masas diferentes de agua (agua de mar, laguna costera y una media-baja de lecho del río). La mejor arquitectura de red neuronal se obtuvo mediante una red FF-MLP con dos capas, cinco nodos de entrada correspondientes a cada coeficiente de extinción, cinco nodos ocultos y un nodo de salida. A continuación, la aplicación y la formación de la red neuronal artificial se hace por medio de MATLAB en un microcontrolador PIC18F4550. La relación entre los valores estimados y los medidos demuestra que éste modelo reproduce bastante bien los valores de turbidez obtenidos por el disco Secchi para los tres cuerpos de agua estudiados ($R= 0.998$).

Con estos resultados, una combinación del LUZEX y del diseño de redes neuronales artificiales se presenta como una alternativa eficaz en el disco Secchi para medir la turbidez de las masas de agua. Es capaz de dar las mismas medidas que el disco Secchi, fielmente de un modo rápido fácil y automático. Además, el uso de este dispositivo tiene dos ventajas importantes: Por un lado desaparece la subjetividad por el disco Secchi, cuyos valores dependen de la observación del operador, y por otro lado, en aguas

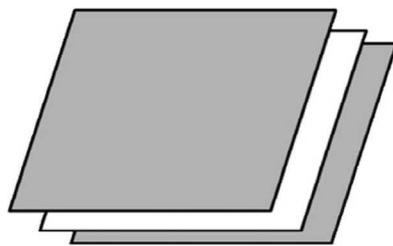
particulares cuando, debido a sus propiedades, utilizar el disco Secchi es imposible.

6. Producción científica.

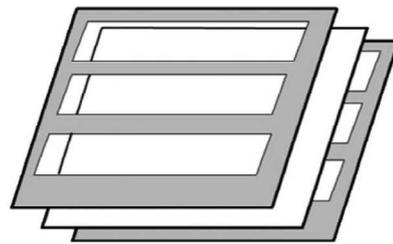
Javier Ibáñez Civera, Eduardo Garcia Breijo, Nicolás Laguarda Miró, Luis Gil Sánchez, José Garrigues Baixauli, Inmaculada Romero Gil, Rafael Masot Peris, Miguel Alcañiz Filloi. *Artificial neural network onto eight bit microcontroller for Secchi depth calculation*. Sensor and Actuators B: Chemical. 156 (2011) pp. 132-139.

Capítulo 4

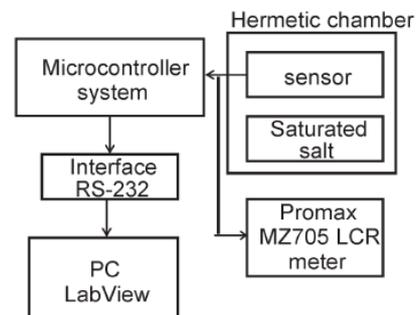
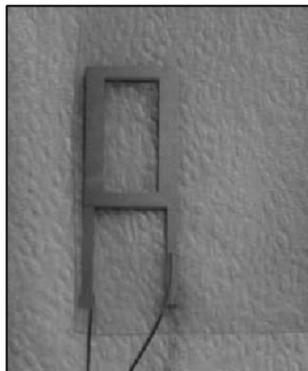
Aplicación de la Red FF-MLP en Microcontroladores de 8 Bits para la Medida de Humedad Utilizando un Sensor de Humedad Capacitivo en Etiquetas Flexibles de RFID



(a)



(b)



En este capítulo se describe la aplicación de una red neuronal artificial, para su incorporación en un microcontrolador de 8 bits utilizada para la medida de la humedad relativa mediante un sensor capacitivo de humedad relativa (HR)

El sensor utiliza procesos de serigrafía para la deposición de las pastas que conforman el sensor. También se detalla el método de medida aplicado, basado en el uso de microcontroladores con un sistema electrónico sencillo y de bajo coste. Además, se muestra una comparación entre dos tipos diferentes de sustratos de poliéster [Melinex (DuPont) y CG3460 (3M)]. Ambos sustratos de poliéster tienen propiedades similares, y sólo les diferencia el espesor (175 micras para Melinexy 100 micras para CG3460). Con este tipo de sensores se ha obtenido una respuesta no lineal. Finalmente, para linealizar la respuesta y reducir el hardware externo, se ha programado una red neuronal artificial en el microcontrolador.

1. Introducción.

Debido a las nuevas tecnologías de impresión, ahora es posible realizar etiquetas de identificación por radiofrecuencia (RFID) que incorporan una etapa de sensado. Estas etiquetas incluyen una antena, un transmisor/receptor, y una memoria de datos (las etiquetas activas o semiactivas, también llevan integradas una batería). Todos estos componentes se pueden incorporar en una película flexible [Laundgren, 2002] [Sankir, 2005] [RFID, 2005][Ferrer et al., 2006] [Oprea et al., 2007]. En consecuencia, es posible conocer el estado de un producto en cualquier momento con el uso de la información que proporcionan estos adhesivos [Rida et al., 2007]. El sensor se encarga de enviar toda la información a la memoria, almacenando en ella todos los eventos de interés, tales como cambios en presión, temperatura y humedad, de acuerdo con el sensor diseñado.

Sensores para parámetros específicos tales como temperatura y humedad son ampliamente utilizados en el control ambiental, producción industrial, etc. Estos sensores están siendo integrados en etiquetas RFID con propiedades de bajo coste y baja potencia.

Hay dos tipos de sensores de humedad relativa (HR) basados en la variación de la impedancia eléctrica: sensores **capacitivos** [Baxter, 1997] [Roman et al., 1995] y sensores **resistivos** [Sakai et al., 1996] [Bearzotti et al., 1992] [Gusmano et al., 1993]. Ambos tipos de sensores pueden funcionar en una amplia gama de HR con una electrónica simple.

Los RFID se construyen utilizando películas de distintos materiales, entre ellos destacan los polímeros. Las películas de polímeros se utilizan en aplicaciones de bajo coste porque son bastante baratas y tienen buena absorción de humedad.

Además del bajo coste del sensor de humedad relativa, también tiene un consumo de energía reducido. El sensor utilizado en el presente capítulo de la tesis ha sido desarrollado sobre un sustrato de poliéster usando tecnología de película gruesa con el fin de integrarlo en una etiqueta RFID. Se utilizan tintas conductoras poliméricas con baja temperatura de curado.

Teniendo en cuenta la necesidad de aplicar tintas conductoras en los sensores fabricados, se han desarrollado varios métodos de inserción de la tinta [Cho et al., 2008], [Harrey et al., 2002]. Hoy en día, los métodos más utilizados en la fabricación de impresión, son: la serigrafía y la inyección de tinta. Además, estos métodos presentan algunas ventajas en el coste final de los sensores, tales como la velocidad de impresión y el cuidado del medio ambiente. Esta última ventaja es la más importante con el fin de reducir el coste final. Esto es así porque no es necesario usar los productos tóxicos y peligrosos que se utilizan en los métodos de impresión tradicionales, y así se pueden ahorrar los costes de este tipo de desechos.

Con el fin de reducir los sistemas electrónicos en los dispositivos RFID es necesario encontrar sistemas básicos para la medición de la señal de los sensores [Reverter & Pallás, 2005].

Hay dos tipos de circuitos de interface para medir un sensor capacitivo HR: 1) circuitos basado en el método de transferencia de carga [Dietz et al., 2002] [Gaitán et al., 2007] y 2) circuitos osciladores; Sin embargo, algunos autores han propuesto un nuevo método de conexión del sensor al circuito de medida donde el sensor capacitivo está conectado directamente al microcontrolador. Los dos tipos clásicos de circuitos de interface no se aplican a sensores

capacitivos RH porque hay una falta de conocimiento acerca de los efectos de las componentes parásitas del sensor. Un amplio estudio sobre este efecto se ha desarrollado en Reverter y Casas [Reverter & Casas, 2008].

2. Descripción del sistema.

2.1. Diseño, fabricación y calibración del sensor.

El diseño del sensor se basa en la medición de las variaciones de la capacitancia que se obtiene en el condensador formado por capas de tintas de diferentes características (metálicas y aislantes). Los dos parámetros más importantes en la impresión del sensor son: el diseño geométrico y el material del elemento activo. Con el fin de configurar el sensor, es muy importante saber la mejor forma y el comportamiento del material utilizado debido a que la forma modifica la capacitancia del sensor y el material modifica tanto la conductividad como la resistencia.

En cuanto al diseño geométrico del sensor, hay dos modelos que podrían modificar el comportamiento del sensor. Se denominan, diseño de doble capa y diseño de una sola capa. El diseño de doble capa tiene un material activo entre las dos capas conductoras de tinta. En cambio, en el diseño de una sola capa, el aire es generalmente el material activo, aunque hay algunas excepciones (por ejemplo, electrodos interdigitados).

Con referencia a las dimensiones de los sensores [Laconte et al., 2003], es necesario prestar atención al tamaño del diseño porque una variación en la zona conductora implica una variación en la capacitancia total y, por consiguiente, una variación en la medición de la humedad. La ecuación (4.1) explica el comportamiento de la capacitancia del sensor [Laundgren, 2002], donde ϵ_0 es la constante dieléctrica para el vacío, ϵ_r es la constante dieléctrica relativa del poliéster, A es el área conductora, y a es el espesor de la capa activa.

$$C = \epsilon_0 \epsilon_r \frac{A}{a} \quad (4.1)$$

Diseño de Doble Capa: La Fig. 4-1 (a) muestra el diseño rectangular de un sensor. La capacitancia, en este caso, responde a la Ec. (4.1). Con el fin de

aumentar la exposición a la humedad, se hacen agujeros en las capas conductoras obteniendo un sensor en forma de bandera o de rejilla, (Fig. 1 (b)).

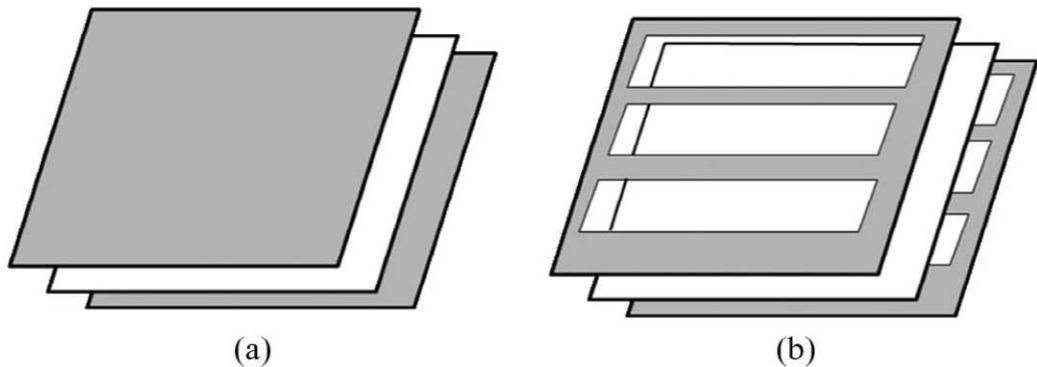


Figura 4-1. Diseño de un sensor de doble capa. (a) rectangular. (b) con agujeros.

La Fig.4.2 muestra el diseño del sensor de doble capa (llamado del tipo "Bandera" debido a su forma) y sus dimensiones (en mm), así como una pista para aumentar el área.

La zona conductora se obtiene de la superficie total, el área de la pista, y el agujero. En este caso, las zonas conductoras son 218 mm^2 para el primero y 259 mm^2 para el segundo.

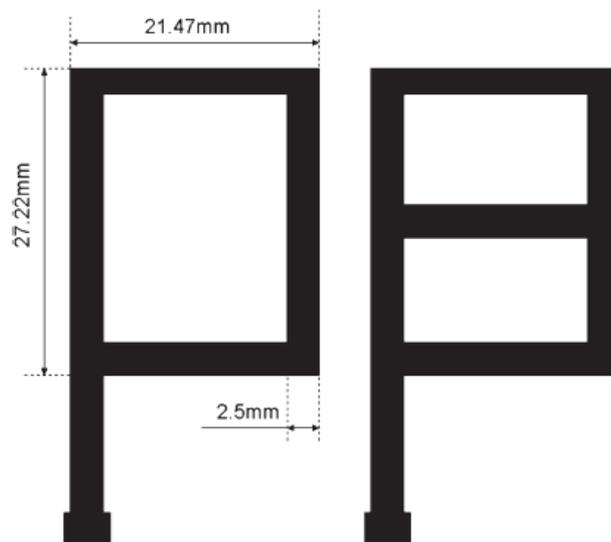


Figura 4-2. Diseño del sensor del tipo "bandera".

Se han utilizados dos películas de poliéster como sustratos, Melinex ST726 (DuPont) con 175 micras de espesor (la Tabla 4.1 muestra sus

propiedades) y CG3460 (3M) con propiedades similares pero sólo 100 micras de espesor.

Tabla 4.1. Parámetros eléctricos del sustrato Melinex ST726 (DuPont)			
Propiedad	Método de prueba	Unidades	Valor
Rigidez dieléctrica	ASTM D-149-81	KV/mm	125
Constante dieléctrica (50Hz)	ASTM D-150-92		2.9
Permeabilidad del vapor de agua	BS 3177	g/m ² /24hrs	4.0
Coefficiente de expansión higroscópica		per 1% rh	8 10 ⁻⁶

Conociendo los parámetros A , a , ϵ_0 , y ϵ_r , se puede obtener el valor de capacidad por medio de la Ec. (4.1). La capacidad teórica es alrededor de 38 pF para el sensor de Melinex y alrededor de 62 pF para el sensor CG3460.

Se ha utilizado una tinta polimérica, 5033 (DuPont). Esta tinta con base en plata ha sido desarrollada para sustratos de poliéster. Los principales parámetros de la tinta son:

- 14 micras de $m\Omega/sq/25\mu m$ de resistividad
- 15-30 Pa·s de viscosidad.

Esta tinta se deposita sobre el sustrato. La Fig.4-3 muestra el sensor de humedad con dos capas conductoras y una capa dieléctrica entre ellas.

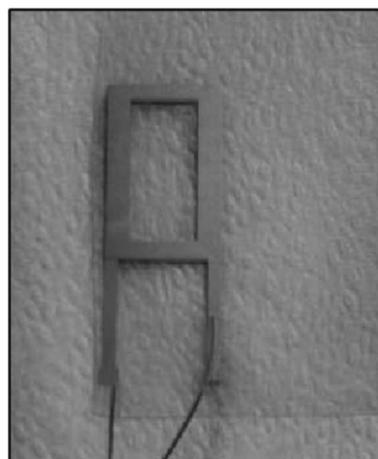


Figura 4-3. Sensor con 5033 sobre ST726.

Los sensores de humedad han sido calibrados utilizando una mezcla de sal sólida con solución acuosa saturada a una temperatura determinada para

obtener un valor de humedad relativa específica [Oprea et al., 2007], [McGaughey et al., 2006], [Papkovskyy et al., 1994]. Cada concentración de HR requiere una sal específica (Tabla 4.2). La solución salina saturada, que es una mezcla de agua destilada y sal químicamente pura, está encerrada en un metal sellado o en una cámara de vidrio. Se ha utilizado un medidor de la humedad para comparar los resultados.

Las pruebas eléctricas de los sensores se llevaron a cabo mediante la conexión de cables a los contactos de los electrodos utilizando plata cargada de adhesivo conductor.

	15°C	20°C	25°C
Cloruro de litio	11.3	11.3	11.3
Cloruro de magnesio	33.3	33.1	32.8
Carbonato potásico	43.1	43.2	43.2
Bromuro de sodio	60.7	59.1	57.6
Cloruro de sodio	75.6	75.7	75.3
Cloruro de potasio	85.9	85.1	84.3
Sulfato potásico	97.9	97.6	97.3

En primer lugar, la capacidad se midió utilizando el medidor LCRMZ705 de Promax [1kHz/2V/8-160pF \pm (3%+ 2pF)] y nuestro propio sistema electrónico (Fig. 4-4).

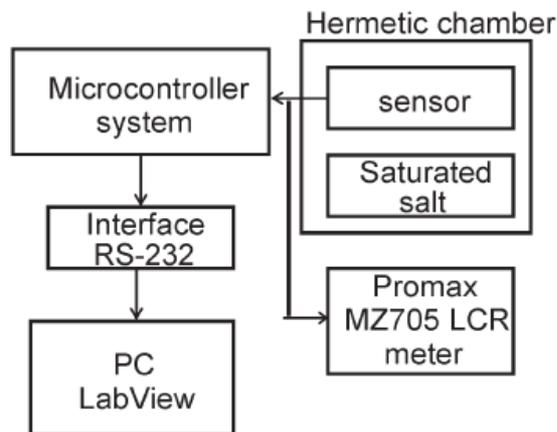


Figura 4-4. Organigrama básico del sistema electrónico.

La respuesta de los sensores de capacidad fue estudiada en el intervalo del 10% al 90% de HR (experimentalmente 9.2% -98.5% de HR). La humedad se midió utilizando el medidor RS-1366 humedad/temperatura (rango de 1% a 99% de humedad relativa, la resolución 0.1%, HR y la precisión $\pm 3\%$ HR).

2.2. Sistema electrónico de medición.

En general, los microcontroladores incluyen temporizadores que permiten medir con precisión los tiempos de carga y descarga en circuitos de resistencia-condensador (RC), lo que permite medir la capacidad C conocido el valor de R . La carga y descarga del condensador dan señales exponenciales con una constante de tiempo RC . Estas señales temporizan hasta que se alcanza la tensión umbral en un pin del puerto de entrada del microcontrolador (Fig. 4-7). El puerto de entrada incluye un disparador de Schmitt (ST) con un umbral de tensión bajo (V_{TL}) y uno máximo (V_H) (Fig. 4-5).

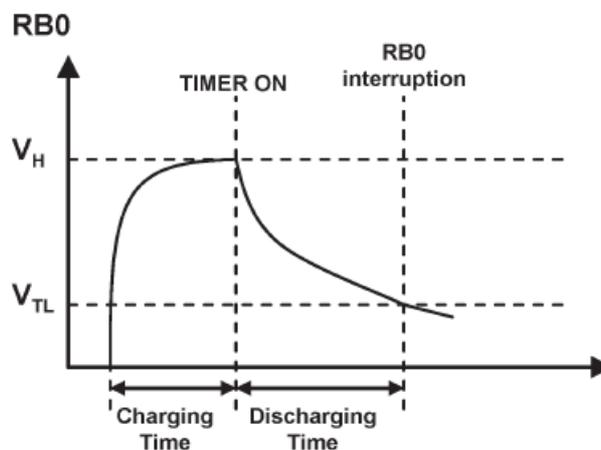


Figura 4-5. Tensión umbral de carga y descarga del condensador.

La tensión de ruido superpuesta en V_{TL} es menor que la superpuesta en V_H , por lo tanto, las medidas de tiempo de descarga sufren menos variaciones que las de tiempos de carga [Pallás & Webster, 1991]. Debido a esto, en este capítulo se estudian los circuitos basados en el tiempo de descarga [Reverter & Casas, 2008].

2.2.1. Circuitos de interface para sensores capacitivos.

La Fig. 4-6 muestra el organigrama de un programa para el microcontrolador PIC16F877 para lograr la interface básica de un sensor capacitivo directo, donde C_x es el sensor de capacidad, C_c es el condensador de calibración y R es la resistencia de descarga.

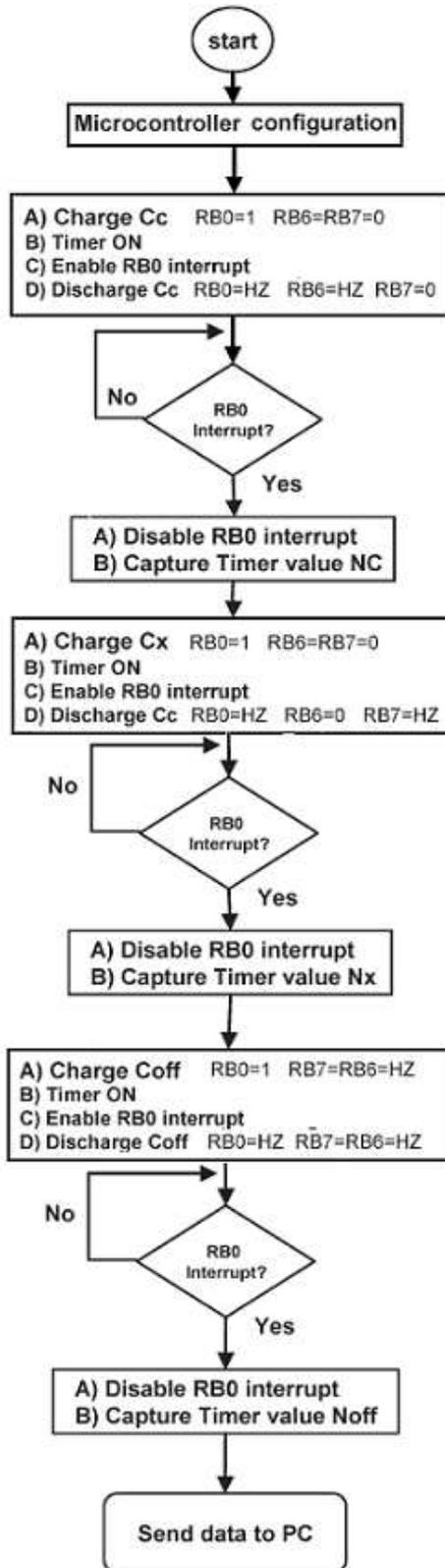


Figura 4-6. Organigrama para la medida de la capacidad.

Se puede utilizar en este diseño el denominado "método de las tres medidas" [Pallás & Webster, 1991] sin componentes adicionales. Este diseño

incluye una calibración de dos puntos. Con el fin de hacer la calibración, el microcontrolador tiene que hacer tres medidas:

- 1) medir el sensor (para obtener N_x),
- 2) medir la referencia (para obtener N_c), y
- 3) una medida de error de cero (para obtener N_{off}).

En donde N es el número de ciclos o temporizaciones en el microcontrolador necesarios para la descarga del condensador, por lo tanto N es proporcional a la capacidad.

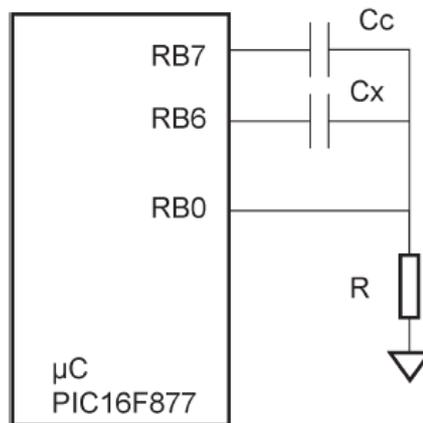


Figura 4-7. Interface básica en el microcontrolador de un sensor capacitivo directo.

Debido a las capacidades parásitas, N_{off} es diferente a cero. En estas circunstancias, se puede hacer una estimación del valor de la capacidad del sensor C_x por medio de la siguiente expresión [Reverter & Casas, 2008].

$$C_x = \frac{N_x - N_{off}}{N_c - N_{off}} C_c \quad (4.2)$$

El valor de la capacidad de referencia debe ser bien conocido para poder hacer la medida. Este dato es el valor medio del rango del sensor capacitivo. Por ejemplo, si tenemos un sensor que tiene unas variaciones de 60 a 100 pF, el valor de la capacidad de referencia debe ser aproximadamente 80 pF. En nuestro caso, se eligió el valor estándar más cercano (82 pF). Para medir los valores en el orden de picofaradios se utilizó una resistencia de 20 M Ω .

Para hacer las mediciones necesarias, se ha desarrollado un programa específico en lenguaje Labview para el microcontrolador utilizado (PIC16F877) según el diagrama de flujo mostrado en la Fig. 4-6.

3. Análisis de datos.

Para el acondicionamiento de la señal del sensor, se va a emplear una red neuronal artificial (RNA) [Dias et al., 1998] [Massicotte et al., 1998] [Medrano et al., 2001].

La respuesta de los sensores puede ser lineal o no lineal. Una respuesta no lineal complica el proceso de medición, por lo tanto, es necesario un procedimiento de linealización. Para linealizar el sensor se emplea la red neuronal artificial perceptrón multicapa con alimentación hacia delante (FF-MLP). El microcontrolador nos permite programar el procedimiento de linealización. Con este dispositivo se puede aplicar un modelo matemático para diseñar rutinas de software que nos ayuden a mejorar la característica del sensor, bastante habitual cuando se trabaja con sistemas de adquisición de datos basados en microprocesadores [Ibañez et al., 2011]. El modelo matemático puede ser programado o implementado numéricamente como una tabla de búsqueda simple [Pallás & Webster, 1991] [Martin & Medrano, 2000].

Debido a su capacidad de auto-aprendizaje basado en ejemplos [Haykin, 1999], las RNAs son herramientas muy útiles para modelizar el sensor en caso de ausencia de modelos matemáticos apropiados (o lo suficientemente precisos) [Medrano et al., 2001] para la calibración del sensor [Massicotte et al., 1998] o para la compensación de la temperatura [Haykin, 1999]. El modelo de red neural aplicado es muy simple y puede ser programado en cualquier microcontrolador de bajo coste, tales como los utilizados en aplicaciones integradas para los registradores de datos o de control industrial, incluyendo el que se utiliza en este capítulo.

En este tipo de redes, la capacidad de aprendizaje es muy útil, especialmente cuando se utilizan sensores sin una exacta representación matemática de su característica de entrada-salida. La red MLP puede linealizar un sensor no lineal a partir de muestras [Ibañez et al., 2011], considerando las muestras como pares de datos obtenidos en las salidas de los sensores y sus

correspondientes comportamientos lineales deseados (o correcciones para linealizar las salidas). Además, la flexibilidad de la RNA permite su fácil adaptación a los cambios en la respuesta del sensor debido a diferentes causas como el envejecimiento, las variaciones en RH, la deriva, y el reemplazo del sensor. El reentrenamiento del sistema con nuevas muestras permite esta adaptación. [Medrano & Martin, 2005]

3.1. Entrenamiento de la red FF-MLP en MATLAB®

Con el fin de obtener la mejor arquitectura de la red neuronal que permita analizar los datos de los sensores, se han realizado varias pruebas cambiando algunos de los parámetros de este tipo de redes. Se ha cambiado el número de capas, el número de nodos, y las funciones de activación.

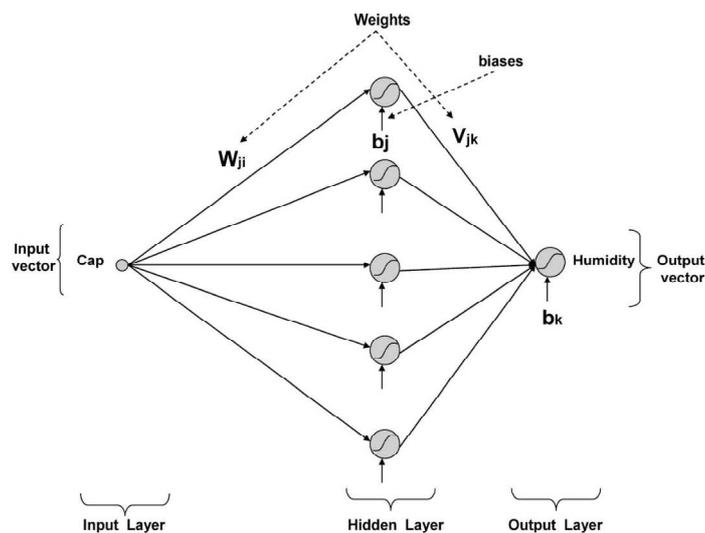


Figura 4-8. Perceptrón multicapa con alimentación hacia delante (FF-MLP).

El mejor resultado se ha obtenido mediante una FF-MLP con tres capas, un nodo de entrada que corresponde al valor del condensador, cinco nodos ocultos, y un nodo de salida (Fig. 4-8).

Se utiliza la función de activación Tansig (Cap. 1) donde los datos se normalizan en el intervalo de $[-1, 1]$. Los pesos y sesgos de la red neuronal se obtienen a partir del ordenador personal, donde se ha utilizado el algoritmo de retropropagación de Levenberg-Marquardt implementado en MATLAB para formar esta red.

El desarrollo y formación de la red neuronal se explican en el diagrama de flujo mostrado en la Fig. 11. Inicialmente se normalizan los datos mediante la función *mapminmax* de MATLAB que procesa matrices para la normalización de los valores mínimo y máximo de cada fila [Ymin, Ymax]. Se define la red Perceptrón Multicapa Feed-Forward mediante la función *newff*. A continuación se indican los parámetros de entrenamiento de la red con *net.trainparam*. Finalmente, se desnormalizan los datos con la función *mapminmax*.

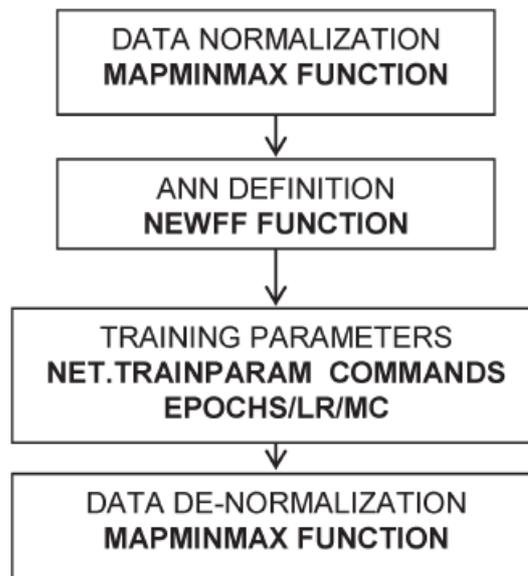


Figura 4-10. Organigrama de entrenamiento de la RNA.

Los datos necesarios para la programación del microcontrolador son los pesos y sesgos, que se pueden obtener usando los comandos NET de MATLAB (Tabla 4.3).

Tabla 4.3. Pesos y sesgos de la capa oculta y de salida.			
Capa oculta		Capa de salida	
Pesos	Sesgos	Pesos	Sesgos
-7.52041532	7.38034635	-2.46144799	1.68141843
-6.58526136	4.11939046	0.736835	
-7.18429992	-0.58900774	-1.25744294	
-11.2393272	-9.15073573	1.1122881	
7.13275488	6.55710454	2.86117801	
		-2.46144799	

3.2. Implementación en el microcontrolador.

El sistema integrado se basa en un microcontrolador PIC16F877. Las características técnicas de este dispositivo han sido comentadas en el capítulo 2. El software fue codificado en lenguaje C y consta de dos rutinas:

- 1) **Control general del sistema** en el que se realiza un cálculo de la capacidad de acuerdo con lo indicado en el apartado 2.2.1
- 2) **Aplicación de la red neuronal.** Los datos capacitivos se normalizan utilizando la misma función de MATLAB [Ibañez *et al.*, 2011]:

Los pesos y los sesgos del entrenamiento de la RNA se obtienen de la memoria del microcontrolador. A partir de los pesos y sesgos, el microcontrolador calcula la salida para cada uno de los cinco nodos ocultos y los valores del nodo de salida.

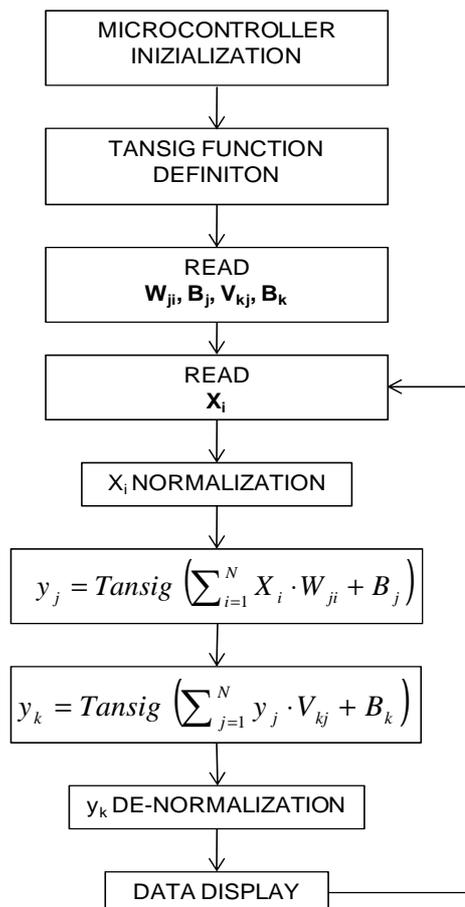


Figura 4-11. Diagrama de flujo de la implementación de la FF-MLP en el microcontrolador

Los datos de salida se desnormalizan utilizando la misma función que en MATLAB. Estos datos finales se muestran al usuario mediante una pantalla de cristal líquido.

Esta rutina se codifica en lenguaje C y se convierte a código hexadecimal (HEX) utilizando un compilador cruzado. El archivo HEX se descarga en la memoria flash del microcontrolador. Para programar la RNA han sido necesarios utilizar 5742 bytes, de memoria de programa y 197 bytes, de memoria de datos y el programa se ejecuta en menos de 1ms. La **fig. 4-11** muestra el organigrama correspondiente.

4. Resultados y discusión.

La capacidad se ha medido por un medidor de impedancia (1 kHz) a 20°C y50% de HR. La capacidad nominal obtenida en el sensor basado en Melinex es de 57pF y en el sensor basado en CG3460 es de 93 pF (Tabla 4.5).

Sensor	Real	Teórico
Basado en Melinex ST726	57 pF	38 pF
Basado en CG3460	93 pF	62 pF

Las diferencias entre los valores reales y teóricos pueden ser debidas a una reducción del espesor del sustrato en el proceso de fabricación. De hecho, las variaciones significativas de espesor se han observado mediante microscopía de electrones secundarios (SEM) (**Fig. 4-12**). Esta reducción puede estar influenciada por algunos factores tales como la temperatura y/o la difusión de tinta en el interior del sustrato.

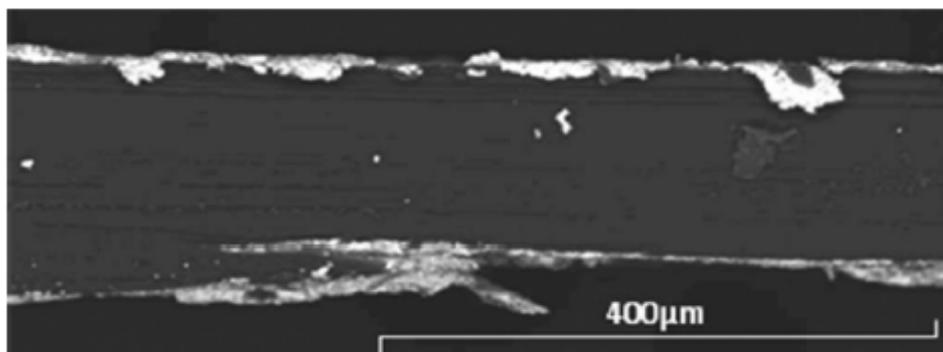


Figura 4-12. Fotografía transversa ISEM de la película de Melinex ST-726.

La Fig.4-13 muestra la respuesta de las medidas de la capacidad realizadas con el medidor LCRMZ705 de Promax y con nuestro sistema microcontrolador propio basado en el sensor de MELINEX ST-726 (125m).

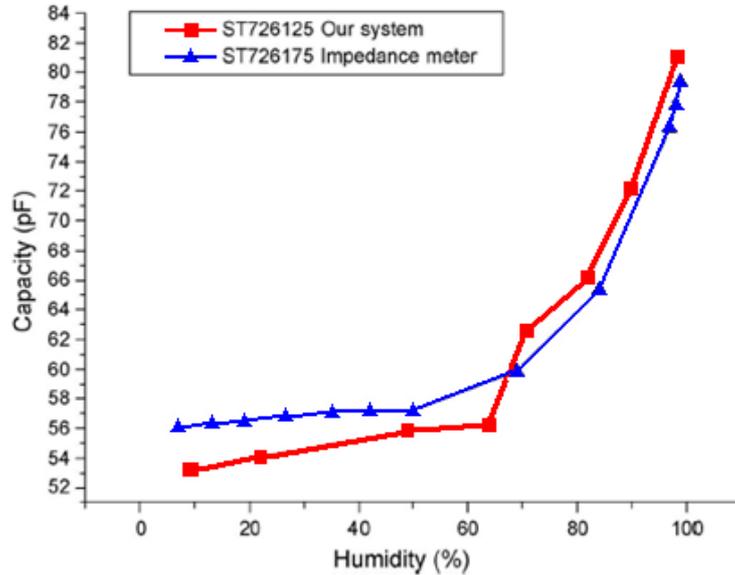


Figura 4-13. Relación entre la capacidad y la humedad (20°C) para el sensor Basado en Melinex ST726 y medidor LCR MZ705 de Promax

La Fig.4-14 muestra la respuesta de las medidas capacitivas con el mismo medidor LCR que el de la figura anterior y con nuestro sistema microcontrolador para del sensor basado en CG3460 (100m).

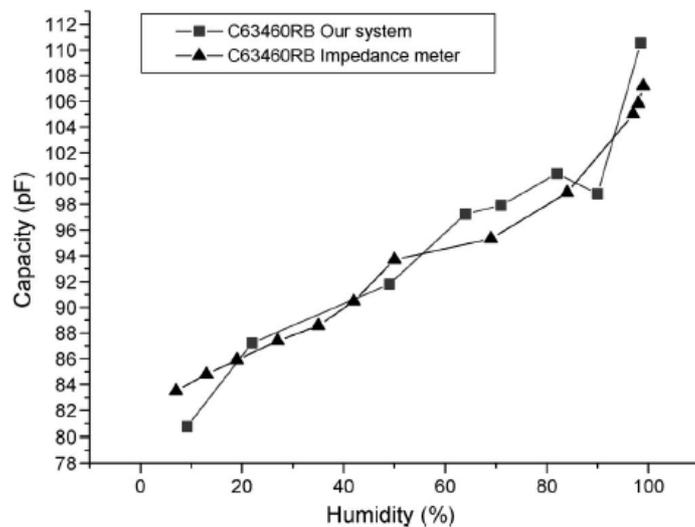


Figura 4-14. Relación entre la capacidad y la humedad (20 °C) para el sensor CG3460 y para medidor LCR MZ705 de Promax.

Como se muestra en las Figs. 4-13 y 4-14, la respuesta de nuestro sistema propio de medición es bastante similar a la respuesta del medidor comercial LCR.

Otro parámetro a considerar es la variación de la capacidad con el tiempo. La Fig. 4-15 muestra la variación de la capacidad con el tiempo del sensor CG3460 cuando es sometido a un cambio de la humedad de 25% -60% de HR (absorción) y del 60%-25% de HR (desabsorción).

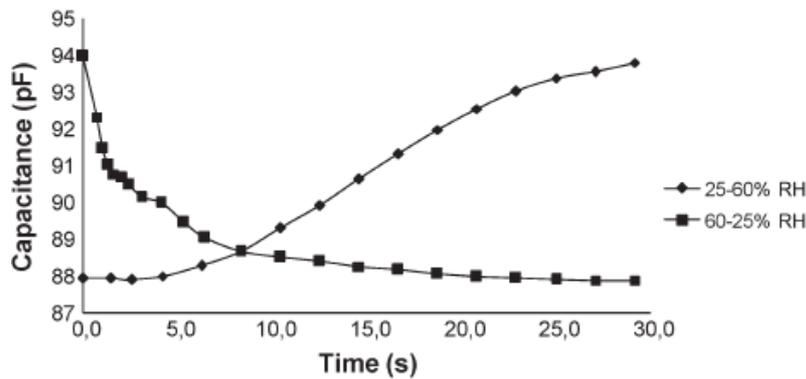


Figura 4-15. Respuesta temporal del sensor CG3460

En este caso, el tiempo de respuesta es alto en comparación con los sensores comerciales tal como el Humirel HS1101 (5 s de 33% a76%HR).

4.1. Linealidad.

El error máximo de no linealidades igual al 12.2% a fondo escala (SFS) para el sensor CG3460 (Fig. 4-16). Este error es considerable en comparación con el de los sensores comerciales como el HS1101 (SFS 2%). Para el sensor basado en ST726, la respuesta es completamente no lineal. Debido a estas respuestas no lineales, hemos decidido utilizar una red neuronal, tal como se ha descrito antes.

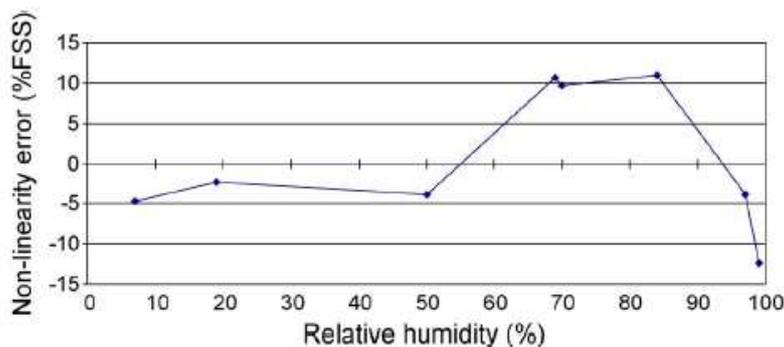


Figura 4-16. La linealidad de la respuesta del circuito cuando se mide el sensor CG3460.

Finalmente, se presentan los resultados de la aplicación de la red neuronal para la respuesta de los sensores: La relación normalizada entre los valores de humedad estimados y los medidos se muestran en la Fig. 4-17 (sensor Melinex ST726). En esta figura, también se presenta la línea de ajuste donde se observa el alto nivel de ajuste ($R=0,998$).

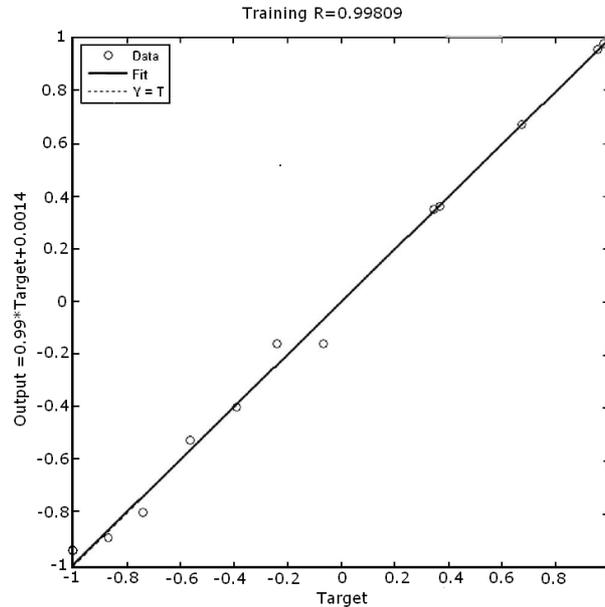


Figura 4-17. Relación entre la respuesta estimada y medida de la humedad relativa.

La Fig. 4-18 muestra el valor estimado(a través de la RNA) y medido (a través del sensor) de los valores de humedad en comparación con la capacidad.

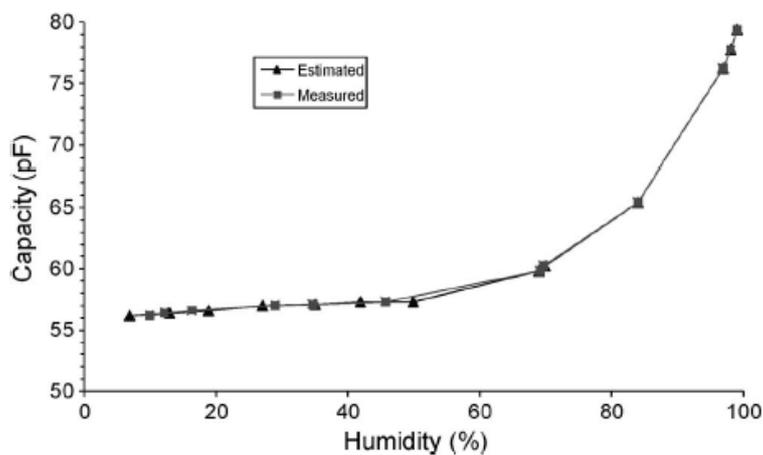


Figura 4-18. Respuesta de la red neuronal para el sensor MelinexST726.

5. Conclusión.

A lo largo del capítulo se ha presentado el desarrollo de dos sensores capacitivo de humedad relativa de bajo coste utilizando distintos materiales. Estos sensores están formados por pistas conductoras impresas en ambas caras de un sustrato de poliéster sensible a la humedad.

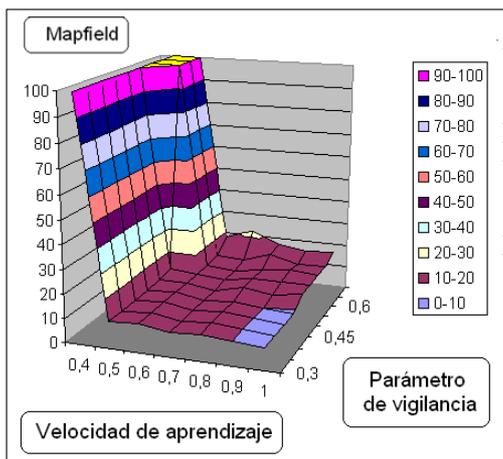
Con el fin de medir la humedad relativa, se ha utilizado un microcontrolador para desarrollar e implementar un sistema de medición de impedancia capacitiva. El diseño del sistema se basa en un sensor directo al microcontrolador. Se ha utilizado el "método de las tres medidas," que incluye un sistema de medición de auto-calibración, sin necesidad de acondicionadores. De esta forma, se pueden desarrollar dispositivos de baja potencia y de bajo coste. Tanto la baja potencia como el coste reducido son un requisito importante para los sensores RFID y sus aplicaciones. Al final, debido a la respuesta no lineal de los sensores, se ha desarrollado y aplicado una red neuronal artificial. Esta RNA ha sido realizada con un microcontrolador.

6. Producción científica.

Pelegri-Sebastia, J.; Garcia-Breijo, E.; Ibanez, J.; Sogorb, T.; Laguarda-Miro, N.; Garrigues, J.; , "*Low-Cost Capacitive Humidity Sensor for Application Within Flexible RFID Labels Based on Microcontroller Systems*," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, nº 2, (2012) pp. 545-553.

Capítulo 5

Aplicación de Redes Neuronales en Microcontroladores de 8 Bits para la Clasificación de Aguas Embotelladas



Confusion Matrix

1	4 14.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	4 14.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	4 14.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	4 14.3%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3 10.7%	0 0.0%	0 0.0%	100% 0.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	4 14.3%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 3.6%	0 0.0%	4 14.3%	80.0% 20.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	75.0% 25.0%	100% 0.0%	100% 0.0%	96.4% 3.6%
	1	2	3	4	5	6	7	
	Target Class							

En este capítulo se describen los análisis realizados con redes neuronales artificiales Fuzzy Artmap para la clasificación de aguas embotelladas de diversas marcas comerciales. Los datos analizados proceden de medidas realizadas con un sistema de lengua electrónica potenciométrica formada por un conjunto de electrodos metálicos de diferentes materiales.

1. Introducción. La técnica potenciométrica tiene como principal característica la simplicidad del método de medición y de los equipos electrónicos. En potencimetría se han utilizado diferentes tipos de electrodos, tales como las membranas [Verrelli et al., 2007] o las superficies metálicas [Lvova et al., 2006]. En este último electrodo, se obtiene un voltaje que es proporcional a la concentración de todas las especies presentes en la solución y por lo tanto, su cuantificación es difícil de determinar cada vez que el medio acuoso es complejo. [Soto et al., 2008]

Un método para la obtención de un multi-electrodo de fácil construcción y operación simple es el empleo de tintas con tecnología de película gruesa de circuito híbrido [Martínez et al., 2005a] porque hay muchos tipos de diferentes tintas y cada uno tiene un elemento químico clave que puede convertirse en el elemento activo del sensor

La tecnología de película gruesa emplea pastas o tintas que contienen dispersos metales ordinarios o nobles, de manera que pueden ser conductoras, resistivas o dieléctricas. Dichas pastas se depositan sobre un sustrato cerámico de acuerdo con la geometría del circuito a realizar y se funden al someterlas a temperaturas elevadas. El proceso de impresión, secado y fusión se repite para las diferentes pastas utilizadas [Pallás, 2003].

La mayoría de los sistemas de lenguas electrónicas permanecen en la versión de laboratorio, ya que requieren la presencia de un equipo y, sobre todo, de dos procesos separados, uno para tomar medidas y otra para el procesamiento de datos. Si se desea que estos sistemas tengan aplicación industrial, es necesario unificar estas dos fases en un solo sistema. El mejor método es el uso de microcontroladores en los sistemas que, además de la medición del potencial, sean capaces de realizar el análisis de los datos

pertinentes mediante un programa de software implementado en la memoria del microcontrolador. Algunos sistemas que utilizan microprocesadores se han presentado como lenguas electrónicas [Abdul et al. 2004]. Pero el sistema que se presenta en esta tesis tiene como principal novedad el desarrollo de una red neuronal artificial del tipo Fuzzy Artmap Simplificada (SFAM) que ha sido descrita con detalle en el capítulo 3 y utilizado en las narices electrónicas [Llobet et al., 1999b].

El objetivo de ese capítulo es presentar un sistema potenciométrico de lengua electrónica que utiliza un conjunto de electrodos construidos en la tecnología de película gruesa, cuyo análisis de datos del sistema consiste en una red neuronal artificial Fuzzy Artmap Simplificada (SFAM) implementada en un sistema microprocesador.

Como ejemplo de aplicación de este sistema, el análisis se ha hecho de los distintos tipos de agua potable que tienen diferentes concentraciones y tipos de sales. La red neuronal artificial es capaz de realizar una clasificación de estas muestras de agua utilizando los datos obtenidos de las mediciones potenciométricas.

2. Descripción del sistema.

2.1. Electroodos.

Para obtener los sensores apropiados para realizar la medidas se han seleccionado una amplia gama de electrodos con diferentes superficies con el fin de explorar su respuesta diferencial en las mediciones potenciométricas. Siguiendo esta pauta se prepararon diversos electrodos fabricados con tecnología de película gruesa. Para este fin, se utilizaron diferentes tintas, cada una con un elemento activo determinado: pastas de RuO_2 de $10\Omega/\text{sq}$ (modelo R8911) y $1\text{M}\Omega/\text{sq}$ (modelo R8961), Cu (modelo C7257), Ag (C8829 modelo), y Pt (modelo C1076D) han sido facilitadas por Heraeus. El AgCl fue fabricado por la mezcla de Ag y AgCl en polvo en una proporción de 1:1 y con baja temperatura de vidrio EG2020 (suministrado por Ferro). La pasta protectora de la capa superior de D2020823D2, modelo suministrado por GWENT.

Los electrodos se apoyan en un sustrato de Al_2O_3 708S RUBALIT (suministrado por Ceramic Tec) con una superficie de 50.8 mm x 25.4 mm y espesor de 0.635 mm. A fin de preparar el mencionado conjunto de electrodos, se imprimieron tres pantallas para realizar la serigrafía, correspondientes a tres niveles: la capa conductora de trabajo como un sistema electrónico de interconexión de la señal, la capa activa y la capa de protección superior (Fig. 5-1). Como pasta conductora se ha utilizado Ag C8829 (suministrado por Heraeus). El trazado de las pistas se ha diseñado para que pueda unirse al sustrato de cerámica mediante un conector de cable plano con una separación de 3 mm entre los terminales.

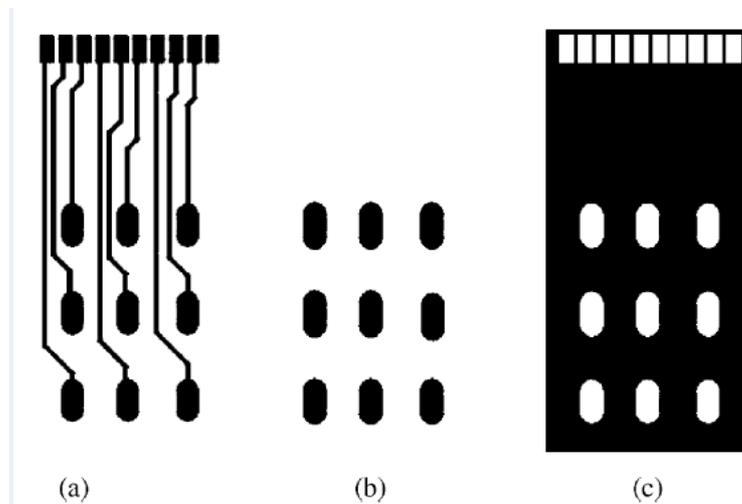


Figura 5-1. Disposición de las capas: conductoras (a), activa (b) y protección (c).

Para realizar las medidas se emplearon en cada ensayo tres electrodos diferentes de cada tipo (tres de Cu, tres de RuO_2 de $10\Omega/\text{sq}$, tres de RuO_2 de $1\Omega/\text{sq}$, tres de Ag, tres de Pt y tres de AgCl). Todos ellos formaron un conjunto de 18 electrodos, que fueron utilizados como un sistema activo para la medición potenciométrica (la Fig. 5-2 muestra la matriz final de electrodos realizada con la tecnología de película gruesa). El sistema de lengua electrónica se completó con dispositivos para la adquisición, control y almacenamiento de los datos medidos, así como con el software para los estudios de reconocimiento de patrones (MATLAB®)

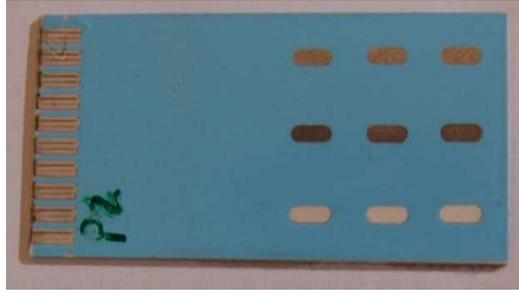


Figura 5-2. Parte superior del multi-electrodo realizado en la tecnología de película gruesa.

2.2. Sistema Electrónico.

Las mediciones se realizaron con un diseño propio denominado registrador de datos. Se utilizaron dos electrodos múltiples, de esta forma se midieron 36 canales de forma simultánea. Como electrodo referencia se empleó un dispositivo comercial de Ag/AgCl (suministrado por CRISON). Las señales de salida del multi-electrodo se adquirieron mediante una arquitectura multiplexora 36:1, realizado con dos dispositivos MAX306 (multiplexor 16:1) y un MAX308 (multiplexor 8:1). La selección de cada canal en el multiplexor fue controlada por el microcontrolador. La frecuencia de muestreo para los 36 canales fue de un electrodo cada 100 ms en períodos de 10 s.

El LMC646, un amplificador CMOS de precisión de cuatro amplificadores operacionales (AO), conectado con el multiplexor de salida. Este AO tiene una impedancia de entrada muy alta (ultra baja corriente de polarización de menos de 16fA) y se adapta la impedancia de la señal potenciométrica generada por los múltiples electrodos.

Debido a que la resolución del convertidor AD del microcontrolador es de 8 bits y admite solo tensiones positivas se ha optado por utilizar un MAX128. Este A/D tiene una resolución de 12 bits y puede trabajar con señales de entrada tanto unipolares como bipolares y utiliza un voltaje de referencia externa o interna a fin de obtener diferentes intervalos de la escala completa. En este caso, se ha utilizado una referencia externa de 2,5 V y una señal de entrada bipolar. Con esta configuración, la resolución (equivalente al bit menos significativo) es de 1.22 mV. En el microcontrolador PIC18F4550 se reunieron los datos del convertidor A/D con un bus I2C.

El software para el microcontrolador PIC18F4550 se ha diseñado para obtener el valor el promedio de cada canal. Se calculan seis vectores de entrada a partir de los 36 canales de datos. Estos seis vectores de entrada corresponden a los seis tipos de electrodos

El proceso de medida se ha dividido en dos fases:

- Periodo de entrenamiento
- Periodo de prueba.

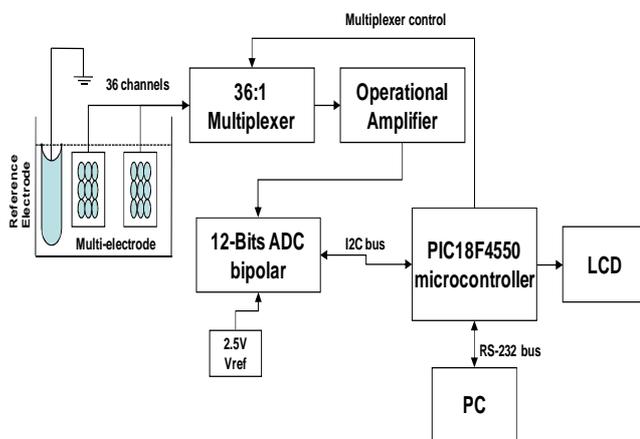


Figura 5-3. Diagrama de bloques del sistema electrónico de medición.

En el periodo de entrenamiento los datos se envían al PC a través de un enlace de comunicaciones serie RS232 para así utilizarlos en el algoritmo de entrenamiento con MATLAB® R2010b. El software de adquisición se ha desarrollado utilizando Visual Basic® 6.0 y Microsoft® Excel 2003. En el periodo de prueba los datos se medían y almacenaban directamente en el microcontrolador para ser utilizados en la red neuronal “embebida”.

El diagrama de bloques del sistema de medición se muestra en la **Fig. 5-3.**

2.3. Medidas.

Para realizar los análisis se han seleccionado un total de cinco aguas minerales naturales comerciales españolas. De ellas, cuatro sin gas (Bezoya, Bronchales, Cortés, Lanjarón, Solán) y un agua con gas (Primavera). Además agua del grifo de la ciudad de Valencia y se han estudiado mediante medidas

realizadas con una matriz de electrodos potenciométricos que se describe a continuación. Los nombres y las concentraciones (en mg/L) de los iones principales de las aguas minerales usados se muestran en la Tabla 5.1.

Tabla 5.1 Concentración (en mg/l) de algunos aniones y cationes en las aguas minerales de acuerdo con la información suministrada por los fabricantes.									
Tipo de agua	CO ₃ H ⁻	SO ₄ ⁻	Cl ⁻	Ca ²⁺	SiO ₂	Mg ²⁺	Na ⁺	F ⁻	K ⁺
Agua del grifo	154	258	84	117	3.3	37	49	-	2.5
Agua con gas	297.2	43.9	35.8	88.7	7.1	23.4	20.7	0.1	1.3
Bezoya	18	15	0.67	4	8.9	1.8	1	-	-
Bronchales	19	6	2	3	9	3	1.35	0.1	1.36
Cortes	268.4	17.6	8.7	81.9	9.2	37	6.4	1	0.7
Solan	279.4	18	8.1	59.3	7.1	25.8	5.2	0.2	1.1
Lanjarón	105	17.3	2.8	27.2	4.8	8.8	4.8	0.2	1.1

Como muestra el cuadro, las aguas Bezoya y Bronchales son muy similares entre si. Ambas tienen la concentración de iones más baja. Esta relación entre las aguas se muestra en la **Fig. 5-4** mediante un análisis PCA realizada con los datos de la concentración de los distintos iones de cada una de las aguas, donde la cercanía o distancia entre los puntos da una idea de la semejanza (o no) de la concentración de iones en las muestras de agua analizadas.

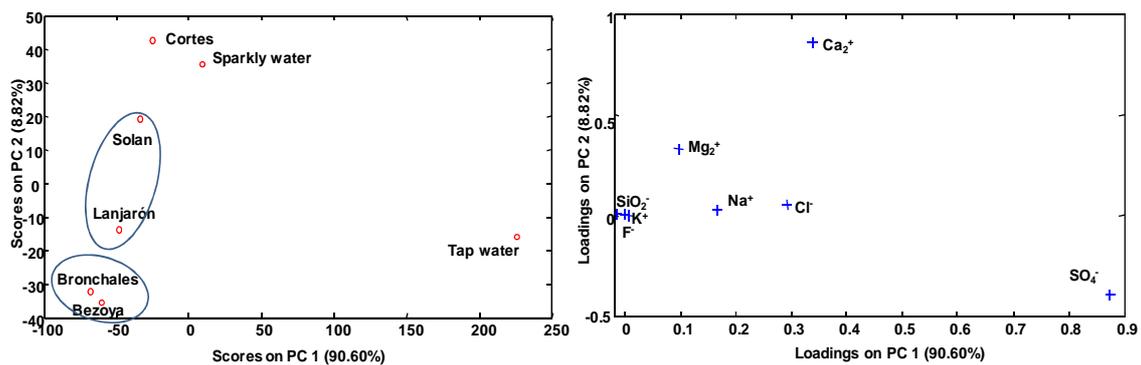


Figura 5-4. Resultados del análisis PCA de las muestras de agua.

Inicialmente el conjunto de los 36 electrodos se sumergieron a 25 °C en una solución de referencia que estaba formada por 300 ml de KNO₃ de concentración 0.01 M. Se estudió su respuesta de los electrodos hasta que

alcanzan un potencial estable, lo que sucede después de quince días. Este periodo fue llamado periodo de acondicionamiento (**Fig. 5-5**). Después de este periodo, las muestras se adquirieron en días alternos durante cuarenta días más con cuatro botellas de cada tipo de agua. Este periodo fue llamado periodo de prueba y entrenamiento (**Fig. 5-5**).

Los grupos de electrodos se sumergen en la muestra correspondiente a una temperatura de 25 °C, durante un período de 10 minutos. El potencial de los electrodos se almacena en los últimos 5 min. El tiempo típico para el estado de equilibrio fue siempre inferior a 5 min. Después de cada medición, los electrodos se limpiaron con agua destilada.

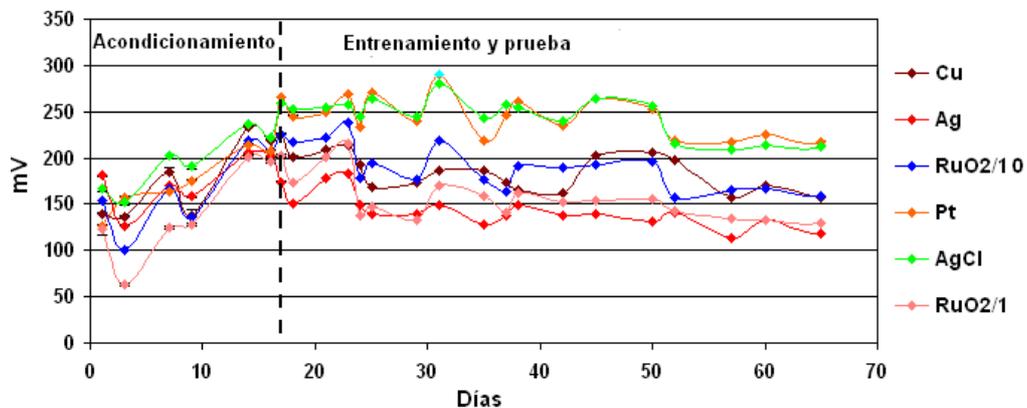


Figura 5-5. Respuesta a la solución de referencia.

Las muestras se midieron en un orden aleatorio. Después de cada medición, el conjunto de electrodos fue sumergido de nuevo en la solución de referencia. La respuesta de la matriz multi-electrodo se considera estable durante las 8 primeras muestras (véase la respuesta a la muestra de agua del grifo en la **Fig. 5-6**). En esta figura se comprueba que los resultados obtenidos no contribuyen a una clara discriminación entre las muestras. Por ello, es necesario realizar un estudio de los resultados con redes neuronales artificiales.

Con éstas 8 primeras muestras, se obtuvieron un total 56 datos, que es el resultado de 8 muestras por 7 tipos de agua. Estos 56 datos son los que se han utilizado para el entrenamiento de las redes neuronales artificiales.

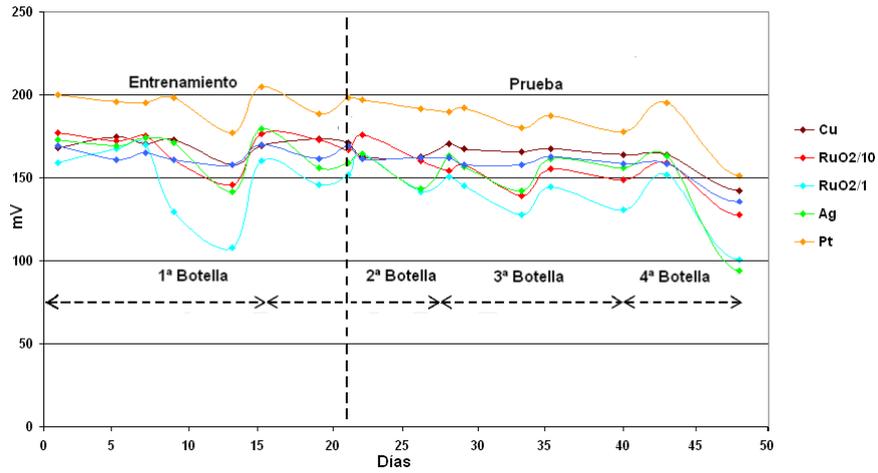


Figura 5-6. Respuesta a la muestra de agua del grifo.

Las Figs. 5-7 a 5-12 muestran las respuestas de los diferentes tipos de aguas embotelladas.

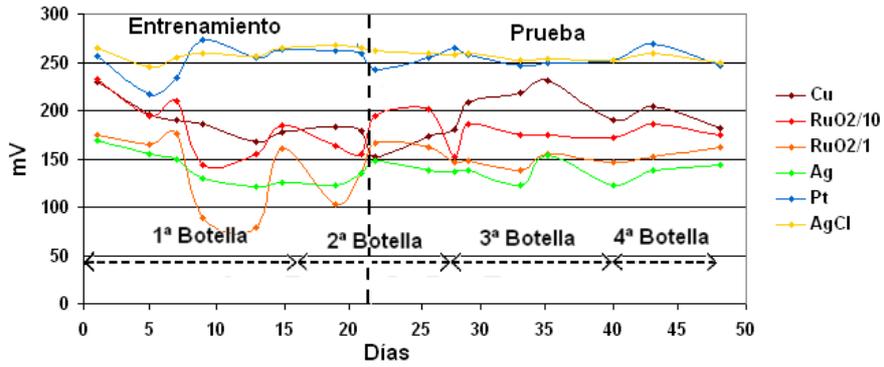


Figura 5-7. Respuesta a la muestra de Bezoya.

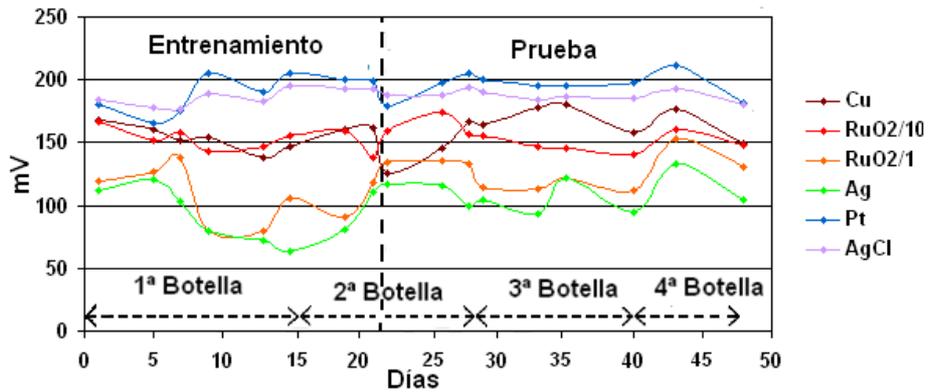


Figura 5-8. Respuesta a la muestra de Cortés.

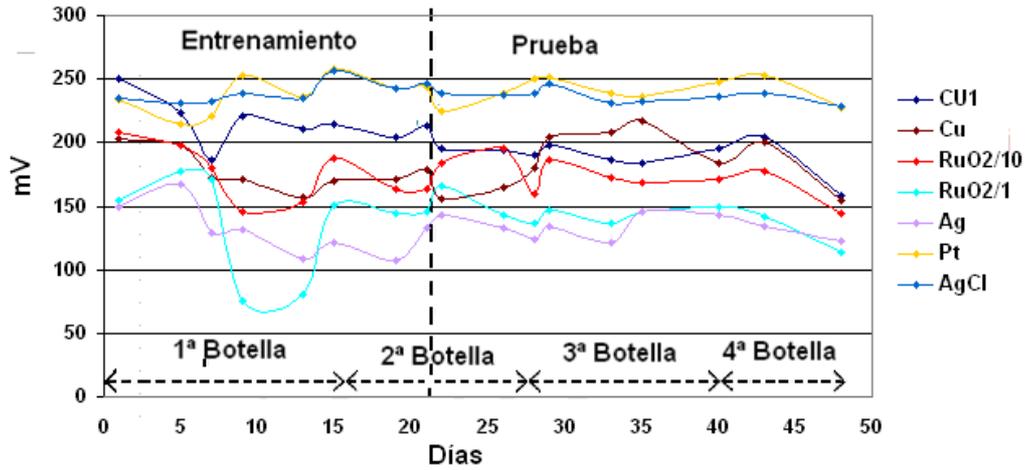


Figura 5-9. Respuesta a la muestra de Bronchales.

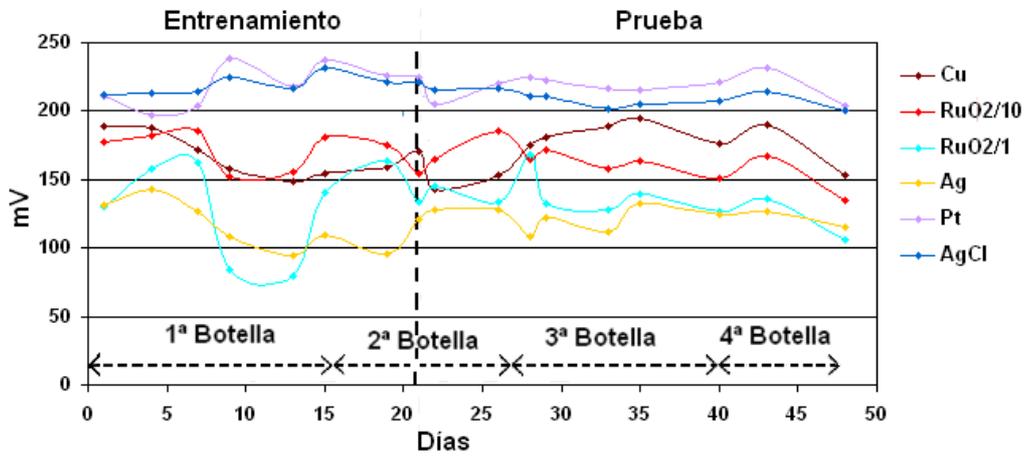


Figura 5-10. Respuesta a la muestra de Lanjaron.

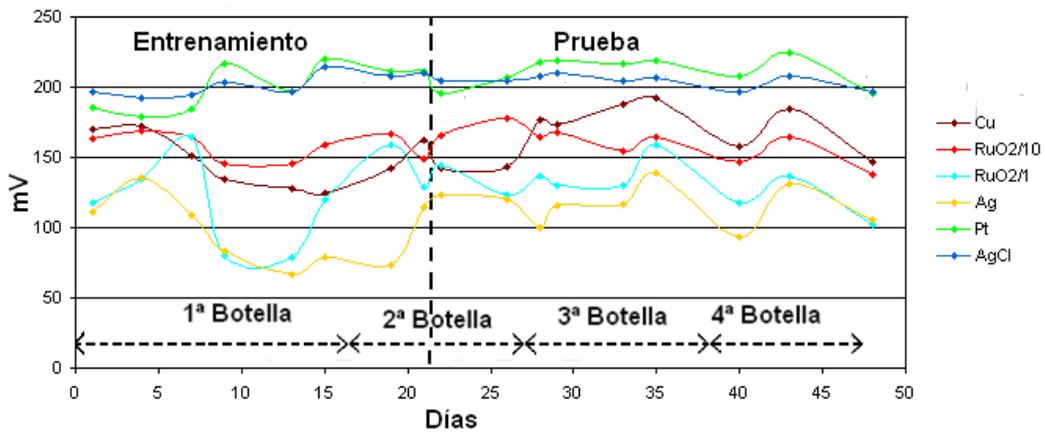


Figura 5-11. Respuesta a la muestra de Solan.

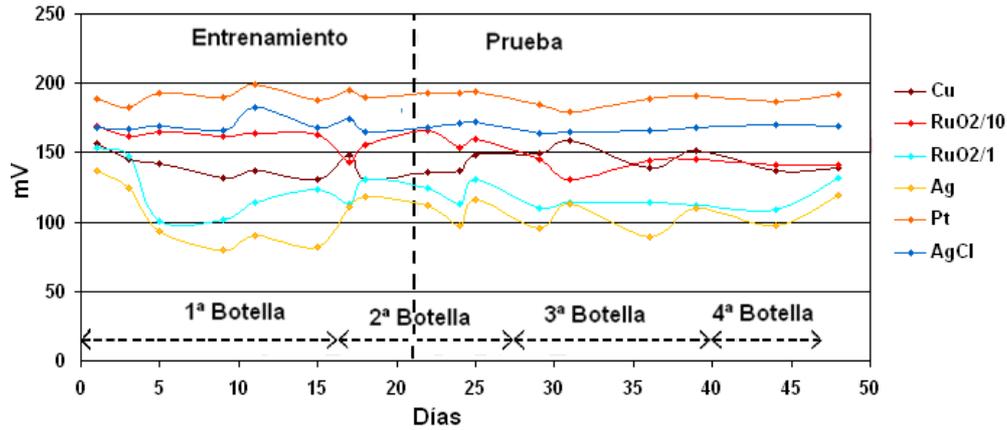


Figura 5-12. Respuesta a la muestra de Primavera.

3. Análisis de datos.

El análisis de datos consiste en la clasificación de los datos de entrada en una serie de categorías preestablecidas. El tipo de red más adecuado para este tipo de análisis es la red Fuzzy Artmap Simplificada (SFAM).

La etapa de entrenamiento de la red neuronal se realiza con una parte de las medidas disponibles. En esta etapa se establecen las categorías de la red (en nuestro caso los siete tipos diferentes de agua). Como vector de entrada se aplican los datos de seis electrodos para cada medición. Con estos datos se calculan los coeficientes del algoritmo que configuran la red. En la etapa de verificación, los datos de las nuevas medidas se aplican a las entradas, comprobando si la salida correcta activa o no la red.

Para el entrenamiento de las redes se ha utilizado el programa MATLAB 2010b®, que se ejecuta en un ordenador PC. El uso de ordenador está determinado por su potencia de cálculo y facilidad de aplicación de los algoritmos de redes neuronales. En cambio, la etapa de verificación se realiza íntegramente en microcontroladores.

Para realizar la verificación se utilizan los resultados obtenidos en la etapa de entrenamiento, que corresponden a los coeficientes de los algoritmos de la red neuronal. Estos coeficientes se incorporan en el programa del microcontrolador y con ellos se realiza la verificación.

Mediante esta forma de trabajar, una vez que la etapa de entrenamiento ha concluido, el sistema desarrollado puede trabajar independientemente de un

PC. Esta es una de las principales características de los equipos presentados en esta tesis.

3.1. Red Fuzzy Artmap Simplificada (SFAM).

3.1.1. Entrenamiento de la red SFAM.

Como se ha explicado en el capítulo 1 de esta memoria, la red Fuzzy Artmap se compone de dos redes del tipo ART, la de entrenamiento (ARTa) y la de verificación (ARTb). La conexión entre las dos redes se realiza por medio de un mapa de memoria denominado "*mapfield*". A través de esta red se realiza una clasificación supervisada.

Los datos de entrada pueden ser digitales o analógicos en el rango de 0 a 1. En el caso de valores analógicos en la red recibe el nombre de Fuzzy Artmap. La red Fuzzy Artmap se entrenó para obtener los máximos y mínimos y "*mapfield*" de cada entrada.

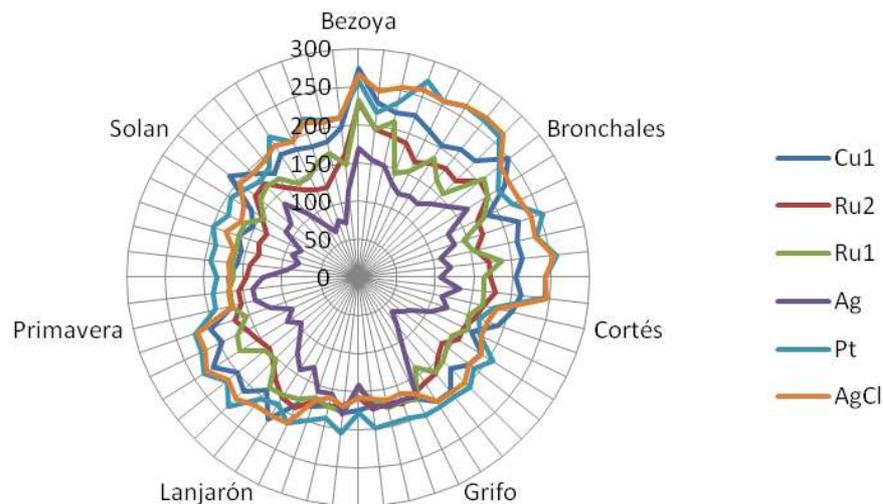


Figura 5-15. Datos de entrada para el entrenamiento de la red neuronal (mV)

Para el entrenamiento de la red se han utilizado los datos de 8 muestreos procedentes de 7 tipos diferentes de agua. Al utilizar 6 electrodos como datos de entrada se obtiene una matriz de 6x56 valores de entrada. Para obtener una visión global de dicha matriz se han representados los valores mediante una gráfica radial, donde se indica una traza para cada uno de los electrodos realizando un barrido por las siete muestras de agua potable (Fig. 5-

15). En esta gráfica se comprueba que no hay una distinción clara de los valores de potencial de cada electrodo para cada una de las muestras. Por este motivo, es necesario aplicar sistemas de redes neuronales para lograr una clasificación de las muestras.

Se ha desarrollado la Interface Gráfica de Usuario (GUI) denominada GUI Fuzzy ARTMAP (capítulo 2). Con objeto de obtener el mayor número posible de clasificaciones correctas, se ha variado el parámetro de vigilancia (ρ) y la velocidad de aprendizaje (β), realizando para ello los programas correspondientes al entrenamiento de la red (capítulo 2) en MATLAB® 2010b. La Fig. 16. muestra organigrama simplificado del proceso de entrenamiento.

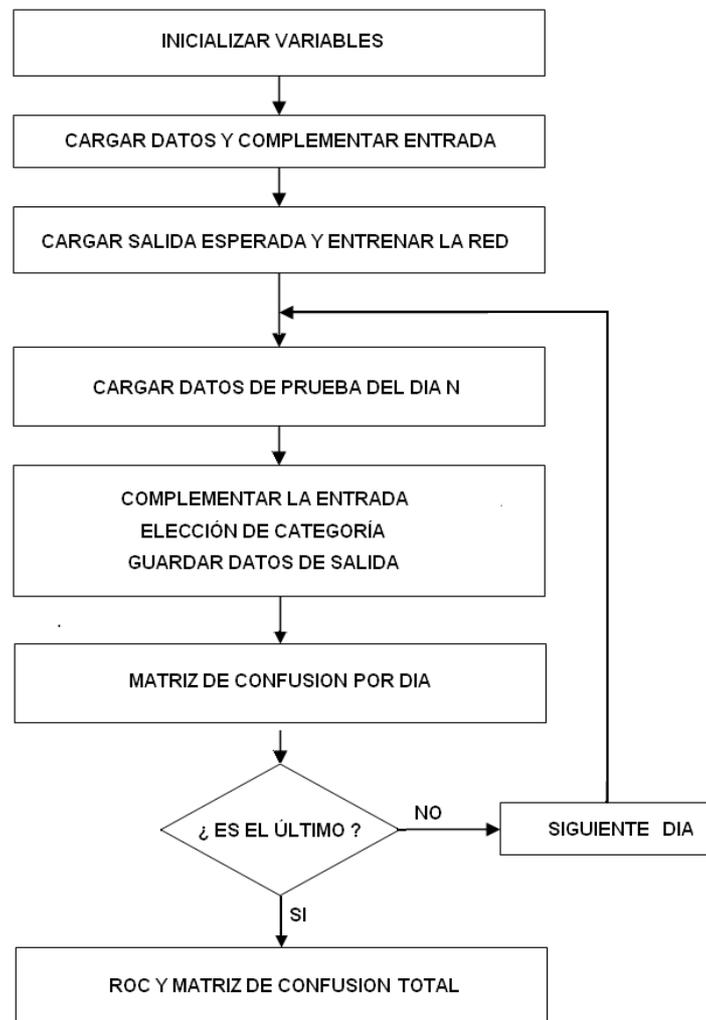


Figura 5-16. Diagrama de flujo del programa de entrenamiento de la GUI en MATLAB.

Para facilitar el manejo de tablas y gráficos a las medidas de **Bezoya** se le ha asignado el número **1**, a **Bronchales** el **2**, a **Cortés** el **3**, al **agua del grifo** el **4**, a **Lanjarón** el **5**, el número **6** a **Primavera** y el **7** a **Solan**.

En primer lugar se ha realizado un barrido general de los valores de ρ y β , utilizando la GUI desarrollada, para así determinar cuáles son los valores de estos dos parámetros con los que se obtiene una matriz de pesos y un mapfield más reducido. Posteriormente, se deberá comprobar si con estas matrices se ha obtenido un porcentaje óptimo de aciertos.

En esta tabla 5.2 se observa que se puede obtener un tamaño reducido de la matriz de pesos y de mapfield trabajando con unos valores de ρ entre 0.3 y 0.7 y una β entre 0.4 y 1.

ρ / β	0,1	0,3	0,5	0,7	0,9
0,10	>12x100	12x20	12x11	12x10	12x10
0,30	>12x100	12x20	12x11	12x10	12x10
0,50	>12x100	12x20	12x12	12x10	12x11
0,70	>12x100	12x28	>12x100	12x22	12x15
0,90	>12x100	>12x100	>12x100	12x54	12x40

Para la fase de validación se han utilizado los valores de ρ y β que obtienen un tamaño más reducido de la matriz de pesos y, además, los datos procedentes de las botellas 2^a y 3^a y 4^a medidos en diferentes días.

Se realizaron un total de 4 muestreos nuevos, cuyos resultados se muestran de forma simplificada en las **Figs. 5-17 a 5-20**. Con todos estos datos se determinará como se obtiene la mayor tasa de aciertos, con objeto de compatibilizar un porcentaje óptimo de aciertos (tabla 5.3) con un valor pequeño de la matriz de pesos (tabla 5.4).

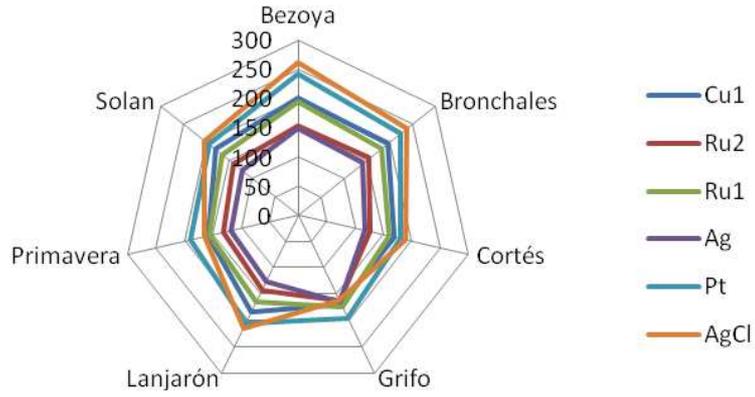


Figura 5-17. Datos de la 9ª muestra (mV).

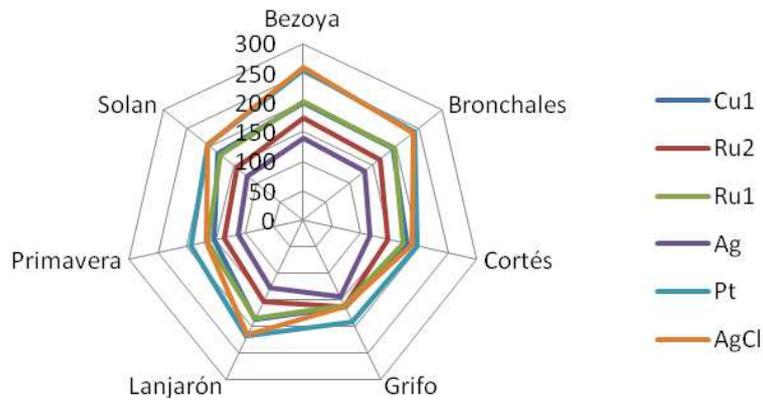


Figura 5-18. Datos de la 10ª muestra (mV).

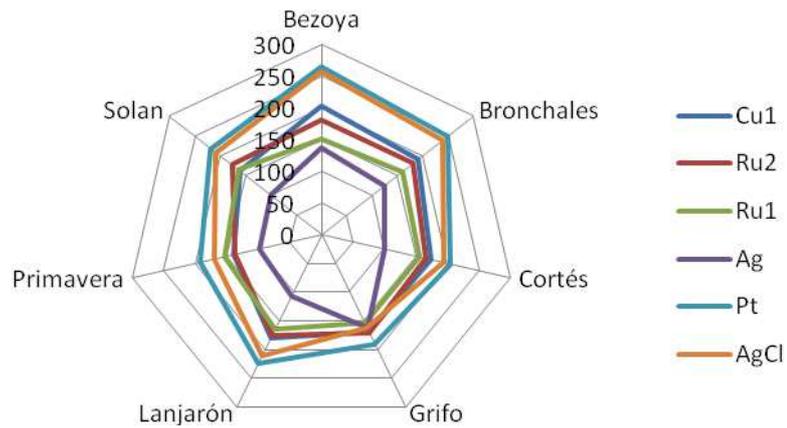


Figura 5-19. Datos de la 11ª muestra (mV).

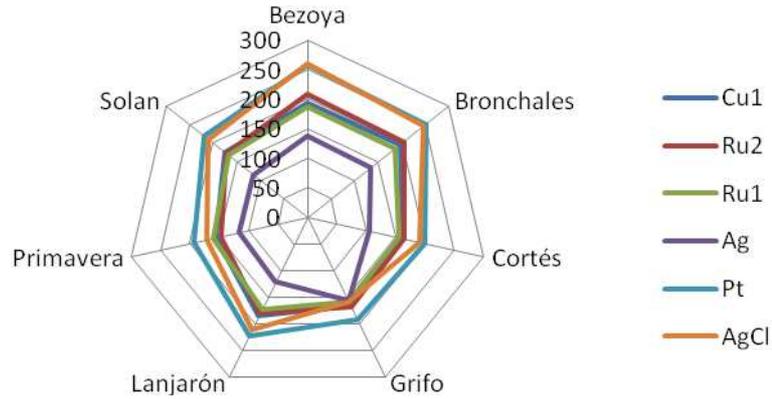


Figura 5-20. Datos de la 12ª muestra (mV).

Es necesario comprobar el porcentaje de aciertos obtenido en función de ρ y β , tal que las matrices de pesos y el mapfield sean mínimos pero no necesariamente los óptimos. Para esta comprobación se ha realizado un estudio de la variación del porcentaje de aciertos en función de los valores de ρ y β .

Tabla 5.3. Tasa de aciertos (%).							
ρ/β	0.4	0.5	0.6	0.7	0.8	0.9	1
0.30	89.3	89.3	89.3	89.3	89.3	85.7	85.7
0.35	89.3	89.3	89.3	89.3	89.3	85.7	85.7
0.40	89.3	89.3	89.3	89.3	89.3	85.7	85.7
0.45	89.3	89.3	89.3	89.3	89.3	85.7	85.7
0.50	85.7	89.3	89.3	89.3	89.3	85.7	82.1
0.55	89.3	85.7	89.3	92.9	89.3	82.1	85.7
0.60	92.9	89.3	85.7	92.9	89.3	96.4	85.7
0.65	82.1	82.1	85.7	85.7	92.9	96.4	85.7
0.70	85.7	82.1	82.1	78.6	75.0	75.0	71.4

En la Tabla 5.3 se puede observar que, la mayor tasa de acierto es del 96.4% que corresponde a dos combinaciones de ρ y β . Una combinación es para un ρ de 0.65 y una β de 0.9 y la segunda combinación corresponde a un ρ de 0.6 y una β de 0.9. En el primer caso, el tamaño de la matriz de pesos es de 12x13 pero en la segunda combinación el tamaño de la matriz de pesos es mayor, de 12x14 (Tabla 5.4).

Tabla 5.4. Tamaño de la matriz de pesos.							
ρ/β	0,4	0,5	0,6	0,7	0,8	0,9	1
0,30	12x100	12x11	12x12	12x10	12x11	12x10	12x9
0,35	12x100	12x11	12x12	12x10	12x11	12x10	12x9
0,40	12x100	12x11	12x12	12x10	12x11	12x10	12x9
0,45	12x100	12x11	12x12	12x10	12x11	12x10	12x9
0,5'0	12x100	12x12	12x12	12x10	12x13	12x11	12x13
0,55	12x100	12x13	12x12	12x11	12x13	12x13	12x13
0,60	12x100	12x100	12x12	12x11	12x13	12x14	12x13
0,65	12x100	12x100	12x13	12x12	12x16	12x13	12x13
0,70	12x100	12x100	12x18	12x22	12x17	12x15	12x16

La Fig. 5-21 muestra la evolución del tamaño de la matriz en función de β y ρ , en donde, se observa la zona de trabajo donde los valores de β y ρ minimizan el tamaño de la matriz.

Los valores óptimos de tasa de éxito y tamaño de la matriz de pesos elegidos son un ρ de 0.65 y una β de 0.9. Ahora es necesario suministrarle al programa la tabla de objetivos de la fase de validación (Tabla 5.5). En esta tabla, el "1" de la columna indica la clase a la que pertenece la muestra. Dicha tabla se compara con la tabla obtenida para así generar la matriz de confusión. Esta tabla es la misma para los diferentes días de prueba.

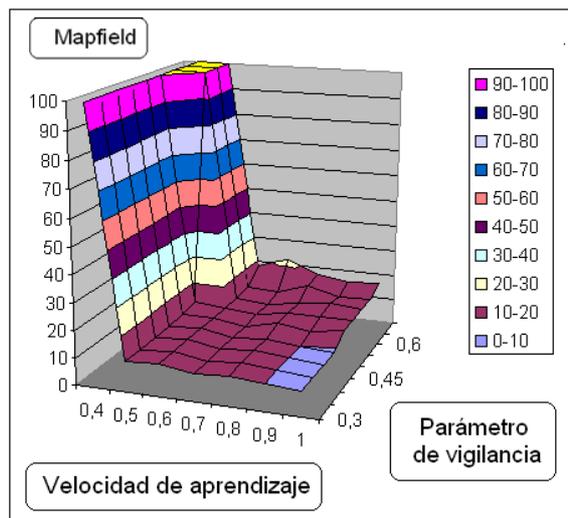


Figura 5-21. Evolución del "mapfield" en función de β y ρ .

Tabla 5.5. Objetivos durante la fase de validación							Clase
1	0	0	0	0	0	0	1
0	1	0	0	0	0	0	2
0	0	1	0	0	0	0	3
0	0	0	1	0	0	0	4
0	0	0	0	1	0	0	5
0	0	0	0	0	1	0	6
0	0	0	0	0	0	1	7

Para cada nueva muestra (días 9, 10, 11 y 12), se obtiene una matriz de salidas en función de la clase obtenida. Por ejemplo para la 9ª muestra se indica en la tabla 5.6, en donde se observa un 85.71% de aciertos (6 de 7).

Tabla 5.6. Tabla de salida para la 9ª muestra							Clase
1	0	0	0	0	0	0	1
0	1	0	0	0	0	0	2
0	0	1	0	0	0	0	3
0	0	0	1	0	0	0	4
0	0	0	0	0	0	0	5
0	0	0	0	0	1	0	6
0	0	0	0	1	0	1	7

La tabla 5.7 indica el resultado de los aciertos para cada una de las muestras. El mejor resultado se obtuvo para las muestras número 10, 11 y 12, en el que todos los datos se clasificaron correctamente (100% de aciertos).

Tabla 5.7. Aciertos por muestra.				
Muestra	9ª	10ª	11ª	12ª
% Aciertos	75	100	100	100

Como para la tasa máxima de aciertos (96.4) hay dos valores del mapfield (13 y 14) muy parecidos obtenemos la matriz de confusión y las curvas características operativas del receptor (ROC) para ambos valores. Ello nos permitirá discernir el de mayor especificidad y/o sensibilidad.

En la tabla de confusión para las 4 muestras se concluye que la tasa de aciertos obtenida es del 96.4% (**Fig. 5-22**). En la matriz de confusión las celdas de la diagonal indican el número de respuestas clasificadas correctamente.

Confusion Matrix

Output Class	1	2	3	4	5	6	7	
1	4 14.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	4 14.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	4 14.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	4 14.3%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3 10.7%	0 0.0%	0 0.0%	100% 0.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	4 14.3%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 3.6%	0 0.0%	4 14.3%	80.0% 20.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	75.0% 25.0%	100% 0.0%	100% 0.0%	96.4% 3.6%
	1	2	3	4	5	6	7	
	Target Class							

Figura 5-22. Matriz de confusión para $\rho=0.6/\beta=0.9$ y $\rho=0.65/\beta=0.9$.

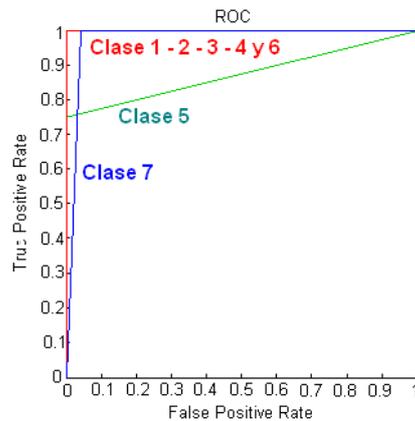


Figura 5-23. ROC para $\rho=0.6/\beta=0.9$ y $\rho=0.65/\beta=0.9$.

La Fig. 5-23 muestra las curvas de características operativas del receptor (ROC), que es una medida utilizada para comprobar la calidad de la clasificación. Cuanto más cerca esté el vértice de la esquina superior izquierda, mejor es la clasificación. En la Fig. 5-23 la clases situadas más arriba, indican mayor tasa de aciertos (clase 7 mejores resultados que las clase 5), mientras que más a la izquierda indican que los fallos se distribuyen en un número menor de clases (clase 5 menor dispersión que la 7). Las clase 1, 2, 3, 4, y 6, fueron clasificadas correctamente (esquina superior izquierda).

En este caso no hay diferencia entre las dos arquitecturas y se elige la de menor tamaño de matriz ($\rho=0.65/\beta=0.9$).

Tabla 5.8. Especificidad y sensibilidad para cada clase.		
Clases	Especificidad	Sensibilidad
1, 2, 3, 4, y 6	1	1
5	1	0.75
7	0.95	1

Una vez obtenidas las matrices de pesos (Tabla 5.9) y el mapfield (Tabla 5.10) durante el entrenamiento se trasladan estos datos al algoritmo Fuzzy Artmap Simplificado programado en el microcontrolador y se pasa a la fase de validación.

Tabla 5.9. Matriz de pesos correspondiente a 56 datos de entrenamiento ($\alpha=0.65$ y $\beta=0.9$)												
0.569	0.420	0.280	0.435	0.517	0.086	0.132	0.260	0.355	0.000	0.175	0.240	0.428
0.000	0.401	0.191	0.434	0.483	0.615	0.779	0.441	0.417	0.872	0.499	0.693	0.572
0.636	0.417	0.455	0.310	0.440	0.127	0.320	0.446	0.224	0.063	0.254	0.000	0.411
0.000	0.405	0.249	0.482	0.560	0.630	0.517	0.385	0.562	0.695	0.539	0.833	0.589
0.599	0.053	0.446	0.072	0.528	0.000	0.077	0.416	0.139	0.055	0.106	0.074	0.297
0.000	0.512	0.260	0.728	0.472	0.732	0.585	0.502	0.548	0.672	0.673	0.700	0.703
0.739	0.495	0.569	0.382	0.496	0.000	0.675	0.546	0.271	0.141	0.389	0.029	0.418
0.093	0.379	0.103	0.398	0.504	0.500	0.000	0.318	0.503	0.362	0.378	0.828	0.582
0.482	0.834	0.449	0.664	0.867	0.000	0.107	0.282	0.482	0.159	0.119	0.284	0.139
0.148	0.000	0.361	0.184	0.133	0.626	0.638	0.580	0.322	0.685	0.581	0.490	0.861
0.799	0.898	0.667	0.707	0.895	0.177	0.000	0.486	0.531	0.074	0.318	0.357	0.243
0.023	0.000	0.295	0.198	0.105	0.655	0.888	0.487	0.330	0.777	0.521	0.484	0.757

Tabla 5.10. Mapfield correspondiente a 56 datos de entrenamiento($\rho=0.65$ y $\beta=0.9$)												
1	1	2	2	2	3	4	5	5	6	7	7	3

Para implementar el algoritmo en el microcontrolador, también es necesario conocer la matriz de máximos y mínimos (tabla 5.11) de las entradas para utilizarlas en la normalización de los datos.

272.73	152.74
228.88	124.41
232.1	138.45
179.55	62.92
272.71	165.88
267.66	157.64

3.1.2. Implementación en el microcontrolador de la SFAM.

Para materializar la red neuronal artificial se utilizó el microcontrolador PIC18F4550 cuyas características fundamentales se indicaron en el capítulo 2 de la presente memoria.

El software para el microcontrolador PIC18F4550 ha sido diseñado para obtener el valor medio en cada canal. El proceso de medida se ha dividido en dos partes: periodo de entrenamiento y periodo de prueba. En el periodo de entrenamiento los datos se mandan al PC a través del canal de comunicaciones RS232. De esta forma los datos se pueden utilizar en el algoritmo de entrenamiento realizado en MATLAB R2010b. El software de adquisición de datos se ha desarrollado utilizando el Visual Basic 6.0 y Microsoft Excel 2003.

En el periodo de prueba, los datos se han almacenado directamente en la memoria del microcontrolador. La función de clasificación mediante la red SFAM se implementó en el microcontrolador como muestra la Fig. 5-24.

El código de esta rutina es en lenguaje C y se convierte en código hexadecimal utilizando un compilador cruzado. El archivo HEX se descarga en la memoria flash del microcontrolador. La red SFAM se ha programado en 12986 bytes de memoria de programa (39%) y 1950 bytes de memoria de datos (79%).

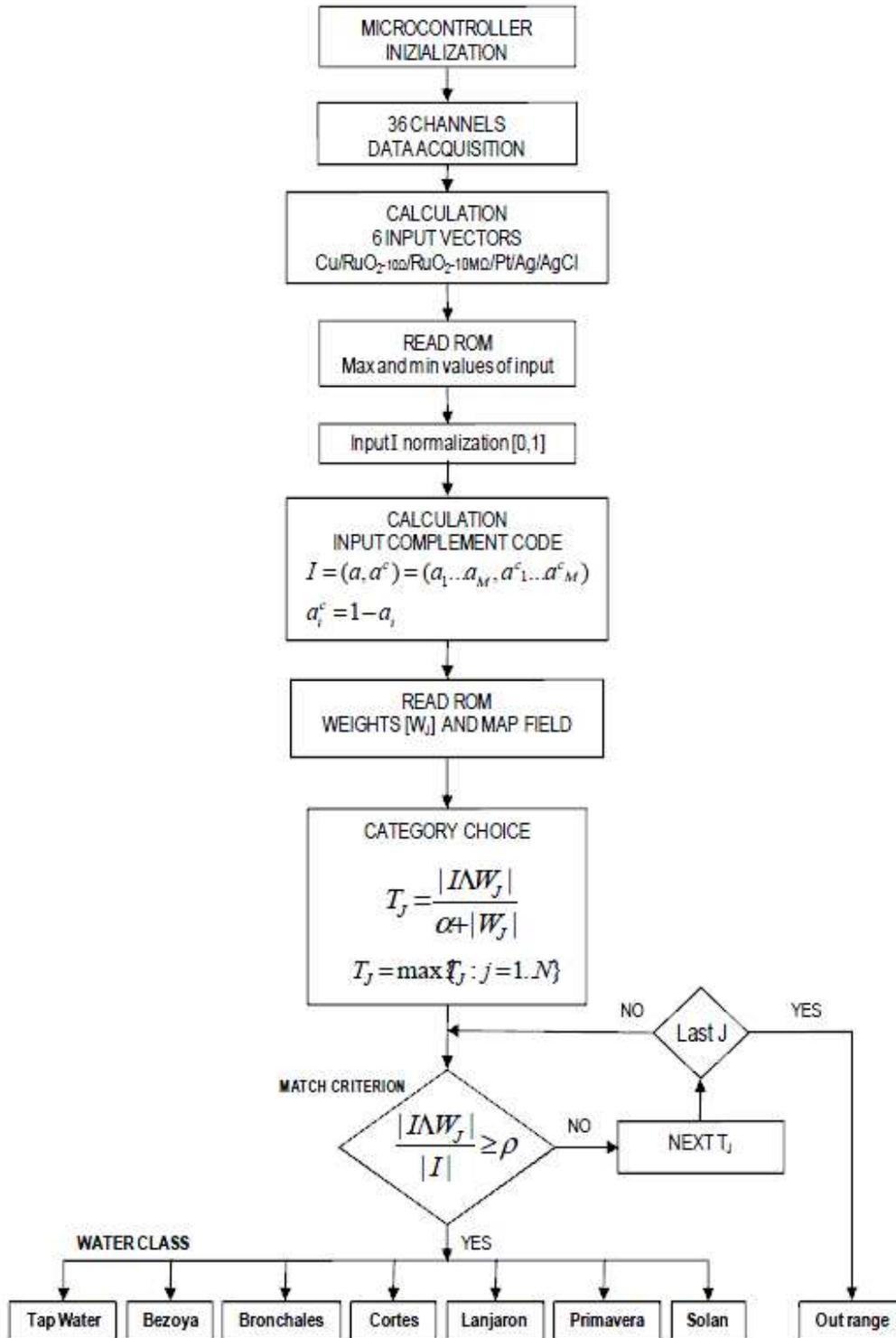


Figura 5-24. Organigrama de la implementación de la red SFAM en el μC .

3.1.3. Resultados y discusión con la red SFAM.

Una vez realizado el entrenamiento y verificación de la red neuronal y determinado los valores de los parámetros necesarios para implementar el

algoritmo de la red neuronal en el microcontrolador se pasa a la fase de prueba. En esta fase de prueba se utilizan los datos de nuevas muestras obtenidas directamente con el microcontrolador. Para esta fase se han utilizado 5 muestras más, de la 13^a a la 17^a muestra (Figs. 5-25 a 5-29), obtenidas a partir de las botellas 3^a y 4^a.

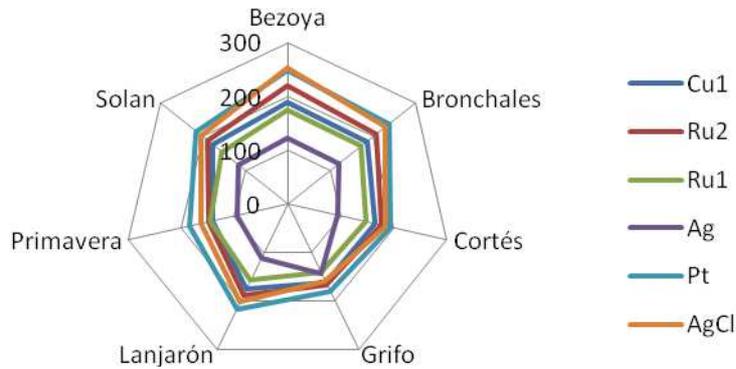


Figura 5-25. Datos de la 13^a muestra (mV).

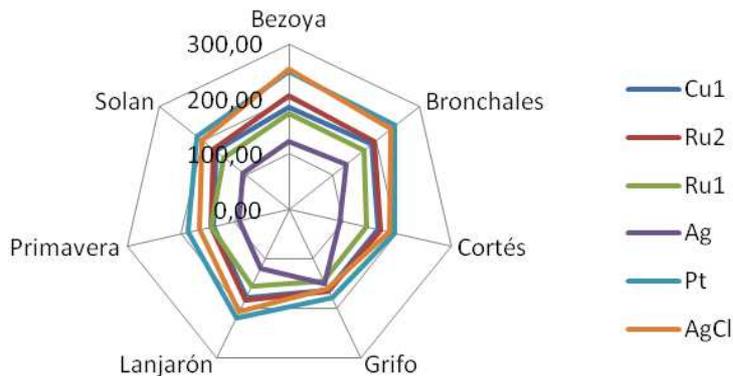


Figura 5-26. Datos de la 14^a muestra (mV).

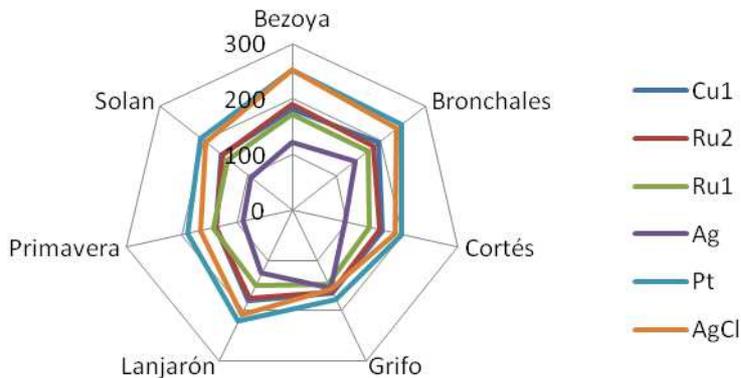


Figura 5-27. Datos de la 15^a muestra (mV).

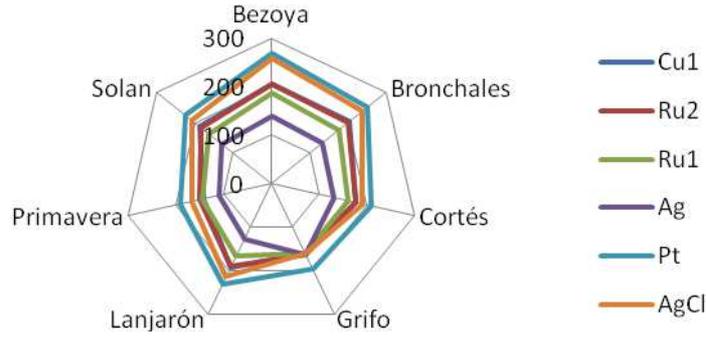


Figura 5-28. Datos de la 16ª muestra (mV).

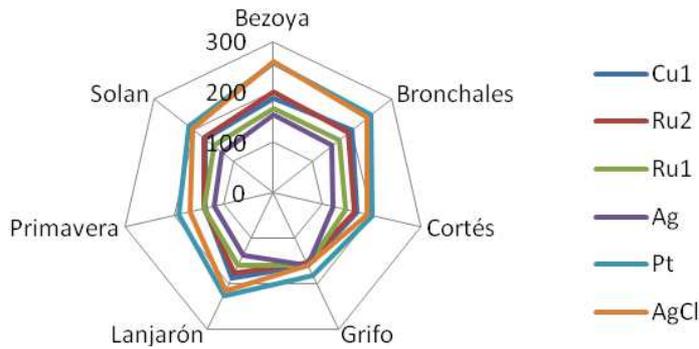


Figura 29. Datos de la 17ª muestra (mV).

Confusion Matrix

1	5 14.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	5 14.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	2 5.7%	0 0.0%	0 0.0%	0 0.0%	1 2.9%	66.7% 33.3%
4	0 0.0%	0 0.0%	0 0.0%	5 14.3%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3 8.6%	0 0.0%	2 5.7%	60.0% 40.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5 14.3%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	3 8.6%	0 0.0%	2 5.7%	0 0.0%	2 5.7%	28.6% 71.4%
	100% 0.0%	100% 0.0%	40.0% 60.0%	100% 0.0%	60.0% 40.0%	100% 0.0%	40.0% 60.0%	77.1% 22.9%
	1	2	3	4	5	6	7	

Target Class

Figura 5-30. Matriz de confusión con 8 muestras de entrenamiento y 5 de prueba. ($\rho=0.65$ y $\beta=0.9$).

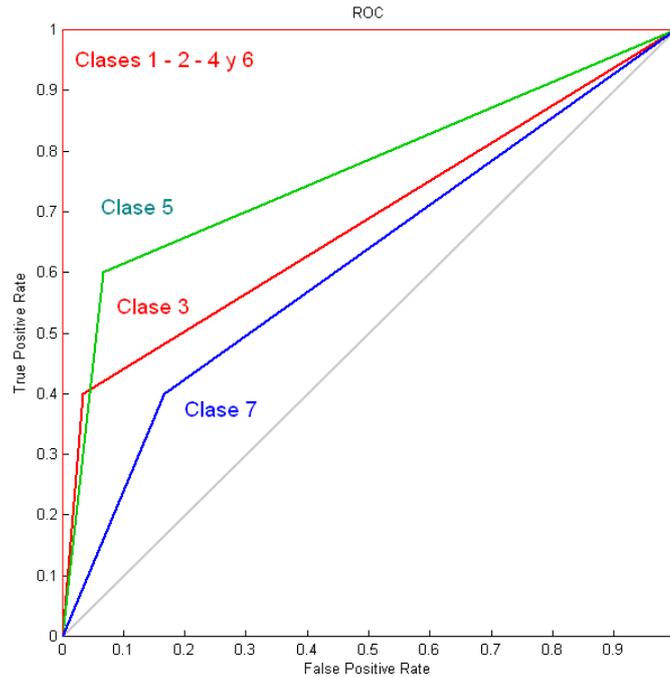


Figura 5-31. ROC con 8 muestras de entrenamiento y 5 de prueba ($\rho=0.65$ y $\beta=0.9$).

Los resultados de la fase de prueba con el microcontrolador se indican mediante la matriz de confusión y el ROC. Las Fig. 5-30 y 5-31 muestran la matriz de confusión y el ROC, respectivamente, obtenidas con $\rho=0.65$ y $\beta=0.9$, lo que da una tasa de aciertos del 77.1%.

Debido a que los últimos resultados de la clasificación no ofrecen una tasa de aciertos muy elevada (77.1), cabe la posibilidad de que aumentando el número de muestras de entrenamiento se obtenga una mejor clasificación. También es lógico pensar que el tamaño de la matriz de datos y el mapfield aumentarán. Es interesante pues, realizar un nuevo entrenamiento con un número mayor de muestras y observar el efecto en el porcentaje de acierto y en el tamaño de las matrices. No hay que olvidar que el tamaño de las matrices podría aumentar por encima de la capacidad de memoria del microcontrolador y por tanto no se podría implementar el algoritmo directamente en él.

Se han añadido como datos para el nuevo entrenamiento de la 9ª a la 12ª muestra (utilizadas anteriormente para la 1ª validación), dejando sólo como muestras para la validación los cinco últimos días (de la 13ª a la 17ª). Los resultados obtenidos en el microcontrolador para $\rho=0.65/\beta=0.9$ se muestran en la tabla 5.12.

Tabla 5.12. Comparación de los cinco últimos días.								Aciertos	Muestras entrenamiento
13ª muestra	1	2	3	4	7	6	7	6	8($\rho=0.65/\beta=0.9$)
	2	2	3	4	5	6	5	5	12 ($\rho=0.65/\beta=0.9$)
14ª muestra	1	2	7	4	5	6	5	5	8($\rho=0.65/\beta=0.9$)
	2	2	3	4	5	6	5	5	12 ($\rho=0.65/\beta=0.9$)
15ª muestra	1	2	3	4	7	6	3	5	8($\rho=0.65/\beta=0.9$)
	2	2	3	4	5	6	7	6	12 ($\rho=0.65/\beta=0.9$)
16ª muestra	1	2	7	4	5	6	5	5	8($\rho=0.65/\beta=0.9$)
	1	2	7	4	5	6	5	5	12 ($\rho=0.65/\beta=0.9$)
17ª muestra	1	2	7	4	5	6	7	6	8($\rho=0.65/\beta=0.9$)
	1	2	7	4	5	6	7	6	12 ($\rho=0.65/\beta=0.9$)

El “*mapfield*” (Tabla 5.13) aumenta al aumentar los datos para el entrenamiento, sin embargo los resultados de la tasa de aciertos (Tabla 5.12) no mejoran como cabría esperar, ya que sólo ha habido aumento para la 15ª muestra, que se compensa con la disminución en la 13ª muestra, no habiendo variación en las restantes muestras.

Tabla 5.13. Mapfield correspondiente a 12 muestras de entrenamiento($\rho=0.65$ y $\beta=0.9$)																	
1	1	1	2	2	2	3	3	4	5	5	6	7	7	7	1	3	1

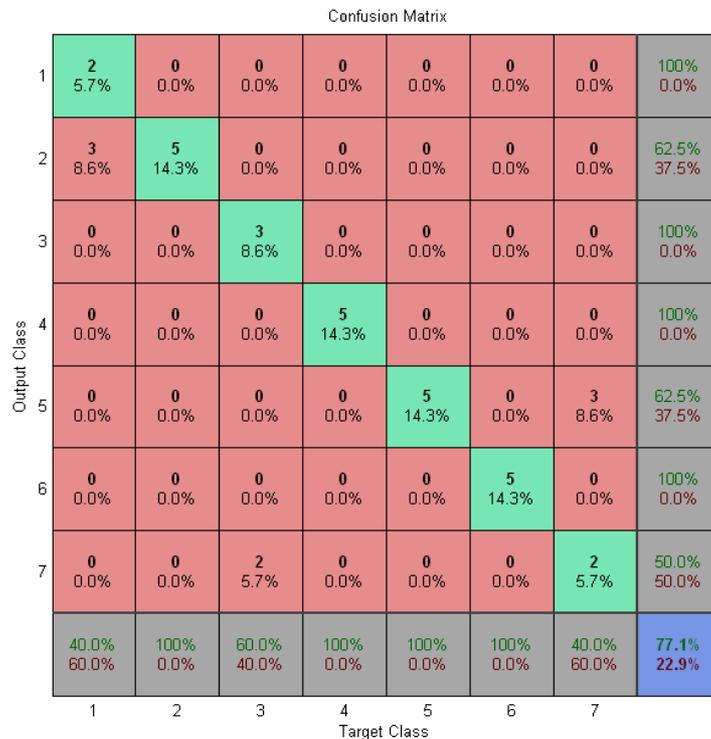


Figura 5-32. Matriz de confusión para 12 muestras de entrenamiento y 5 de prueba. ($\rho=0.65$ y $\beta=0.9$).

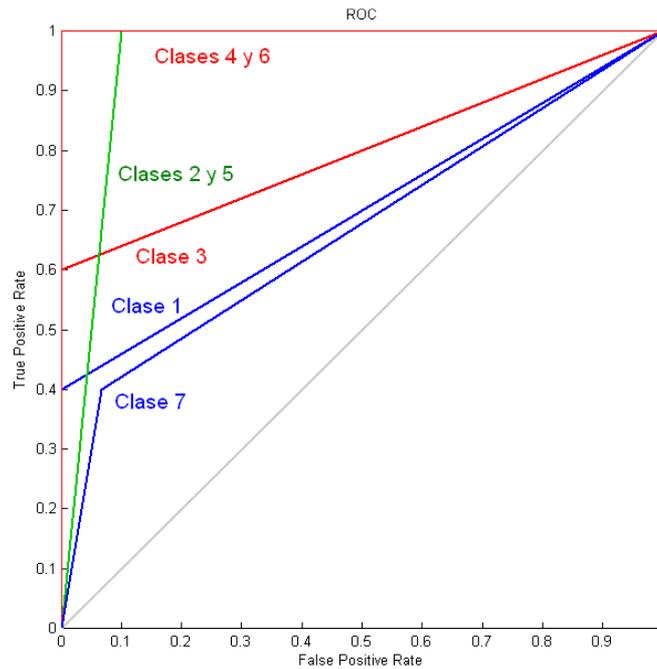


Figura 5-33. ROC para 12 muestras de entrenamiento y 5 de prueba. ($\rho=0.65$ y $\beta=0.9$).

Primeramente se ha entrenado con 8 muestras y se ha obtenido una tasa de aciertos del 96.4% con 4 muestras de validación en MATLAB. A continuación, con las matrices de pesos, máximos y mínimos obtenidos, se ha probado la red SFAM en el microcontrolador, obteniendo una tasa de aciertos del 77.1% sobre un total de 5 muestras nuevas. En tercer lugar, como la tasa de aciertos ha disminuido bastante, se ha creído conveniente utilizar 12 muestras para el entrenamiento (las 8 muestras de entrenamiento, más las 4 utilizadas en la validación del primer entrenamiento), y las 5 muestras de prueba en el microcontrolador como datos de validación, que se han comparado (Tabla 5.12) con lo obtenido en el microcontrolador. Los resultados obtenidos con 12 muestras, han empeorado con respecto a los obtenidos con 8 muestras, en contra de lo que cabría esperar.

Dado que los resultados no mejoran como cabría esperar, se ha hecho un estudio de la tasa de variación con 12 muestras (Tabla 5.14) y con 8 muestras (Tabla 5.15) y de la matriz de pesos (Tabla 5.16) en función de ρ y β . En dichas tablas se ha resaltado en rojo el aumento de la tasa de aciertos y del mapfield con las correspondientes a 8 muestras de entrenamiento y 5 para validación (Tablas 5.16 y 5.17). Los valores marcados en azul indican donde no

ha habido variación y en negro las celdillas donde ha habido una disminución, tanto de la tasa de aciertos como de la matriz de pesos.

Tabla 5.14. Tasa de aciertos (%) para 12 muestras de entrenamiento y 5 en validación.

ρ/β	0.4	0.5	0.6	0.7	0.8	0.9	1
0.3	80.0	80.0	80.0	77.1	74.3	74.3	68.6
0.35	80.0	80.0	80.0	77.1	74.3	74.3	68.6
0.4	80.0	80.0	80.0	77.1	74.3	74.3	68.6
0.45	80.0	77.1	80.0	77.1	74.3	74.3	68.6
0.5	80.0	80.0	80.0	74.3	74.3	74.3	82.9
0.55	80.0	80.0	82.9	80.0	74.3	65.7	82.9
0.6	80.0	80.0	80.0	77.1	71.4	77.1	82.9
0.65	77.1	71.4	62.9	71.4	77.1	77.1	65.7
0.7	65.7	65.7	77.1	71.4	62.9	68.6	71.4

Tabla 5.15. Tasa de aciertos (%) para 8 muestras de entrenamiento y 5 en validación.

ρ/β	0.4	0.5	0.6	0.7	0.8	0.9	1
0.3	82.9	80.0	77.1	77.1	74.3	77.1	77.1
0.35	82.9	80.0	77.1	77.1	74.3	77.1	77.1
0.4	82.9	80.0	77.1	77.1	74.3	77.1	77.1
0.45	82.9	80.0	77.1	77.1	74.3	77.1	77.1
0.5	74.3	80.0	77.1	71.4	74.3	80.0	74.3
0.55	80.0	80.0	68.6	80.0	77.1	71.4	82.9
0.6	80.0	85.7	71.4	80.0	71.4	74.3	82.9
0.65	74.3	68.6	68.6	62.9	77.1	77.1	82.9
0.7	80.0	74.3	77.1	74.3	77.1	71.4	71.4

Tabla 5.16. Mapfield para 12 muestras de entrenamiento y 5 en validación.

ρ/β	0.4	0.5	0.6	0.7	0.8	0.9	1
0.3	14	12	11	14	12	10	9
0.35	14	12	11	14	12	10	9
0.4	14	12	11	14	12	10	9
0.45	14	12	11	14	12	10	9
0.5	15	13	11	15	13	12	13
0.55	15	14	12	16	15	12	13
0.6	16	14	12	16	17	16	13
0.65	100	16	15	22	23	16	14
0.7	100	28	26	29	20	19	18

No hay una región claramente definida donde la tasa de aciertos sea claramente mayor, lo que parece indicar que la cantidad de datos tomados (8

muestras) es suficiente para una buena clasificación. Tampoco el mapfield es mayor en todos los casos, lo que parece indicar que existen otros parámetros que también influyen en el tamaño de la matriz de pesos.

Tabla 5.17. Mapfield para 8 muestras de entrenamiento y 5 en validación.

ρ/β	0.4	0.5	0.6	0.7	0.8	0.9	1
0.3	100	11	12	10	11	10	9
0.35	100	11	12	10	11	10	9
0.4	100	11	12	10	11	10	9
0.45	100	11	12	10	11	10	9
0.5	100	12	12	10	13	11	13
0.55	100	13	12	11	13	13	13
0.6	100	100	12	11	13	14	13
0.65	100	100	13	12	16	13	13
0.7	100	100	18	22	17	15	16

Finalmente la tabla 5.18, muestra una comparación de la tasa de aciertos, con los valores de ρ y β iniciales, con 8 y 12 muestras de entrenamiento y 5 de validación. Los resultados obtenidos con $\rho=0.6$ y $\beta=0.9$ para 12 muestras de entrenamiento y 5 de prueba coinciden, pero mejoran ligeramente los resultados con $\rho=0.6$ y $\beta=0.9$ para 8 muestras de entrenamiento y 5 de prueba (tabla 5.18).

Tabla 5.18. Comparación de tasa de aciertos

	8 muestras	12 muestras
$\rho=0.6$ y $\beta=0.9$	74.3%	77.1%
$\rho=0.65$ y $\beta=0.9$	77.1%	77.1%

También se ha hecho un estudio eliminando el agua del grifo dando los siguientes resultados:

- a) en el mejor de los casos los resultados son exactamente iguales
- b) en el resto de los casos se obtiene un número menor de aciertos

Lo que parece indicar que el agua del grifo actúa como discriminante.

3.2. Red FF-MLP.

3.2.1. Entrenamiento de la red FF-MLP.

El segundo tipo de red neuronal artificial que se ha empleado para realizar los análisis es la red Perceptrón Multicapa feed-forward cuyas características se han descrito en el capítulo 2.

Con el fin de entrenar a la red neuronal multicapas “*feed-forward*” se ha utilizado la interface gráfica de usuario para el reconocimiento de patrones “*nprtool GUI*” (*Neural Network Pattern Recognition tool Graphical User Interface*) de MATLAB® 2010b.

La red posee dos capas de neuronas con alimentación hacia adelante con las funciones de transferencia por defecto de tipo tangente sigmoidea, tanto en las capas ocultas como en las de salida. Se denomina “*patternnet*” (red de reconocimiento de patrones) en MATLAB ® (Fig.34).

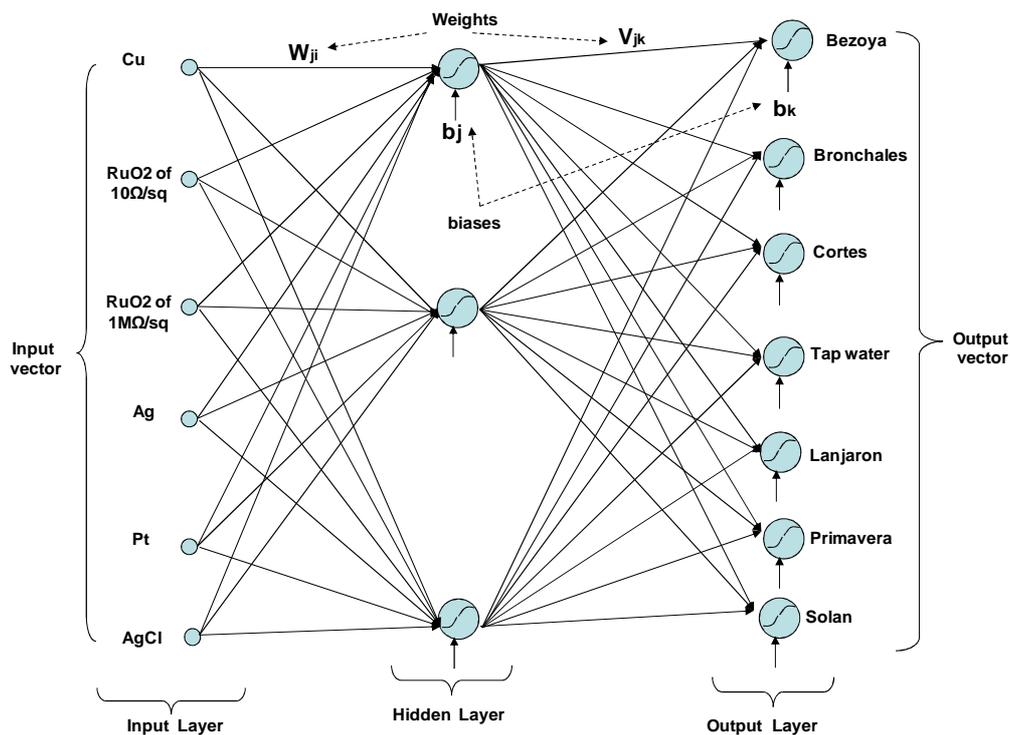


Figura 5-34. Red neuronal artificial multicapa feed forward.

Los datos objetivo para las redes de reconocimiento de patrones deben consistir en los vectores de todos los valores de entrada a cero a excepción de un 1 en el elemento i , donde i es la clase que va a representar. Cuando un

vector de entrada de la categoría adecuada se aplica a la red, la neurona correspondiente debe producir un 1, y la salida de las otras neuronas un 0.

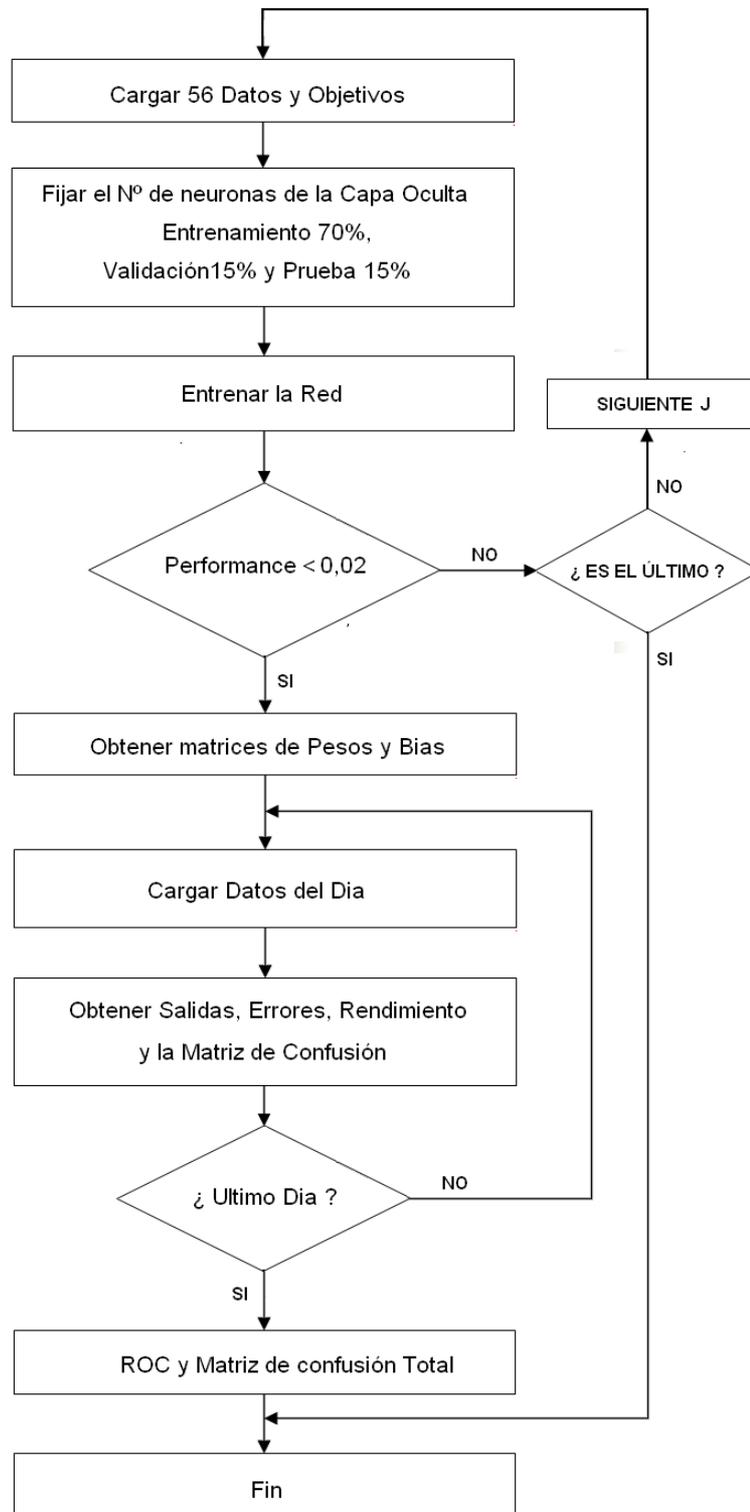


Figura 5-35. Diagrama de flujo de la perceptrón multicapa.

Para trabajar con la red se han utilizado 56 muestras, de las cuales 40 (70%) lo han sido para el entrenamiento, 8 (15%) para la validación y 8 (15%) para el test (tabla 5.29). La división de las muestras ha sido aleatoria usando la “*random data división function*”.

Para el entrenamiento de esta red se ha utilizado el algoritmo “*scaled conjugate gradient back-propagation*” implementado en MATLAB®. Para el entrenamiento de la red, se ha variado el número de neuronas de la capa oculta realizando para ello el programa (script de MATLAB) MLP_FF, cuyo organigrama se muestra en la Fig. 5-35.

Los resultados de la clasificación con este tipo de red neuronal se muestran en la Tabla 5.19, donde se indica la tasa de éxito total y para cada una de las clases (tipo de agua) según el número de número de neuronas en la capa oculta.

Tabla 5.19	CLASE							Total	%
	Neuronas	1	2	3	4	5	6		
10	6	4	6	8	5	6	0	35	62.5
12	6	5	5	6	5	6	3	36	64.3
14	8	5	5	6	4	4	2	34	60.7
16	7	8	3	7	7	8	2	42	75.0
18	3	8	6	7	2	7	2	35	62.5
20	8	7	8	8	7	8	8	54	96.4
22	6	8	3	8	6	8	2	41	73.2
24	6	8	3	8	6	7	3	41	73.2
26	8	6	3	8	6	7	2	40	71.4
28	5	6	1	8	7	8	4	40	71.4
30	8	7	5	7	7	7	2	43	76.8

El número de neuronas ocultas de 20 es el que mejor resultado ha dado (Tabla 5.19), la columna Total indica el número de clasificaciones correctas (51 para 20 neuronas) de un total de 63 datos (muestras de la 1ª a la 8ª), mientras que la fila Total indica el numero de clasificaciones correctas por cada clase.

En esta misma tabla se observa que el número de clasificaciones correctas de la clase 4 (agua del grifo) es del 100% (de 20 a 28 neuronas), mientras que la clase 7 (Solan) se confunde con las clases 3 (Cortés) y 5 (Lanjarón).

La Fig. 5-36 muestra la evolución del número de clasificaciones correctas en función del número de neuronas de la capa de entrada, mientras que en la Fig. 5-37 se muestra en función de la clase.

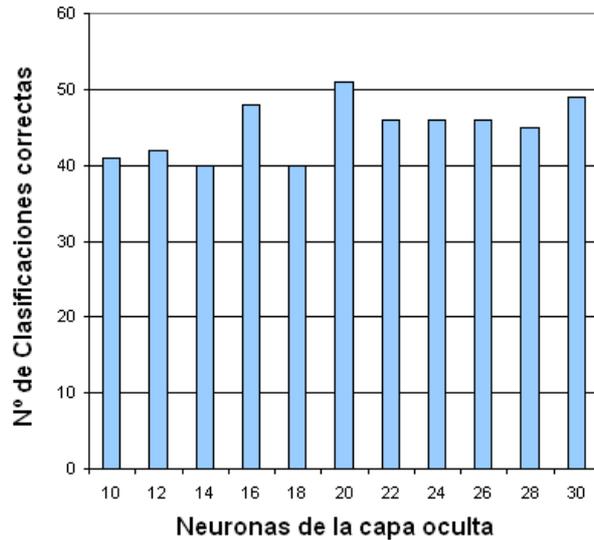


Figura 5-36. Evolución el número de aciertos en función de las neuronas de la capa oculta.

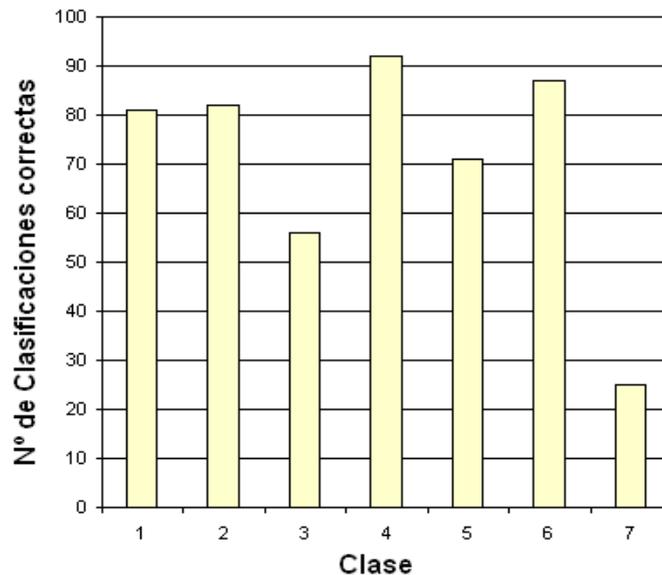


Figura 5-37. Evolución el número de las clasificaciones correctas en función de la clase.

Los valores de MSE (Mean Squared Error) y el porcentaje de error para el entrenamiento, validación y prueba se muestran en la tabla siguiente (5.20).

Tabla 5.20. MSE y % de Error para el entrenamiento, validación y prueba.			
	Muestras	MSE	%E
Entrenamiento	40	$5.83530 \cdot 10^{-3}$	0
Validación	8	$6.80533 \cdot 10^{-3}$	0
Prueba	8	$6.59341 \cdot 10^{-2}$	25.0000

La Fig. 5-38 muestra la matriz de confusión para el entrenamiento, validación y prueba.

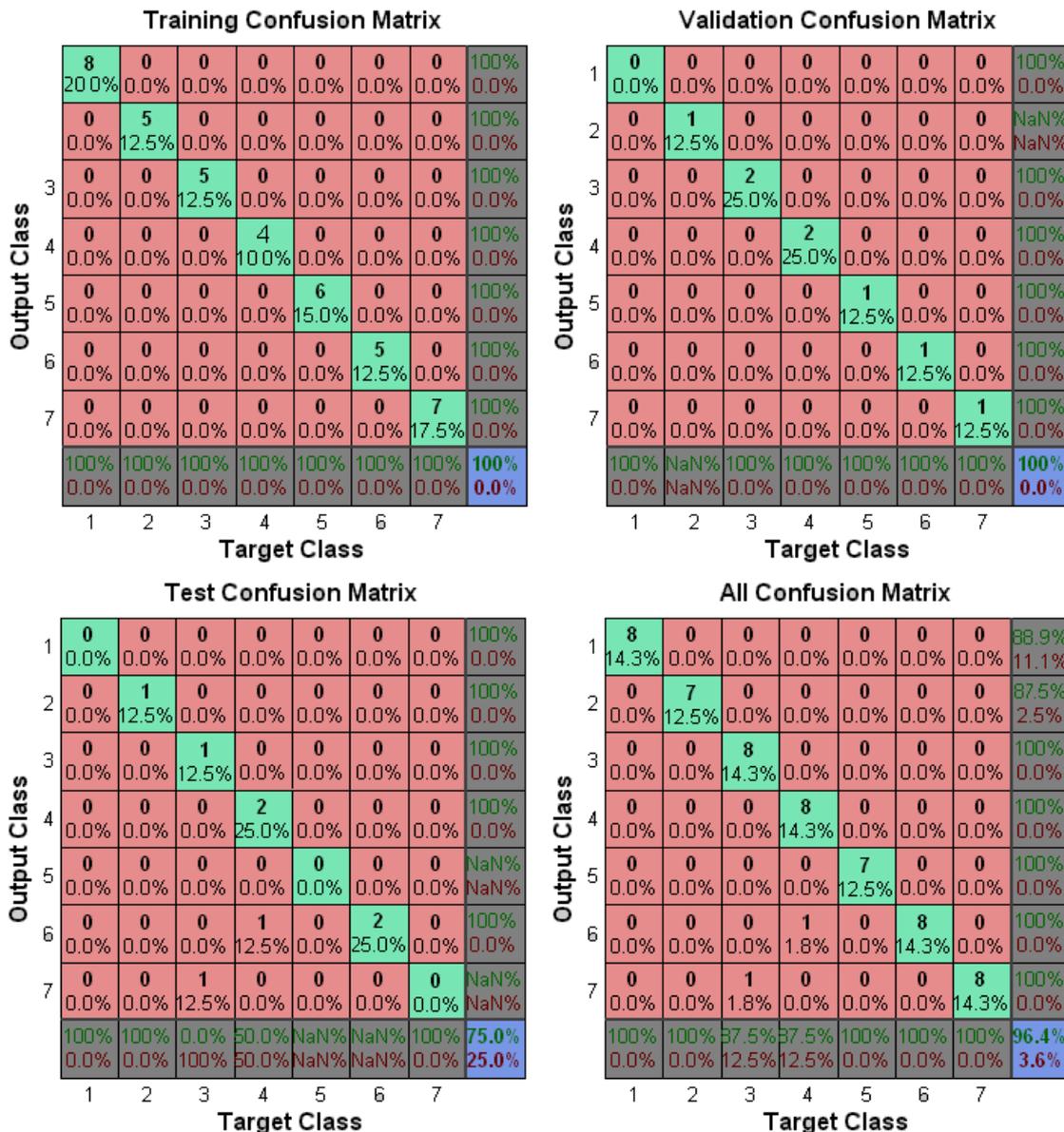


Figura 5-38. Matriz de confusión durante el entrenamiento, validación y prueba.

La **Fig. 5-39** muestra la curva de las características operativas del receptor (ROC), la tasa de verdaderos positivos (sensibilidad) frente a la tasa de falsos positivos ($1 - \text{especificidad}$).

La **Fig. 5-40** muestra el histograma de error, en donde las barras azules representan los datos de entrenamiento, las barras verdes representan los datos de devalidación, y las barras rojas representan los datos de prueba. El histograma puede dar una indicación de los valores atípicos, que son los puntos de datos donde el ajuste es significativamente peor que en la mayoría de los datos. En dicha figura se observa la distribución del tamaño de error. Por lo general la mayoría de errores se encuentran cerca de cero, con muy pocos errores lejanos de cero.

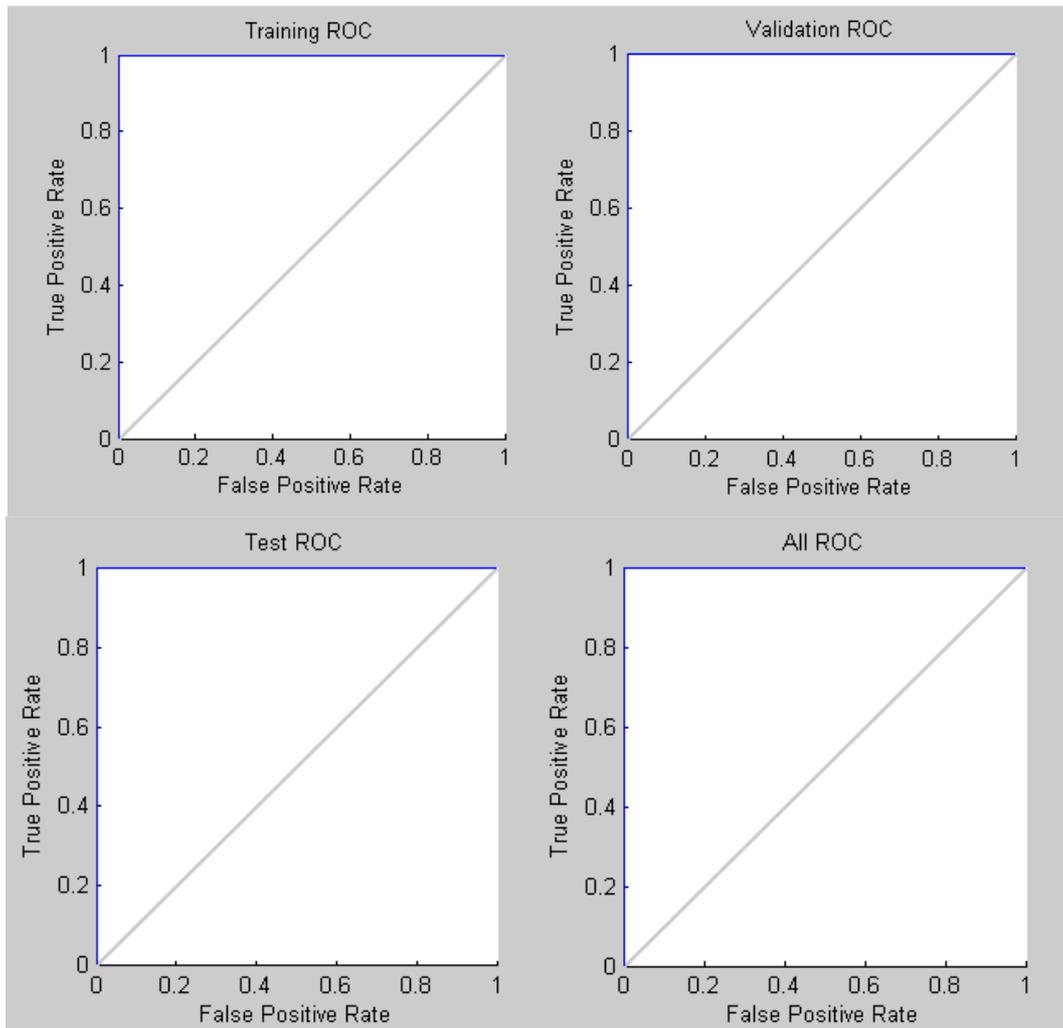


Figura 5-39. Curva de características operativas del receptor (ROC).

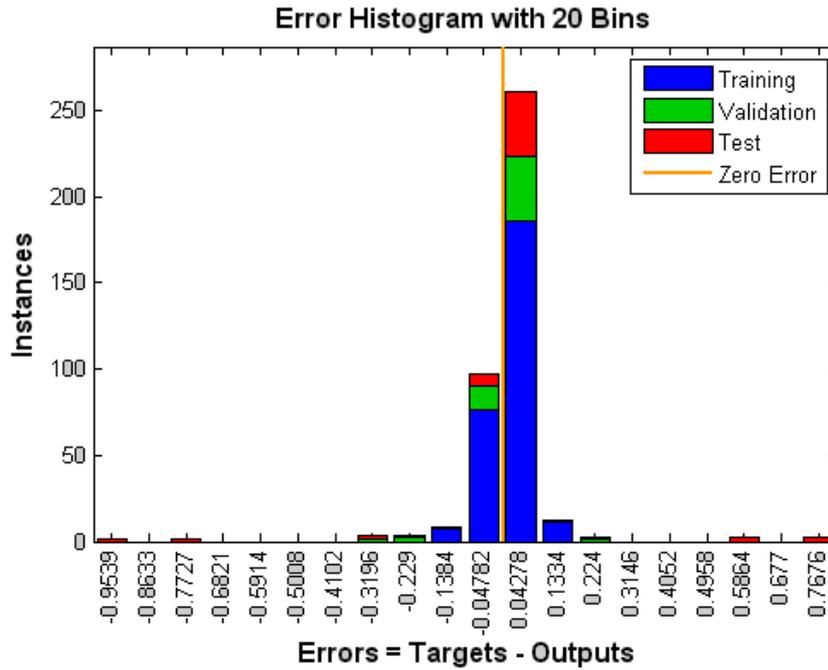


Figura 5-40. Histograma de error.

La Fig. 5-41 muestra el esquema del perceptrón multicapa proporcionado por MATLAB® R2010b.

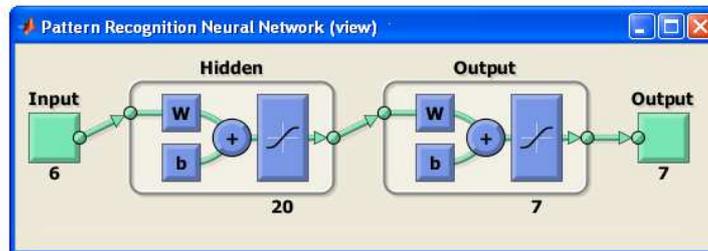


Figura 5-41. Esquema de la red Perceptrón multicapa en MATLAB®.

Después de la fase de entrenamiento se obtuvieron 2 matrices de pesos. una de 6×20 (capa de entrada a capa oculta) y otra de 20×7 (capa oculta a capa de salida) así como dos matrices de polarización de 1×20 (capa oculta) y de 1×7 (capa de salida). Estas matrices han sido utilizadas para la implementación de la red en el microcontrolador.

Los pesos y sesgos se pueden obtener utilizando los comandos NET de MATLAB. Para los pesos y sesgos de la capa oculta, $net.iw\{1,1\}$ y $net.b\{1\}$, respectivamente (Tabla 5.31). Para los pesos y sesgos de la capa de salida, $net.iw\{2,1\}$ y $net.b\{2\}$, respectivamente (Tabla 5.32).

Tabla 5.21. Pesos y sesgos (B1) de entrada.							Tabla 5.22. Pesos y sesgos (B2) de salida							
B1	Matriz de pesos de entrada						B2	Matriz de pesos de salida						
2.79	-0.31	-1.78	1.10	-0.68	-0.39	-2.64	1.18	-2.21	2.41	0.25	-0.41	0.67	-0.13	0.31
2.34	0.05	-1.16	-0.40	1.18	1.21	-0.91	-0.21	-0.30	-1.02	-0.63	-0.69	-1.00	-0.61	0.15
1.14	-1.38	0.03	1.53	2.12	1.75	-0.75	-1.09	-0.64	-0.08	-1.78	-0.43	1.06	1.83	-0.85
-1.50	0.99	-0.96	0.08	1.22	-1.10	1.66	-0.56	1.54	-0.16	0.42	-0.14	0.63	0.11	0.03
-0.97	1.48	1.38	-0.32	0.57	1.27	-0.21	-1.01	-0.18	0.44	0.40	-0.19	-0.77	0.54	-0.07
-1.64	1.38	0.13	-0.98	-0.47	0.51	-1.13	0.18	0.54	1.09	0.97	0.76	0.56	-0.25	0.08
0.80	0.13	1.21	-0.90	-0.80	-0.95	-2.23	-1.98	-0.43	0.22	0.15	0.85	1.63	-0.22	0.36
-1.29	1.27	-1.06	1.15	-0.64	0.23	0.04		-0.06	0.37	0.11	-0.11	0.63	0.87	0.75
0.51	-0.76	1.08	0.74	0.53	-0.44	1.61		0.27	0.54	0.04	-0.32	0.17	-0.34	0.12
-0.57	0.09	1.82	-0.64	-0.81	-1.44	-2.05		-0.64	-1.02	1.62	0.78	-0.76	-0.09	-0.89
0.76	-0.69	-1.48	0.74	-1.25	-1.00	1.84		0.91	-1.03	-0.34	-1.88	-0.24	0.98	0.32
-0.55	-1.34	-0.19	-1.28	0.47	-1.61	-0.99		-0.91	-0.01	0.36	0.22	-0.35	0.22	1.28
-1.50	-0.75	-0.55	-1.15	-0.44	-0.02	-1.99		0.59	-0.91	0.50	0.50	-1.34	0.40	-0.41
-0.74	-0.61	-1.71	-0.73	0.04	-1.60	-2.09		-0.45	-1.38	-0.41	0.52	0.12	0.22	1.70
1.16	1.21	-0.78	-0.08	1.06	0.12	2.16		0.25	0.55	-1.06	-0.01	0.37	-0.86	1.33
-0.57	0.08	1.33	0.51	2.09	-0.18	-1.19		-0.08	0.04	-0.46	1.39	-0.55	-0.89	0.35
-0.68	-0.63	1.34	-0.02	1.94	-0.33	1.51		0.89	0.47	0.03	0.43	-1.18	-0.66	-0.39
1.06	1.12	-1.12	-1.39	-0.67	1.06	-0.66		-0.77	0.09	-0.19	-0.27	0.05	-0.60	-0.36
2.18	1.25	-0.71	-0.48	-0.92	1.24	-0.45		0.00	-1.15	0.31	-0.66	-0.53	-0.23	-0.78
1.24	1.65	1.39	0.85	-1.14	-0.90	-1.27		-0.68	-1.37	-0.18	-0.63	0.77	-0.78	0.20
2.07	1.23	0.79	-1.21	0.43	-0.39	2.29		-0.64	-0.36	1.10	0.19	0.02	-1.78	0.42
-2.35	-0.16	-0.87	1.63	1.04	0.97	-0.48		0.72	0.49	0.13	0.25	0.73	0.49	0.30

3.2.2. Implementación en el microcontrolador de la FF-MLP.

Los pesos (W_{ij}) y sesgos (B_j) de la red neuronal se obtienen del PC durante el período de entrenamiento. A partir de los vectores de entrada (I_N) y de los pesos y sesgos, el microcontrolador calcula la salida para cada uno de los veinte nodos ocultos.

Mediante el uso de estos datos y de los valores de los pesos se obtienen los valores de los nodos de salida.

Al ser la función de salida una tansig el valor de las salidas se encuentra entre [-1.1] por lo que deben ser ajustadas a [0.1]. La salida con valor 1 o lo más próximo posible a 1 señalará la clase a la que pertenece el vector de entrada. El número correspondiente a la clase de salida se muestra en un panel de tipo LCD.

El código de esta rutina, que escrita en lenguaje C, se convierte en código hexadecimal utilizando un compilador cruzado. El archivo HEX generado se descarga en la memoria flash del microcontrolador. La red neuronal ha sido programado en 12.160 bytes de memoria de programa (ROM 37%) y 624 bytes de memoria de datos (30% de RAM).

Todo el proceso esquematizado se muestra en el diagrama de flujo visto en el capítulo 2.

3.2.3. Resultados y discusión con la red FF-MLP.

Una vez realizado el entrenamiento y verificación de la red neuronal y determinado los valores de los parámetros necesarios para implementar el algoritmo de la red neuronal en el microcontrolador se pasa a la fase de prueba. En esta fase de prueba se utilizan los datos de nuevas muestras obtenidas directamente con el microcontrolador. Para esta fase se han utilizado los datos de las muestras 9^a a la 12^a muestra y 5 muestras más, de la 13^a a la 17^a muestra, obtenidas a partir de las botellas 3^a y 4^a.

Para obtener la matriz de confusión, además de los parámetros de la red neuronal perceptrón, es necesario suministrarle al programa la tabla de objetivos durante la fase de prueba (Tabla 5.6). El uno de la columna indica la clase a la que pertenece la muestra. Dicha tabla se compara con la obtenida para generar la matriz de confusión, que es la misma que la utilizada para la Fuzzy Artmap para los diferentes días de prueba.

Tabla 5.33. Tabla de salida para el día 9.							Clase
1	0	0	0	0	0	0	1
0	1	0	0	0	0	0	2
0	0	0	0	0	0	0	3
0	0	0	1	0	0	0	4
0	0	0	0	0	0	0	5
0	0	0	0	0	1	0	6
0	0	1	0	1	0	1	7

Para cada día de prueba se obtiene una matriz de salida en función de la clase obtenida. Los resultados para los datos del día 9 se muestran en la tabla 5.23, en donde se observan dos fallos de siete.

La tabla 5.24 muestra el resultado de clasificaciones correctas por día. El mejor resultado se obtuvo para el día 17, en el que todos los datos se clasificaron correctamente, lo que da un total de 51 datos clasificados correctamente de un total de 63 datos de entrada.

Tabla 5.24. Tabla de clasificaciones correctas por día.

Día	9	10	11	12	13	14	15	16	17
Correctos	5	5	6	5	6	6	6	5	7

Finalmente, las figuras, 5-42 y 5-43 muestran la curva característica del receptor y la matriz de confusión, respectivamente.

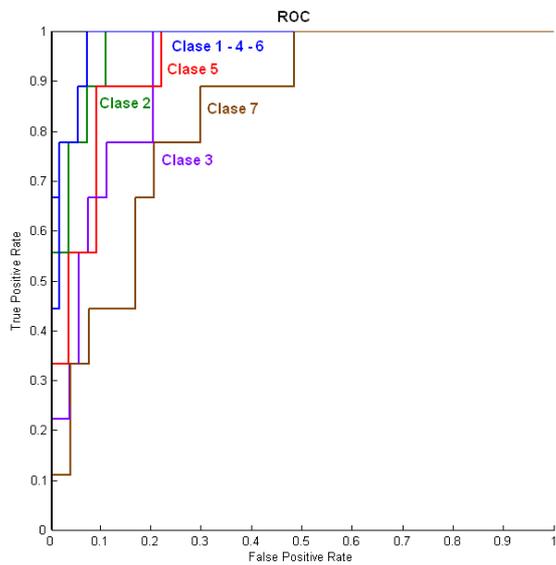


Figura 5-42. Curva de características operativas del receptor (ROC), en la fase de prueba.

Confusion Matrix

	1	2	3	4	5	6	7	
1	9 14.3%	1 1.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90.0%
2	0 0.0%	8 12.7%	0 0.0%	0 0.0%	1 1.6%	0 0.0%	0 0.0%	88.9%
3	0 0.0%	0 0.0%	6 9.5%	0 0.0%	0 0.0%	0 0.0%	1 1.6%	85.7%
4	0 0.0%	0 0.0%	1 1.6%	9 14.3%	0 0.0%	0 0.0%	0 0.0%	90.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	7 11.1%	0 0.0%	5 7.9%	58.3%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	9 14.3%	0 0.0%	100%
7	0 0.0%	0 0.0%	2 3.2%	0 0.0%	1 1.6%	0 0.0%	3 4.8%	50.0%
	100%	88.9%	66.7%	100%	77.8%	100%	33.3%	81.0%
	0.0%	11.1%	33.3%	0.0%	22.2%	0.0%	66.7%	19.0%
	1	2	3	4	5	6	7	

Figura 5-43. Matriz de confusión en la fase de prueba de los 9 días.

Hay que hacer constar que en este caso, como los datos ya se habían obtenido en la implementación y prueba de la SFAM en el microcontrolador, se ha hecho entrenamiento y prueba en MATLAB y posteriormente comprobar que los resultados en MATLAB coinciden con la simulación en el microcontrolador, en donde los datos de prueba no se han tomado del exterior sino que estaban previamente almacenados en la memoria del microcontrolador.

3.3. Comparación entre redes.

La comparación de resultados entre la red SFAM y la FF-MLP se muestra en la tabla 5.25. En dicha tabla se observa, que en la FF-MLP solamente se consigue una tasa de aciertos del 100% en un único día, frente a tres de la SFAM, y que en los días 14^a y 15^a la tasa de aciertos es ligeramente superior en la FF-MLP. Finalmente, la tasa total de la red SFAM es del 87.3% frente al 80.9% de la red FF-MLP.

Tabla 5.25. Comparación de resultados.																			
Número muestra	Número botella	Fuzzy Artmap Simplificada							%	FF-MLP									
		Clase								Clase									
		1	2	3	4	5	6	7		1	2	3	4	5	6	7			
9 th	2 nd	1	2	3	4	7	6	7	85.7	1	2	7	4	7	6	7	71.4		
10 th	2 nd	1	2	3	4	5	6	7	100.0	1	2	7	4	2	6	7	71.4		
11 th	3 rd	1	2	3	4	5	6	7	100.0	1	2	3	4	5	6	5	85.7		
12 th	3 rd	1	2	3	4	5	6	7	100.0	1	1	3	4	5	6	5	71.4		
13 th	3 rd	1	2	3	4	7	6	7	85.7	1	2	3	4	5	6	5	85.7		
14 th	3 rd	1	2	7	4	5	6	5	71.4	1	2	3	4	5	6	5	85.7		
15 th	4 th	1	2	3	4	7	6	3	71.4	1	2	3	4	5	6	5	85.7		
16 th	4 th	1	2	7	4	5	6	7	85.7	1	2	4	4	5	6	5	71.4		
17 th	4 th	1	2	7	4	5	6	7	85.7	1	2	3	4	5	6	7	100		
Total									87.3	Total									80.9

Para aumentar el número de muestras del entrenamiento de los sistemas de reconocimiento de patrones, hay que tener presente que trabajar sólo sobre PC no presenta problemas pero en sistemas que van a trabajar sobre microcontrolador se plantea un problema de ocupación de memoria. El tamaño de memoria RAM y ROM no supera en las dos redes creadas el tamaño máximo de memoria del microcontrolador utilizado. La ocupación de memoria de programa es muy similar en las dos redes, no así la ocupación de memoria RAM, la cual es mucho mayor en la red SFAM que en la FF-MLP (Tabla 5.26).

Tabla 5.26. Memoria utilizada		
	Ocupación de memoria	
RNA	ROM	RAM
Fuzzy Artmap	39%	80%
FF-MLP	37%	30%

Si se aumenta el número de muestras de entrenamiento, las dos redes se comportan de un modo diferente en lo referente a la ocupación de memoria.

En la red neuronal FF-MLP el tamaño de la memoria sólo depende del número de neuronas de entrada, internas y de salida; éste número no varía aunque se aumente el número de muestras para el entrenamiento. Por consiguiente el tamaño de memoria será constante aunque el número de muestras del entrenamiento aumente. En la **Fig. 5-44** se observa como el tamaño de memoria es constante para la FF-MLP utilizada de 20 nodos internos.

La red SFAM tiene un comportamiento distinto, cuantas más muestras se utilicen para el entrenamiento mayor será el tamaño de la matriz de pesos y el mapfield. Este incremento se observa en la tabla 5.27. Al aumentar el tamaño de las matrices aumenta el tamaño de la memoria ocupada. En la **Fig. 5-44** se muestra esta variación para matrices de distinto tamaño. Se puede ver que a partir de matrices de pesos de 12x20 el tamaño de la memoria RAM supera los 2K, lo que implicaría en este caso cambiar de microcontrolador.

Como conclusión desde el punto de vista de programación en un microcontrolador, es más eficiente la utilización de la FF-MLP que la red SFAM.

Tabla 5.27. Matriz de pesos en función del número de muestras	
Número de muestras en el entrenamiento	Medida de la matriz de pesos
8	12x16
10	12x19
12	12x31
14	12x36

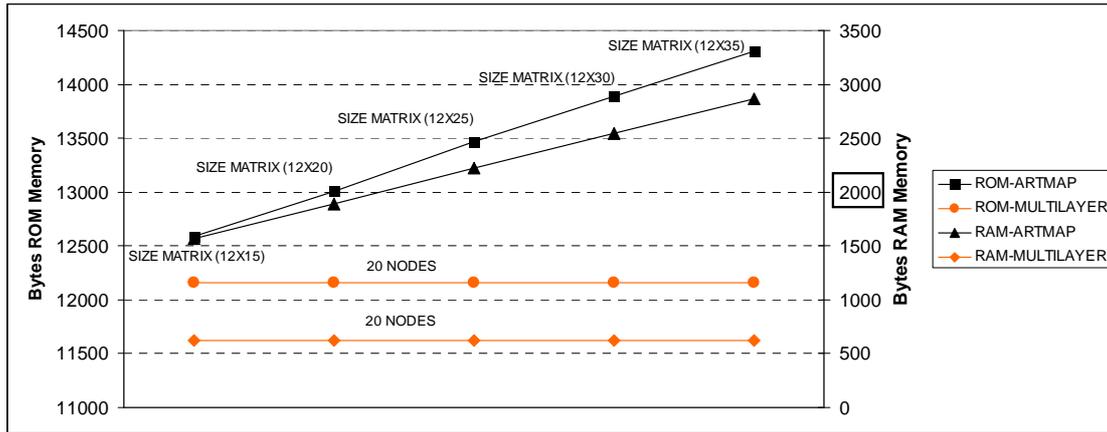


Figura 5-44. Variación del tamaño de memoria ocupada

4. Selección de variables.

Como último estudio de este capítulo se va a realizar un análisis de las variables de entrada para determinar si conviene realizar una selección de variables que corresponda a una disminución del número de electrodos utilizados en las medidas.

El objetivo va a consistir en el estudio, implementación y validación en la base de datos de experiencias reales para la seleccionar el menor número de variables en problemas de clasificación supervisada, pero manteniendo las condiciones óptimas:

- Tasa de aciertos elevada
- Reducido mapfield

Una vez implementadas las diferentes técnicas de selección, se va a realizar un estudio comparativo de todas ellas.

Como se ha comentado en el capítulo 2, la GUI Fuzzy Artmap se basa en cinco algoritmos desarrollados por Lapiedra [Lapiedra, 2004] en MATLAB®.

Los métodos Forward Selección y Backward Eliminación son rápidos y es el usuario quien elige el número de variables mediante menú desplegable.

En los métodos Forward PNN y Backward PNN los datos son validados mediante red neuronal probabilística pero una vez seleccionada (Forward) o eliminada (Backward) una variable esta no es nuevamente utilizada.

Finalmente el método Stepwise PNN es una combinación de los dos anteriores.

A cada electrodo se le ha asignado un número de acuerdo con la tabla 5.28.

Electrodo	Cu1	Ru2	Ru1	Ag	Pt	AgCl
Variable nº	1	2	3	4	5	6

Los resultados obtenidos por los diferentes métodos y utilizando las 12 primeras muestras (8 para el entrenamiento y 4 para la validación) se muestran en la tabla 5.29. En esta tabla se indica cuales son las variables elegidas para diferentes métodos de análisis multivariante. También se indica el tiempo que ha tardado la red en completar el análisis (minutos y segundos) y la tasa de aciertos. En dicha tabla se observa que la variable 5 (Pt) se selecciona en todos los métodos y la variable 6 (AgCl) en cuatro de los cinco métodos.

Método	Variables elegidas	mm:ss	Tasa
<i>Forward</i> (3 variables)	5 2 4		-----
<i>Backward</i> (3 variables)	1 5 6		-----
<i>Forward PNN</i>	6 5 4	02:21	86.9%
<i>Backward PNN</i>	6 5 4	02:14	86.9%
<i>Stepwise PNN</i>	5 6		87.5%

ρ \ β	Tasa (%)							Mapfield						
	0.4	0.5	0.6	0.7	0.8	0.9	1	0.4	0.5	0.6	0.7	0.8	0.9	1
0.3	60.7	46.4	53.6	53.6	46.4	50.0	67.9	100	100	100	100	24	23	30
0.4	60.7	46.4	53.6	53.6	46.4	50.0	67.9	100	100	100	100	24	23	30
0.5	60.7	46.4	53.6	53.6	46.4	50.0	67.9	100	100	100	100	24	23	30
0.6	60.7	46.4	53.6	53.6	46.4	50.0	67.9	100	100	100	100	24	23	30
0.7	64.3	53.6	53.6	57.1	46.4	57.1	64.3	100	100	100	100	24	26	30
0.8	57.1	53.6	50.0	46.4	50.0	60.7	53.6	100	100	100	100	100	28	30
0.9	39.3	50.0	46.4	50.0	53.6	60.7	60.7	100	100	48	100	43	42	38

El método Forward selecciona las variables 2, 4 y 5 correspondientes a los electrodos de Ru2, Ag y Pt. Realizando nuevamente un entrenamiento con 12 muestras (84 datos) con un 66.67% de los datos para el entrenamiento se

obtiene una tasa máxima de aciertos del 60.7% (tabla 5.30) pero con un mapfield de 1x100, mayor que el número de datos.

El método Backward selecciona las variables 1, 5 y 6 correspondientes a los electrodos de Cu1, Pt y AgCl. Realizando nuevamente un entrenamiento con 12 muestras (84 datos) que corresponde con un 66.67% de los datos para el entrenamiento. se obtiene una tasa máxima de aciertos del 92.9% (tabla 5.31) pero con un mapfield de 6x100, mayor que el número de datos. por lo que, evidentemente el mapfield no es reducido. En cambio, el mínimo mapfield de 1x19, proporciona una tasa de aciertos del 85.7%.

Tabla 5.31 Tasa de aciertos y mapfield utilizando los electrodos de Cu1, Pt y AgCl.														
ρ / β	Tasa (%)							Mapfield						
	0.4	0.5	0.6	0.7	0.8	0.9	1.0	0.4	0.5	0.6	0.7	0.8	0.9	1
0.3	82.1	92.9	89.3	89.3	82.1	78.6	85.7	100	100	100	100	20	21	19
0.4	82.1	92.9	89.3	89.3	82.1	78.6	85.7	100	100	100	100	20	21	19
0.5	82.1	92.9	89.3	89.3	82.1	78.6	85.7	100	100	100	100	20	21	19
0.6	82.1	92.9	89.3	89.3	82.1	78.6	85.7	100	100	100	100	20	21	19
0.7	78.6	92.9	82.1	89.3	89.3	82.1	85.7	100	100	100	100	21	22	19
0.8	78.6	82.1	92.9	85.7	82.1	75.0	85.7	100	100	100	23	24	22	20
0.9	53.6	85.7	78.6	75.0	82.1	82.1	82.1	100	100	100	40	36	35	31

Los métodos Forward_PNN y Backward_PNN seleccionan las variables 4, 5 y 6 correspondientes a los electrodos de Ag, Pt y AgCl. Realizando nuevamente un entrenamiento con 12 muestras (84 datos) con un 66.67% de los datos para el entrenamiento se obtiene una tasa máxima de aciertos del 96.4% (tabla 5.32) pero con un mapfield de 1x100, mayor que el número de datos, por lo que, evidentemente el mapfield no es reducido. En cambio, el mínimo mapfield de 1X15, proporciona una tasa de aciertos del 92.9%, que comparado con el inicial con 6 electrodos (96.4% y 1x13), puede ser una opción válida.

Tabla 5.32 Tasa de aciertos y mapfield utilizando los electrodos de Ag. Pt y AgCl

ρ / β	Tasa (%)							Mapfield						
	0.4	0.5	0.6	0.7	0.8	0.9	1.0	0.4	0.5	0.6	0.7	0.8	0.9	1
0.3	92.9	92.9	92.9	92.9	92.9	85.7	85.7	100	16	100	15	15	19	22
0.4	92.9	92.9	92.9	92.9	92.9	85.7	85.7	100	16	100	15	15	19	22
0.5	92.9	92.9	92.9	92.9	92.9	85.7	85.7	100	16	100	15	15	19	22
0.6	92.9	92.9	92.9	92.9	92.9	85.7	85.7	100	16	100	15	15	19	22
0.7	96.4	92.9	92.9	92.9	92.9	85.7	85.7	100	100	100	15	15	20	22
0.8	89.3	75.0	89.3	85.7	89.3	82.1	82.1	100	22	32	25	25	22	24
0.9	71.4	53.6	53.6	57.1	82.1	57.1	89.3	100	100	100	100	42	34	35

El método `stepwise_PNN` selecciona las variables 5 y 6 correspondientes a los electrodos de Pt y AgCl. Realizando nuevamente un entrenamiento con 12 muestras (84 datos) con un 66.67% de los datos para el entrenamiento se obtiene una tasa máxima de aciertos del 100% (Tabla 5.33) pero con un mapfield de 1x100, mayor que el número de datos, por lo que, evidentemente el mapfield no es reducido. En cambio, el mínimo mapfield de 1x17 proporciona una tasa de aciertos del 85.7%, que comparado con el inicial con 6 electrodos (96.4% y 1x13) se observa una disminución excesiva de la tasa de aciertos.

Tabla 5.33 Tasa de aciertos y mapfield utilizando los electrodos de Pt y AgCl.

ρ / β	Tasa (%)							Mapfield						
	0.4	0.5	0.6	0.7	0.8	0.9	1	0.4	0.5	0.6	0.7	0.8	0.9	1
0.3	85.7	96.4	100	85.7	92.9	85.7	85.7	100	100	100	100	100	17	18
0.4	85.7	96.4	100	85.7	92.9	85.7	85.7	100	100	100	100	100	17	18
0.5	85.7	96.4	100	85.7	92.9	85.7	85.7	100	100	100	100	100	17	18
0.6	85.7	96.4	100	85.7	92.9	85.7	85.7	100	100	100	100	100	17	18
0.7	85.7	96.4	100	85.7	92.9	85.7	85.7	100	100	100	100	100	17	18
0.8	82.1	89.3	89.3	85.7	92.9	85.7	85.7	100	100	100	100	100	19	18
0.9	78.6	75.0	67.9	75.0	82.1	89.3	78.6	100	100	100	100	100	27	25

La tabla 5.34 se muestra un resumen de la variación de la tasa, mapfield y matriz de pesos en función del número total de electrodos. En dicha tabla se observa que la matriz de pesos, se puede reducir prácticamente a la mitad, pero a costa de disminuir ligeramente (3.5%) la tasa de aciertos (de 96.4% a 92.8%).

Nº de electrodos	Tasa	Mapfield	Matriz de pesos
6	96.4%	1x13	12x13
3 (Cu1. Pt y AgCl)	85.7%	1x19	6x19
3 (Ag. Pt y AgCl)	92.9%	1x15	6x15
2 (Pt y AgCl)	85.7%	1x17	4x17

5. Conclusiones.

A lo largo de este capítulo se ha descrito los análisis para la discriminación de aguas con dos tipos de redes neuronales. Con la red de tipo Fuzzy Artmap Simplificada se logra los mejores resultados (96.4%) con un tamaño de la matriz de pesos de 12x13.

Los análisis con la red FF-MLP muestran que el número óptimo de neuronas de la capa oculta es de 20 logrando una tasa de éxito de 80.9%. También se ha realizado un análisis de selección de las variables de entrada llegando a la conclusión que eligiendo solo tres electrodos (Ag, Pt y AgCl) se alcanza una tasa de aciertos del 92.9% con una matriz de pesos de 6x15.

6. Producción científica.

Artículos

Eduardo Garcia-Breijo. John Atkinson. Luis Gil-Sanchez. Rafael Masot. Javier Ibañez. Jose Garrigues. Monika Glanc. Nicolas Laguarda-Miro. Cristian Olguin. *A comparison study of pattern recognition algorithms implemented on a microcontroller for use in an electronic tongue for monitoring drinking waters.* Sensors and Actuators A: Physical nº 172 (2011) 570-582

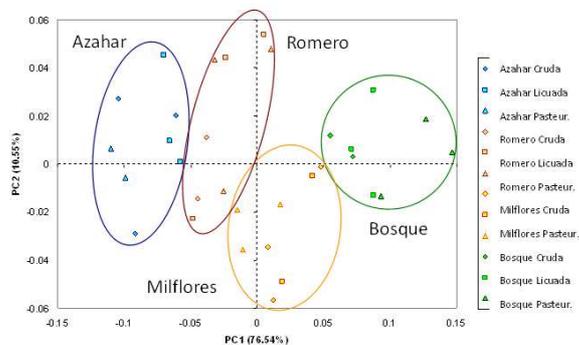
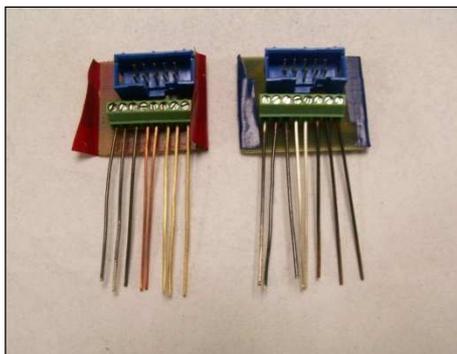
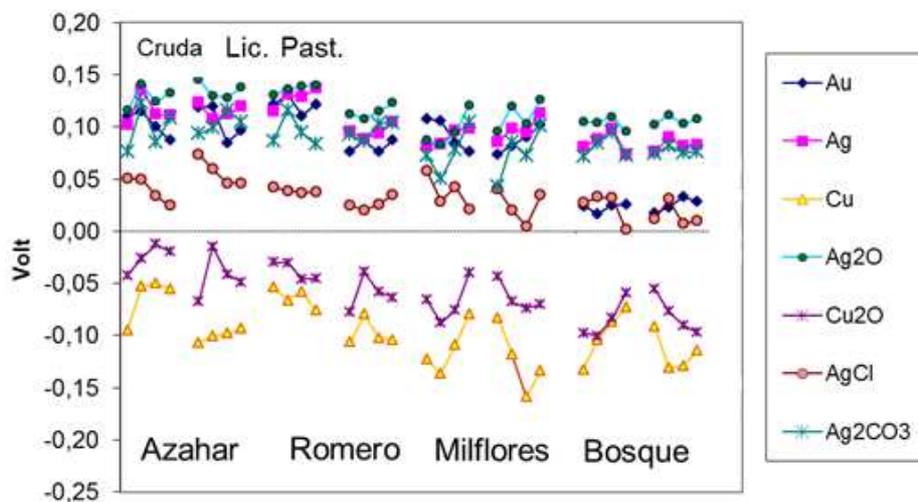
Congresos

Garcia-Breijo. E.; Atkinson. J.; Garrigues. J.; Gil. L.; Ibanez.J.; Glanc. M.; Olguin. C. "An electronic tongue for monitoring drinking waters using a fuzzy ARTMAP neural network implemented on a microcontroller." *2011 IEEE International Symposium on Industrial Electronics (ISIE).* (2011) pp.1270-1275.

J. Garrigues, E. Garcia-Breijo, L. Gil-Sánchez, J. Ibáñez. *Implementación de redes neuronales artificiales en microcontroladores*. V Workshop on Sensors and Molecular Recognition, Valencia – España. Julio 2011

Capítulo 6

Aplicación de las Redes Neuronales en Microcontroladores de 8 Bits para la Clasificación de Mieles



Aplicación de las Redes Neuronales con μ C de 8 Bits para la Determinación de Glifosato
Mediante la Técnica de Voltamperometría de Pulsos

En este capítulo se describen los análisis realizados con redes neuronales artificiales para la discriminación de los datos procedentes de medidas con diversas variedades de miel. La discriminación consiste en determinar cuatro tipos de origen floral de la miel (azahar, romero, milflores y bosque) y además se ha teniendo en cuenta tres tratamientos físicos: crudo (sin tratamiento), licuación y pasteurización. Las medidas fueron realizadas con un sistema de lengua electrónica potenciométrica formada por un conjunto de electrodos metálicos de diferentes materiales.

1. Introducción.

Las empresas dedicadas a la producción y comercialización de la miel deben de indicar su origen floral, porque según dicho origen poseen unas determinadas características físicas y organolépticas que son de interés para el consumidor. Algunas mieles se denominan monoflorales porque se considera que proceden del polen de un solo tipo de flor. De esta forma existe la miel de romero, bosque o azahar porque las colmenas se encuentran cercanas a determinado tipo de vegetación. En cambio, otras mieles proceden de varios tipos de polen, o simplemente no está determinado su origen floral, en estos casos se suelen denominar milflores. En general, las mieles monoflorales tienen mayor valor comercial para los productores porque se suele garantizar unas características determinadas (color, consistencia, olor, sabor, etc.). Por este motivo los empresarios del sector deben de ser exigentes a la hora del control de calidad y deben de poner medios para el estudio de los sistemas de identificación del origen botánico de las mieles.

Por otro lado, la miel se suele tratar con procesos físicos, de esta forma podemos obtener tres tipos de mieles:

Cruda: miel obtenida directamente de la colmena, sin que sufra ningún tratamiento térmico. El único tratamiento que sufren es un lavado como operación previa de desinfección a la entrada de la planta industrial.

Licuada: miel que ha sufrido una operación de limpieza para hacerla conservable y de aspecto estable, para así poder transvasarla, fraccionarla, mezclarla y envasarla. El licuado se realiza mediante calentamiento, introduciéndola en una cámara a una temperatura entre 45°C y 55°C, por un tiempo aproximado de dos días. El licuado comprende dos fases: la fluidificación, donde la miel sólida y cristalina se convierte en un fluido espeso, y la fusión propiamente dicha, donde la miel es recogida en el exterior en un recipiente provisto de un agitador lento y sometido a un moderado calentamiento.

Pasteurizada: La pasterización es un proceso que se realiza después de la licuación, por el que la miel se somete a un choque térmico elevado (75 °C-85°C) y de reducida duración (2-6 minutos), que destruye la mayor parte de las estructuras cristalinas iniciales que favorecen la total o parcial cristalización de la miel, permitiendo que ésta permanezca líquida durante más tiempo. El principal objetivo de la pasterización, es retrasar durante el almacenamiento los fenómenos de la cristalización y de la fermentación, de manera que la presencia del producto de cara al consumidor sea la mejor posible a lo largo del tiempo. También se realiza para reducir la carga microbiana

Pero es interesante señalar que estos procesos no deberían afectar a la calidad y las propiedades intrínsecas de la miel.



Figura 6-1. Mieles de diferentes orígenes florales.

El método clásico para determinar el origen botánico de la miel es el denominado análisis de melisopalinología, que se basa en la identificación y la cuantificación del porcentaje de cada tipo de polen que constituye la miel por medio de un examen por microscopio. Sin embargo, esta técnica es muy tediosa y requiere de analistas altamente calificados y equipos especializados, por lo que el análisis del 100% de las muestras no es posible llevarlo a la práctica. Además, para algunos tipos de miel la interpretación de este análisis puede ser especialmente difícil y, a veces puede ser impreciso y ambiguo, como es el caso de la miel de cítricos. Esto es debido a la secreción máxima de néctar a veces no coincide con la madurez de la antera, y por lo tanto la cantidad de polen presente en la miel de cítricos es en ocasiones menor que la cantidad esperada considerando las características organolépticas [Serra & Ventura, 2003]. También se han sugerido como información complementaria para la caracterización de la miel determinados parámetros físico-químicos (color, actividad de la diastasa, la conductividad eléctrica, la actividad antioxidante total, etc.) [Beretta et al., 2005]. Sin embargo, su valor en la clasificación de miel es cuestionable, ya que se ha demostrado que algunos de estos parámetros cambian considerablemente, incluso para el mismo tipo de miel [Soria et al., 2005]. Otro inconveniente de las técnicas tradicionales de reconocimiento del origen floral de la miel es que se precisa personal especializado y trabajando en instalaciones lejanas del lugar de producción.

Sin embargo, la industria del envasado y comercialización de la miel necesita de técnicas rápidas, sencillas y fiables que permitan la determinación del origen y la autenticidad de la miel, sin necesidad de equipo costoso y trabajadores altamente cualificados.

En muchas ocasiones, más que un exhaustivo análisis cuantitativo, es conveniente realizar un análisis cualitativo para poder discriminar las muestras en grupos. En estos casos, es cuando resulta interesante la aplicación de sistemas de nariz o lengua electrónica.

Se han aplicado diferentes sistemas de lenguas electrónicas al control de calidad de la miel utilizando diversos métodos de medidas: sensores de membrana [Dias et al., 2008], sistema comercial de lengua electrónica con electrodos sensibles a los distintos sabores [Wei et al., 2009], voltametría a

diferentes frecuencias [Wei & Wang, 2011]. La novedad de la experiencia de donde se han obtenido los datos para realizar los análisis de este capítulo, consiste en el uso de electrodos de metal, que permiten una forma sencilla, barata y fácil de usar [Escriche et al., 2012].

2. Descripción del sistema de medida.

2.1. Las muestras.

Las experiencias se llevaron a cabo por medio de una colaboración entre el Instituto de Reconocimiento Molecular y Desarrollo Tecnológico (IDM) y el Instituto de Ingeniería de Alimentos para el Desarrollo (IUIAD), ambos de la Universidad Politécnica de Valencia (UPV).

Cuatro tipos de mieles españolas han estado en estudio: tres de distinto origen floral (azahar, romero y milflores) y una de origen de bosque. El origen botánico de las muestras fue determinado por análisis melisopalinológico. La elección de estas cuatro variedades de miel fue determinada porque representan los tipos más populares de consumo en España.

2.2. Sistema Electrónico.

Los bloques que constituyen el sistema general de medida están representados en la Fig. 6.2 y son los siguientes:

- Sensores.
- Sistema de Acondicionamiento de la Señal.
- Sistema de Adquisición de Datos.
- Análisis Multivariante de los datos.

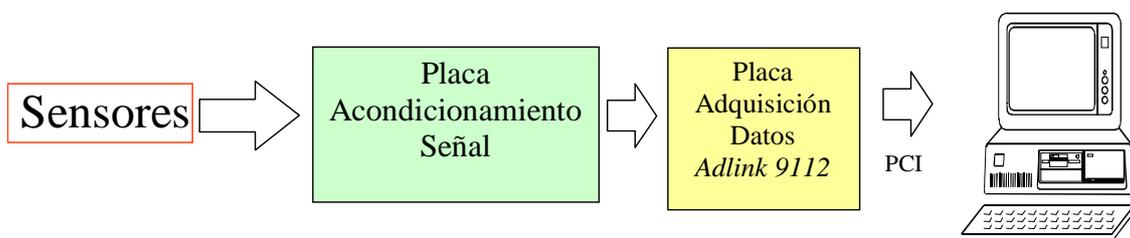


Figura 6-2. Diagrama de bloques del Sistema de Medida.

2.3. Sensores.

Para esta experiencia se utilizaron siete electrodos de diferentes metales, algunos puros como el oro, plata y cobre y otros tratados químicamente mediante electrólisis (AgO_2 , CuO_2 , AgCl y Ag_2CO_3) para determinar si tenían propiedades distintas que los metales puros. Algunos de los electrodos estaban repetidos. Los electrodos estaban formados por cables de 0,8 mm de diámetro y unos 5 cm de longitud, todos ellos unidos a un conector de cable plano (Fig. 6-3).

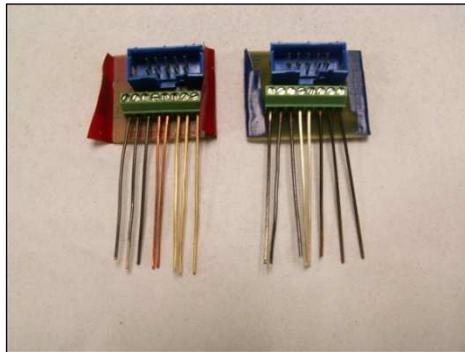


Figura 6-3. Electrodos metálicos de diferentes materiales.

2.4. Acondicionamiento de la Señal.

La señal eléctrica que genera un sensor de tipo potenciométrico, aun teniendo dimensiones de tensión eléctrica, posee características que dificultan su medida de forma directa. Esta dificultad estriba en la debilidad de la señal generada, con baja amplitud. Además, muchos electrodos poseen gran impedancia de salida, lo que conlleva que la señal generada sea muy sensible al ruido eléctrico. Por este motivo, se incorpora en la cadena de medida una etapa que realiza un acondicionamiento de la señal eléctrica procedente del electrodo para adecuarla al posterior sistema de adquisición de datos.

Para lograr el acondicionamiento de señal potenciométrica se diseñó un sistema compuesto fundamentalmente por dos partes: una primera etapa de muy alta impedancia de entrada y baja impedancia de salida y una segunda etapa de filtrado para eliminar el ruido procedente de la red eléctrica (50 Hz), así como otras señales de ruido de alta frecuencia. Para obtener un sistema con las características reseñadas, la primera tarea es la búsqueda de sistemas comerciales que realicen las tareas propuestas, pero en el mercado de

dispositivos electrónicos no se encuentran elementos con las características adecuadas. Además hay que tener en cuenta que se va a trabajar con un número considerable de electrodos. Por este motivo se decide la construcción de un sistema electrónico con las cualidades óptimas para el tipo de medida a realizar.

Para construir el sistema electrónico de acondicionamiento de la señal, una decisión inicial es determinar el número de canales de medida. Este número está condicionado por el sistema de adquisición de datos. La placa utilizada es de 16 canales, luego este era el número canales que en principio debe tener el sistema de acondicionamiento de la señal. Además hay que tener en cuenta el rango de tensión del potencial de los electrodos, que suele estar en el rango de ± 1 V. En la **Figura** se muestra el interior del sistema electrónico de acondicionamiento de la señal, con las dos placas de circuito impreso, que se acaban de describir, y la fuente de alimentación.

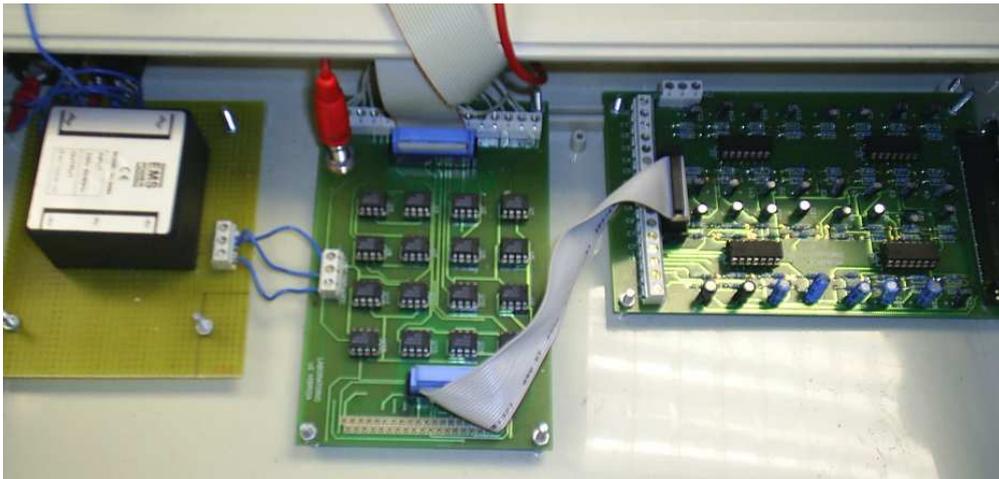


Figura 6-4. Prototipo de Sistema de Acondicionamiento de la Señal.

La adquisición de datos consiste en la captura de las señales eléctricas procedentes de los diferentes sensores y su almacenamiento en un ordenador. El elemento primordial para la adquisición de datos es una tarjeta multicanal de adquisición de datos. Este sistema permite de forma automática la visualización de la información en la pantalla del ordenador en tiempo real, así como su almacenamiento para un posterior tratamiento matemático.

La tarjeta de adquisición de datos se inserta en un slot de bus PCI de un ordenador personal PC. La tarjeta que se utilizó en los experimentos es de la marca ADLINK® modelo PCI-9112.

Para visualizar y registrar los datos se aplica el lenguaje de programación visual VEE Pro®. Este lenguaje contiene un entorno gráfico que permite la representación en pantalla de las trazas de cada electrodo con tiempo de muestreo programado por el usuario. El programa permite guardar el valor de cada medida de forma automática en un fichero ASCII.



Figura 6-5. Ejemplo de la representación en pantalla del ordenador de las señales potenciométricas de algunos electrodos.

La Fig. 6-5 muestra las trazas de las señales de los diferentes electrodos. Para realizar las medidas se sumergían los electrodos junto con un electrodo de referencia de Ag/AgCl en un recipiente donde se encontraban las muestras de miel y se iba tomando medidas de forma continua durante aproximadamente 5 minutos o hasta que la señal quedaba estabilizada porque se llegaba al equilibrio electroquímico entre el electrodo y la disolución. Una vez alcanzado dicho equilibrio se tomaba la media de las últimas 10 muestras,

de esta forma se aminoraban los posibles ruidos. Es decir, que en cada medida se obtenía un solo dato por cada uno de los siete electrodos.

2.5. Medidas.

Para determinar la repetitividad del sistema de medida, se realizaron diversas mediciones en días diferentes, El número total de mediciones realizadas fueron: 4 orígenes florales x 3 tratamientos físicos x 4 repeticiones = 48 medidas. De los cuales, con los primeros 32 datos se ha realizado el entrenamiento y prueba en MATLAB. Posteriormente, en el periodo de prueba se obtuvieron 16 datos más en el microcontrolador.

La representación de los potenciales de cada una de las medidas y para cada uno de los electrodos queda reflejada en la Fig. 6-6. En ella se puede observar que cada electrodo posee un nivel característico de potencial y que varía según el tipo de la muestra de miel. Esta variación no es determinante porque hay casos en que el potencial de un electrodo varía con respecto a un tipo de muestra pero no lo hace respecto a otra, y otros casos al contrario. También es de destacar que las medidas potenciométricas con muestras de naturaleza química tan compleja como es nuestro caso, suelen adolecer de falta de repetitividad. Por todo ello debemos de echar mano de las técnicas de análisis multivariante.

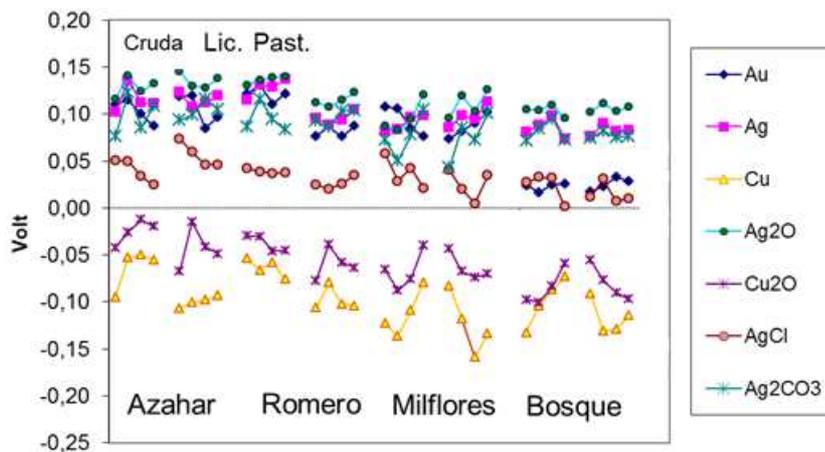


Figura 6-6. Potencial eléctrico de cada uno de los electrodos para cada una de las muestras.

El tipo de análisis multivariante más utilizado es el análisis de componentes principales (PCA), que consiste, básicamente, en una reducción

de la dimensión de una matriz de datos de forma que la información que aporta un conjunto de sensores (en nuestro caso los 7 electrodos) puedan reducirse a dos y así pueda representarse en una gráfica bidimensional. En la tarea de reducción de dimensiones se pierde información, pero justamente el análisis PCA hace que esta pérdida sea la mínima, para ello se crean nuevas variables ficticias, llamadas componentes principales (PC).

En la **Fig. 6-7** se representa los valores de las dos primeras componentes principales (PC1 y PC2) de los datos que fueron representados en la gráfica anterior. En esta nueva gráfica se puede observar que surgen agrupaciones espontáneas de los datos según el origen floral de la miel, aunque también es verdad que la separación entre los grupos no es demasiado grande, e incluso hay algún solapamiento. En cambio no es fácil determinar si existe discriminación de las muestras según el tratamiento térmico (en la figura corresponde a diferente forma de los puntos de las medidas).

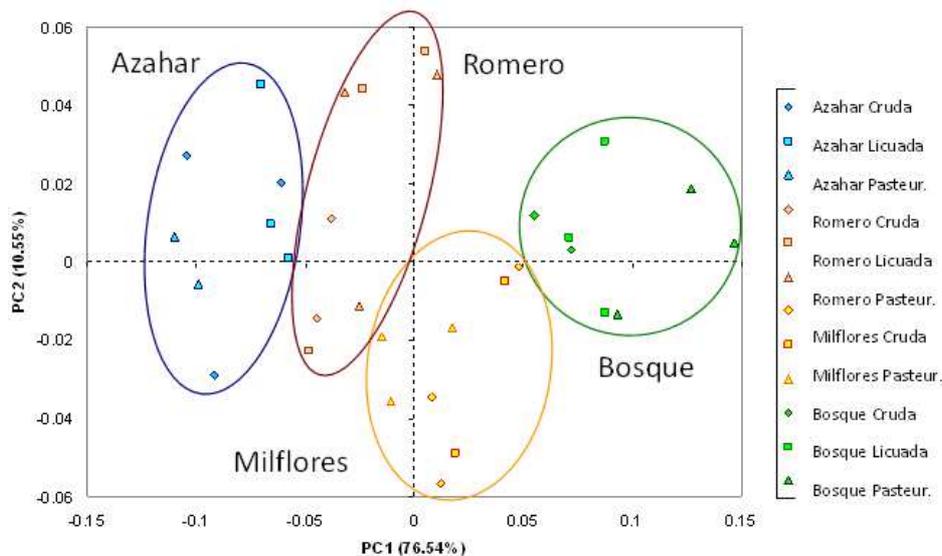


Figura 6-7. Gráfica bidimensional del análisis de componentes principales (PCA).

Los resultados del análisis PCA se pueden considerar como análisis cualitativos porque casi siempre se representan mediante gráficas, donde la cercanía o lejanía de unos puntos con respecto a otros determina lo parecido o diferente que son las medidas obtenidas con el conjunto de electrodos. Además hay que tener en cuenta que la información que representa la gráfica no es la total de los electrodos, por ejemplo, la información que representa la

figura anterior es $76.54\% + 10.55\% = 87.09\%$ del total de la varianza en los datos originales, por lo que se puede decir que las conclusiones son solo aproximadas.

La eficacia de la clasificación se contrastó comparando con otras medidas físico-químicas que se utilizan para el control de la calidad de la miel. Este contraste se realizó con técnicas de regresión como el *Partial Least Squares* (PLS) donde se obtuvo una buena correlación entre ambos tipos de variables.

Para obtener conclusiones completas de los resultados y sobretodo lograr que estas conclusiones sean cuantitativas se va a utilizar redes neuronales artificiales, donde es necesario fijar previamente las categorías de salida y realizar un entrenamiento de la red para fijar unos pesos en las ecuaciones de los algoritmos que las materializan. Al igual que en otros capítulos de esta memoria, el tipo de red a utilizar es el tipo Fuzzy Artmap Simplificada.

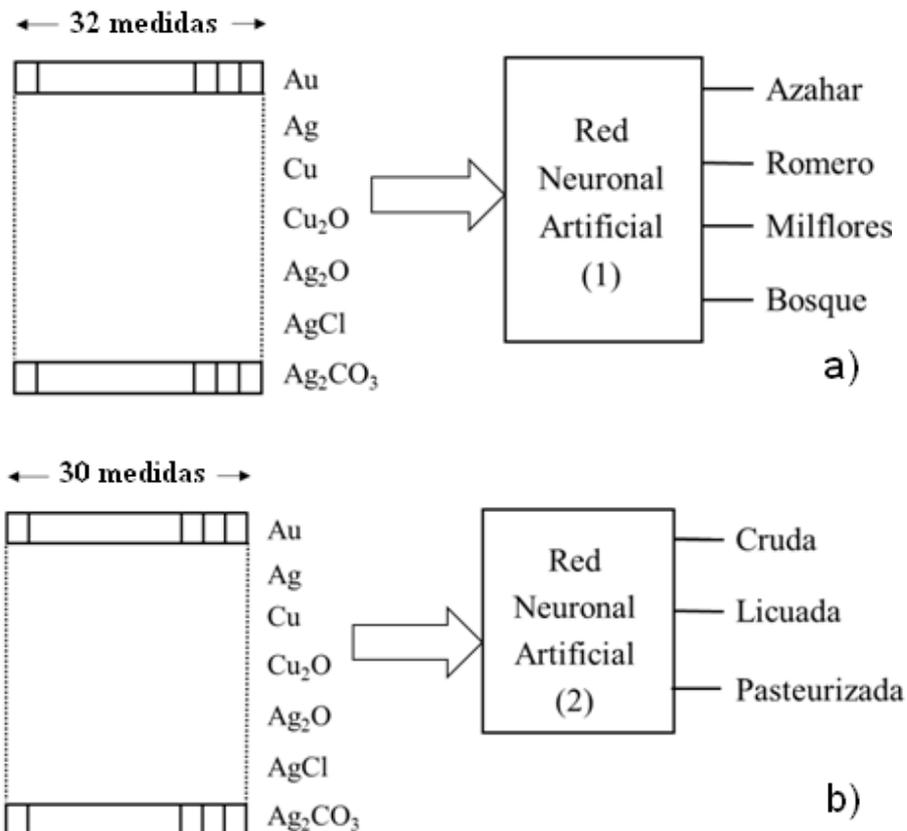


Figura 6-8. Clasificación de los datos mediante RNA en función de dos parámetros.

En nuestro caso se utilizaron dos redes neuronales (Fig. 6-8). La primera red busca la determinación del origen floral, por lo que la red cuenta con cuatro

salidas (**Fig. 6-8-a**), una para cada grupo de miel. La segunda red neuronal consiste en tres salidas, una para cada tratamiento físico (**Fig. 6-8-b**). En estas figuras cada vector de entrada corresponde con una columna aunque al realizar el análisis matricial se tome la matriz transpuesta.

Los datos de partida son los mismos que se han utilizado para el análisis PCA, es decir una matriz de 7 columnas (electrodos) y 32 filas (mediciones). (En la **Fig. 6-8** las matrices de entrada están transpuestas con respecto a las matrices de entrada de datos en la GUI).

3. Análisis de datos.

Para realizar el análisis de los datos con redes neuronales artificiales se dividieron los datos en dos grupos uno para realizar el **entrenamiento** y **validación** de la red formado por 32 datos y otro grupo diferente de datos para realizar la **prueba** de la red formado por 16 datos.

Para seleccionar los datos que van a formar el grupo de entrenamiento-validación de la red se procuró que fueran lo más representativo posible de los diferentes tipos de muestras. Por ello se eligió 8 muestras de cada una de los cuatro orígenes florales. Al mismo tiempo, también se procuró que fuera equilibrado en cuanto al tratamiento térmico recibido por la miel. De esta forma, se tomaron 11 muestras de miel cruda, 11 licuada y 10 pasteurizada.

3.1. Entrenamiento de la red SFAM para clasificación según origen floral de la miel.

El primer análisis con redes neuronales consistía en determinar el origen floral de la miel, por ello se asignó a cada medida la clase 1, 2, 3 y 4 para azahar, romero, milflores y bosque respectivamente.

Para el entrenamiento de la red se ha utilizado la técnica de validación cruzada, pero en lugar de ser de orden uno, en este caso, se realiza de orden 4. Es decir se ha dividido el conjunto de las 32 medidas en grupos de cuatro medidas, en donde en cada grupo hay una muestra de cada una de una categoría (origen floral). De este modo, se obtiene un total de 8 grupos que le

llamamos muestras, de M1 a M8. Cada muestra contiene al menos un dato de cada uno de los tratamientos.

En la tabla 6.1 se muestran los diferentes tipos de mieles y clases asignadas a cada grupo: Azahar (A), Romero (R), Milflores (MF) y bosque (B) y de cada uno de los tratamientos: crudo (C), licuada (L) y pasteurizada (P).

Tabla 6.1. Tipos de mieles y clases asignadas a cada grupo.																
I Miel	AC	RL	MFP	BC	AL	RP	MFC	BL	AP	RC	MFL	BP	AC	RL	MFP	BC
Clase	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Grupo	M1				M2				M3				M4			
Miel	AL	RP	MFC	BL	AP	RC	MFL	BP	AC	RL	MFP	BC	AL	RP	MFC	BL
Clase	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Grupo	M5				M6				M7				M8			

En la tabla 6.2 se muestran los datos de entrenamiento y la salida esperada (Clase) para dichos datos.

El objetivo es obtener un mapfield lo más reducido posible pero que tenga una tasa de aciertos óptima ya que se va a implementar en un microcontrolador, para lo cual es necesario obtener los valores del mapfield, los valores máximos y mínimos y la tabla de pesos.

En primer lugar se ha realizado un entrenamiento con 7 de los 8 grupos dejando el último para validar la red. Para ello se han variado los parámetros ρ y β de 0.1 a 0.9 y de 0.1 a 1, respectivamente. Dicho entrenamiento, al ser realizado con una única muestra nos servirá de base para ver como varia el mapfield y la tasa de entrenamiento al aumentar el número de muestras para validación o reducir el número de variables.

A continuación se ha realizado una validación cruzada de orden 4, con objeto de obtener el conjunto de muestras óptimo. Finalmente se realizó una selección de variables, con el objetivo de mantener la tasa máxima y el mapfield y así reducir el número de electrodos, lo que disminuiría el tamaño de la memoria del microcontrolador.

Para el entrenamiento de la red es necesario introducir tanto los datos de entrenamiento como los de validación, así como la salida esperada en un único fichero en formato texto (.txt). El programa GUI realizado separa los datos de entrenamiento y de validación de la salida esperada y los presenta en

pantalla. También es necesario introducir el % de entrenamiento, que en este caso es del 87.5% (28 datos para entrenamiento y 4 para validación). El número de categorías se calculan y se presentan en pantalla. Finalmente se obtienen las matrices binarias de entrenamiento y validación (Tabla 6.1) necesarias para obtener el PlotROC y el Plotconfusion.

Tabla 6.2. Matrices binarias de objetivos de salida															
Entrenamiento										Validación					
1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0
0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1

A continuación fijamos mediante menú desplegable los valores inicial y final de parámetros ρ y β , así como el incremento de estos parámetros (Tabla 6.3).

Tabla 6.3. Valores de los parámetros ρ y β .			
Parámetros	Inicial	Final	Incremento
ρ	0.1	0.9	0.1
β	0.1	1	0.1

La tabla 6.4 muestra la tasa de aciertos y el mapfield obtenido cuando se valida con una muestra de las 8 totales. Para un mapfield de 1x6 la tasa de aciertos es de 100% (con ρ entre 0.1 y 0.5 y β entre 0.3 y 1).

Tabla 6.4. Tasa de aciertos y mapfield en función de ρ y β con un 87,5% de datos en el entrenamiento																					
		Tasa de aciertos (%)									Mapfield										
ρ/β		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0.1		100	75	100	100	100	100	100	100	100	100	100	100	7	6	6	6	6	6	6	6
0.2		100	75	100	100	100	100	100	100	100	100	100	100	7	6	6	6	6	6	6	6
0.3		100	75	100	100	100	100	100	100	100	100	100	100	7	6	6	6	6	6	6	6
0.4		100	75	100	100	100	100	100	100	100	100	100	100	7	6	6	6	6	6	6	6
0.5		100	75	100	100	100	100	100	100	100	100	100	100	7	6	6	6	6	6	6	6
0.6		75	75	75	50	75	50	75	75	100	100	100	100	9	7	8	9	9	7	7	7
0.7		25	25	25	50	50	25	25	75	25	25	100	100	15	14	14	15	14	14	9	8
0.8		75	25	50	75	50	50	50	75	50	50	61	100	28	21	24	21	19	20	15	12
0.9		25	25	25	25	50	25	50	50	25	25	100	100	31	26	27	27	26	24	23	21

3.1.1. Validación cruzada de la Red SFAM.

Para garantizar que los resultados de un análisis estadístico son independientes de la partición entre datos de entrenamiento y prueba, se debe realizar una validación cruzada.

La validación cruzada de orden 4, equivale a utilizar un 75% de datos para el entrenamiento y un 25% de datos para validación. El proceso se repite 4 veces, de esta forma obtenemos el conjunto de datos con mejor tasa de aciertos y menor mapfield. El K_Fold igual a 4 divide los 32 datos obtenidos en dos grupos, 24 para el entrenamiento y 8 para la validación.

Tanto el número de muestras como los datos para el entrenamiento, validación y el total se calculan y se muestran en pantalla. Es necesario generar las tablas de objetivos de salida tanto para el entrenamiento como para la validación (tabla 6.5) para obtener la tasa de aciertos y el mapfield en función de los parámetros ρ y β .

Tabla 6.5. Matrices binarias de objetivos de salida.	
Entrenamiento	Validación
1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0	1 0 0 0 1 0 0 0
0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0	0 1 0 0 0 1 0 0
0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0	0 0 1 0 0 0 1 0
0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1	0 0 0 1 0 0 0 1

El primer entrenamiento de la validación cruzada al igual que en el apartado anterior, se realiza con las muestras colocadas al final, por lo tanto en este caso se ha entrenado la red con las muestra M1 a M6 y se ha validado con las muestras M7 y M8.

Los resultados obtenidos se muestran en la tabla 6.6. En dicha tabla se observa que la tasa de aciertos del 100% se mantiene y el mapfield ha disminuido con respecto al entrenamiento con una muestra, lo que era de esperar ya que ha disminuido el número de datos para el entrenamiento.

Tabla 6.6. Tasa de aciertos y mapfield en función de ρ y β para la validación con las muestras M7 y M8.

		Tasa de aciertos (%)										Mapfield									
ρ/β		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0.1		100	100	100	100	100	100	100	100	87.5	87.5	15	8	7	7	5	5	5	5	6	6
0.2		100	100	100	100	100	100	100	100	87.5	87.5	15	8	7	7	5	5	5	5	6	6
0.3		100	100	100	100	100	100	100	100	87.5	87.5	15	8	7	7	5	5	5	5	6	6
0.4		100	100	100	100	100	100	100	100	87.5	87.5	15	8	7	7	5	5	5	5	6	6
0.5		100	100	100	100	100	100	100	100	87.5	87.5	15	8	7	7	5	5	5	5	6	6
0.6		62.5	62.5	50	50	62.5	62.5	50	75	62.5	50	37	9	10	8	96	9	8	9	8	7
0.7		37.5	37.5	62.5	75	50	50	75	87.5	50	62.5	100	15	16	14	15	16	13	15	11	8
0.8		75	12.5	62.5	75	50	75	75	75	62.5	75	42	100	22	20	19	17	18	16	13	11
0.9		25	25	25	25	62.5	50	25	25	50	62.5	100	100	100	24	22	24	21	19	20	20

A continuación las muestras colocadas en primer lugar (M1 y M2) se colocan al final de la tabla de entrada y se realiza el entrenamiento. El proceso se repite hasta que todos los grupos han sido utilizados para validación.

La tabla 6.7 muestra los resultados obtenidos para la validación con las muestra M1 y M2. En este caso la tasa ha disminuido al 87.5%, siendo el mapfield de 1x5 y 1x6.

Tabla 6.7. Tasa de aciertos y mapfield en función de ρ y β para la validación con las muestras M1 y M2.

		Tasa de aciertos (%)										Mapfield									
ρ/β		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0.1		62.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5	100	10	5	5	6	6	6	6	6	7
0.2		62.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5	100	10	5	5	6	6	6	6	6	7
0.3		62.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5	100	10	5	5	6	6	6	6	6	7
0.4		62.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5	100	10	5	5	6	6	6	6	6	7
0.5		62.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5	100	10	5	5	6	6	6	6	6	7
0.6		75	75	75	75	62.5	75	75	75	87.5	87.5	100	82	9	10	9	8	7	7	7	7
0.7		25	37.5	62.5	50	37.5	50	50	50	50	75	100	19	14	14	18	11	13	13	10	10
0.8		37.5	37.5	75	37.5	50	62.5	62.5	75	50	75	100	22	21	21	19	19	16	18	12	11
.9		37.5	50	25	37.5	50	37.5	25	25	25	37.5	100	49	23	27	23	24	22	19	20	18

La tabla 6.8 muestra los resultados obtenidos para la validación con las muestra M3 y M4. En este caso la tasa es del 100% siendo el mapfield de 1x4 y 1x5, el mínimo.

Tabla 6.8. Tasa de aciertos y mapfield en función de ρ y β para la validación con las muestras M3 y M4.

		Tasa de aciertos (%)										Mapfield									
ρ/β		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0.1		100	75	100	100	100	100	100	100	100	100	17	100	6	5	5	5	4	4	6	6
0.2		100	75	100	100	100	100	100	100	100	100	17	100	6	5	5	5	4	4	6	6
0.3		100	75	100	100	100	100	100	100	100	100	17	100	6	5	5	5	4	4	6	6
0.4		62.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5	17	100	6	5	5	5	4	4	6	6
0.5		62.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5	17	100	6	5	5	5	4	4	6	6
0.6		75	75	75	75	62.5	75	75	75	87.5	87.5	37	100	10	10	9	9	7	8	7	7
0.7		25	37.5	62.5	50	37.5	50	50	50	50	75	31	100	14	14	14	14	13	13	10	8
0.8		37.5	37.5	75	37.5	50	62.5	62.5	75	50	75	100	100	20	19	18	21	18	16	15	13
0.9		37.5	50	25	37.5	50	37.5	25	25	25	37.5	100	100	23	23	23	23	22	22	22	21

La tabla 6.9 muestra los resultados obtenidos para la validación con las muestra M5 y M6. En este caso la tasa ha disminuido al 87.5%, siendo el mapfield de 1x5 y 1x6. La tasa máxima del 100% se consigue con un mapfield mayor, que varía de 1x9 a 1x13.

Tabla 6.9. Tasa de aciertos y mapfield en función de ρ y β para la validación con las muestras M5 y M6.

		Tasa de aciertos (%)										Mapfield									
ρ/β		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0.1		87.5	75	75	87.5	87.5	87.5	87.5	87.5	87.5	87.5	100	100	100	100	6	6	6	6	5	5
0.2		87.5	75	75	87.5	87.5	87.5	87.5	87.5	87.5	87.5	100	100	100	100	6	6	6	6	5	5
0.3		87.5	75	75	87.5	87.5	87.5	87.5	87.5	87.5	87.5	100	100	100	100	6	6	6	6	5	5
0.4		87.5	75	75	87.5	87.5	87.5	87.5	87.5	87.5	87.5	100	100	100	100	6	6	6	6	5	5
0.5		87.5	75	75	87.5	87.5	87.5	87.5	87.5	87.5	87.5	100	100	100	100	6	6	6	6	5	5
0.6		75	75	75	75	100	100	75	100	87.5	75	52	100	100	17	8	8	9	9	6	6
0.7		62.5	75	75	62.5	100	100	75	100	75	75	40	90	34	17	15	13	13	12	9	8
0.8		62.5	62.5	62.5	100	75	50	50	75	75	87.5	73	100	28	19	19	16	18	18	16	15
0.9		37.5	37.5	37.5	37.5	25	50	50	50	37.5	50	48	100	100	23	22	23	24	22	21	20

Por lo tanto el mejor conjunto de muestras para entrenar la red es el formado por las muestras M1, M2 y M5 a M8, ya que para ρ de 0.1 a 0.3 y β de 0.7 a 0.8 se obtiene una tasa de aciertos del 100% con el mínimo mapfield 1x4.

A partir de los resultados expresados en las cuatro últimas tablas, se puede observar que la validación que tiene mayor tasa de éxitos (100%) y menor mapfield (4) es la realizada con las muestras M3 y M4. Por ello se va a

realizar un análisis más exhaustivo de variación de los parámetros α y β , de forma que se realice un salto de ρ cada centésima entre 0.1 y 0.3.

Tabla 6.10. Tasa de aciertos y <i>mapfield</i> en función de ρ y β .													
Validación con las muestras M3 y M4													
Tasa de aciertos (%)				<i>Mapfield</i>			Tasa de aciertos (%)				<i>Mapfield</i>		
ρ/β	0.7	0.75	0.8	0.7	0.75	0.8	ρ/β	0.7	0.75	0.8	0.7	0.75	0.8
0.10	100	100	100	4	4	4							
0.11	100	100	100	4	4	4	0.21	100	100	100	4	4	4
0.12	100	100	100	4	4	4	0.22	100	100	100	4	4	4
0.13	100	100	100	4	4	4	0.23	100	100	100	4	4	4
0.14	100	100	100	4	4	4	0.24	100	100	100	4	4	4
0.15	100	100	100	4	4	4	0.25	100	100	100	4	4	4
0.16	100	100	100	4	4	4	0.26	100	100	100	4	4	4
0.17	100	100	100	4	4	4	0.27	100	100	100	4	4	4
0.18	100	100	100	4	4	4	0.28	100	100	100	4	4	4
0.19	100	100	100	4	4	4	0.29	100	100	100	4	4	4
0.20	100	100	100	4	4	4	0.30	100	100	100	4	4	4

En la tabla 6.10 se comprueba que dentro del margen de variación de los parámetros ρ y β los resultados son óptimos y el tamaño del *mapfield* no puede disminuir de 4 porque el número de categorías es 4. Con esta tabla se confirma que se mantiene la tasa de aciertos del 100% para valores intermedios de ρ y β .

3.1.2. Selección de variables.

Para lograr un tamaño de memoria lo más pequeño posible, además de disminuir el *mapfield*, también es interesante disminuir si es posible el tamaño del vector de cada entrada. Es decir, disminuir el número de electrodos. En el caso de las lenguas electrónicas donde se utilizan sensores no específicos es muy probable que haya información redundante, porque hayan varios sensores que ofrezcan información muy similar.

En este apartado se va a realizar un análisis para determinar, si eligiendo los datos de parte de los electrodos, se obtiene un resultado similar a cuando se eligen todos o incluso puede ser que se obtenga un resultado mejor.

El análisis se va a realizar utilizando diversos métodos de selección de variables y se elegirá aquel que obtenga una mayor tasa de resultados.

Como se ha comentado en el capítulo 2, la GUI Fuzzy Artmap se basa en cinco algoritmos desarrollados por Lapiedra [Lapiedra, 2004] en MATLAB®.

Los métodos Forward Selección y Backward Eliminación son rápidos y es el usuario quien elige el número de variables mediante menú desplegable.

En los métodos Forward PNN y Backward PNN los datos son validados mediante red neuronal probabilística pero una vez seleccionada (Forward) o eliminada (Backward) una variable esta no es nuevamente utilizada.

Finalmente el método Stepwise PNN es una combinación de los dos anteriores.

A cada electrodo se le ha asignado un número de acuerdo con la tabla 6.11.

Tabla 6.11. Número asignados a cada electrodo.							
Electrodo	Au	Ag	Cu	Ag ₂ O	Cu ₂ O	AgCl	Ag ₂ CO ₃
Variable nº	1	2	3	4	5	6	7

Los resultados obtenidos por los diferentes métodos se muestran a continuación. No se han considerado los tiempos de ejecución debido a que al ser un fichero de 32 datos y con pocas variables, dicho tiempo ha sido inferior a un minuto.

Tabla 6.12. Resumen de los datos obtenidos		
Método	Variables elegidas	Tasa
<i>Forward</i> (3 variables)	1 3 6	-----
<i>Forward</i> (4 variables)	1 2 3 6	-----
<i>Backward</i> (3 variables)	1 5 6	-----
<i>Backward</i> (4 variables)	1 3 5 6	-----
<i>Forward PNN</i>	1 2 3	96.875%
<i>Backward PNN</i>	1 2 3	96.875%
<i>Stepwise PNN</i>	1 2 3 5	100%

La validación cruzada con 7 electrodos ha dado un 100% de aciertos al validar con las muestras M3 y M4, con un *mapfed* de 1x4, mientras que en el

método *Stepwise PNW* ha dado también un 100% con los electrodos de Au, Ag, Cu y Cu₂O.

Vamos a combinar las muestras M3 y M4 con las variables 1 2 3 5 y realizar un nuevo entrenamiento con la red Fuzzy Artmap con parámetros de $\rho=0.1$ a $\rho=0.3$ y $\beta=0.7$ a $\beta=0.9$, con incrementos de 0.01 y 0.05 respectivamente.

Los resultados se muestran en la tabla 6.13. Como se observa en dicha tabla, en todos los casos se ha obtenido una tasa de aciertos del 100% con un *mapfield* muy reducido de 1x6. Todo ello reduce considerablemente las necesidades de memoria del microcontrolador.

Tabla 6.13. Tasa de aciertos y <i>mapfield</i> en función de ρ y β . Validación con las muestras M3 y M4 y con electrodos de Au, Ag, Cu y Cu ₂ O.													
Tasa de aciertos (%)				<i>Mapfield</i>			Tasa de aciertos (%)				<i>Mapfield</i>		
ρ/β	0.7	0.75	0.8	0.7	0.75	0.8	ρ/β	0.7	0.75	0.8	0.7	0.75	0.8
0.10	100	100	100	6	6	6							
0.11	100	100	100	6	6	6	0.21	100	100	100	6	6	6
0.12	100	100	100	6	6	6	0.22	100	100	100	6	6	6
0.13	100	100	100	6	6	6	0.23	100	100	100	6	6	6
0.14	100	100	100	6	6	6	0.24	100	100	100	6	6	6
0.15	100	100	100	6	6	6	0.25	100	100	100	6	6	6
0.16	100	100	100	6	6	6	0.26	100	100	100	6	6	6
0.17	100	100	100	6	6	6	0.27	100	100	100	6	6	6
0.18	100	100	100	6	6	6	0.28	100	100	100	6	6	6
0.19	100	100	100	6	6	6	0,29	100	100	100	6	6	6
0.20	100	100	100	6	6	6	0,30	100	100	100	6	6	6

Las tablas 6.14, 6.15 y 6.16 muestran la matriz de pesos, el *mapfield* y los máximos y mínimos que se necesitan para implementar la red Fuzzy Artmap Simplificada en un microcontrolador.

Tabla 6.14. Matriz de pesos definitivos							
Directa	Au	0.651145	0.567176	0.130534	0.000000	0.185496	0.720916
	Ag	0.000000	0.176336	0.679389	0.749618	0.746565	0.479084
	Cu	0.619296	0.354628	0.434159	0.000000	0.342894	0.796610
	CuO ₂	0.000000	0.432855	0.430248	0.691004	0.646675	0.403390
Complem.	Au	0.501322	0.039648	0.475771	0.000000	0.274009	0.662379
	Ag	0.000000	0.262555	0.306608	0.806167	0.695154	0.537621
	Cu	0.577160	0.513889	0.324846	0.000000	0.342593	0.821605
	CuO ₂	0.000000	0.204475	0.368056	0.652778	0.597222	0.378395

1	2	3	4	3	1
---	---	---	---	---	---

0.1307	-0.0003
0.1314	0.0547
-0.05	-0.1635
-0.0126	-0.1422

Con los datos definitivos de la red neuronal Fuzzy Artmap: matriz de pesos, mapfield y máximos y mínimos, se vuelve a realizar un análisis con verificación de la red para determinar si clasifica adecuadamente los 8 datos de los grupos M3 y M4. Los resultados se muestran mediante una matriz de confusión que está representada en la Fig. 6-9 donde se comprueba que, en efecto, todas las medidas las clasifica en su categoría adecuada.

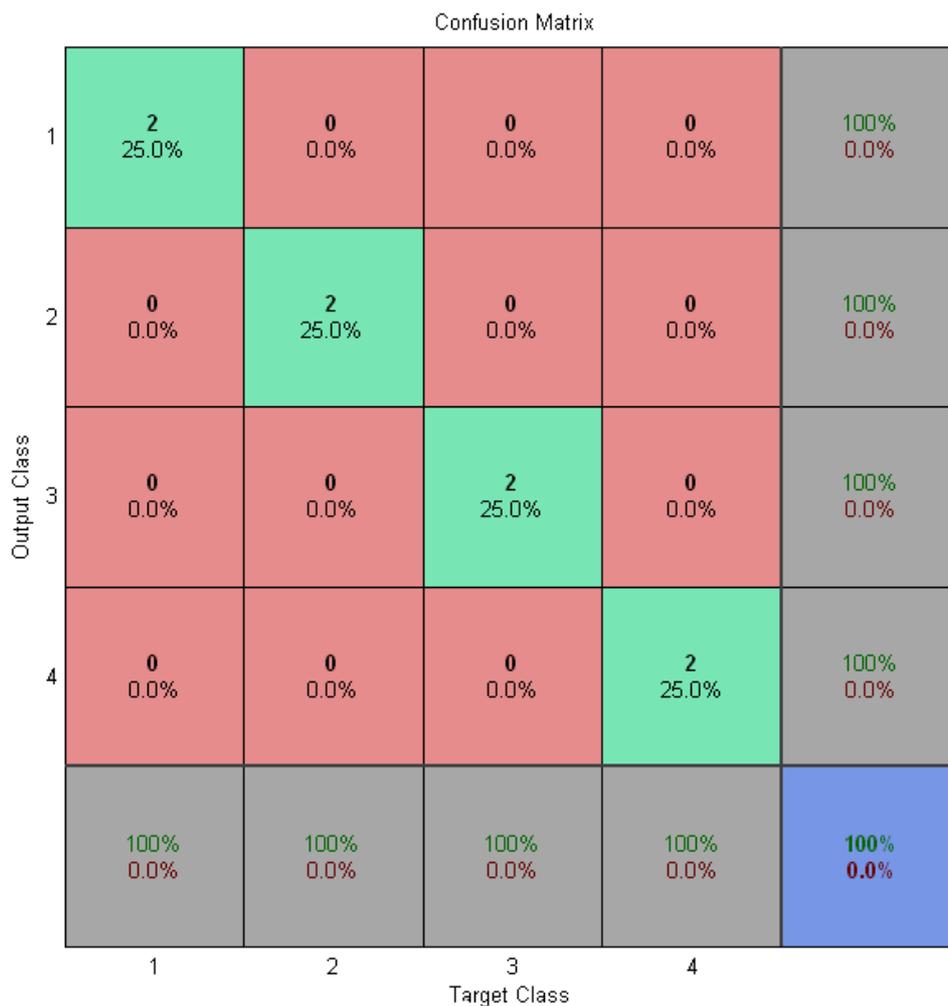


Figura 6-9. Matriz de confusión para $\rho=0,3$ y $\beta=0,8$ obtenida en el entrenamiento.

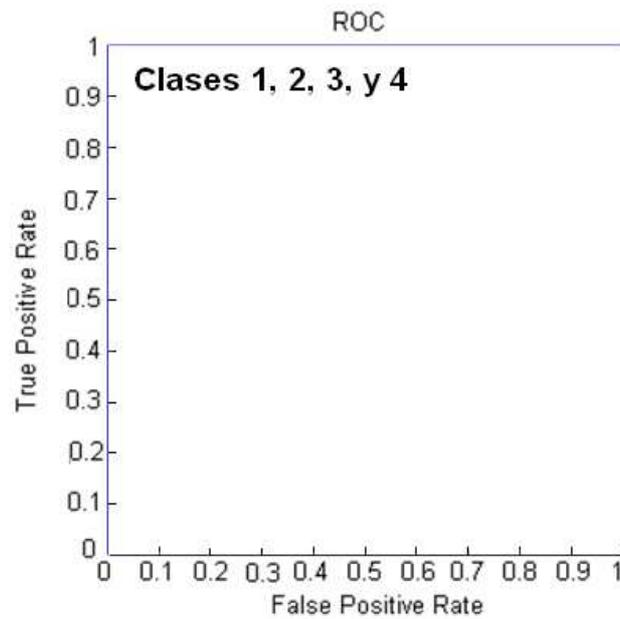


Figura 6-10. Curvas ROC para $\rho=0,3$ y $\beta=0,8$ obtenidas en el entrenamiento.

En la Fig. 6-9 se muestra la matriz de confusión y en la Fig.6-10 las curvas ROC obtenidas para $\rho=0,3$ y $\beta=0,8$. En este caso como el porcentaje de aciertos es del 100%, tanto la sensibilidad (tpr) como la especificidad es 1 (fpr=0), todas las curvas terminan en el vértice superior izquierdo.

3.2. Validación cruzada de los tratamientos térmicos de la miel.

La segunda parte del análisis de las medidas con miel es comprobar sin con los datos obtenidos la red neuronal es capaz de discriminar la muestras según el tratamiento físico-térmico recibido. La red neuronal consiste en tres salidas, una para cada tratamiento físico. Para obtener un número múltiplo de tres se han tomado los 30 primeros datos de los 32 datos iniciales. Estos datos se han dividido en 10 muestras de tres datos cada una, de modo que cada muestra tenga un dato de cada una de las tres categorías. Se ha realizado una validación cruzada de orden 5. Para ello se han realizado agrupaciones de dos muestras. Con los datos de estas muestras se ha realizado un barrido de los valores de los parámetros ρ y β para determinar la tasa de aciertos y el tamaño del mapfield. Los resultados obtenidos con 24 datos (8 muestras) de entrenamiento y 6 datos (2 muestras) de prueba se muestran en la tabla 6.16.

Tabla 6.17. Tasa de aciertos y mapfield en función de ρ y β para los tratamientos físicos de la miel											
		Tasa de aciertos (%)					Mapfield				
ρ	β	M9-M10	M1-M2	M3-M4	M5-M6	M7-M8	M9-M10	M1-M2	M3-M4	M5-M6	M7-M8
0,1	0,1	16.67	33.33	0.00	33.33	16.67	39	100	26	25	100
0,1	0,2	16.67	66.67	0.00	0.00	0.00	18	100	15	100	23
0,1	0,3	0.00	33.33	0.00	66.67	16.67	39	100	12	21	14
0,1	0,4	0.00	66.67	0.00	33.33	16.67	100	13	14	23	10
0,1	0,5	0.00	50.00	0.00	33.33	16.67	100	17	14	17	11
0,1	0,6	50.00	50.00	0.00	50.00	16.67	15	100	15	16	11
0,1	0,7	33.33	50.00	0.00	50.00	0.00	16	15	14	16	12
0,1	0,8	33.33	50.00	16.67	16.67	0.00	15	16	15	15	12
0,1	0,9	33.33	50.00	0.00	16.67	0.00	14	16	12	15	14
0,1	1	50.00	50.00	0.00	33.33	0.00	18	15	13	10	15
0,2	0,1	16.67	33.33	0.00	33.33	16.67	39	100	26	25	100
0,2	0,2	16.67	66.67	0.00	0.00	0.00	18	100	15	100	23
0,2	0,3	0.00	33.33	0.00	66.67	16.67	39	100	12	21	14
0,2	0,4	0.00	66.67	0.00	33.33	16.67	100	13	14	23	10
0,2	0,5	0.00	50.00	0.00	33.33	16.67	100	17	14	17	11
0,2	0,6	50.00	50.00	0.00	50.00	16.67	15	100	15	16	11
0,2	0,7	33.33	50.00	0.00	50.00	0.00	16	15	14	16	12
0,2	0,8	33.33	50.00	16.67	16.67	0.00	15	16	15	15	12
0,2	0,9	33.33	50.00	0.00	16.67	0.00	14	16	12	15	14
0,2	1	50.00	50.00	0.00	33.33	0.00	18	15	13	10	15
0,3	0,1	16.67	33.33	0.00	33.33	16.67	39	100	26	25	100
0,3	0,2	16.67	66.67	0.00	0.00	0.00	18	100	15	100	23
0,3	0,3	0.00	33.33	0.00	66.67	16.67	39	100	12	21	14
0,3	0,4	0.00	66.67	0.00	33.33	16.67	100	13	14	23	10
0,3	0,5	0.00	50.00	0.00	33.33	16.67	100	17	14	17	11
0,3	0,6	50.00	50.00	0.00	50.00	16.67	15	100	15	16	11
0,3	0,7	33.33	50.00	0.00	50.00	0.00	16	15	14	16	12
0,3	0,8	33.33	50.00	16.67	16.67	0.00	15	16	15	15	12
0,3	0,9	33.33	50.00	0.00	16.67	0.00	14	16	12	15	14
0,3	1	50.00	50.00	0.00	33.33	0.00	18	15	13	10	15
0,4	0,1	16.67	33.33	0.00	33.33	16.67	39	100	26	25	100
0,4	0,2	16.67	66.67	0.00	0.00	0.00	18	100	15	100	23
0,4	0,3	0.00	33.33	0.00	66.67	16.67	39	100	12	21	14
0,4	0,4	0.00	66.67	0.00	33.33	16.67	100	13	14	23	10
0,4	0,5	0.00	50.00	0.00	33.33	16.67	100	17	14	17	11
0,4	0,6	50.00	50.00	0.00	50.00	16.67	15	100	15	16	11
0,4	0,7	33.33	50.00	0.00	50.00	0.00	16	15	14	16	12
0,4	0,8	33.33	50.00	16.67	16.67	0.00	15	16	15	15	12
0,4	0,9	33.33	50.00	0.00	16.67	0.00	14	16	12	15	14
0,4	1	50.00	50.00	0.00	33.33	0.00	18	15	13	10	15
0,5	0,1	50.00	33.33	0.00	33.33	16.67	100	100	26	26	100
0,5	0,2	33.33	66.67	0.00	0.00	0.00	21	100	15	100	23
0,5	0,3	16.67	33.33	0.00	33.33	16.67	40	100	12	22	14

Aplicación de las Redes Neuronales con μC de 8 Bits para la Clasificación de Mieles

0,5	0,4	0.00	66.67	0.00	33.33	16.67	100	13	14	23	10
0,5	0,5	16.67	50.00	0.00	33.33	16.67	100	18	14	17	11
0,5	0,6	50.00	50.00	0.00	50.00	16.67	15	100	15	16	11
0,5	0,7	33.33	50.00	0.00	50.00	0.00	16	15	14	16	12
0,5	0,8	33.33	50.00	16.67	16.67	0.00	15	16	15	15	12
0,5	0,9	33.33	50.00	0.00	16.67	0.00	14	16	12	15	14
0,5	1	50.00	50.00	0.00	33.33	0.00	18	15	13	10	15
0,6	0,1	16.67	33.33	0.00	33.33	66.67	100	100	29	93	57
0,6	0,2	16.67	50.00	0.00	33.33	16.67	23	100	15	100	24
0,6	0,3	33.33	66.67	16.67	66.67	16.67	53	21	13	100	14
0,6	0,4	0.00	66.67	0.00	50.00	16.67	100	14	13	23	10
0,6	0,5	33.33	50.00	0.00	33.33	16.67	18	17	14	17	11
0,6	0,6	50.00	50.00	0.00	50.00	16.67	15	100	15	16	11
0,6	0,7	33.33	50.00	0.00	50.00	0.00	16	15	14	16	12
0,6	0,8	33.33	50.00	16.67	50.00	0.00	15	16	15	15	12
0,6	0,9	33.33	50.00	0.00	16.67	0.00	14	16	12	15	14
0,6	1	50.00	50.00	0.00	33.33	0.00	18	15	13	12	15
0,7	0,1	33.33	16.67	33.33	50.00	33.33	100	100	100	49	27
0,7	0,2	33.33	33.33	33.33	33.33	0.00	100	100	16	100	100
0,7	0,3	16.67	83.33	50.00	33.33	16.67	100	15	16	100	14
0,7	0,4	0.00	83.33	16.67	66.67	33.33	100	100	14	24	12
0,7	0,5	33.33	50.00	16.67	50.00	33.33	18	17	15	17	11
0,7	0,6	33.33	66.67	33.33	50.00	33.33	15	22	17	100	11
0,7	0,7	33.33	50.00	33.33	50.00	16.67	16	16	16	16	12
0,7	0,8	33.33	50.00	16.67	33.33	16.67	15	20	16	19	12
0,7	0,9	33.33	50.00	33.33	33.33	16.67	14	17	13	16	14
0,7	1	50.00	50.00	16.67	66.67	16.67	18	16	13	16	15
0,8	0,1	33.33	16.67	33.33	50.00	50.00	100	100	80	87	98
0,8	0,2	33.33	50.00	0.00	66.67	33.33	100	100	100	100	100
0,8	0,3	16.67	33.33	16.67	66.67	33.33	100	18	22	29	19
0,8	0,4	33.33	0.00	0.00	66.67	33.33	100	100	23	24	19
0,8	0,5	16.67	33.33	16.67	66.67	33.33	18	100	21	19	19
0,8	0,6	33.33	0.00	33.33	66.67	33.33	17	100	22	20	19
0,8	0,7	16.67	50.00	0.00	66.67	66.67	19	17	17	16	18
0,8	0,8	0.00	16.67	16.67	66.67	66.67	15	18	17	16	19
0,8	0,9	0.00	16.67	0.00	66.67	33.33	15	20	16	17	18
0,8	1	33.33	50.00	0.00	66.67	50.00	19	18	16	15	18
0,9	0,1	16.67	33.33	33.33	33.33	50.00	100	100	28	52	39
0,9	0,2	16.67	50.00	16.67	50.00	33.33	100	100	36	100	100
0,9	0,3	33.33	16.67	33.33	50.00	16.67	100	100	100	100	27
0,9	0,4	16.67	0.00	16.67	16.67	0.00	28	100	26	100	25
0,9	0,5	16.67	50.00	33.33	33.33	16.67	23	24	27	25	24
0,9	0,6	16.67	33.33	16.67	50.00	16.67	24	23	23	25	24
0,9	0,7	16.67	50.00	0.00	33.33	16.67	23	23	24	24	24
0,9	0,8	16.67	33.33	0.00	50.00	16.67	23	24	22	24	22
0,9	0,9	16.67	50.00	0.00	16.67	16.67	23	23	23	23	23

0,9	1	16.67	50.00	0.00	33.33	16.67	22	23	22	24	22
-----	---	-------	-------	------	-------	-------	----	----	----	----	----

En la tabla anterior sólo cabe destacar una tasa de aciertos del 83.3% para $p=0.7$ y $\beta=0.3$, con un *mapfield* de 1x15 y validando con las muestras M1 y M2, lo que indica que no hay agrupación de los datos. Aplicando la misma técnica de validación que en la anterior red, el éxito de la clasificación es mucho menor y el *mapfield* muy elevado, lo que sugiere que el tratamiento térmico no afecta a las cualidades intrínsecas de la miel.

4. Implementación en el microcontrolador de la red SFAM.

4.1. Introducción.

Una vez determinada cual es la arquitectura óptima de la red neuronal SFAM se va a proceder a la implementación de la red neuronal en un sistema integrado formado por el microprocesador PIC18F4550 de Microchip, cuyas características generales han sido expuestas en el capítulo 2 de la presente memoria.

El software para el microcontrolador fue codificado en lenguaje C y consta de dos rutinas principales:

- Sistema de adquisición de datos, en el que el microcontrolador lee los datos del convertidor A/D y los procesa con el fin de obtener el promedio.
- Realización de la red neuronal. El programa sigue el organigrama que se ha especificado en el capítulo 1 de la presente memoria

4.2. Realización de las redes SFAM.

En esta rutina los vectores de entrada I , son normalizados para establecer su rango de $[0,1]$, con la misma función del MATLAB®. Con el fin de preservar la amplitud de la información el dato de entrada se complementa.

La función de clasificación que se implementó en el microcontrolador, se codifica en el lenguaje C y se convierte en código hexadecimal usando un compilador cruzado. El archivo HEX se descarga en la memoria flash del microcontrolador. La red neuronal SFAM se ha programado en 12.662 bytes de

memoria de programa (ROM 39%) y en 1.632 bytes de memoria de datos (RAM 79%).

Tabla 6.18. Datos de prueba en el microcontrolador (mV).

Electrodo/Clase	1	2	3	4	Muestra
Au	0.1105	0.0762	0.0173	-0.0085	M9
Ag	0.1287	0.0939	0.0762	0.0640	
Cu	-0.0578	-0.1020	-0.0912	-0.1879	
Ag ₂ O	0.1388	0.1147	0.1022	0.0927	
Cu ₂ O	-0.0460	-0.0580	-0.0555	-0.1308	
AgCl	0.0367	0.0259	0.0118	-0.0022	
Ag ₂ CO ₃	0.0946	0.1030	0.0742	0.0447	
Au	0.0872	0.0761	0.0279	0.0213	M10
Ag	0.1112	0.0986	0.0925	0.0920	
Cu	-0.0553	-0.0790	-0.1077	-0.1539	
Ag ₂ O	0.1328	0.1206	0.1076	0.1095	
Cu ₂ O	-0.0196	-0.0401	-0.0810	-0.1123	
AgCl	0.0246	0.0212	0.0167	0.0276	
Ag ₂ CO ₃	0.1084	0.1052	0.0923	0.0879	
Au	0.0969	0.1016	0.0256	0.0029	M11
Ag	0.1194	0.1132	0.0736	0.0666	
Cu	-0.0934	-0.1341	-0.0733	-0.1426	
Ag ₂ O	0.1377	0.1263	0.0956	0.0815	
Cu ₂ O	-0.0490	-0.0706	-0.0589	-0.1091	
AgCl	0.0463	0.0347	0.0015	-0.0061	
Ag ₂ CO ₃	0.1051	0.1008	0.0724	0.0693	
Au	0.1217	0.0870	0.0286	-0.0077	M12
Ag	0.1367	0.1053	0.0823	0.0317	
Cu	-0.0760	-0.1041	-0.1145	-0.1681	
Ag ₂ O	0.1400	0.1228	0.1079	0.0648	
Cu ₂ O	-0.0456	-0.0634	-0.0972	-0.1440	
AgCl	0.0374	0.0345	0.0102	-0.0224	
Ag ₂ CO ₃	0.0835	0.1041	0.0767	0.0335	

Una vez realizado el entrenamiento en el PC y obtenida la arquitectura óptima de la red, se procede a realizar nuevas medidas de prueba para comprobar si la red las clasifica adecuadamente. La prueba se realizó con

cuatro muestras (M9, M10, M11 y M12) con cuatro datos cada una, lo que hace un total de 16 datos.

Los datos de las muestras de prueba están formados por los siete electrodos iniciales del entrenamiento. Esto permite hacer dos clasificaciones diferentes, la primera utilizando los 7 electrodos y una segunda clasificación utilizando únicamente los 4 electrodos elegidos en la selección de variables.

La tabla 6.18 muestra los datos de prueba, mientras que la tabla 6.19 muestra la clasificación obtenida.

Tabla 6.19. Salidas obtenidas en el μ C.										
Muestra	a) con 7 electrodos					b) con 4 electrodos				
	Clase				Tasa	Clase				Tasa
M9	1	1	3	4	75%	1	2	3	4	100%
M10	1	1	3	3	50%	1	1	3	3	50%
M11	1	1	3	4	75%	1	1	3	4	75%
M12	1	1	3	4	75%	1	1	3	4	75%

En la Fig. 6-11 se muestra la matriz de confusión con 7 y 4 electrodos, mientras que la Fig. 6-12, muestra las curvas ROC.

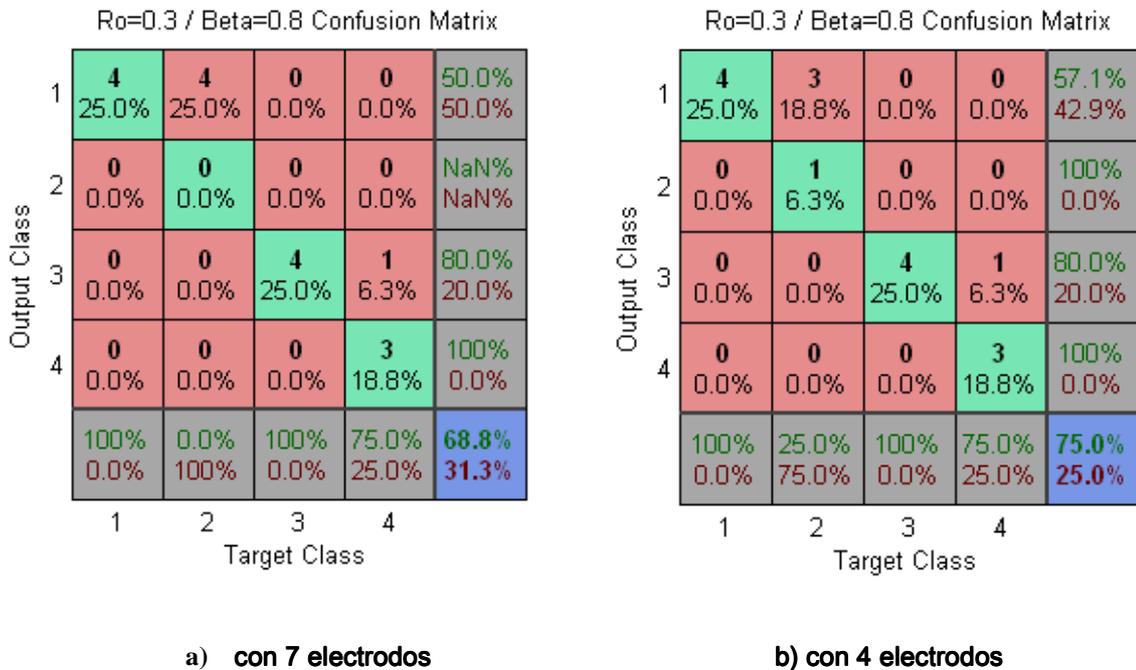


Figura 6-11. Matriz de Confusion para $\rho=0,3$ y $\beta=0,8$ obtenidas en el microcontrolador.

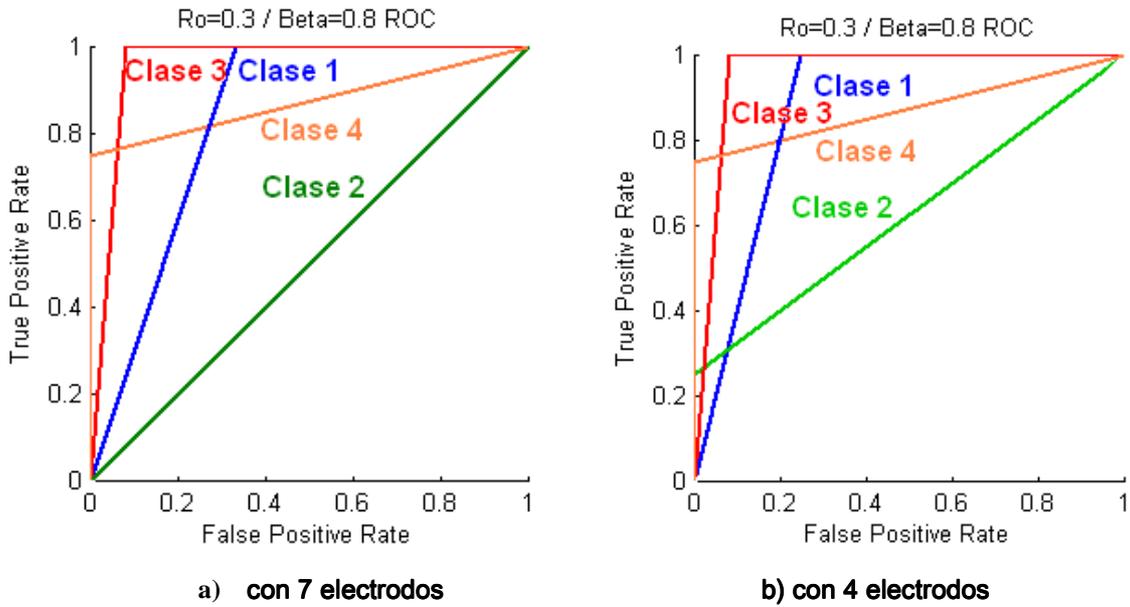


Figura 6-12. Curvas ROC para $\rho=0,3$ y $\beta=0,8$ obtenidas en el microcontrolador.

Como se observa en la tabla 6.20, la tasa de aciertos es bastante baja (75%), ello es posiblemente debido a que la cantidad total de datos para el entrenamiento (24) sea insuficiente. Podemos tomar 2 ó 3 muestras obtenidas en el microcontrolador (M9-M12) para entrenar junto con las seis iniciales y validar con 2 ó 1 muestra restante.

Tabla 6.20. Tasa de aciertos para $\rho=0,3$ y $\beta=0,8$			
Muestras		Nº Electrodo	
Entrenamiento	Validación	4	7
8	2 (M11,M12)	87.5%	75%
9	1 (M12)	100%	75%

Tabla 6.21. Mapfield para $\rho=0,3$ y $\beta=0,8$			
Muestras		Nº Electrodo	
Entrenamiento	Validación	4	7
8	2 (M11,M12)	1x6	1x6
9	1 (M12)	1x6	1x6

La tabla 6.20 muestra los resultados obtenidos al entrenar con 8 ó 9 muestras y validar con 2 ó 1 muestra para $\rho=0.3$ y $\beta=0.8$. En dicha tabla se observa que la tasa de aciertos mejora considerablemente cuando el número de electrodos utilizados es de 4. Es decir, cuando previamente se ha hecho una selección de variables. El mapfield obtenido es el mismo en todos los casos (tabla 6.21).

4.3. Comparación entre las dos redes.

En el origen floral de la miel, tenemos un mapfield de 6, sobre un total de 8 datos de entrada, lo que indica que 4 datos de entrada se han agrupado en dos grupos, siendo la tasa de aciertos del 100%. En el caso del tratamiento térmico de la miel, tenemos un mapfield de 15 sobre un total de 24 datos de entrada, lo que indica que no hay agrupamiento.

5. Conclusión.

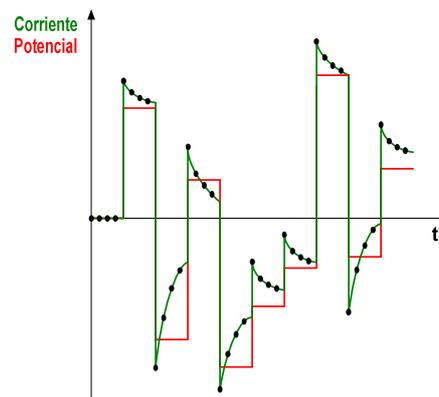
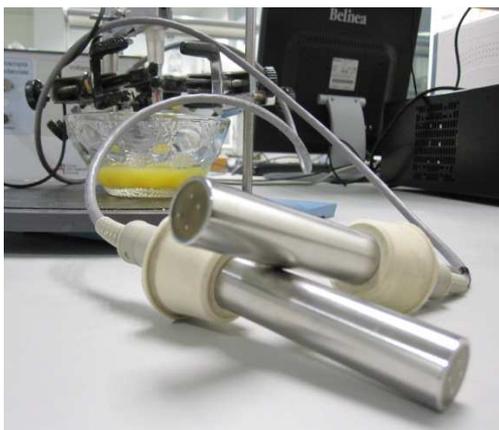
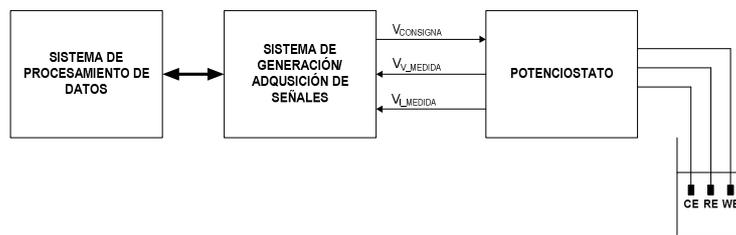
Gracias a un sistema de lengua electrónica potenciométrica, se ha clasificado la miel de diferentes orígenes florales (azahar, romero, milflores y bosque). Al mismo tiempo, las mieles no se han discriminado de acuerdo con el tratamiento térmico a que fueron sometidas. Esto puede significar que el calor de tales tratamientos no afecta a sus propiedades bioquímicas y por lo tanto, no tiene repercusiones en el comportamiento electroquímico.

6. Producción científica.

Luis Gil-Sánchez, Eduardo Garcia-Breijo, José Garrigues, Miguel Alcañiz, Isabel Escriche, Melinda Kadar. *Classification of honeys of different floral origins by artificial neural networks*. IEEE SENSORS 2011 CONFERENCE, Limerick (Irlanda)

Capítulo 7

Aplicación de las Redes Neuronales en Microcontroladores de 8 Bits para la Determinación de Glifosato Mediante la Técnica de Voltamperometría de Pulsos



A lo largo de este capítulo se describen los análisis realizados con redes neuronales artificiales para la determinación de la concentración de glifosato en disoluciones a partir de los datos procedentes de medidas en muestras con diferentes concentraciones de glifosato en una disolución tampón. Las mediciones se han realizado con un dispositivo electrónico desarrollado por el Grupo de Microelectrónica Híbrida (GMH) del Instituto de Reconocimiento Molecular y Desarrollo Tecnológico (IDM) de la UPV llamado FraPlusMini, que es capaz de ejecutar las pruebas de voltametría de pulsos, de voltametría cíclica y de amperometría.

1. Introducción.

El glifosato es un herbicida, no selectivo, de amplio espectro desarrollado para la eliminación de hierbas y arbustos. Químicamente es nombrado como fosfometilamino ácido acético oN-(fosfometil) glicina (CAS RN 1071-83-6). Este producto químico es polar, no volátil, muy soluble en agua y prácticamente insoluble en disolventes orgánicos. La forma en que este producto actúa como un herbicida es mediante la inhibición de la enzima 5-enolpiruvilsiquimato-3-fosfato sintasa, también conocido por el acrónimo EPSP, que tiene una función esencial en el metabolismo de las plantas, ya que participa en la síntesis de algunos aminoácidos como el triptófano y la tirosina [Sierra et al. 2008], que son fundamentales para la vida vegetal.

El consumo del glifosato ha aumentado espectacularmente en las vastas áreas rurales de todo el mundo, asociado a la aparición de variedades de cultivos genéticamente modificados y específicamente diseñados para ser resistentes a este herbicida. El problema con este herbicida es está relacionado con el hecho de que su uso está controlado en la mayoría de los países desarrollados [B.O.E. 83. 2007][D.O.U.E. 206. 2006], pero todavía se utiliza sin mucho control en otras regiones.

En algunas regiones, se utilizan aviones ligeros para aplicar este herbicida rociándolo sobre el campo. Pero el área de aplicaciones poco selectiva y el producto, además de establecerse en el cultivo a tratar, también precipita en otras áreas no deseadas que afectan al agua [Nirascou, 2006] [C. Cox 1998], al suelo, a las plantas y a los cultivos [Sawchuk et al., 2006], a la

fauna y también directa e indirectamente a los seres humanos [Contardo et al., 2009].

El glifosato tiene un periodo de descomposición que va desde 7 a 60 días [Accinelli et al., 2004], pero puede persistir más de 174 días bajo ciertas condiciones ambientales [Estelar, 1999] y además, la toxicología del producto es objeto de debate intenso. Es bien sabido que, a pesar de que no se ha demostrado que sea cancerígeno, estudios recientes han demostrado su toxicidad y su poder como un disruptor endocrino [Gasnier et al., 2009]. Mientras tanto, lo que puede suceder con su uso a largo plazo y los compuestos derivados de su degradación es aún desconocido.

Por otro lado, las técnicas actuales de análisis para la detección y cuantificación de glifosato y sus derivados en el medio físico (agua y suelo), en los animales o en las plantas, son complejas y costosas. En este sentido, uno de los métodos más utilizados es la cromatografía líquida de alta espectrometría de masas (HPLC-MS) [Sánchez et al., 2010], aunque existen otros métodos como pruebas de inmuno absorción (ELISA) [Sánchez et al., 2010], colorimetría [Coutinho et al., 2008], *Capillary Electrophoresis* [Chiu et al., 2008] y las técnicas de electroquimioluminiscencia, que hacen posible un análisis efectivo de glifosato [Jin et al., 2010]. Sin embargo, estos procedimientos son generalmente lentos, costosos y tienen que ser realizados en un laboratorio especializado, por lo que el avance en el desarrollo de la sensibilidad rápida, e "in situ" de los procedimientos analíticos es una obra de gran interés [Jin et al., 2010].

En este sentido, hay avances muy importantes, como los presentados por Aquino [Aquino & Andrade, 2009] sobre la base de electroquímica y con sensores hechos con diferentes materiales como plata y platino [Méndez et al., 2007], cobre [Coutinho et al., 2007] o las últimas investigaciones con enzimas reactivas [Songa et al. 2009] y de doble capa de hidróxidos [Khenifi et al., 2009], que podría ser la base para la futura detección "in situ" del glifosato. En este sentido, parece que las técnicas de voltamperometría/amperométrica pueden ser la base para el desarrollo "in situ" de los procedimientos de medición. De hecho, la voltamperometría es la base del procedimiento propuesto para predecir la concentración de glifosato y se ha utilizado un dispositivo de

voltamperometría / amperométrica llamado “**FraPlus**” para hacer las mediciones de voltamperometría / amperométrica en un laboratorio.

Finalmente, el objetivo de este capítulo de la tesis es presentar los resultados del trabajo desarrollado para implementar una red neuronal artificial en un microcontrolador con el fin de detectar concentraciones de glifosato en una disolución tampón por voltamperometría de pulsos. Para lograr este objetivo, se han realizado varios ensayos con diferentes concentraciones del herbicida y que han sido analizados mediante diversos metales (nobles y no nobles) como electrodos. Se han utilizado los datos obtenidos por voltametría con FraPlus para entrenar la red neuronal artificial presentada en esta tesis.

2. Descripción del sistema.

2.1. Voltametría.

La potenciometría se basa en la medida del potencial de un sistema en equilibrio, con un paso de corriente despreciable, para determinar las actividades de unas determinadas especies presentes en una disolución.

En cambio, en los métodos voltamétricos, las medidas se efectúan cuando en el sistema el estado de equilibrio no se ha alcanzado y, por lo tanto, la señal obtenida es una relación intensidad-potencial. Un sistema de voltametría emplea como mínimo tres electrodos, un electrodo de referencia, uno de trabajo y un electrodo auxiliar, cada uno tiene su propia función [*Escrivà, 2011*]:

- El electrodo auxiliar es el encargado de suministrar la corriente a la celda.
- El electrodo de trabajo es donde se producen los procesos electroquímicos y donde se mide la señal de corriente.
- El electrodo de referencia es el elemento con respecto al cual se aplica la tensión al electrodo de trabajo.

El potencial de electrodo de referencia es constante, por lo que, el flujo de corriente ocurre entre el electrodo de trabajo y el auxiliar. Los potenciales se aplican al electrodo de trabajo midiendo la corriente resultante. Los sensores voltamétricos son dispositivos que ofrecen ventajas importantes para el análisis multicomponente, como son su alta selectividad y sensibilidad, alta relación

señal-ruido, bajo límite de detección y además que con ellos se pueden utilizar varios modos de medición (voltametría de onda cuadrada, voltametría cíclica, voltametría de pulsos, etc.).

En las técnicas voltamétricas de los datos procesados se pueden extraer de la representación de los voltagramas, pero también existe la posibilidad de utilizar la información correspondiente a la evolución temporal de la corriente. En la mayoría de las técnicas voltamétricas la medida de la corriente se utiliza para proceder a la representación gráfica del voltagrama. Este es el caso de la voltametría de barrido lineal, de la voltametría de disolución, de la voltametría cíclica y de la mayoría de las técnicas voltamétricas de pulsos ya que los puntos en los que la corriente se mide corresponden a valores crecientes de tensión y, por tanto, es posible la representación para esos puntos de la gráfica de corriente frente a tensión. Sin embargo, en algunas técnicas el objetivo no es la obtención de un voltagrama sino el estudio de la evolución temporal de corriente en el electrodo cuando éste es sometido a escalones de potencial.

Los valores de las muestras de corriente son procesados mediante herramientas de análisis multivariante con objeto de clasificar las muestras o generar modelos predictivos que permitan cuantificar un componente determinado de la disolución. En la siguiente figura se representan las señales típicas de este tipo de ensayos. En rojo se muestran los pulsos de tensión aplicados al electrodo de trabajo mientras que en verde aparece la señal de corriente que circula por el mismo. Los puntos son los valores de corriente que son muestreados, y por lo tanto, se utilizan para el posterior análisis multivariante.

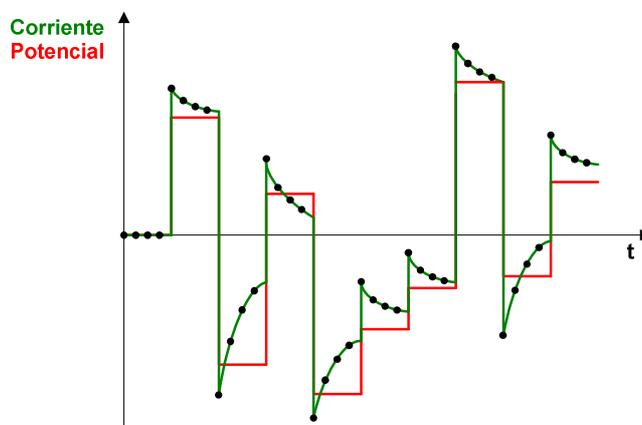


Figura 7-1. Voltametría de corriente muestreada.

Si se estudia cada pulso de forma individual se observa que la señal de corriente tiene un comportamiento exponencial debido a la corriente farádica (debida a las reacciones de oxidación-reducción que ocurren en el interfaz electrodo-disolución) y la corriente no farádica (debida a la reorganización de las cargas en la doble capa eléctrica). La corriente total es la suma de los dos tipos de corriente. Si el potencial aplicado produce una reacción de oxidación-reducción, la corriente obtenida será la suma de la corriente no farádica y la corriente farádica resultante de la reacción. Ambas corrientes están relacionadas con la naturaleza del electrodo y la naturaleza y concentración de las especies presentes en la disolución. Por lo tanto, el estudio de la evolución temporal de la corriente total permite obtener información sobre la composición química de la muestra.

Para la determinación cuantitativa de analitos por voltametría dentro de un intervalo comprendido entre 10^{-4} y 10^{-5} M, no tenemos que preocuparnos de la corriente no farádica. Las corrientes farádicas de la electrolisis del analito son mucho mayores y duran más tiempo. Sin embargo, para determinar la concentración de la disolución más diluida (del orden de 10^{-6} M o inferiores) hay que poner especial cuidado en separar las corrientes farádicas de las no farádicas y de las corrientes residuales. Para realizar esto se utilizan dos métodos diferentes de voltametría de impulsos (polarografía y voltametría diferencial de impulsos).

Las corrientes farádicas tienen dos propiedades que ayudan a disminuir la influencia de corrientes no farádicas [*Escrivá, 2011*]:

1. Cuanto más rápido es el cambio de potencial, mayores son las corrientes no farádicas.
2. Después de un cambio de potencial, las corrientes no farádicas disminuyen muy rápido en relación con las corrientes farádicas.

Para minimizar la interferencia de las corrientes no farádicas, la corriente se mide justo antes del impulso y de nuevo hacia el final del mismo. Esta manipulación del potencial en el electrodo de trabajo y la toma de datos se realizan electrónicamente.

En la presente tesis doctoral la metodología utilizada consiste en aplicar una serie de pulsos a los diversos electrodos utilizados (electrodos de trabajo) cuando se sumergen en disoluciones de glifosato, y luego medir la corriente eléctrica que circula a través del electrodo. Este hecho se debe a que el flujo de la corriente a través de la disolución es una función del voltaje aplicado y la concentración de químicos en la disolución. En esta técnica las medidas no se utilizan para construir un voltagrama, sino que se procesan directamente los datos correspondientes a la evolución temporal de la corriente mediante técnicas multivariante.

2.2. Sistema electrónico.

Con el fin de llevar a cabo las pruebas utilizando voltametría de pulsos, se han preparado diez disoluciones diferentes de glifosato al 96% (adquirido de Molekula), que van desde $5 \cdot 10^{-3}$ M a $5 \cdot 10^{-5}$ M, utilizando un buffer de disolución de fosfato monosódico 0,1 M (Sharlau, CAS: 7558-80-7) en agua destilada, pH 6,7 que se consigue añadiendo unas gotas de NaOH 1M.

En la siguiente figura se muestran los distintos elementos necesarios para llevar a cabo un análisis mediante voltametría:

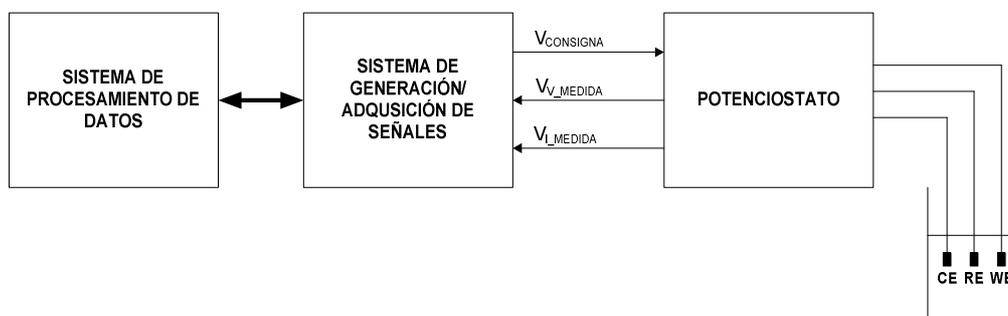


Figura 7-2. Diagrama de bloques de un sistema electrónico para medidas voltamétricas.

Para realizar las medidas se necesitan distintos elementos capaces de generar señales de tensión, aplicar estas señales a la celda electroquímica, recoger los datos de la corriente generada y almacenar dichos datos para su posterior tratamiento. El funcionamiento es el siguiente:

1. En primer lugar, el sistema de procesamiento de datos establece qué tipo de señales deben ser aplicadas a la celda electroquímica (constantes, pulsos, triangulares, etc.)
2. El sistema de generación y adquisición de señales establece la tensión analógica que el potencióstato debe aplicar entre el electrodo de referencia y el de trabajo (V_{CONSIGNA}).
3. El potencióstato es el encargado de controlar la tensión de la celda y la medida de la corriente. Mide las señales correspondientes a la tensión aplicada ($V_{\text{V_MEDIDA}}$) y a la corriente que circula por el electrodo de trabajo ($V_{\text{L_MEDIDA}}$). Por lo tanto, un potencióstato es un dispositivo electrónico que controla el potencial aplicado al electrodo de trabajo con respecto al electrodo de referencia y además mide la corriente que circula en el electrodo de trabajo.
4. El potencióstato transmite estas señales al sistema de generación y adquisición de señales para que las digitalice.
5. Por último, el sistema de procesamiento de datos recibe los datos digitalizados de la tensión y la corriente y los almacena, visualiza y/o procesa.

Las mediciones se han realizado con un dispositivo desarrollado por el GMH del IDM, llamado FraPlusMini [Masot et al. 2010], que es capaz de ejecutar las pruebas de voltametría de pulsos, voltametría cíclica y amperometría [Barat et al., 2008b][Campos et al., 2010b] y ha sido utilizado con éxito en el análisis de alimentos [Barat et al., 2008b], análisis de agua [García et al., 2011] y también en la determinación de los parámetros del medio ambiente [Ibañez et al., 2011].

El sistema consiste en una aplicación de software que se ejecuta en un PC y los equipos electrónicos necesarios. Para la voltametría de pulsos, el equipo genera una secuencia de hasta 50 pulsos con una amplitud que se puede configurar para cada uno de los pulsos dentro del rango de [-2V, 2V]. El ancho de los pulsos también puede ser configurado con un valor mínimo de 1 ms y un valor máximo de 50 ms.

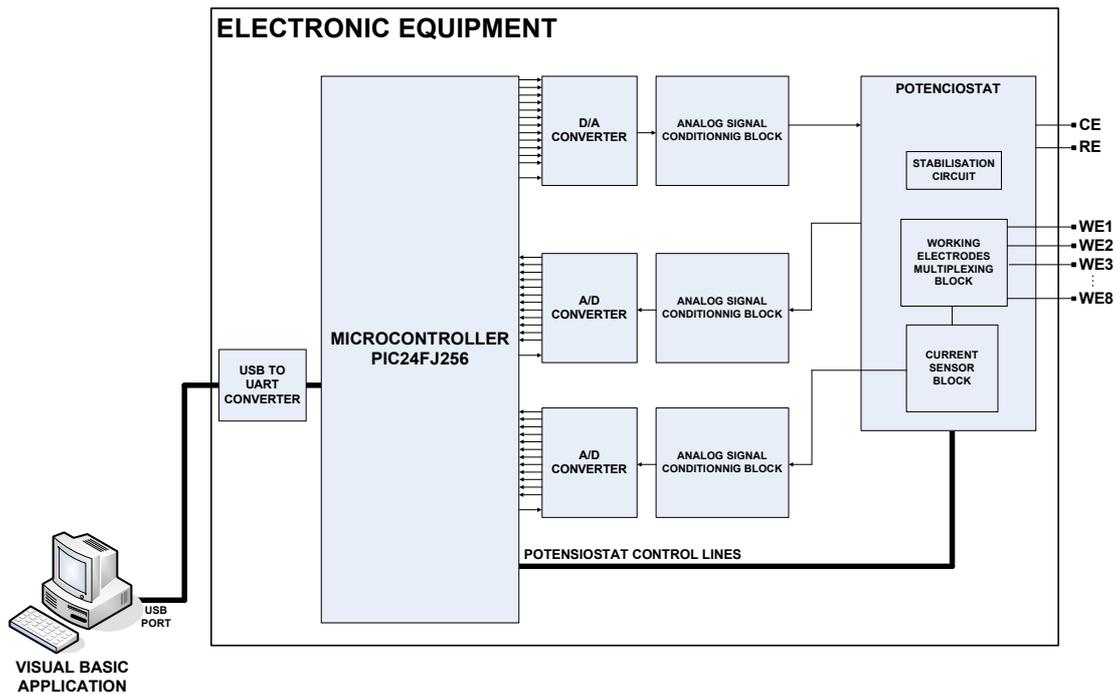


Figura 7-3. Diagrama de bloques del equipo electrónico diseñado.

El sistema incluye un potencióstat que aplica el voltaje al electrodo contador de la célula electroquímica y mide el voltaje en el electrodo de referencia y la corriente en los electrodos de trabajo. El potencióstat permite mediciones de hasta 8 electrodos de trabajo multiplexados, que pueden ser seleccionados automáticamente desde la aplicación de software.

Como los potencióstats son sistemas bastante inestables, el diseño incluye un circuito de estabilización que se pueden configurar de acuerdo a las necesidades de cada aplicación. El circuito de medición de corriente del potencióstat dispone de 32 escalas de corriente programable. La **Fig. 7-3** muestra el diagrama de bloques del sistema diseñado.

La aplicación de software incluye una sección para la voltamperometría de pulsos. En esta sección, se puede programar una secuencia de pruebas para varios electrodos de trabajo. La aplicación permite la configuración de los parámetros del tren de pulsos: el número de pulsos, la amplitud de cada pulso y el ancho de los pulsos. De esta manera, se puede configurar para cada electrodo de trabajo un patrón de pulsos diferentes. La escala actual y el nivel de estabilización también se pueden establecer. Una vez que la prueba se ha configurado, la aplicación envía esta información a los equipos electrónicos. La aplicación calcula 1.000 puntos correspondientes a la evolución temporal de la

señal del tren de impulsos y envía esta información (junto con el resto de los datos necesarios para llevar a cabo la prueba) a los equipos electrónicos a través de un puerto USB. Entonces, el equipo electrónico genera y aplica el tren de pulsos a la celda electroquímica, a las muestras de la tensión y señales de corriente. Por último, el equipo envía las muestras al PC donde se muestran los datos y se almacena. Este procedimiento se repite para el siguiente electrodo de trabajo de la secuencia. Los datos de la secuencia completa se almacena en un archivo para que se pueden aplicar los estudios estadísticos.

El equipo electrónico incluye un microcontrolador de 16 bits (PIC24FJ256), un convertidor digital a analógico de 12-bit (DAC), dos convertidores de 12 bits de analógico a digital (ADC) y un potencióstato que incorpora un circuito de medición de corriente, un electrodo de trabajo en el bloque de multiplexación y un circuito de estabilización. Algunos circuitos de acondicionamiento de señal analógica se utilizan para adaptar las señales que conectan el potencióstato al DAC y a los ADC.

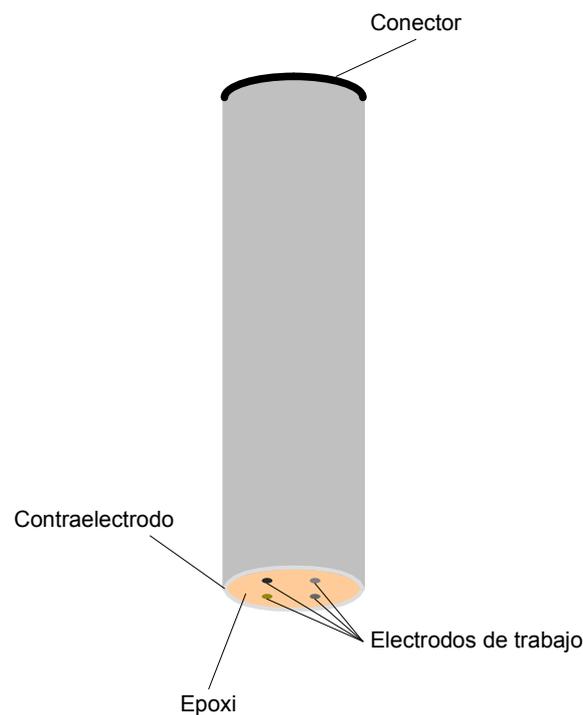


Figura 7-4. Partes de los electrodos utilizados en voltametría de pulsos.

El microcontrolador recibe los datos enviados por el PC. Cuando se reciben todos los datos correspondientes a una prueba, el microcontrolador configura la escala actual y el nivel de estabilización del potencióstato y se

selecciona el electrodo de trabajo deseado. A continuación, nos devuelve el valor correspondiente a la evolución temporal de la señal en el DAC a un ritmo que cumpla con los requisitos de tiempo de la señal. El mismo bucle del programa, muestrea las señales correspondientes a la tensión en el electrodo de referencia y a la corriente que circula a través del electrodo de trabajo seleccionado. Los datos recogidos son enviados al PC donde se procesa y se almacena.

En las **figuras 7-4 y 7-5** [Alcañiz, 2011] se muestran las partes de estos dos electrodos:



Figura 7-5. Electrodo de metales nobles y no nobles para medidas voltamperométricas.

2.3. Medidas.

Utilizando el dispositivo FraPlus y un tren de pulsos determinado se ha obtenido una importante base de datos de los análisis realizados con glifosato disuelto en tampón en diez diferentes concentraciones. Se han hecho ensayos preliminares con ocho electrodos de trabajo diferentes, realizados con metales nobles y no nobles, como se describió anteriormente. Se ha hecho el siguiente análisis: utilizando únicamente con los electrodos de rotación más sensibles a estas concentraciones de glifosato, debido a su mejor respuesta en comparación con los estáticos. Luego, se han hecho tres series diferentes de los ensayos (A, B y C) con el fin de tener un número estadísticamente significativo de muestras y poder comprobar su repetibilidad.

Después de considerar varios trenes de pulsos en los ensayos preliminares y la atención de trabajos anteriores en esta área [Sánchez *et al.*, 2010] [Chiu *et al.*, 2008] [Coutinho *et al.*, 2007], se ha decidido utilizar un tren de pulsos que se compone de nueve diferentes pulsos positivos y negativos en el rango de tensión [-500 mV, 600 mV] a fin de captar la respuesta de glifosato a diferentes voltajes. El tren de pulsos seleccionado que se aplica a las disoluciones de glifosato en cada ensayo se muestra en la Fig. 6-6.

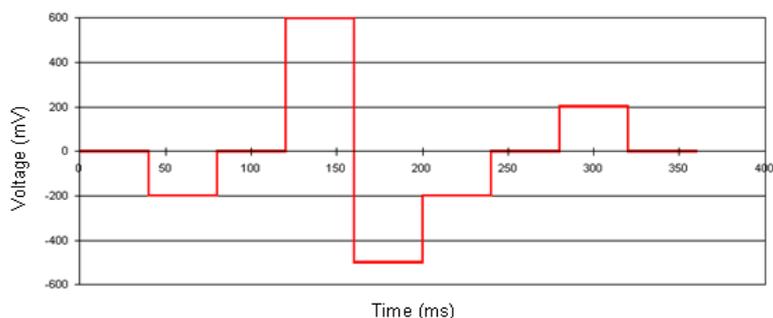


Figura 7-6. Pulsos aplicados para el análisis de las muestras de Glifosato.

Los resultados de la aplicación de este tren de pulsos a los diez diferentes concentraciones de glifosato seleccionados ($5 \cdot 10^{-3}$ M, $2.5 \cdot 10^{-3}$ M, $1.25 \cdot 10^{-3}$ M, $0.625 \cdot 10^{-4}$ M, $5 \cdot 10^{-4}$ M, $2.5 \cdot 10^{-4}$ M, $1.25 \cdot 10^{-4}$ M, $0.625 \cdot 10^{-4}$ M, $5 \cdot 10^{-5}$ M, $2.5 \cdot 10^{-5}$ M) son 999 medidas de voltamperometría correspondiente a la respuesta voltamperométrica de cada disolución a la aplicación de tensiones. Luego, al comparar los gráficos de los ocho diferentes electrodos y sus respuestas a las diferentes concentraciones de glifosato, se han seleccionados los electrodos de Co y Cu por su mayor sensibilidad a la presencia de glifosato. Después de decidir con qué electrodos vamos a trabajar, se han hecho los ensayos con los electrodos de rotación de Co y Cu. Finalmente, se ha construido la base de datos con los resultados, incluyendo los 999 datos de voltamperometría de 140 ensayos diferentes para el electrodo de Cu y 111 ensayos diferentes para el electrodo de Co.

En este sentido, se han obtenido varias tablas, como la que se muestra en la tabla 7.1, para cada serie de análisis (A, B y C).

Tabla 7.1. Un ejemplo de datos promediados en los ensayos de voltamperometría para cada concentración de glifosato en la serie A para el electrodo de cobalto.

Muestras analizadas		Medidas voltamperométricas (mA)					
Concentración de Glifosato (M)	Concentración de Glifosato (mg/l)	1	2	998	999
$5 \cdot 10^{-3}$	845.50	133.539	131.861			66.230	66.230
$2.5 \cdot 10^{-3}$	422.75	126.200	120.538			60.988	60.988
$1.25 \cdot 10^{-3}$	211.38	109.635	108.377			49.036	49.036
$0.625 \cdot 10^{-3}$	105.69	97.054	93.699			44.003	44.003
$5 \cdot 10^{-4}$	84.55	94.957	92.021			41.067	41.067
$2.5 \cdot 10^{-4}$	42.28	93.978	88.946			38.062	38.062
$1.25 \cdot 10^{-4}$	21.14	91.392	85.731			37.922	37.922
$0.625 \cdot 10^{-4}$	10.57	88.666	84.89			36.245	36.245
$5 \cdot 10^{-5}$	8.46	86.360	80.488			34.358	34.358
$2.5 \cdot 10^{-5}$	4.23	84.193	76.924			34.707	34.707
Buffer	0.0	85.101	81.747			36.664	36.664

Un ejemplo de las respuestas globales electroquímicas obtenidas se muestra en las Fig. 7-7 y 7-8, que corresponde con los resultados promedio de voltamperometría en la serie A de todos los pulsos y las diez concentraciones de glifosato analizado con los dos electrodos más sensibles (Co y Cu).

Por último, la base de datos de la respuesta voltamperométrica para las diez concentraciones de glifosato y nueve pulsos con los electrodos de cobalto y cobre han sido recogidas en un archivo de texto para ser modelada por dos procedimientos diferentes.

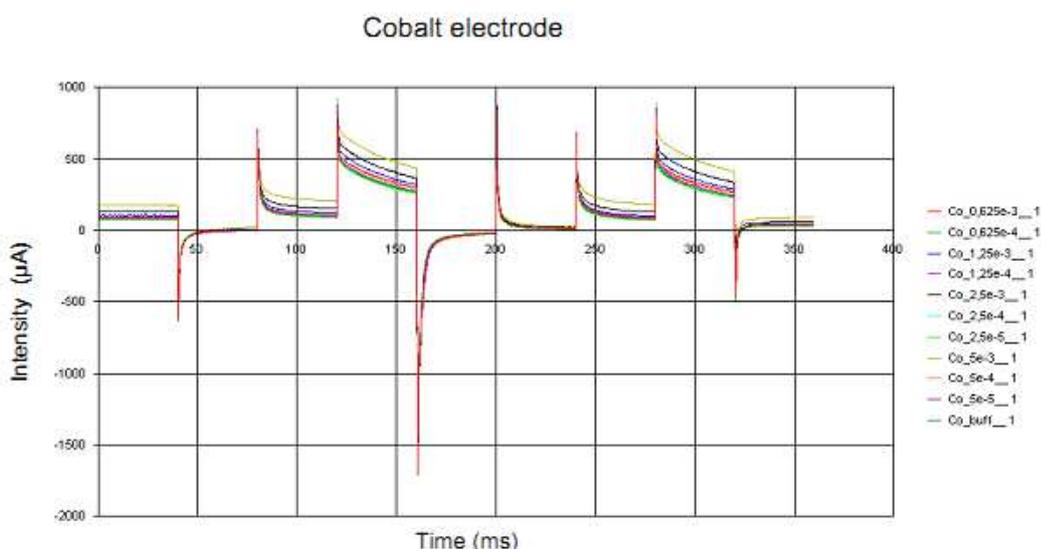


Figura 7-7. Respuesta voltamperométrica de diferentes concentraciones de glifosato con electrodos de cobalto.

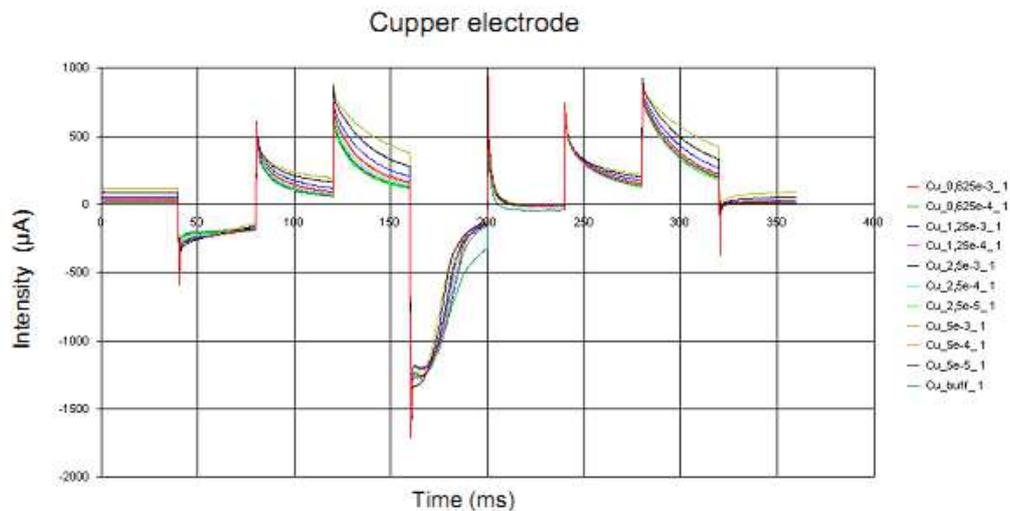


Figura 7-8. Respuesta voltamperométrica de diferentes concentraciones de glifosato con electrodos de cobre

3. Análisis de datos.

Al igual que en los capítulos precedentes, en este capítulo también se va a utilizar redes neuronales artificiales para intentar lograr una clasificación de las medidas, realizadas con técnicas voltamétricas, en función de la concentración de glifosato,

El procedimiento para trabajar con redes neuronales artificiales consta, tal como se ha comentado, de dos fases, una primera etapa de formación de la red y una segunda etapa para su verificación. La etapa de formación o capacitación se realiza con una parte de las medidas disponibles. En esta etapa, se han establecido las categorías de la red (en este caso, diez diferentes concentraciones de glifosato) y se aplican los datos procedentes de ocho electrodos para cada medición como un vector de entrada inicial que se ha simplificado más tarde. Con estos datos, se han calculado los coeficientes del algoritmo que configura la red. En la etapa de verificación, se han aplicado a las entradas los datos de las nuevas medidas, para comprobar si la salida de la red activa es correcta o no.

3.1. Entrenamiento de la red FF-MLP.

Con el fin de desarrollar el modelo, se ha utilizado la red neuronal artificial perceptrón multicapa con alimentación hacia adelante (FF-MLP). En el entrenamiento de la red neuronal, algunas de sus principales características se pueden cambiar, como son: el número de capas, el número de neuronas en la capa oculta, el tipo de algoritmo de entrenamiento, la función de transferencia, etc. El uso de una sola capa oculta y el algoritmo de entrenamiento [Martin, 2006] Levenberg-Marquardt es suficiente para la conexión de la red. El proceso de formación de la red consiste en cambiar el número de neuronas de la capa oculta, para así obtener la solución óptima. El proceso de formación de la red se puede hacer fácilmente con la interfaz de programas de MATLAB® GUI llamado NFtool (*neural network fitting tool graphical user interface*). Este programa trabaja con dos capas de neuronas, y utiliza la función de activación *Tansig* para las neuronas en la capa oculta y *Purelin* (función de transferencia lineal) para las neuronas de la capa de salida (Fig. 7-9).

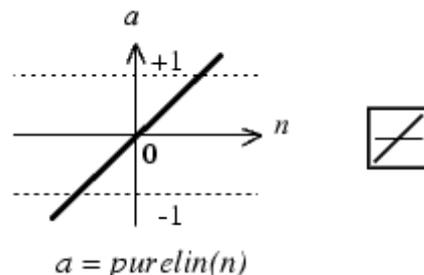


Figura 7-9. Función de transferencia lineal).

El método de entrenamiento es el denominado algoritmo de retropropagación de Levenberg-Marquardt. En ese programa, el usuario simplemente deberá cambiar el número de neuronas de la capa oculta, pero no el número de capas, tampoco se pueden cambiar las funciones pero sí el algoritmo de entrenamiento.

Se han realizado varios entrenamientos con el fin de obtener la mejor arquitectura de la red neuronal. Se ha cambiado el número de nodos y el algoritmo de entrenamiento.

La primera capa tiene una conexión de la entrada de la red, en este caso, tenemos cuatro entradas para el cobre, correspondiente a los impulsos

P3, P4, P8 y P9 y tres para cobalto correspondiente a los impulsos P2, P4 y P8.

Los resultados se van analizar de dos formas, primero utilizando solo los datos de los electrodos de cobre y posteriormente con los datos del electrodo de cobalto.

3.1.1. Resultados y discusión. Electrodo de cobre.

El mejor resultado se ha obtenido mediante la red Perceptrón Multicapa con alimentación hacia adelante (FF-MLP) con dos capas, 4-entradas con 127 datos correspondientes a las diferentes concentraciones en mg/l (Tabla 7.1) utilizadas en los ensayos, 10 neuronas ocultas y un nodo de salida.

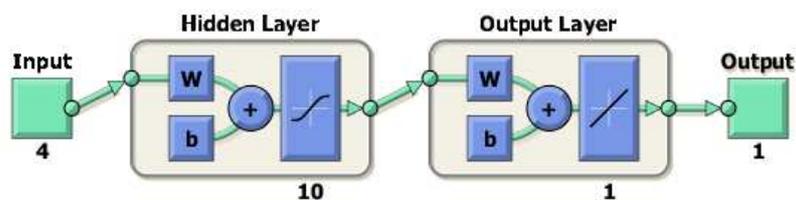


Figura 7-10. Modelo de redes neuronales de MATLAB®.

Con el fin de seleccionar las muestras para la formación de la red, el programa MATLAB® divide los datos aleatoriamente, el 70% de las muestras se usan para los datos de entrenamiento, el 15% para la validación de datos y el 15% restante se ha utilizado como datos de prueba. Después del entrenamiento, se obtienen dos matrices de pesos de la red neuronal: una matriz de 20×4 de pesos de la capa oculta y una matriz de 4×1 para los pesos de la capa de salida. Además, también se obtienen dos matrices de sesgo: una matriz de 20×1 correspondiente a los sesgos de la capa oculta y una matriz de sesgos de la capa de salida de 1×1 . También se obtienen el valor máximo y mínimo de los datos de entrada.

Los resultados del análisis de las redes neuronales con los datos de los electrodos de cobre se reflejan en la Tabla 7.2. En ella se indican el número de muestras utilizadas para cada una de las tres tareas (entrenamiento, validación y prueba), así como los valores de MSE (error cuadrático) y R (coeficiente de regresión) de los valores predichos frente a los medidos. Estos mismos valores estadísticos también quedan expuestos en la Fig. 7-11 donde se muestra la

gráfica de los valores y sobre ella se ha dibujado la recta de regresión y se ha indicado los valores de R para cada muestra (entrenamiento, validación y prueba), así como para todo el grupo de muestras.

Tabla 7.2. Número de muestras, MSE y valores de R para el modelo diseñado con los datos del electrodo de cobre			
	Nº Muestras	MSE	R
Entrenamiento	89	0.5658	0.99905
Validación	19	4.8256	0.99918
Prueba	19	20.1458	0.99911

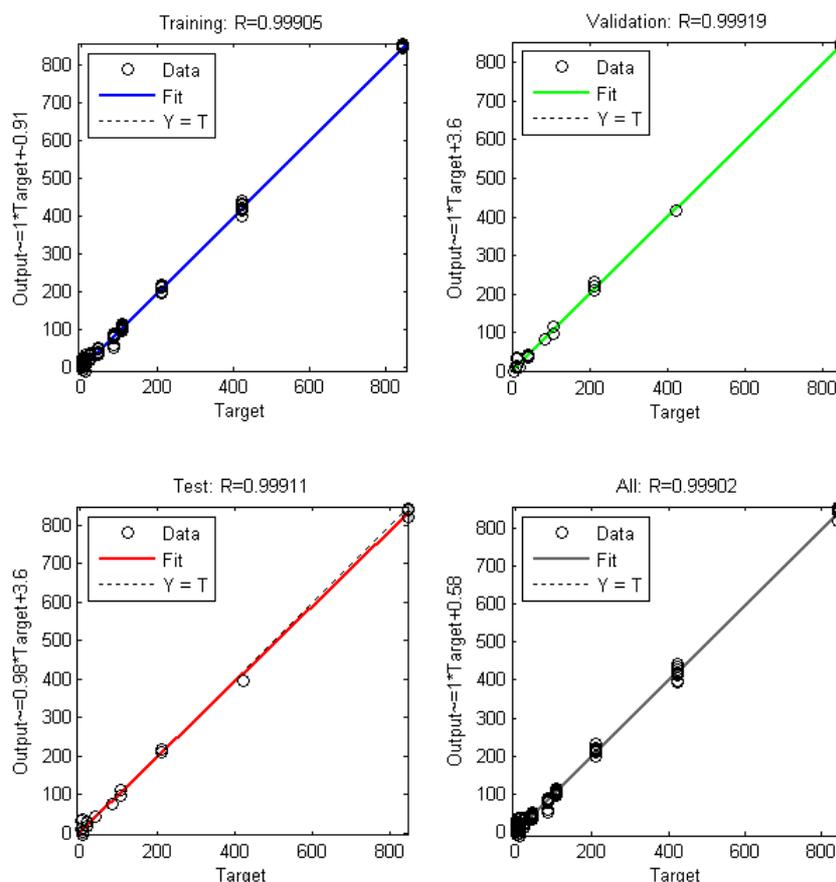


Figura 7-11. Curvas de regresión para el entrenamiento, validación, prueba y para todo el conjunto de datos con MATLAB® Fitting Tools.

Aunque la recta de regresión total da un valor $R=0.999$, que se puede considerar como muy bueno, también se observa en la gráfica (Fig. 7-11) una dispersión bastante elevada para valores de concentración baja, lo que indica que pueden haber bastantes valores con una desviación elevada. Para distinguir los resultados según la concentración, en las figuras 12, 13 y 14, se

muestra los valores predichos frente a los valores medidos para las concentraciones baja, media y alta.

Para las concentraciones bajas (Fig. 7-12) y media (Fig. 7-13) aparecen bastantes valores con errores apreciables, de los cuales algunos presentan una desviación incluso mayor que el valor de la concentración.

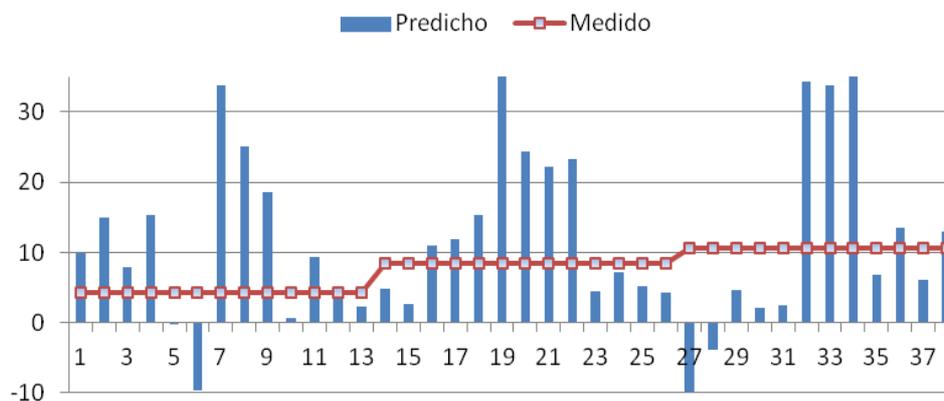


Figura 7-12.- Salidas con concentraciones bajas para el electrodo de cobre.

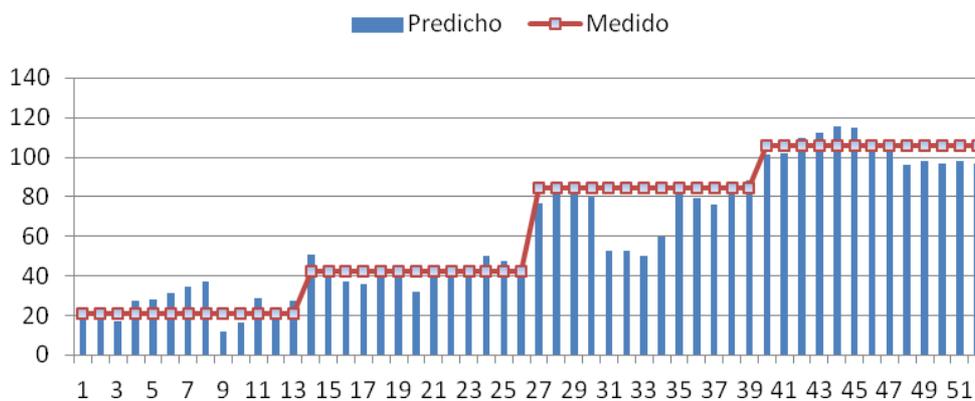


Figura 7-13.- Salidas con concentraciones medias para el electrodo de cobre.

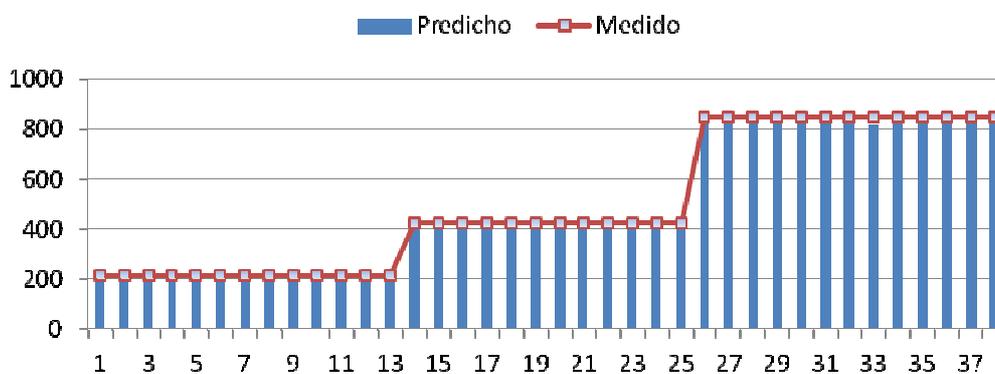


Figura 7-14.- Salidas con concentraciones altas para el electrodo de cobre.

Para la concentración alta (Fig. 7-14) aparecen tres o cuatro valores con muy poca desviación.

Al representar los valores medidos en comparación con los previstos por la red neuronal, se puede observar que hay una muy buena relación en concentraciones alta y media ($R = 0.9504$ para la concentración media y $R = 0.9992$ para la concentración alta), pero la relación es mala en baja concentración (Fig. 7-15).

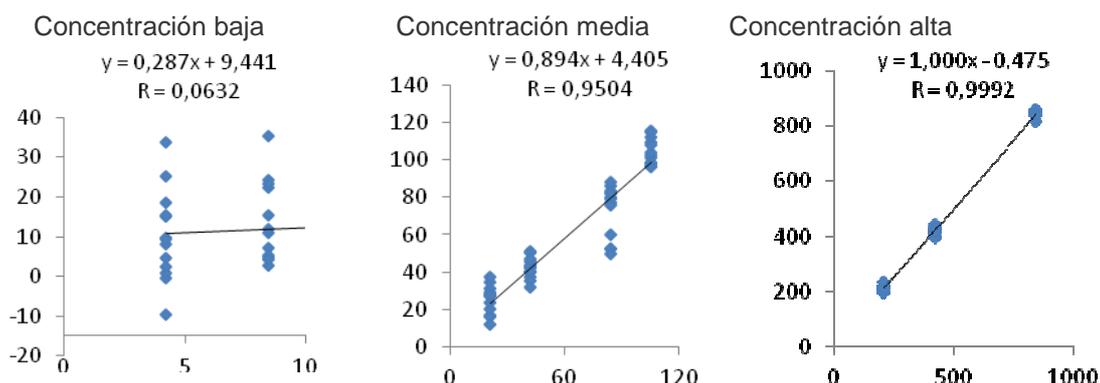


Figura 7-15. Resultados observados frente a los predichos para las concentraciones baja, media y alta de glifosato del modelo construido con RNA (MATLAB®Fitting Tools) para el electrodo de Cu.

3.1.2. Resultados y discusión. Electrodo de cobalto.

El segundo bloque de datos a analizar son los obtenidos con los electrodos de cobalto. También se realizaron diversos tanteos para clasificar las muestras con redes neuronales artificiales.

Se han realizado varios entrenamientos con el fin de obtener la mejor arquitectura de red neuronal. Se ha cambiado el número de capas, el número de nodos y las funciones de activación. El mejor resultado se ha obtenido mediante la red FF-MLP con dos capas, 3 de entrada, 101 valores diferentes de concentración en mg/l (tabla 7.1), 16 nodos ocultos y un nodo de salida (Fig. 7-16).

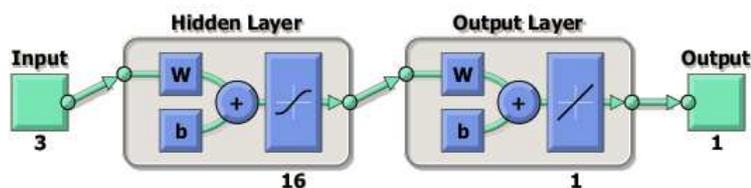


Figura 7-16. Modelo de redes neuronales de MATLAB®.

Con el fin de seleccionar las muestras para la formación de la red, el programa de MATLAB[®] divide los datos aleatoriamente, el 70% de las muestras se usan para los datos de entrenamiento, el 15% como la validación de datos y el 15% restante se ha utilizado como datos de prueba. Después del entrenamiento, se obtienen dos matrices de pesos: una matriz de 16x3 de pesos ocultos y una matriz de 3x1 para los pesos de salida. Además, también se obtienen dos matrices de sesgo: una matriz de 16x1 correspondiente a los sesgos ocultos y una matriz de sesgos de salida de 1x1. También se obtienen el valor máximo y mínimo de los datos de entrada.

Los valores de MSE (error cuadrático) y R (coeficiente de regresión) se muestran en la tabla 7.3. La **Fig. 7-17** muestra los valores de R para cada muestra (entrenamiento, validación y prueba), así como para todo el grupo de muestras correspondiente a los datos obtenidos con el electrodo de cobalto para todo el conjunto de datos, realizado con MATLAB[®] Fitting Tools.

Tabla 7.3. Número de muestras, MSE y valores de R para el modelo diseñado con todos los datos del electrodo de cobalto.			
	Muestras	MSE	R
Entrenamiento	71	2394.5517	0.97970
Validación	15	277.6353	0.99894
Prueba	15	995.2416	0.98894

Aunque la curva de regresión total da un valor $R=0.985$, muy bueno, se observa en la gráfica (**Fig. 7-17**) una dispersión bastante elevada para valores bajos de la concentración, lo que indica que pueden haber bastantes valores con una desviación elevada. La tabla 7.5 muestra los valores predichos frente a los valores medidos para las concentraciones baja, media y alta. Para las concentraciones baja (**Fig. 7-18**) y media (**Fig. 7-19**) aparecen bastantes valores con errores apreciables, de los cuales algunos presentan una desviación incluso mayor que el valor de la concentración. Para la concentración alta (**Fig. 7-20**) aparecen tres o cuatro valores con muy poca desviación.

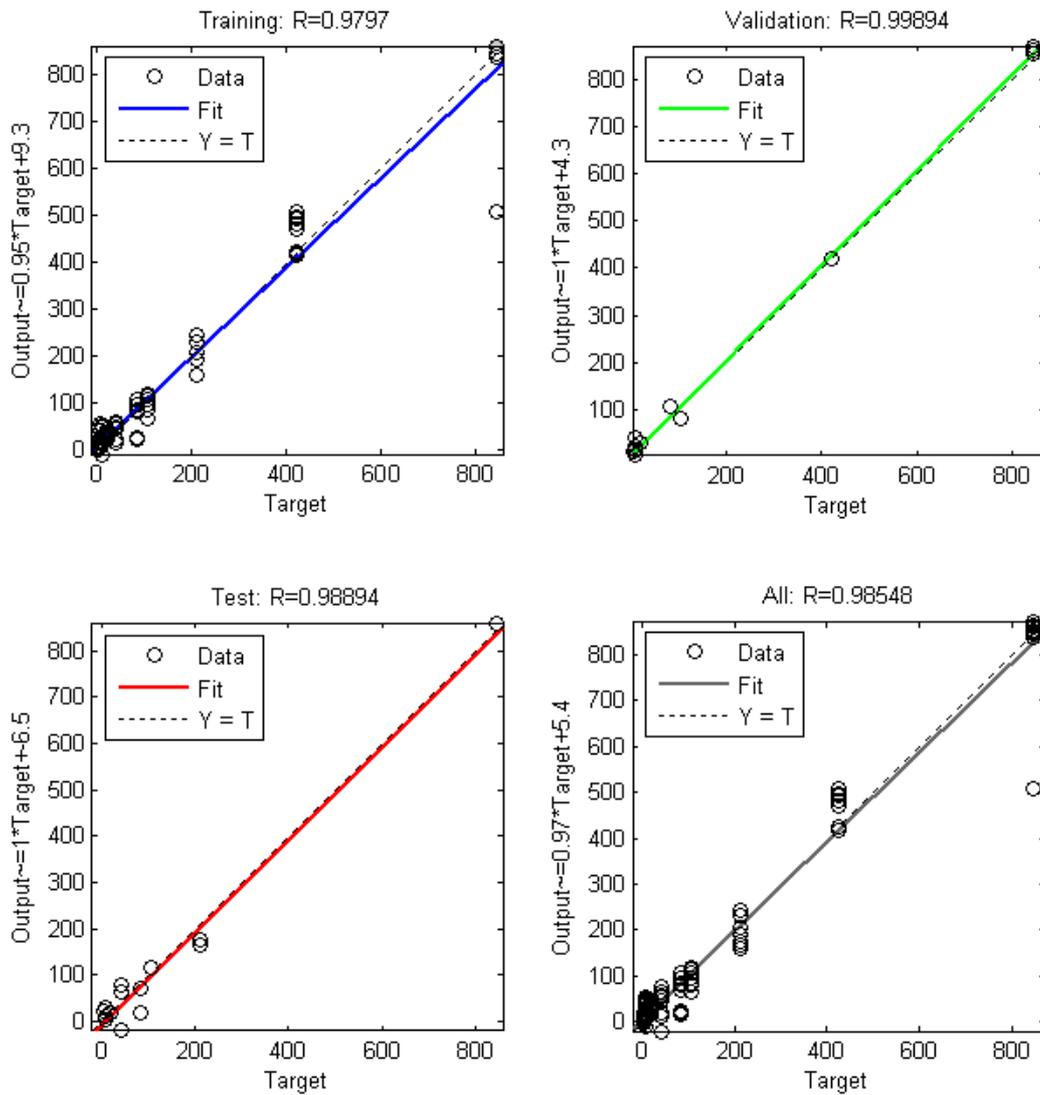


Figura 7-17. Curvas de regresión para el entrenamiento, validación, prueba y para todo el conjunto de datos con MATLAB® Fitting Tools.

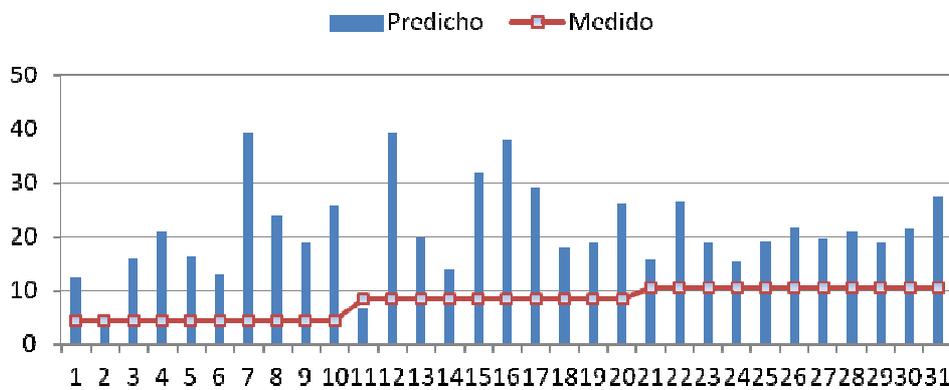


Figura 7-18. Valores de salida y medidos con concentración baja para el electrodo de cobalto.

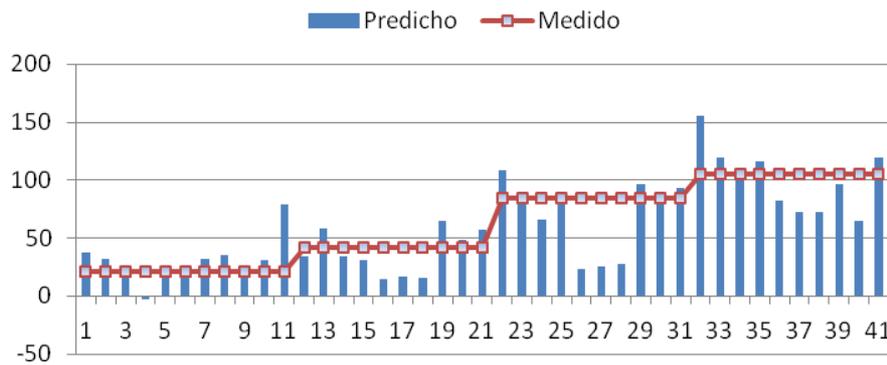


Figura 7-19. Valores de salida y medidos con concentración media para el electrodo de Co.

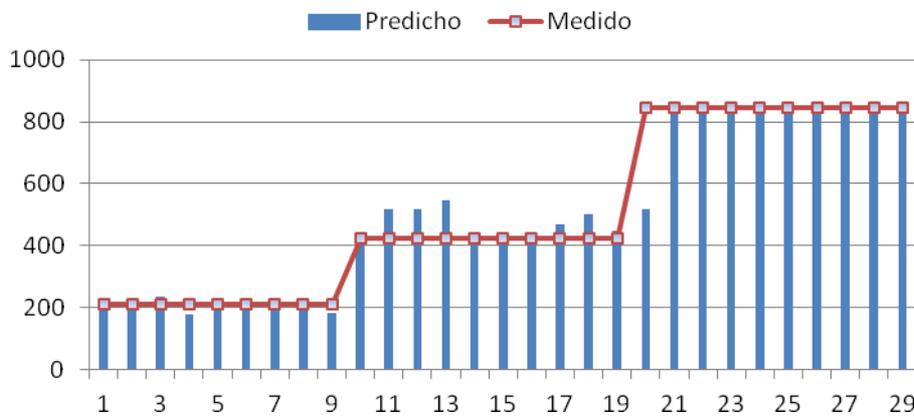


Figura 7-20. Valores de salida y medidos con concentración alta para el electrodo de Co

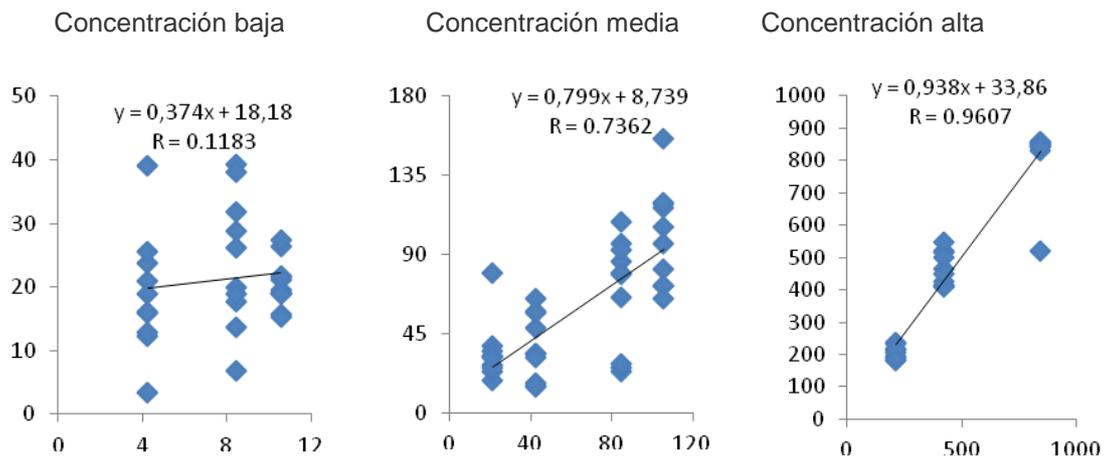


Figura 7-21. Resultados observados frente a los predichos para las concentraciones bajas, medias y altas de glifosato del modelo construido con RNA (MATLAB® Fitting Tools) para el electrodo de Co.

Al representar los valores medidos en comparación con los predichos por la red neuronal, se puede observar que hay una muy buena relación en

concentraciones altas ($R = 0.9607$), regular en medias ($R = 0.7362$), pero la relación es mala ($R=0.1183$) en bajas concentraciones (**Fig. 7-21**).

Dado que los resultados en bajas concentraciones no son los deseados, se decidió utilizar una fase previa de partición de los datos antes de la fase de modelado con la idea de mejorar el valor de R . En nuestro caso, hemos dividido los datos en tres clases diferentes: Clase 1 (concentraciones bajas), Clase 2 (concentración media) y la clase 3 (altas concentraciones) (Tabla 7.5). Esta clasificación se realizó mediante el uso de la red neuronal Fuzzy Artmap Simplificada (SFAM).

3.2. Entrenamiento de la red SFAM.

En el apartado anterior se ha entrenado una única red FF-MLP para hacer una predicción de la concentración de glifosato. La recta de regresión obtenida presenta un valor muy próximo a uno. Sin embargo como el rango de variación de la concentración es 200/1, resulta que una gran variación en el valor predicho con respecto al valor medido (Tabla 7.4) en concentraciones bajas afecta poco a la recta de regresión.

Como se explicó anteriormente, con el fin de capacitar a la red, se ha hecho una clasificación de los datos obtenidos en tres clases diferentes (baja concentración (Clase 1), concentración media (Clase 2) y una concentración alta (Clase 3), como se muestra en la Tabla 7.4.

Concentración (mg/l)	Clase
4.22750	1
8.45500	1
10.56875	1
21.13750	2
42.27500	2
84.55000	2
105.68750	2
211.37500	3
422.75000	3
845.50000	3

Una vez hecha la clasificación mediante la red SFAM se van a dividir los datos en tres grupos (uno por cada clase) y se va a proceder a realizar un nuevo entrenamiento con la red FF-MLP para cada una de las clases. En total se van a utilizar tres redes FF-MLP para la predicción y una red SFAM para hacer la clasificación previa.

3.2.1. Electrodo de cobre

Con objeto de llegar a un compromiso entre una tasa de aciertos alta y un mapfield bajo, los valores de ρ se han variado entre 0.3 y 0.9 y los valores de β entre 0.4 y 1.

Tabla 7.5. Mapfield función de ρ y β

ρ/β	0.4	0.5	0.6	0.7	0.8	0.9	1
0.3	21	100	8	7	7	7	7
0.4	21	100	9	8	8	7	8
0.5	21	100	8	7	7	7	8
0.6	21	100	9	7	7	7	8
0.7	21	100	10	10	9	9	9
0.8	26	100	12	12	10	10	11
0.9	31	100	18	21	19	21	17

Tabla 7.6. Tasa de aciertos (%) función de ρ y β

ρ/β	0.4	0.5	0.6	0.7	0.8	0.9	1
0.3	91.67	87.50	83.33	87.50	87.50	87.50	83.33
0.4	91.67	87.50	83.33	87.50	87.50	87.50	83.33
0.5	91.67	87.50	83.33	87.50	87.50	87.50	83.33
0.6	91.67	87.50	83.33	87.50	87.50	87.50	83.33
0.7	91.67	87.50	83.33	87.50	87.50	87.50	83.33
0.8	87.50	95.83	83.33	83.33	91.67	91.67	83.33
0.9	87.50	87.50	95.83	95.83	91.67	87.50	95.83

Tabla 7.7. Matriz de pesos para $\rho=0.9$ y $\beta=1$

0.001751	0.945480	0.000000	0.941579	0.000000	0.854236	0.000000	0.972623
0.040351	0.832862	0.044371	0.869149	0.045050	0.803259	0.019950	0.955843
0.469839	0.477941	0.339886	0.558968	0.303757	0.513740	0.184764	0.754140
0.000000	0.935612	0.040837	0.853596	0.146435	0.674891	0.026284	0.934305
0.535831	0.431500	0.418757	0.530030	0.470851	0.381479	0.202777	0.729039
0.095836	0.798346	0.119782	0.816952	0.148484	0.671336	0.031713	0.925927
0.043398	0.927136	0.104688	0.869201	0.221032	0.750278	0.056741	0.932252
0.723755	0.195686	0.601351	0.327412	0.546903	0.293298	0.425023	0.489815
0.701691	0.298309	0.572437	0.427563	0.485736	0.514264	0.403699	0.596301
0.269818	0.691549	0.185245	0.770200	0.158314	0.781326	0.065966	0.913039
0.056351	0.943649	0.141347	0.858653	0.343499	0.656501	0.063945	0.936055
0.085443	0.914557	0.091124	0.908876	0.198278	0.801722	0.032922	0.967078
0.293529	0.683130	0.253323	0.697810	0.272452	0.543919	0.090912	0.842736
0.871669	0.128331	0.792633	0.207367	0.670222	0.329778	0.724471	0.275529
0.914732	0.058335	0.888013	0.102194	0.843828	0.138125	0.880256	0.105476
1.000000	0.000000	1.000000	0.000000	1.000000	0.000000	1.000000	0.000000
0.040351	0.945849	0.093721	0.897402	0.174762	0.812852	0.039492	0.955843

A partir de los 114 datos de entradas, se han tomado 90 datos para entrenamiento y 24 para validación.

Un mapfield reducido (de 7 a 9) se obtiene para valores de ρ entre 0.3 y 0.6 y valores de β entre 0.6 y 1, logrando una tasa de aciertos del 87,5%. La tasa de aciertos máxima se obtiene para ρ de 0.9 y β de 0.6; 0.7 y 1, obteniéndose un tamaño del mapfield de 18, 21 y 17 respectivamente. El menor mapfield con una tasa de aciertos de 95.83% se obtiene para $\rho=0.9$ y $\beta=1$. Lo que da lugar a una matriz de pesos de 17x8.

Los datos necesarios para programar el microcontrolador son: los pesos (Tabla 7.7), el mapfield (Tabla 7.8), los valores máximos y mínimos (Tabla 7.9).

1	2	3	1	3	2	2	3	3	2	1	1	2	3	3	3	2
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

31277.76	15271.6
63234.99	26578.28
69449.79	40077.85
8663.01	153.68

En la matriz de confusión (Fig. 7-22) las celdas de la diagonal indican el número de respuestas clasificadas correctamente. Las celdas fuera de la diagonal indican el número de respuestas que fueron clasificadas erróneamente.

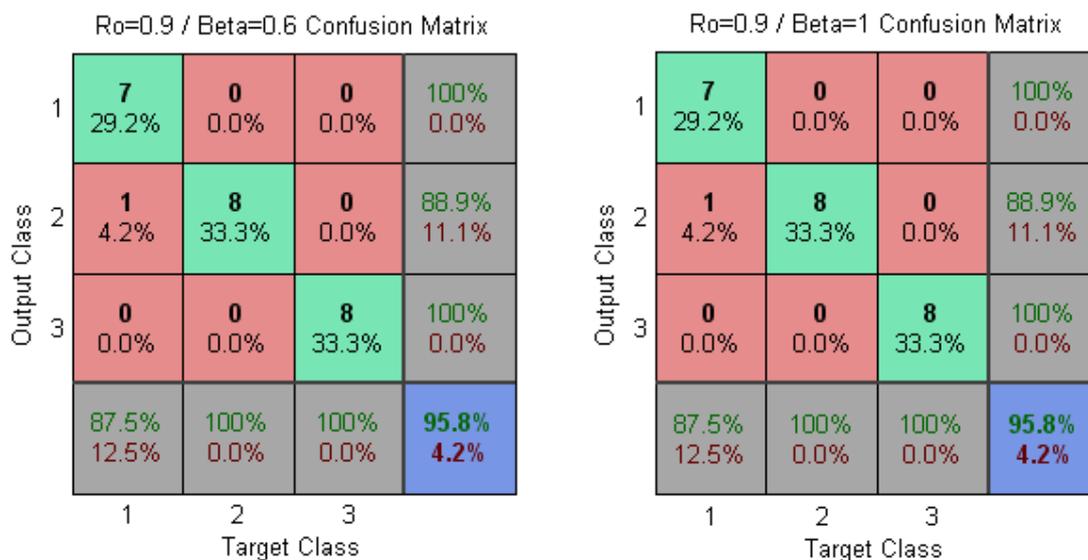


Figura 7-22. Matriz de confusión obtenida con la red Fuzzy ARTMAP.

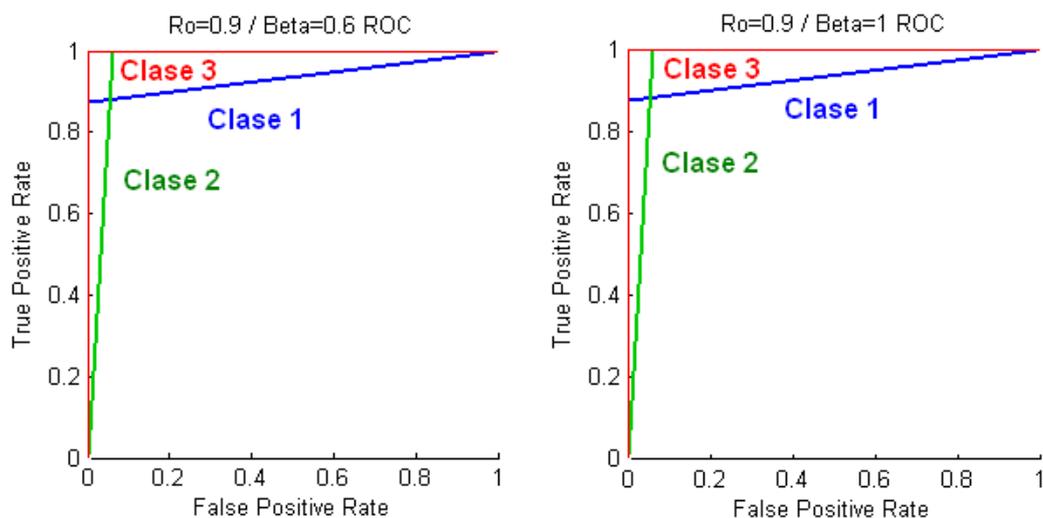


Figura 7-23. Curvas de características operativas del receptor(ROC).

La Fig. 7-23 muestra las curvas de características operativas del receptor (ROC), que es una medida utilizada para comprobar la calidad de la clasificación.

La tabla 7.10 muestra la clasificación obtenida de los 24 datos de validación para $\rho=0.9$ y $\beta=1$.

Tabla 7.10. Clasificación para $\rho=0,9$ y $\beta=1$		
1	2	3
1	2	3
1	2	3
1	2	3
1	2	3
1	2	3
1	2	3
2	2	3

3.2.2. Electrodo de cobalto

De los 101 datos de entradas se han tomado 74 datos para entrenamiento y 27 para validación. Los datos se han clasificado previamente en tres categorías, Baja, Media y Alta, clases 1, 2 y 3 respectivamente. En la tabla 7.6 se muestra la clasificación elegida en función de la concentración.

Al igual que con el electrodo de cobre, para hacer la clasificación previa se ha utilizado la GUI_FUZZY_ARTMAP (Interface Gráfica de Usuario),

programada en MATLAB. De esta forma se pueden obtener los pesos, mapfield y los máximos y mínimos de cada entrada, necesarios para la implementación en un microcontrolador.

ρ/β	0.4	0.5	0.6	0.7	0.8	0.9	1
0.3	12	9	10	12	10	10	10
0.4	12	9	10	12	10	11	11
0.5	16	16	14	17	13	14	12
0.6	100	100	100	17	18	19	14
0.7	18	18	17	100	21	17	14
0.8	100	100	27	28	25	21	22
0.9	100	100	100	49	38	34	34

ρ/β	0.4	0.5	0.6	0.7	0.8	0.9	1
0.3	62.96	55.56	62.96	62.96	55.56	51.85	59.26
0.4	62.96	55.56	62.96	62.96	48.15	48.15	59.26
0.5	66.67	59.26	55.56	66.67	51.85	74.07	81.48
0.6	66.67	74.07	74.07	70.37	59.26	70.37	77.78
0.7	74.07	74.07	66.67	74.07	77.78	74.07	59.26
0.8	77.78	66.67	59.26	74.07	70.37	70.37	74.07
0.9	37.04	62.96	81.48	55.56	59.26	48.15	74.07

La tasa de aciertos máxima se obtiene para ρ 0.5 y β 1, obteniéndose un tamaño del mapfield de 12, lo que da lugar a una matriz de pesos de 12×6 .

Los datos necesarios para programar el microcontrolador son: los pesos (Tabla 7.13), el mapfield (Tabla 7.14), los valores máximos y mínimos (Tabla 7.15).

0.175412	0.14618428	0.19924508	0.61936471	0	0.41691307
0.40458438	0.30857039	0.33516038	0.65721292	0.8344175	0
0	0.17159164	0	0.50303203	0.56129864	0.43362626
0.31128601	0.25195617	0.20396327	0.62008345	0.56319574	0.0017741
0.62791122	0.20643474	0.25023886	0.66286506	0.05617978	0.07332939
0.04197737	0.11539169	0.42427586	0	0.56551745	0.42857422
0.96497284	0	0.82271271	0.11265229	0.56991543	0.42958545
0.175412	0.25195617	0.19924508	0.61936471	0.25901833	0.43487581
0.64310037	0.35689963	0.33713494	0.66286506	0.92667061	0.07332939
0.19584822	0.38147843	0.42427586	0.50499869	0.56551745	0.43379007
0.39243307	0.25278459	0.22531213	0.70576318	0.56337907	0.41691307
0.27879042	0.72120958	0.43386921	0.56613079	0.56560024	0.43439976

1	1	2	2	2	2	3	3	1	2	3	1	2
---	---	---	---	---	---	---	---	---	---	---	---	---

869.2174	-649.3269
12615.678	-5083.3423
737	-954

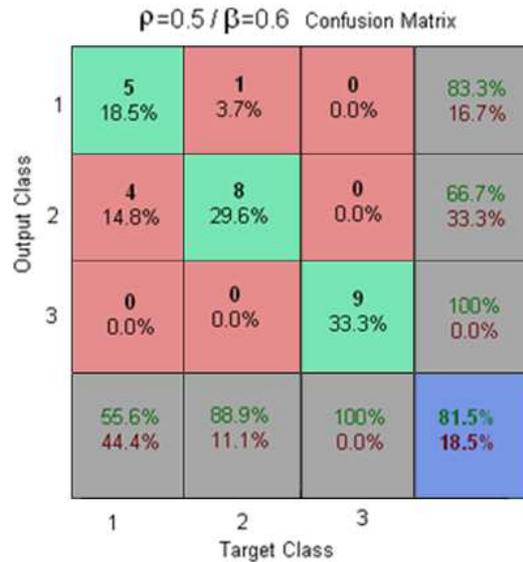


Figura 7-24. Matriz de confusión obtenida con la red Fuzzy ARTMAP.

La Fig. 7-24 muestra la matriz de confusión y la Fig. 7-25 muestra las curvas de características operativas del receptor.

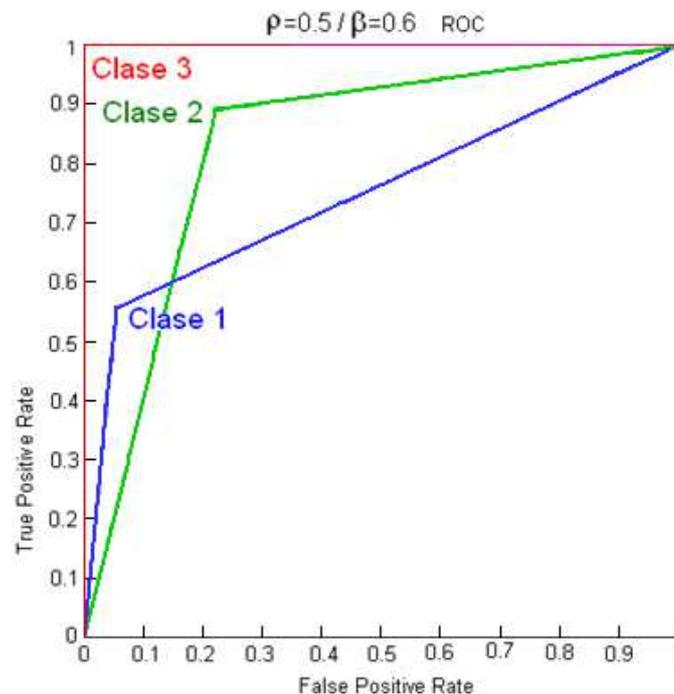


Figura 7-25. Curvas de características operativas del receptor (ROC).

La tabla 7.16 muestra la clasificación obtenida de los 27 datos de validación para $\rho=0.5$ y $\beta=1$.

Tabla 7.16. Clasificación para $\alpha=0,5$ y $\beta=1$

1	2	3
2	2	3
2	2	3
1	2	3
1	1	3
2	2	3
1	2	3
1	2	3
2	2	3

3.3. Entrenamiento de las tres redes FF-MLP. Electrodo de cobre.

Una vez clasificados los datos de entrada con la red SFAM se va a calcular la predicción de la concentración. Para ello, se ha empleado una red FF-MLP para cada una de las clases, por lo que se obtuvieron tres redes independientes para cada una de las concentraciones: baja, media y alta.

3.3.1. Concentración Baja.

La capa de entrada de la red neuronal para determinar los datos de la clase 1 (baja concentración) está formada por 4 entradas (pulsos P3, P4, P8 y P9), 16 neuronas en la capa oculta y una neurona en la capa de salida.

Con esta red se va a determinar si una nueva muestra es de esta clase o no lo es. Esta arquitectura da lugar a una matriz de pesos de entrada de 16×4 y sesgos de 16×1 . Los pesos de salida forman una matriz de 1×16 y los sesgos de 1×1 . Todos los valores de pesos y sesgos de la red neuronal para bajas concentraciones se indican en la tabla 7.17. Los datos necesarios para programar el microcontrolador son los pesos y sesgos que se pueden obtener utilizando los comandos NET del programa MATLAB.

Con los valores de los pesos y sesgos calculados se realiza un nuevo entrenamiento y validación de la red neuronal. Posteriormente se aplican los valores de prueba para determinar los valores predichos por la red y poder compararlos con los reales.

Capa oculta				Capa de salida		
Pesos			Sesgos	Pesos	Sesgos	
-1.90482	2.08649	-0.04269	-0.19907	2.67664	-0.02692	-0.44979
0.39143	1.58256	1.84495	-1.46335	-2.15239	1.09684	
-1.32607	-3.28525	2.77574	1.87954	1.42089	-2.64519	
1.44874	-2.81490	1.32243	-0.93430	-1.60382	-1.23965	
-2.38914	0.15167	0.55178	2.85798	1.47799	1.71454	
1.91402	-1.55886	-1.14610	0.91212	-1.01547	0.80578	
-0.33713	4.61508	-1.45551	-1.67022	-0.73596	-3.02955	
-3.77110	-1.39038	-0.62513	0.58501	-0.90007	-2.35212	
1.28426	2.37383	-0.42208	-1.64574	1.20858	0.40105	
2.12148	-0.77691	-1.70265	2.67649	0.67824	-1.79566	
-0.31952	2.13352	-0.29593	-1.82732	1.10595	0.52922	
-0.45492	-0.29954	1.00391	2.77418	-1.64017	1.37656	
-1.91427	2.04365	0.13849	-1.69759	-1.68431	-1.12142	
-2.38358	-0.94179	-1.16805	-1.86973	-1.72794	0.68796	
-1.37778	-1.90140	1.11053	1.24411	-2.35758	-0.44839	
1.97165	-1.07220	1.05829	-1.55480	2.52522	0.32492	

La Fig. 7-26 muestra la gráfica correspondiente a los valores predichos y medidos para concentraciones bajas.

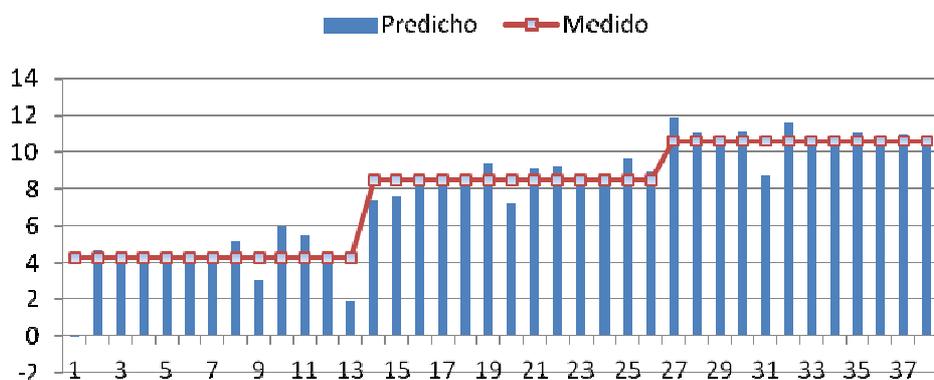


Figura 7-26. Valores predichos y medidos para concentraciones bajas (Cu clasificado).

En la Fig. 7-27 se representan los errores relativos en % de los valores predichos por la RNA para las concentraciones bajas sin clasificar y después de hacer la clasificación. En dicha figura se observa que la mayoría de los errores son inferiores a 20% una vez hecha la clasificación.

Los valores de MSE (Mean Squared Error) y R (regresión valor R) para el entrenamiento, validación y prueba se muestran en la Tabla 7.18.

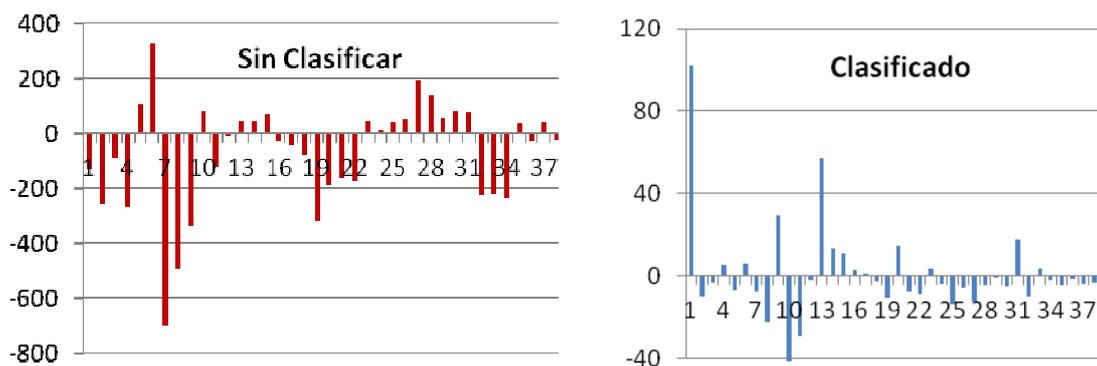


Figura 7-27. Errores relativos en % para las concentraciones bajas (Cu). Sin clasificar y clasificado.

La Fig. 7-28 muestra el valor de R para todo el grupo de muestras correspondientes a la concentración baja.

Tabla 7.18 .Número de muestras, MSEy valores deRpara el modelodiseñado con los datos deelectrodo de Cobre para la concentración baja			
	Nº Muestras	MSE	R
Entrenamiento	26	1,18154	0,80200
Validación	6	2,81879	0,99219
Prueba	6	11,49543	0,75389

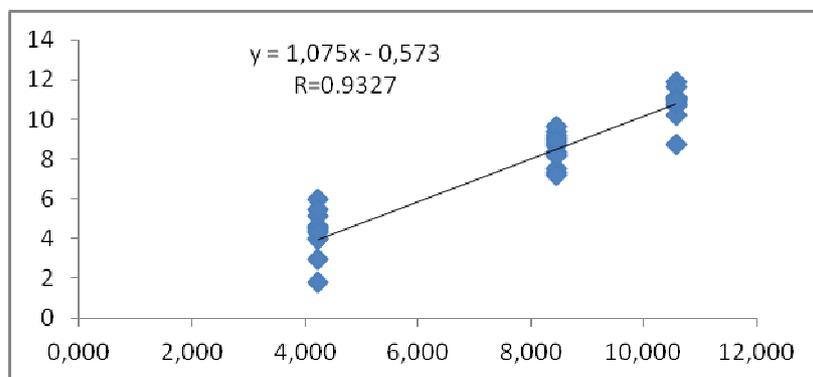


Figura 7-28 .Medidas y valores previstos para el rango de concentración baja de glifosato con la clasificación anterior (Clase 1: concentración baja).

El valor obtenido de R es 0.9327, es notablemente mejor que el obtenido sin clasificación de datos preliminares, como se muestra en la Fig. 7-15 (R = 0.0632) y confirma que la clasificación anterior da mejores resultados.

3.3.2. Concentración Media.

Para predecir las muestras de la clase 2 (concentraciones medias) el mejor resultado se obtuvo empleando una red con 10 neuronas en la capa de oculta y, lógicamente, una en la capa de salida. Esta estructura da lugar a una matriz de pesos de 10×4 y sesgos de 10×1 para la capa oculta. Los pesos de salida forman una matriz de 1×10 y los sesgos de 1×1 (Tabla 7.19). Estos datos son necesarios para programar el microcontrolador.

Capa oculta					Capa de salida	
Pesos				Sesgos	Pesos	Sesgos
-0.69026	0.00665	1.63363	0.62626	2.21865	-0.86501	0.71907
-1.37866	0.61747	-1.52021	2.37290	2.15921	-1.53229	
0.68295	-1.06119	-2.79535	0.10847	-2.17596	-0.03340	
-0.91834	0.79146	-2.59335	4.08072	0.93922	0.62928	
3.75223	1.13765	-2.64889	-2.06748	2.52812	-1.41444	
0.86283	1.85280	-0.74483	1.12830	-0.23702	0.46543	
0.45518	1.45730	-1.85670	-2.83195	-0.08861	0.02830	
-2.09522	-1.84219	-1.49540	0.78228	-2.14261	-0.23701	
-2.53311	-1.28303	1.29444	0.84136	0.56035	0.10840	
-2.23862	-1.52367	2.89752	-0.52721	-3.42669	-2.77407	

La Fig. 7-29 muestra la gráfica correspondiente a los valores predichos y medidos para las concentraciones medias con el electrodo de cobre una vez hecha la clasificación.

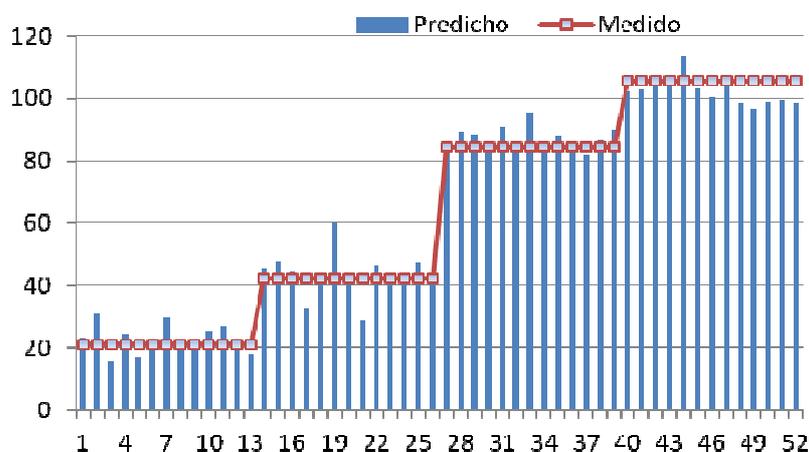


Figura 7-29. Valores predichos y medidos para concentraciones medias (Cu clasificado).

En la Fig. 7-30 se representan los errores relativos de los valores predichos por la RNA para las concentraciones medias sin clasificar y después de hacer la clasificación. En dicha figura se observa que la mayoría de los errores han disminuido notablemente una vez hecha la clasificación.

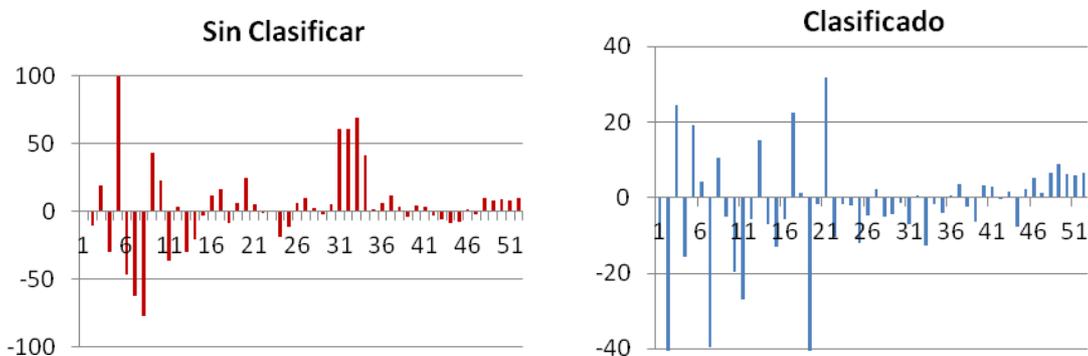


Figura 7-30. Errores relativos en % para la concentración media (Cu). Sin clasificar y clasificado.

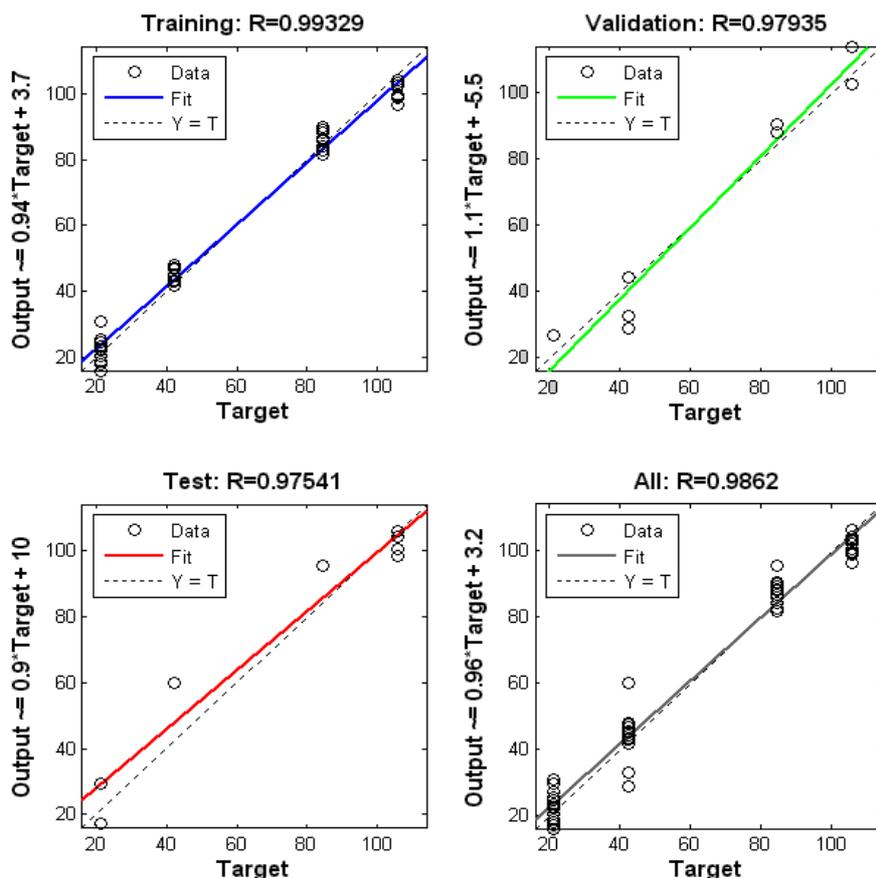


Figura 7-31. Curvas de regresión para el entrenamiento, validación, prueba y para todo el conjunto de datos con MATLAB® Fitting Tools, del electrodo de Cobre para la concentración media

Los valores de MSE (*Mean Squared Error*) y R (coeficiente de regresión) para el entrenamiento, validación y prueba se muestran en la Tabla 7.20. La Fig. 7-31 muestra el valor de R para todas las muestras (entrenamiento, validación y prueba, así como para todo el grupo de muestras.

	Nº Muestras	MSE	R
Entrenamiento	36	16.41289	0.99329
Validación	8	53.4744	0.97935
Prueba	8	75.1912	0.97541

En este caso, el coeficiente de regresión es 0.9862, que es mucho mejor que en el caso anterior, sin ninguna clasificación (0.9504, Fig. 7-15), para las clase 2.

3.3.3. Concentración Alta.

Para predecir las muestras de la clase 3 (alta concentración) el mejor resultado se obtuvo empleando una red neuronal con 10 neuronas en la capa de oculta y una en la capa de salida. Lo que da lugar a una matriz de pesos de 10x4 y sesgos de 10x1. Los pesos de salida forman una matriz de 1x10 y los sesgos de 1x1 (tabla 7.21). Estos datos son necesarios para programar el microcontrolador.

Capa oculta				Capa de salida	
Pesos			Sesgos	Pesos	Sesgos
1.13256	-1.68410	0.99732	0.91119	-2.86338	-0.57154
-0.37244	0.92081	-0.64273	-2.64732	-2.32614	-0.28967
-1.26123	2.11425	1.00554	-0.24494	1.03731	-0.39870
0.26538	-1.38263	2.93663	-2.89127	-1.40530	0.37987
-1.74986	0.92333	0.84020	-3.63533	-5.11593	0.41222
0.31655	-3.38171	1.73409	-0.28125	-0.84252	-1.02989
-0.95876	-1.32039	3.29015	0.31750	-3.27266	0.61336
0.31489	2.35494	-0.70407	-0.94560	1.38400	-0.62543
0.50867	1.42876	0.08341	4.57039	-0.47126	0.60988
1.93252	1.18478	-1.66407	-1.81636	2.99771	1.05459

Predicho (mg/l)	Medido (mg/l)	Predicho (mg/l)	Medido (mg/l)	Predicho (mg/l)	Medido (mg/l)
211.8390	211.3750	412.6098	422.7500	845.5180	845.5000
211.4361	211.3750	431.8459	422.7500	845.2016	845.5000
211.9253	211.3750	422.9751	422.7500	844.7593	845.5000
211.3405	211.3750	423.2632	422.7500	845.8013	845.5000
210.0861	211.3750	423.9962	422.7500	845.3240	845.5000
204.4693	211.3750	422.1055	422.7500	846.3141	845.5000
212.6388	211.3750	428.1822	422.7500	845.5696	845.5000
211.8649	211.3750	421.6664	422.7500	845.7334	845.5000
211.4509	211.3750	433.5333	422.7500	844.7394	845.5000
211.9979	211.3750	422.9108	422.7500	847.8216	845.5000
211.1369	211.3750	418.1855	422.7500	845.6797	845.5000
212.1202	211.3750	422.3068	422.7500	844.4171	845.5000
213.8830	211.3750			844.4176	845,5000

La tabla 7.22 muestra los valores predichos y medidos mientras que la Fig. 7-32 muestra la gráfica correspondiente a dichos valores.

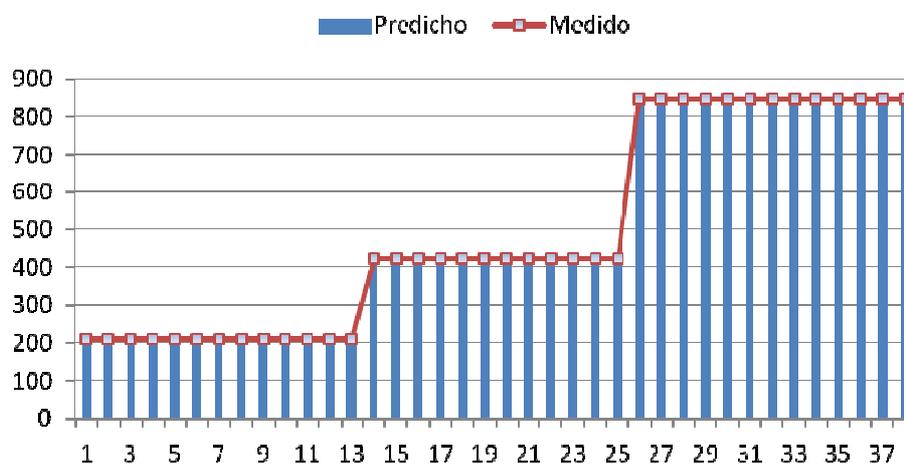


Figura 7-32. Valores predichos y medidos para concentraciones altas (Cu clasificado).

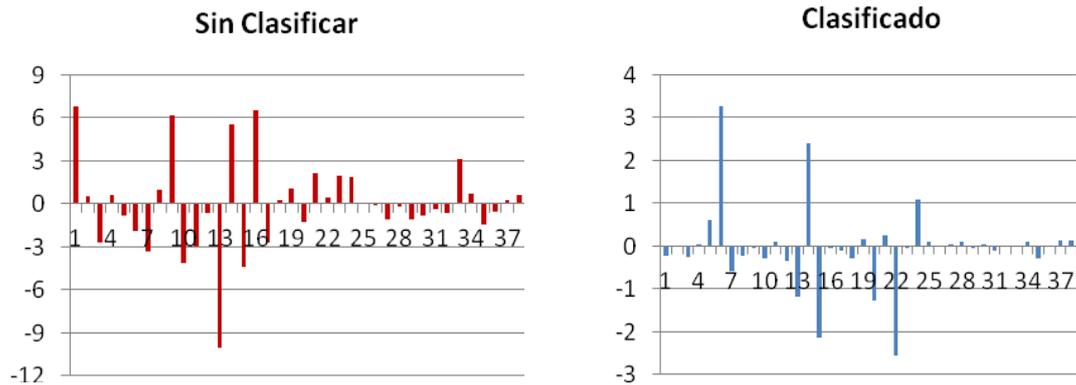


Figura 7-33. Errores absolutos para la concentración alta (Cu). Sin clasificar y clasificado.

En la Fig. 7-33 se representan los errores relativos de los valores predichos por la RNA para las concentraciones altas sin clasificar y después de hacer la clasificación. En dicha figura se observa que la mayoría de los errores han disminuido más de un 60% una vez hecha la clasificación.

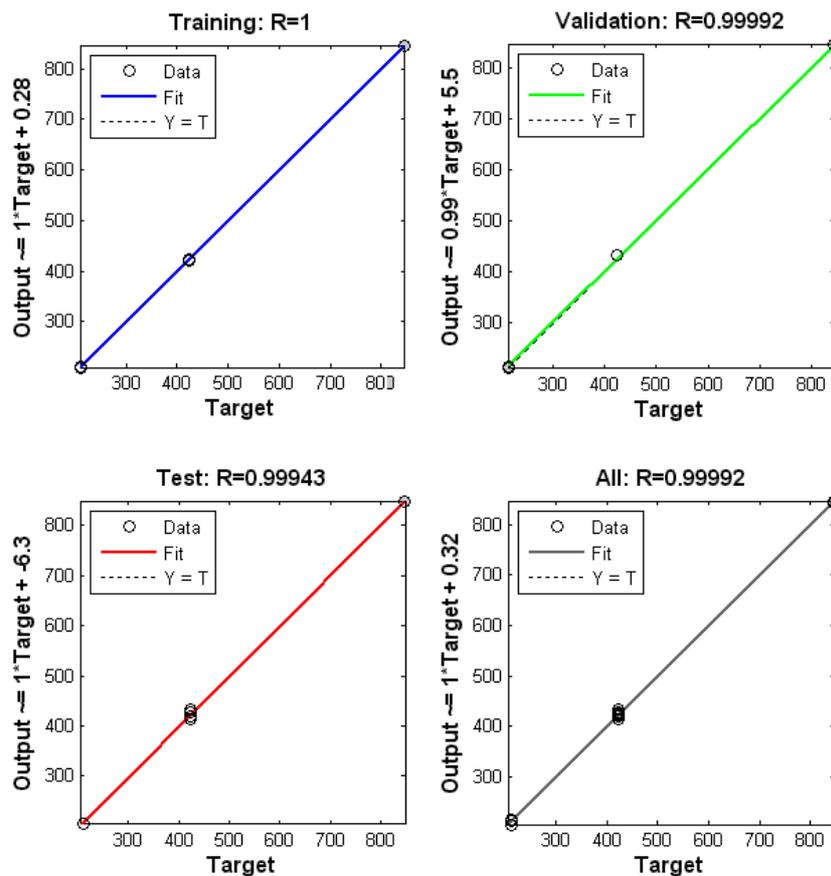


Figura 7-34. Curvas de regresión para el entrenamiento, validación, prueba y para todo el conjunto de datos con MATLAB® Fitting Tools, del electrodo de Cu para la concentración alta.

La **Fig. 7-34** muestra el valor de R para todas las muestras (entrenamiento, validación y prueba), así como para todo el grupo de muestras correspondiente a los datos obtenidos con el electrodo de cobre par la concentración alta, realizado con MATLAB® Fitting Tools. Los valores de MSE (*Mean Squared Error*) y R (coeficiente de regresión) para el entrenamiento, validación y prueba se muestran en la Tabla 7.23.

	Nº Muestras	MSE	R
Entrenamiento	25	0.40838	0.999997
Validación	6	20.8845	0.999918
Prueba	6	48.1633	0.999427

En este caso, el coeficiente de regresión es 0,99992, que es mejor que el correspondiente al caso anterior sin ninguna clasificación (0,9992, **Fig. 7-15**) para la clase 3.

	Clase baja	Clase media	Clase alta
Sin clasificar	0.063	0.9504	0.9992
Clasificado	0.9327	0.9862	0.99992

3.4. Entrenamiento de las tres redes FF-MLP. Electrodo de cobalto.

Al igual que con el electrodo de cobre, una vez clasificados los datos de entrada con la red SFAM se va a calcular la predicción de la concentración, empleando una red FF-MLP por cada una de las clases.

3.4.1. Concentración baja.

Para las concentraciones bajas se ha empleado una red neuronal con 3 entradas, 16 neuronas en la capa oculta y un nodo de salida. Lo que da lugar a una matriz de pesos de entrada de 16x3 y sesgos de 16x1. Los pesos de salida forman una matriz de 1x16 y los sesgos de 1x1 (Tabla 7.25).

Tabla 7.25. Pesos y sesgos de la capa oculta y de salida para la concentración baja					
Capa oculta			Capa de salida		
Pesos		Sesgos	Pesos	Sesgos	
-1.29225	2.45146	2.17856	3.70107	-0.45617	-0.11161
2.62034	-1.39478	1.78519	-3.13431	0.21155	
-2.59572	2.38787	-0.38115	2.50033	0.34456	
-2.02733	2.33184	-1.73580	2.15684	-0.82728	
2.31226	1.91848	-2.01558	-1.42975	-0.87785	
-2.27804	0.28630	-2.64494	1.81116	-0.71408	
2.55880	1.86456	2.64662	-0.83595	-0.73866	
-0.05163	-2.58305	-1.11967	0.05316	0.01248	
-3.33412	-0.59422	-0.40841	0.18931	-0.62538	
2.80509	-1.89736	1.23563	0.43348	-0.19162	
3.26086	1.29711	2.36135	1.57989	0.28738	
0.98753	-3.41281	-0.92329	1.39854	-0.62175	
-0.61329	-3.24703	1.31157	-1.74307	-0.51024	
-2.15041	1.92100	2.41658	-2.88326	-0.94439	
-1.50403	2.78704	1.63216	-3.16154	0.51622	
0.05485	-1.24693	3.22241	3.68493	0.19329	

La Fig. 7-35 muestra la gráfica correspondiente a los valores predichos y medidos para las concentraciones bajas correspondientes al electrodo de cobalto con clasificación previa.

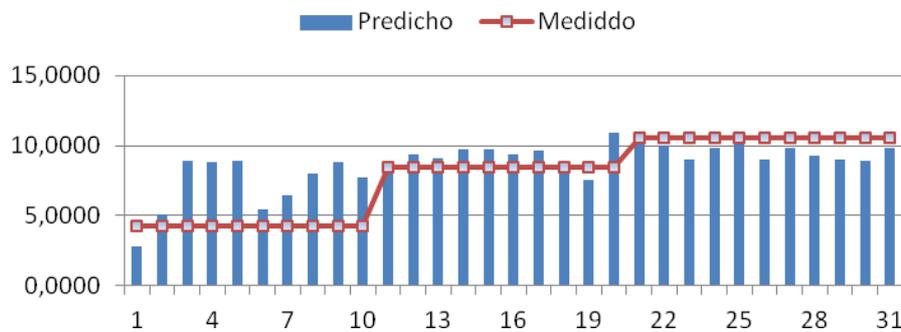


Figura 7-35. Valores predichos y medidos para concentraciones bajas (Co clasificado).

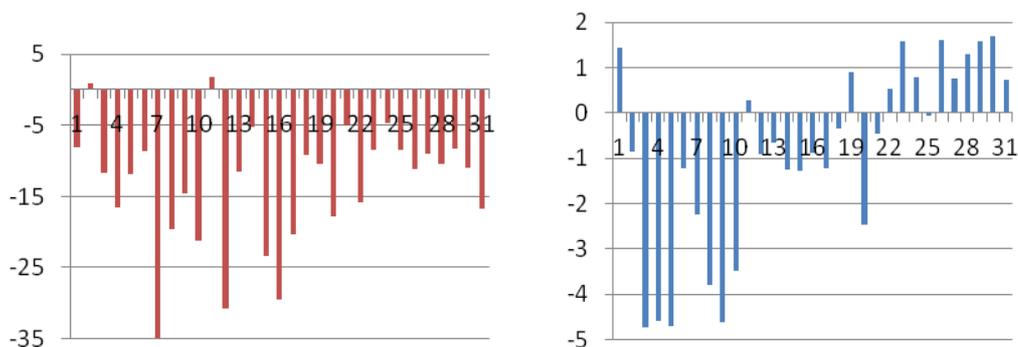


Figura 7-36. Errores absolutos para las concentraciones bajas (Co). Sin clasificar y clasificado.

En la **Fig. 7-36** se representan los errores absolutos de los valores predichos por la RNA para las concentraciones bajas sin clasificar y después de hacer la clasificación. En dicha figura se observa que la mayoría de los errores han disminuido notablemente una vez hecha la clasificación.

Los valores de MSE (*Mean Squared Error*) y R (regresión valor R) para el entrenamiento, validación y prueba se muestran en la Tabla 7.26.

Tabla 7.26. Número de muestras, MSE y valores de R para el modelo diseñado con los datos del electrodo de Cobalto para la concentración baja.			
	Nº Muestras	MSE	R
Entrenamiento	25	3.66334	0.709930
Validación	6	1.59499	0.846670
Prueba	6	13.10295	0.867374

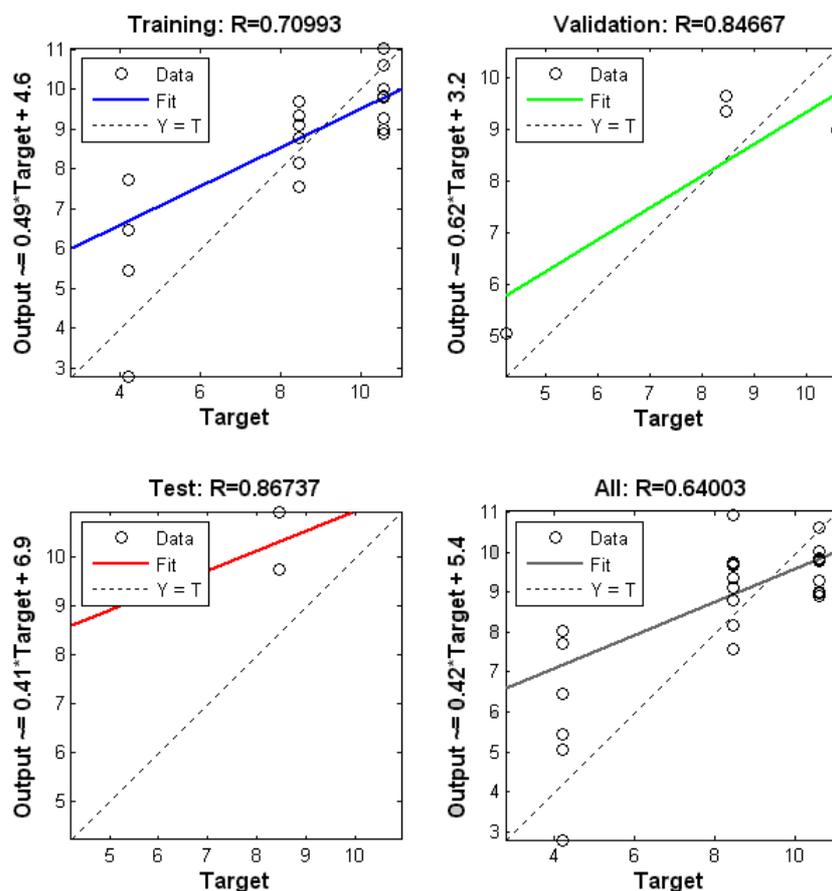


Figura 7-37. Curvas de regresión para el entrenamiento, validación, prueba y para todo el conjunto de datos con MATLAB® Fitting Tools del electrodo de Cobalto para la concentración baja.

La **Fig. 7-37** muestra el valor de R para todas las muestras (entrenamiento, validación y prueba, así como para todo el grupo de muestras).

En este caso, el coeficiente de regresión es 0.64003, que es mucho mejor que en el caso anterior sin ninguna clasificación (0.1183, **Fig. 7-21**), para la clase 1.

3.4.2. Concentración media.

Para las concentraciones medias se ha empleado una red neuronal con 3 entradas, 16 neuronas en la capa oculta y un nodo de salida. Lo que da lugar a una matriz de pesos de entrada de 16×3 y sesgos de 16×1 . Los pesos de salida forman una matriz de 1×16 y los sesgos de 1×1 . Los pesos y sesgos para la capa oculta y capa de salida se muestran en la tabla 7.27.

Tabla 7.27. Pesos y sesgos para la capa oculta y capa de salida					
Capa oculta				Capa de salida	
Pesos			Sesgos	Pesos	Sesgos
2.12895	2.55149	3.49622	-2.67267	1.77257	-0.15619
-1.60247	-3.32788	-0.87696	3.00194	0.37588	
0.90756	0.86839	-2.76608	-3.33102	0.02293	
-2.12266	2.98136	-0.99420	2.29368	0.92947	
-3.26291	0.90013	-0.55877	2.01950	-0.50779	
1.58871	2.85298	-1.49735	-0.79709	-0.84541	
-2.37775	2.28710	0.10061	0.93024	1.39151	
-0.35619	5.04493	0.35614	-0.63815	-2.15126	
-1.74581	-2.50892	-1.31291	-0.11445	-0.46966	
-1.97435	-0.57391	-2.84318	-0.99428	0.41443	
2.50934	-2.24549	-1.22327	1.94038	-0.77992	
3.01581	-2.29902	0.28031	1.19526	0.76955	
-2.16186	-2.22401	-1.20213	-2.51155	-0.60221	
2.21035	2.34258	1.34193	2.47023	0.12801	
2.76654	0.88957	0.00417	3.71221	0.21655	
2.38973	2.50157	-0.45464	3.58830	-0.83721	

La **Fig. 7-38** muestra la gráfica correspondiente a los valores predichos y medidos para las concentraciones medias.

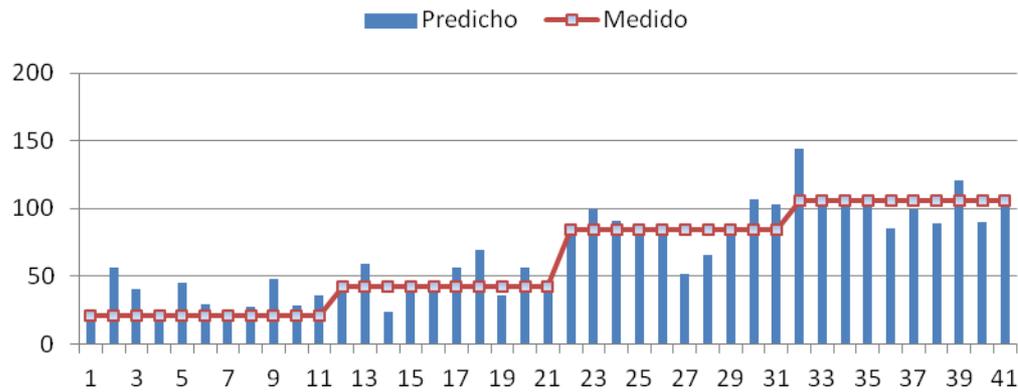


Figura 7-38. Salidas con concentraciones medias para el electrodo de cobalto.

En la Fig. 7-39 se representan los errores relativos en % de los valores predichos por la RNA para las concentraciones medias sin clasificar y después de hacer la clasificación. En dicha figura se observa que la mayoría de los errores han disminuido a la mitad una vez hecha la clasificación.

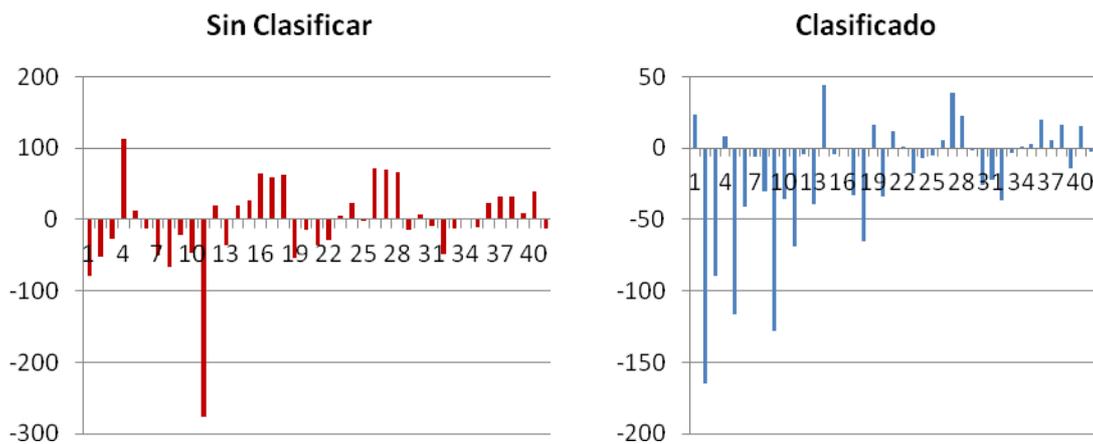


Figura 7-39. Errores relativos en % para la concentración media (C_0). Sin clasificar y clasificado.

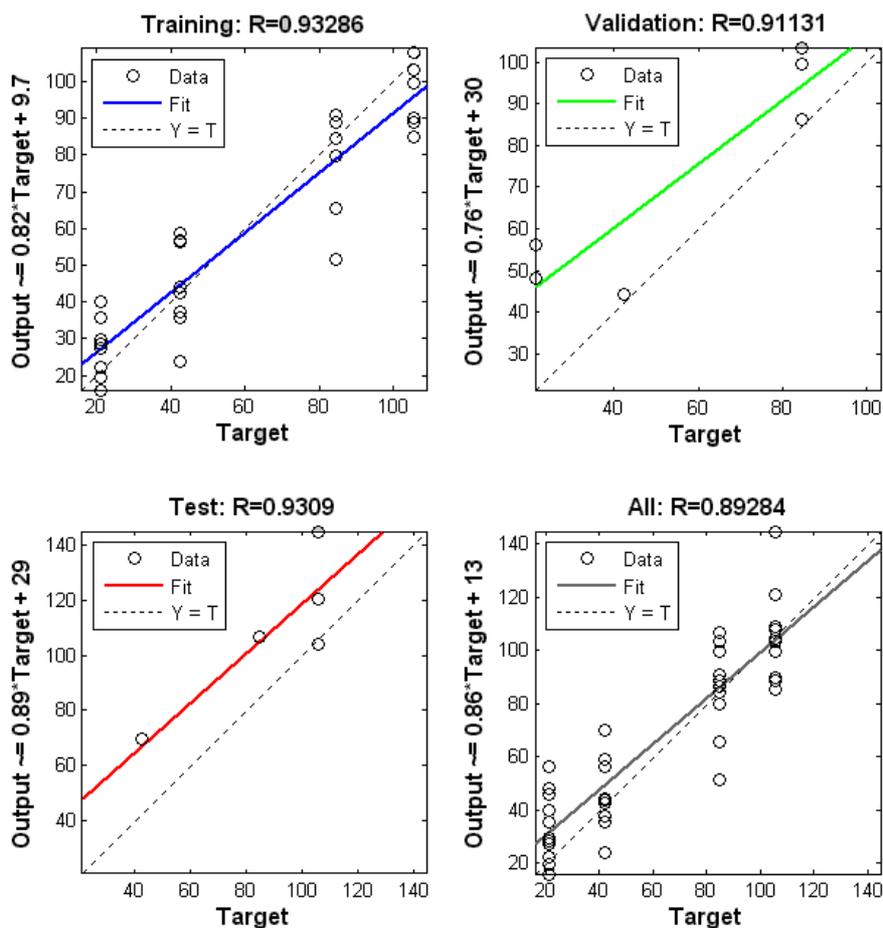


Figura 7-40. Curvas de regresión para el entrenamiento, validación, prueba y para todo el conjunto de datos con MATLAB® Fitting Tools del electrodo de Cobalto para la concentración media.

Tabla 7.28. Número de muestras, MSE y valores de R para el modelo diseñado con los datos del electrodo de Cobalto para la concentración media.			
	Nº Muestras	MSE	R
Entrenamiento	29	151.9268	0.93286
Validación	6	421.6118	0.91131
Prueba	6	597.3278	0.93090

Los valores de MSE (*Mean Squared Error*) y R (regresión valor R) para el entrenamiento, validación y prueba se muestran en la Tabla 7.28. La Fig. 7-40 muestra el valor de R para todas las muestras (entrenamiento, validación y prueba, así como para todo el grupo de muestras.

En este caso, el coeficiente de regresión es 0.89284, que es mejor que en el caso anterior sin ninguna clasificación (0.7362, Fig. 7-21), para la clase 2.

3.4.3. Concentración alta.

Para las concentraciones altas se ha empleado una red neuronal con 3 entradas, 10 neuronas en la capa oculta y un nodo de salida. Lo que da lugar a una matriz de pesos de entrada de 10×3 y sesgos de 10×1 . Los pesos de salida forman una matriz de 1×10 y los sesgos de 1×1 . Los pesos y sesgos para la capa oculta y capa de salida se muestran en la tabla 7.29.

Tabla 7.29. Pesos y sesgos para la capa oculta y capa de salida					
Capa oculta				Capa de salida	
Pesos			Sesgos	Pesos	Sesgos
0.09504	2.60322	-1.56617	-2.97235	-0.2869	-0.2051
1.75557	-2.41631	-0.77095	-2.27521	-0.3311	
2.74073	-0.01491	-1.15729	-1.93821	0.2661	
0.38223	0.60279	2.93616	-1.07235	0.2748	
-1.36529	-2.44067	1.50562	-0.89681	-1.6175	
-1.27817	1.61844	-2.11273	-0.48419	0.3985	
-0.06136	2.09842	2.07956	-1.78395	0.8336	
2.36969	0.08111	3.12947	1.50703	-1.8713	
1.60029	2.28410	1.66108	2.37750	1.1061	
1.10412	1.78493	1.56769	3.41879	-0.688	

La Fig. 7-41 muestra la gráfica correspondiente a los valores predichos y medidos para las concentraciones altas.

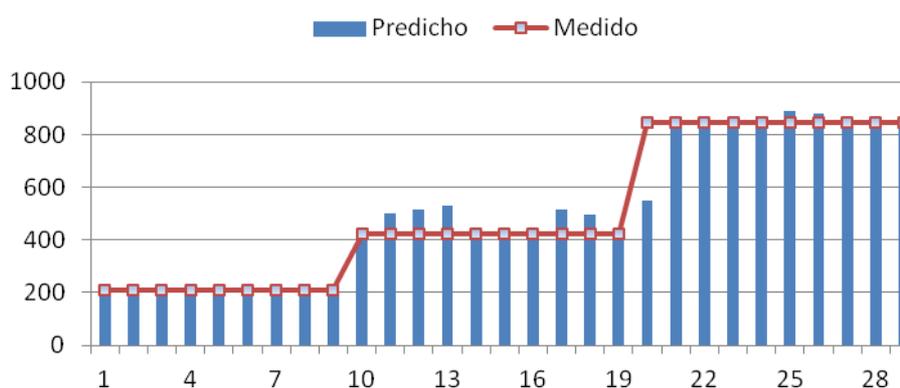


Figura 7-41. Valores predichos y medidos para concentraciones altas (Co clasificado).

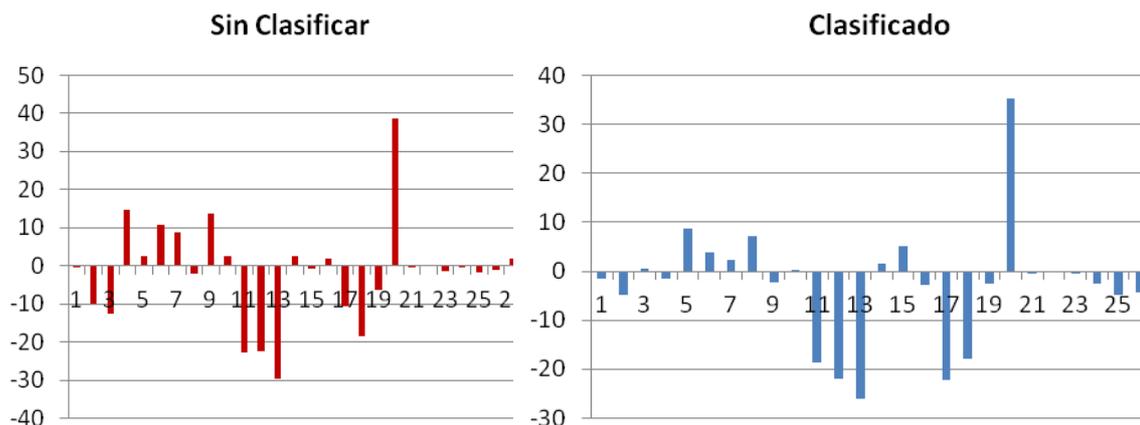


Figura 7-42. Errores absolutos para las concentraciones altas (Co). Sin clasificar y clasificado.

En la Fig. 7-42 se representan los errores relativos en % de los valores predichos por la RNA para las concentraciones altas sin clasificar y después de hacer la clasificación. En dicha figura se observa que la mayoría de los errores han disminuido una vez hecha la clasificación.

Tabla 7.30. Número de muestras, MSE y valores de R para el modelo diseñado con los datos del electrodo de Cobalto.			
	Nº Muestras	MSE	R
Entrenamiento	21	5418.73273	0.958411
Validación	4	2355.64457	0.988633
Prueba	4	3106.7204	0.981957

Los valores de MSE (*Mean Squared Error*) y R (regresión valor R) para el entrenamiento, validación y prueba se muestran en la Tabla 7.30.

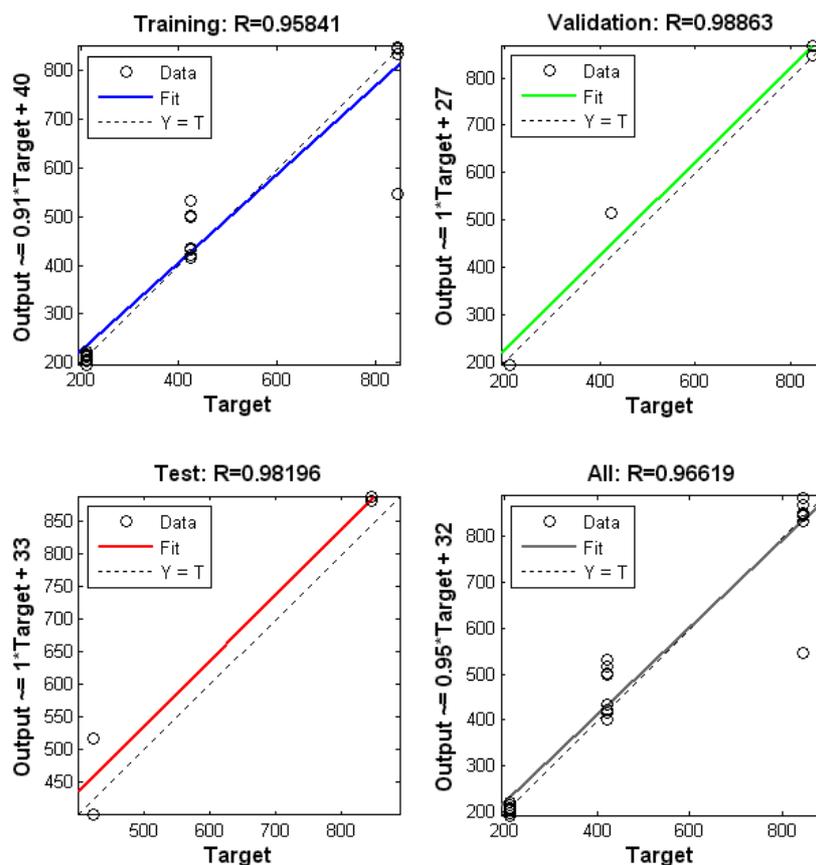


Figura 7-43. Curvas de regresión para el entrenamiento, validación, prueba y para todo el conjunto de datos con MATLAB® Fitting Tools del electrodo de Cobalto para la concentración alta.

La Fig. 7-43 muestra el valor de R para todas las muestras (entrenamiento, validación y prueba, así como para todo el grupo de muestras).

En este caso, el coeficiente de regresión es 0.96619, que también mejora apreciablemente con respecto al caso anterior sin ninguna clasificación (0.9607, Fig. 7-21), para la clase 3.

En el caso específico de la utilización de un electrodo de cobalto, los resultados no son tan buenos como los obtenidos mediante el Cobre, así que se ha decidido desarrollar la red neuronal artificial teniendo en cuenta sólo el electrodo de cobre.

3.5. Comparación entre las dos redes.

La tabla 7.31 muestra la comparación de resultados obtenidos con los electrodos de cobre y cobalto.

Tabla 7.31.Comparación de resultados				
Electrodo	Tasa de aciertos en %	Valores de R		
		Clase baja	Clase media	Clase alta
Cobre	95.8	0.9327	0.9862	0.9999
Cobalto	81.5	0.6400	0.8928	0.9661

La clasificación previa mediante la red SFAM, da una tasa de aciertos del 95.8%, con unos coeficientes de correlación R (tabla 7.31) muy buenos, utilizando el electrodo de cobre. En cambio, con el electrodo de cobalto, la tasa de aciertos disminuye bastante, y el coeficiente de correlación R es muy bajo para la clase baja.

En el caso específico de la utilización del electrodo de cobalto, los resultados no son tan buenos como los obtenidos mediante el cobre, así que se ha decidido desarrollar la red neuronal artificial teniendo en cuenta sólo el electrodo de cobre.

Si nos fijamos en la tabla 7.16, vemos que los resultados con el electrodo de cobalto son muy buenos para la clase media y alta, es decir para concentraciones comprendidas entre 21.138 mg/l y 845.5 mg/l.

4. Conclusión.

Se ha desarrollado un dispositivo de voltamperometría llamado FraPlus que nos permite detectar concentraciones de glifosato en un búfer. Los ensayos con diferentes concentraciones de glifosato han dado una base de datos para construir un modelo matemático.

Después de diversos ensayos y análisis, la RNA es la mejor opción para crear un modelo matemático, ya que da los mejores resultados estadísticos. Se ha realizado el análisis con redes neuronales utilizando todo los datos de todas las concentraciones de glifosato. La recta de regresión que relaciona los datos reales frente a los predichos da un valor $R=0,999$, que se puede considerar como muy bueno, pero también se observa una dispersión bastante elevada para valores de concentración baja, lo que indica que pueden haber bastantes valores con una desviación elevada. Para distinguir los resultados según la concentración, se ha realizado unos nuevos análisis pero dividiendo los datos

en tres grupos para concentraciones bajas (clase 1), medias (clase 2) y altas (clase 3). Los resultados obtenidos indican una buena correlación.

En este caso específico, el valor de coeficiente de regresión y el valor del peso el mejor de todos los modelos estudiados ($R = 0,9998$ y p -valor = 0,0) lo que se recomienda este procedimiento para un mayor desarrollo de modelos matemáticos para predecir las concentraciones de glifosato en muestras reales de agua.

Debido a que la variación en la concentración de glifosato de las muestras es superior a dos órdenes de magnitud, se requiere una clasificación previa antes de determinar la concentración.

5. Producción científica.

Nicolás Laguarda-Miro, Francesca Werner Ferreira, Eduardo García-Breijo, Javier Ibáñez-Civera, Luis Gil-Sánchez, José Garrigues-Baixauli. *Glyphosate detection by voltammetric techniques. A comparison between statistical methods and an artificial neural network.* Sensors and Actuators B: Chemical n° 171–172 (2012). pp 528-536.

Capítulo 8

*Conclusiones Finales y
Futuras Líneas de Investigación.*

Al final de cada capítulo de esta tesis doctoral se ha ido recogiendo, las principales conclusiones derivadas en cada uno de ellos. En este último capítulo se pretende comparar las conclusiones frente a los objetivos de la tesis y dar algunas ideas para desarrollar futuras investigaciones.

1. Resumen de Resultados y Conclusiones.

En esta tesis se han estudiado diversas aplicaciones de las redes neuronales Fuzzy ArtMap Simplificada y Perceptrón Multicapa, que se pueden agrupar en función del tipo de aplicación: clasificación y/o predicción.

La RNA del tipo Perceptrón Multicapa se ha utilizado en las aplicaciones del tipo predicción para calcular la profundidad del disco Secchi, para la medida de humedad en etiquetas flexibles de RFID y para la detección de la cantidad de glifosato. También se ha utilizado en la clasificación de aguas embotelladas.

- a) En el cálculo de la profundidad del disco Secchi, la relación ($R=0.998$) entre los valores estimados y los medidos demuestra que éste modelo reproduce bastante bien los valores de turbidez obtenidos por el disco Secchi para los tres tipos de agua estudiados. La implementación de la RNA en un microcontrolador del tipo PIC18F4550 para calcular la profundidad del disco Secchi es una alternativa eficaz para medir la turbidez de las masas de agua.
- b) Con el fin de medir la humedad relativa, se ha utilizado un microcontrolador para desarrollar e implementar un sistema de medición de impedancia capacitiva. Debido a la respuesta no lineal de los sensores, se ha desarrollado y aplicado una RNA realizada con un microcontrolador del tipo PIC16F877.
- c) Para determinar la cantidad de glifosato, se ha hecho un primer estudio con una RNA FF-MLP, obteniendo una recta de regresión total con un valor $R=0.999$, que se puede considerar como muy bueno, pero al representar los valores medidos en comparación con los previstos por la red neuronal, se puede observar que hay una muy buena relación en concentraciones altas y medias ($R = 0.9504$ para la concentración media y $R = 0.9992$ para la concentración alta), pero la relación es mala en bajas concentraciones ($R=0,0632$). Por

ello, se ha considerado oportuno, realizar una clasificación previa de los datos en tres categorías, mediante la red SFAM.

- d) El análisis de las aguas embotelladas que se ha realizado mediante la red FF-MLP ha dado una tasa de aciertos del 80.9% inferior a la clasificación SFAM. Este tipo de redes tienen la ventaja de que la cantidad de memoria utilizada no varía con el número de muestras de entrenamiento, en cambio en la red SFAM la matriz de pesos aumenta con las muestras de entrenamiento.

En cuanto a la RNA del tipo Fuzzy Artmap Simplificada se ha empleado para la clasificación de aguas embotelladas, clasificación de mieles y también se ha hecho una clasificación previa de la concentración de glifosato con objeto de mejorar el coeficiente de regresión R, especialmente en concentraciones bajas.

- a) El análisis de las aguas embotelladas que se ha realizado mediante la red SFAM con una tasa de aciertos del 96.4% y una matriz de pesos de 12x13. La selección de variables permite reducir la matriz de pesos a 6x15, pero disminuyendo la tasa de aciertos a 92.9. Con los datos obtenidos en el entrenamiento se ha implementado el algoritmo de la red SFAM en el microcontrolador PIC18F4550 dando una tasa de aciertos del 77.1% sobre 5 muestras nuevas.
- b) En la clasificación de la miel según el origen floral se ha conseguido una tasa de aciertos del 100%, tanto con los 7 electrodos iniciales como con los 4 electrodos elegido en la selección de variables. La implementación con el microcontrolador PIC18F4550 ha dado una tasa de aciertos del 68.75% con 7 electrodos y del 75% con 4 electrodos. Con los resultados obtenidos en la clasificación de la miel según el tratamiento térmico, podemos concluir que son debidos al azar.
- c) En la detección del glifosato se ha hecho una clasificación previa en función de la clasificación, en tres categorías, baja media y alta, con una tasa de aciertos del 95.8%. A continuación se ha realizado la detección del glifosato en mediante tres redes FF-MLP, dando lugar a

un coeficiente de regresión R de 0.9327, 0.9862 y 0.9999 para las clases baja, media y alta, respectivamente, lo que indica claramente que el rango de concentración estudiado es demasiado elevado para su detección mediante RNA.

2. Conclusiones.

El objetivo principal de la tesis era utilizar e integrar los algoritmos de las redes neuronales artificiales de tipo Perceptrón multicapa y Fuzzy Artmap Simplificada en un sistema microcontrolador y aplicarlo a distintos sistemas. Así se ha logrado integrar con éxito estas redes en sistemas tan variados como sensores de humedad, detección de la profundidad de Secchi, clasificación de aguas y miel y detección de glifosatos. En todos ellos se ha obtenido una muy buena clasificación o predicción y se han realizados los sistemas electrónicos basados en los microcontroladores PIC18F4550 y el PIC16F877.

El siguiente objetivo de la tesis era desarrollar programas de optimización de las redes neuronales con el objetivo de mejorar los parámetros de integración de dichas redes en sistemas programables. Como conclusión se puede señalar que se ha desarrollado una GUI basada en MATLAB que cumple con los requisitos iniciales. Dicho programa permite optimizar los parámetros de integración de las redes en los microcontroladores tal como se ha comentado en el objetivo anterior.

3. Futuras Líneas de Investigación.

De los ensayos con glifosato se derivan dos conclusiones: por un lado es necesario poner en marcha experiencias sobre sistemas reales como aguas del río y, por otro, integrar en un único microcontrolador los 8 pulsos correspondientes a la voltametría de pulsos realizada con el FraPlus y las redes neuronales de clasificación y detección.

Otras futuras líneas serían:

- Estudio de la viabilidad de la utilización de lenguas electrónicas potenciométricas como herramienta de monitorización en los procesos de

eliminación biológica de fósforo y nitrificación parcial en el tratamiento de aguas residuales.

- Posible implementación de otro tipo de redes neuronales como por ej. PCA, SOM etc.
- Posibilidad de implementar el algoritmo de entrenamiento en el μ C.
- Comparativa entre el μ C, FPGA y micros dedicados a redes neuronales.
- Utilizar microprocesadores neuronales.

Producción Científica.

Hasta la fecha de la publicación de esta tesis, la producción científica derivada de la misma ha sido la siguiente:

Publicaciones SCI.

Javier Ibáñez Civera, Eduardo Garcia Breijo, Nicolás Laguarda Miró, Luis Gil Sánchez, José Garrigues Baixauli, Inmaculada Romero Gil, Rafael Masot Peris, Miguel Alcañiz Fillo, *Artificial neural network onto eight bit microcontroller for Secchi depth calculation*, Sensors and Actuators B: Chemical nº 156 (2011) pag.132-139.

Eduardo Garcia-Breijo, John Atkinson, Luis Gil-Sanchez, Rafael Masot, Javier Ibañez, Jose Garrigues, Monika Glanc, Nicolas Laguarda-Miro, Cristian Olguin, *A comparison study of pattern recognition algorithms implemented on a microcontroller for use in an electronic tongue for monitoring drinking waters*, Sensors and Actuators A: Physical nº 172 (2011) 570-582.

Pelegri-Sebastia, J.; Garcia-Breijo, E.; Ibanez, J.; Sogorb, T.; Laguarda-Miro, N.; Garrigues, J.; , "*Low-Cost Capacitive Humidity Sensor for Application Within Flexible RFID Labels Based on Microcontroller Systems*," *Instrumentation and Measurement, IEEE Transactions on Instrumentation and Measurement*, vol.61, no.2, (2012) pp.545-553.

Nicolás Laguarda-Miro, Francesca Werner Ferreira, Eduardo García-Breijo, Javier Ibáñez-Civera, Luis Gil-Sánchez, José Garrigues-Baixauli. *Glyphosate detection by voltammetric techniques. A comparison between statistical methods and an artificial neural network.*Sensors and Actuators B: Chemical nº 171–172 (2012) pag. 528-536.

Capítulo de Libro .

Eduardo Garcia-Breijo, Jose Garrigues and Luis Gil-Sanchez

Book Title: *Pattern Recognition: Practices, Perspectives and Challenges*

Chapter Title: Optimization of an Embedded Simplified Fuzzy ARTMAP Implemented on a Microcontroller using MATLAB GUI Environment

NOVA SCIENCE PUBLISHERS, INC. 2013. ISBN: **978-1-62618-196-0**

Congresos.

Luis Gil-Sánchez, Eduardo Garcia-Breijo, José Garrigues, Miguel Alcañiz, Isabel Escriche, Melinda Kadar. *Classification of honeys of different floral origins by artificial neural networks*. IEEE SENSORS 2011 CONFERENCE, Limerick (Irlanda)

Garcia-Breijo, E.; Atkinson, J.; Garrigues, J.; Gil, L.; Ibanez, J.; Glanc, M.; Olguin, C.; , "An electronic tongue for monitoring drinking waters using a fuzzy ARTMAP neural network implemented on a microcontroller," *Industrial Electronics (ISIE), 2011 IEEE International Symposium on Industrial Electronics*, (2011) pp.1270-1275,

L. Gil, M. Alcañiz, R. Masot, E. García-Breijo, J. Ibáñez, J. Garrigues. *Desarrollo de sistemas de lenguas electrónicas con distintas técnicas de medida. Aplicación al análisis de aguas y de alimentos*. III Workshop on Sensors and Molecular Recognition. Valencia – España. Julio 2009.

J. Garrigues, E. Garcia-Breijo, L. Gil-Sánchez, J. Ibáñez. *Implementación de redes neuronales artificiales en microcontroladores*. V Workshop on Sensors and Molecular Recognition, Valencia – España. Julio 2011

J. Garrigues Baixauli, E. García Breijo, L. Gil Sánchez. *Interface Gráfica de Usuario (GUI) para desarrollo de redes neuronales artificiales de tipo Fuzzy Artmap*. VI Workshop on Sensors and Molecular Recognition. Valencia – España. Julio 2012.

REFERENCIAS:

- [Abdelmalik et al., 2010] *Evaluación de modelos de clasificación supervisada.*
Abdelmalik Moujahid, Iñaki Inza y Pedro Larrañaga
Departamento de Ciencias de la Computación e
Inteligencia Artificial
Universidad del País Vasco (2010)
- [Abdul et al., 2004] *A microcontrolled-based taste sensing system for the verification
of Eurycoma longifolia*
A.S. Abdul Rahman, M.M. Sim Yap, A.Y. Md. Shakaff, M.N.
Ahman, Z. Dahari, Z. Ismail, M.S. Hitam,
Sensors and Actuators B, vol. 101 (2004) pp. 191-198,
- [Accinelli et al., 2004] *Influence of insecticidal toxins from Bacillus thuringiensis subsp.
kurstakion the degradation of glyphosate and glufosinate-
ammonium in soil samples.*
C. Accinelli, C. Screpanti, A. Vicari, P. Catizone.
Agr. Ecosyst & Environ. 103 Issue 3, (2004) pp. 497-507
- [Ackley et al., 1985] *A learning algorithm for Boltzmann machine*
D. Ackley, G. Hinton, T. Sejnowski
Cognitive Science, vol.9, pp.147-169, 1985
- [Acosta & Zuloaga, 2010] *Tutorial de redes neuronales*
Universidad Tecnológica de Pereira
ohm.utp.edu.co/neuronales/ (2010)
- [Acuña, 2002] *Dr. Edgar Acuña Fernández*
Departamento de Matemáticas
Universidad de Puerto Rico. (2002)
- [Alcañiz, 2011] *Diseño de un sistema de lengua electrónica basado en técnicas
electroquímicas voltamétricas y su aplicación en el ámbito
agroalimentario.*
Alcañiz Fillol, Miguel (2011)
Tesis Doctoral dirigida por: Soto Camino, Juan ; Barat Baviera,
José Manuel ; Gil Sánchez, Luís
Departamento de Ingeniería Electrónica.
Universitat Politècnica de València
- [Amari et al., 2006] *Monitoring the Freshness of Moroccan Sardines with a Neural-
Network Based Electronic Nose,*
A. Amari, N. El Barbri, E. Llobet, N. El Bari, X. Correig,
Bouchikhi,

- Sensors* 6 (2006) 1209-1223.
- [Andina, 2010] *Tutorial de Redes Neuronales*
Dr. Diego Andina de la Fuente
Universidad Politécnica de Madrid (24-11-2010)
- [Andonie & Cataron, 2008]
Fuzzy ARTMAP with feature weighting.
R. Andonie, A. Cataron,
Proceedings of the 26th IASTED International Conference. Artificial Intelligence and Applications, (2008) pp. 91-96.
- [Apetrei et al., 2004]
Array of voltammetric sensors for the discrimination of bitter solutions
Apetrei, M. L. Rodríguez-Méndez, V. Parra, F. Gutierrez, J. A. de Saja
Sensors and Actuators B, 103 (2004) 145-152
- [Aquino & Andrade, 2009]
Electrooxidation of glyphosate herbicide at different DSA compositions: pH, concentration and supporting electrolyte effect.
S. Aquino Neto, A.R. de Andrade.
Electrochim. Acta 54 (2009), pp. 2039 – 2045.
- [Arrieta et al., 2004] *Voltammetric sensor array based on conducting polymer-modified electrodes for the discrimination of liquids.*
A. C. Apetrei, M. L. Rodríguez-Méndez, J. A. de Saja.
Electrochimica Acta, 49,(2004) pp. 4543-4551,
- [Barat et al., 2008a] *An electronic tongue for fish freshness analysis using a thick-film array of electrodes.*
J.M. Barat, I. Escriche, E. GarciaBreijo, L. Gil, R. MartinezManez, J. Soto.
Microchimica Acta 163 (1-2) (2008) pp. 121-129.
- [Barat et al. 2008b] *Freshness monitoring of sea bream (*Sparus aurata*) with a potentiometric sensor*
J.M. Barat, L. Gil, E. García-Breijo, M-C. Aristoy, F. Toldrá, R. Martínez-Máñez, J. Soto..
Food Chem. 108 (2008) pp. 681–688.
- [Bartfai, 1995] *An improved Learning Algorithm for the Fuzzy ARTMAP Neural network,*
G. Bartfai,
Victoria University of wellington. Department of Computer Science. Technical Report CS-TR-95/10 (1995).

- [Bartfai & White, 1997] *A fuzzy ART-based modular neuro-fuzzy architecture for learning hierarchical clusterings*,
G. Bartfai, R. White,
Fuzzy Systems Proceedings of the Sixth IEEE International Conference on, 3 (1997) 1713-1718.
- [Baxter, 1997] *Capacitive Sensors: Design and Applications*.
L. K. Baxter,
New York: IEEE Press, (1997)
- [Bearzotti et al., 1992] *“Integrated optic sensor for the detection of H₂ concentrations,”*
A. Bearzotti, D. Caliendo, E. Verona, and A. D’Amico,
Sens. Actuators B, Chem. vol. 7, no. 1/3, (1992) pp. 685–688
- [Beretta et al., 2005] *“Standardization of antioxidant properties of honey by a combination of spectrophotometric/fluorimetric assays and chemometrics”*.
Beretta, G., Granata, P., Ferrero, M., Orioli, M., &
Facino, R. M.
Analytica Chimica Acta, 533, (2005) 185-191
- [Bhadeshia 2008] *Neural Networks and Information in Materials Science*,
H. Bhadeshia,
University of Cambridge, 2008.
- [B.O.E. 83. 2007] *Ministerio de la Presidencia. Orden PRE/876/2007 de 30 de Marzo por la que se modifican los anexos II de los Reales Decretos 569/1990 de 27 de abril y 280/1994 de 18 de febrero, por los que se establecen los límites máximos de residuos plaguicidas y su control en determinados productos de origen animal y vegetal, respectivamente.*
Boletín Oficial del Estado 83 (2007) pp 37
- [Bortolozzi et al., 1996] *A fuzzy ARTMAP-based classification system for detecting cancerous cells, based on the one-class problem approach.*
F. Bortolozzi, N. Murshed, R. Sabourin,
Proceedings of 13th International Conference on Pattern Recognition IV (1996) 478-482.
- [Bradley, 1997] *The use of the area under the ROC curve in the evaluation of machine learning algorithms.*
Bradley, A. P. (1997).
Pattern Recognition, 30 (7), 1145 - 1159.

[Breiman et al.,1984]

Classification and regression trees.
Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984).
Wadsworth International Group, Belmont, CA.

[Brezmes et al]

Evaluation of there modified Fuzzy ARTmap algorithms to be used by an electronic nose
J. Brezmes, P. Margalef, E. Llobet, X. Vilanova, X. Correig.
9th International Symposium on Olfaction and Electronic Nose ISOEN '02 Roma.

[Brezmes et al., 2005]

Discrimination between different samples of olive oil using variable selection techniques and modified fuzzy artmap neural networks,
J. Brezmes, P. Cabre, S. Rojo, E. Llobet, X. Vilanova, X. Correig,
Sensors Journal IEEE , 5(3) (2005) 463- 470.

[Burgueño et al., 1995]

Las curvas ROC en la evaluación de las pruebas diagnósticas
M.J. Burgueño (a), J.L. García-Bastos (b) y J.M. González-Buitrago,(b,c,d)
a)Servicio de Análisis Clínicos. Hospital General Yagüe. Burgos.
b)Servicio de Bioquímica. c)Unidad de Investigación. Hospital Universitario. Salamanca. d)Departamento de Bioquímica y Biología Molecular.
Universidad de Salamanca.
MEDICINA CLÍNICA VOL. 104 NÚM. 17. (1995)

[Camman, 1979]

Working with Ion-Selective Electrodes
K. Camman
Springer, Berlin (1979)

[Campos et al., 2010a]

Electronic Tongue for Detection and Classification of Nerve Agent Mimics,
I. Campos, L. Gil, R. Martínez-Manez, J. Soto, J.L. Vivancos.
Use of a Voltammetric Electroanalysis, 22 (2010) 1643–1649.

[Campos et al., 2010b]

Accurate concentration determination of anions nitrate, nitrite and chloride in minced meat using a voltammetric electronic tongue
I. Campos, R. Masot, M. Alcañiz, L. Gil, J.L. Vivancos, E. García-Breijo, R. H. Labrador, J. M. Barat, R. Martínez-Mañez, J. Soto..
Sens. Actuators B 149 (2010) pp. 71–78.

- [Carpenter & Grossberg, 1987]
A massively parallel architecture for a selforganizing neural pattern recognition machine
G.A. Carpenter, S. Grossberg (1987)
Computer Vision, Graphics and image Processing 37, pp.54-115,
- [Carpenter & Grossberg, 1987b]
ART2:Stable self-organization of pattern recognition codes for analog input patterns
G.A. Carpenter, S. Gossberg (1987)
Applied Optics 26 pp. 4919-4930.
- [Carpenter & Grossberg, 1987c]
ART3 hierarchical search: Chemical transmitters in self organizing pattern recognition architectures.
G.A. Carpenter, S. Gossberg. (1987)
International joint conference on neural networks. Washington DC. Eribaun Associates.
- [Carpenter & Ross, 1995]
ART-EMAP: A neural network architecture for object recognition by evidence accumulation,
G. A. Carpenter, W. D. Ross
IEEE Transactions on Neurall Networks 6(4) (1995)
805-818.
- [Carpenter et al., 1991]
Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system.
G.A. Carpenter, S. Gossberg, D. Rosen. (1991)
Neural Networks, 4 pp. 759-771.
- [Carpenter et al., 1991b]
Artmap: Supervised realtime learning and classification of nonstationary data by a self-organizing neural network.
G.A. Carpenter, S. Gossberg, J. Reynolds. (1991)
Neural Networks, 4 pp. 565-588.
- [Carpenter et al., 1992]
Fuzzy Artmap: A Neural Network architecture for incremental supervised learning of analog multidimensional maps.
G.A. Carpenter, S. Gossberg, N. Markuzon, J. Reynolds, D. Rosen. (1992)
*IEEE Transactions on neural networks.*3 pp. 698-713.
- [Carpenter, 1997] *Distributed Learning, Recognition and Prediction by ART and ARTMAP Neural Networks.*

Carpenter, Gail A. (1997)
Neural Networks pp. 1473– 1494.

[Cauwenberghs, 1993]

A fast stochastic Error-Descent Algorithm for supervised learning and optimization.

Cauwenberghs, G. (1993)
Advances in Neural Information Processing Systems,
S. J. Hanson, J. D. Cowan, C. L. Giles (eds).
pp. 244-251. Morgan Kaufmann, San Mateo, CA

[Chappelier &Grumbach, 1994]

Time in Neural Networks,
J. C. Chappelier and A. Grumbach (1994)
SIGART Bulletin, vol. 5, no. 3, pp. 3-11, Jul.

[Chiu et al. 2008]

Analysis of glyphosate and aminomethylphosphonic acid by capillary electrophoresis with electrochemiluminiscence detection.

H.Y. Chiu, Z.Y. Lin, H. Tu, Ch. Whang.
J. Chromatogr. A 1177 (2008) pp. 185 – 198.

[Chralampidis et al., 2001]

Classification of noisy signals using fuzzy ARTMAP neural networks.

D. Chralampidis, T. Kasparis, M. Georgiopoulos.
Neural Networks, IEEE Transactions on , 12(5) (2001) 1023-1036.

[Concejero, 2009]

Comparación de modelos de curvas ROC para la evaluación de procedimientos estadísticos de predicción en investigación de mercados.

Pedro Concejero Cerezo. (2009)
Tesis Doctoral dirigida por Rosario Martínez Arias
Dpto. Metodología de las Ciencias del Comportamiento
Facultad de Psicología
Universidad Complutense de Madrid

[Contardo et al., 2009]

Bioaccumulation of glyphosate and its formulation Roundup Ultra in Lumbriculus variegatus and their effects on biotransformation and antioxidant enzymes.

V. Contardo-Jara, E. Klingelmann, C. Wiegand,
Environ. Pollut.157 (2009) pp. 57-63.

[Coutinho et al., 2007]

Investigation of copper dissolution inthe presence of glyphosate using hydrodynamic voltammetry and chronoamperometry.

- C.F.B. Coutinho, M.O. Silva, M.L. Calegaro, S.A.S. Mahcado, L.H. Mazo.
Solid State Ionics 178 (2007) pp. 161 -164.
- [Coutinho et al., 2008]
Camara. *Rapid and direct determination of glyphosate and aminomethyl phosphonic acid in wáter using anion-exchange chromatography with coloumetric detection.*
C.F.B. Coutinho, L.F.M. Coutinho, L.H. Mazo, S.L. Nixdorf, C.A.P.
J. Chromatogr. A 1208 (2008), pp. 246 – 249.
- [Cox, 1998] C. Cox. *Glyphosate (Roundup).*
J. Pest. Reform. 18 (1998) pp 3-17
- [Cristhian et al., 2011]
Data acquisition, analysis and processing tool for multisensory system and mass spectrometry,
M. Cristhian, A. Duran , G. Oscar Gualdron, F. Adrian Carvajal,
Revista Colombiana de Tecnologias de Avanzada 1(17) (2011) 16-23.
- [Dagher et al., 1998a]
Fuzzy ARTVar: An Improved Fuzzy ARTMAP Algorithm. Neural Networks Proceedings,
I. Dagher, M. Georgiopoulos, G.L. Heileman, G. Bebis,
IEEE World Congress on Computational Intelligence 3 (1998) 1688 – 1693
- [Dagher et al., 1998b]
Ordered Fuzzy ARTMAP: A Fuzzy ARTMAP algorithm with a fixed order of pattern presentation.
Dagher, M. Georgiopoulos, G.L. Heileman, G. Bebis
Neural Networks Proceedings, IEEE World Congress on Computational Intelligence 3 (1998) 1717 – 1720.
- [DARPA, 1988] *Neural Network Study, ,*
AFCEA International Press, p. 60, (1988)
- [Demetgul et al., 2001]
Conditioning Monitoring and Fault Diagnosis for a Servo-Pneumatic System with Artificial Neural Network Algorithms.
M. Demetgul, S. Taskin, I. Nur Tansel,
Artificial Neural Networks, Industrial and Control Engineering Applications, Ed. Kenji Suzuki (2001).
- [Dhawam et al., 2009]
Recent developments in urea biosensors
G. Dhawan, G. Sumana, B.D. Malhotra
Biochemical Engineering Journal, 44, (2009) 42-52

- [Dias et al., 1998] “A temperature compensated system for magnetic field measurements based on artificial neural networks,”
J. Dias Pereira, O. Poastolache, and P. Silva Girão,
IEEE Trans. Instrum. Meas., vol. 47, no. 2, (1998) pp. 494–498
- [Dias et al., 2008] “An electronic tongue for honey classification”.
Dias, L.A.; Peres, A.M.; Vilas-Boas, M; Rocha, M.A.;
Estevinho, L. & Machado A.
Microchimica Acta 163 (2008) pp. 97-102
- [Dietz et al., 2002] *Wireless liquid level sensing for restaurant applications*,
P. H. Dietz, D. Leigh, and W. S. Yerazumis,
in Proc. IEEE Sens., Orlando, FL, Jun. 12–14, (2002)
pp. 715–720.
- [D.O.U.E. 206. 2006]
Comisión de las Comunidades Europeas. *DIRECTIVA 2006/60/CE DE LA COMISION de 7 de julio de 2006 que modifica los anexos de la Directiva 90/642/CEE del Consejo por lo que respecta a los limites maximos de residuos de trifloxistrobin, tiabendazol, abamectina, benomilo, carbendazima, tiofanato-metil, miclobutanil, glifosato, trimetilsulfonio, fenpropimorf y cloromequat.*
Diario Oficial de la Unión Europea 206(2006) pp 11.
- [Downs et al., 1996]
Application of the fuzzy ARTMAP neural network model to medical pattern classification tasks.
J. Downs, R.F. Harrison, R.L. Kennedy, S.S. Cross.
Artif Intell Med. 8(4) (1996) 403-28.
- [Egan, 1975] *Signal Detection Theory and ROC Analysis*
Egan, J. P. (1975)..
Series in Cognition and Perception. Academic Press, New York.
- [Escriche et al., 2012]
A potentiometric electronic tongue for the discrimination of honey according to the botanical origin. Comparison with traditional methodologies: Physicochemical parameters and volatile profile
Isabel Escriche, Melinda Kadar, Eva Domenech, Luis Gil-Sánchez,
Journal of Food Engineering 109, April 2012, pp. 449–456
- [Escrivá, 2011] *Estudio de la degradación de la vitamina C en zumos de fruta mediante el método reflectrométrico y propuesta de una nueva metodología basada en la voltametría de pulsos.*
Escrivá Pons, Ana. *Proyecto Final de Carrera Ingeniería Técnica Industrial,*
Universidad Politécnica de Valencia, Septiembre de 2011.

- [Estelar, 1999] *Hoja de manejo seguro. ESTELAR 480 SL. Dow Chemical. (1999)*
- [Fawcett, 2003] *ROC Graphs: Notes and Practical Considerations for Data Mining Researchers*
Tom Fawcett
Intelligent Enterprise Technologies Laboratory
HP Laboratories Palo Alto
*HPL-2003-4. January 7th , 2003**
- [Faggin, 1991] *VLSI implementation of neural networks*
F. Faggin (1991)
International Joint Conference on Neural Networks.
Seattle, WA,
- [Fernandez&Barro 1990] *MART: a multichannel ART-based neural network.*
M. Fernandez-Delgado, S. Barro Ameneiro,
Neural Networks IEEE Transactions on , 9(1) (1990)
pp. 139-150.
- [Ferrer et al., 2006] *Integration of sensors and RFID's on ultra-low-cost paper-based substratesfor wireless sensor networks applications,*
A. Ferrer-Vidal, A. Rida, S. Basat, L. Yang, and M. M. Tentzeris
inProc. IEEE WiMesh, (2006), pp. 126–128.
- [Fritsch, 1996] *Modular Neural Networks for Speech Recognition.*
Fritsch, J. (1996). *Master's Thesis.*
Carnegie Mellon University.
- [Fuente & Calonge, 1999] *Aplicaciones de las redes de neuronas en supervisión diagnosis y control de procesos*
María Jesús de la Fuente Aparicio, T. Calonge Cano
Equinocio (1999)
Ediciones de la universidad Simón Bolivar.
- [Gaitán et al., 2007] *“Direct interface for capacitive sensors based on the charge transfer method,”*
J. E. Gaitán-Pire, M. Gasulla, and R. Pallàs-Areny,
inProc. IEEE IMTC, Warsaw, Poland, May 1–3, (2007),
pp. 1–5.
- [Gallardo et al., 2005] *Application of a potentiometric electronic tongue as a classification tool in food analysis.*
J. Gallardo, S. Alegret, M del Valle.
Talanta, 66 (2005) pp. 1303-1309

- [Garcia et al., 2011] *A comparison study of pattern recognition algorithms implemented on a microcontroller for use in an electronic tongue for monitoring drinking waters.*
E. Garcia-Breijo, J. Atkinson, L. Gil-Sanchez, R. Masot, J. Ibanez, J. Garrigues, M. Glanc, N. Laguarda-Miro, C. Olguin
Sensors and Actuators A: Physical, 172(2) (2011) 570-582.
- [Gardner, 1991] *Detection of vapours and odours from a multisensor array using pattern recognition. Part 1: principal components and cluster analysis.*
J.W. Gardner
Sensors and Actuators B, 4 (. 1991) pp. 106-116
- [Gardner, 1992] *Sensors and Sensory Systems for an Electronic Nose*
W. Gardner, P.N. Bartlett
NATO ASI Series E: Applied Sciences; Dor drecht: Kluwer (1992), Vol. 212.
- [Gardner et al., 1996] *Detection of vapours and odours from a multisensor array using pattern recognition: self-organising adaptive resonant*
Gardner J.W., Hines E.L., Pang C.(1996)
Measurement + Control, vol 29,
- [Garret, 2003] *Fuzzy ART and Fuzzy ARTMAP Neural Networks*
Disponible en el sitiowebMATHWORKS
<http://www.mathworks.com/matlabcentral/fileexchange/4306-fuzzy-art-and-fuzzy-artmap-neural-networks>
- [Gasnier et al. 2009] *Glyphosate-based herbicides are toxic and endocrine disruptors in human cell lines.*
C. Gasnier, C. Dumont, N. Benachour, E. Clair, M-C.Chagnon, G-E. Séralini
Toxicology 262 (2009) pp. 184–191.
- [Georgiopoulos et al., 1996] *Order of serarch in Fuzzy ART and Fuzzy ARTMAP: Effect of the choice parameter*
Georgiopoulos, Michael and Fernlund, Hans and Bebis, George and Heilemman, Gregory L. (1996)
Neural Networks. pp. 1541–1559.

- [Gil, 2007] *Diseño De Lenguas Electrónicas Potenciométricas Construidas Con Electrodo Metálicos y Con Tecnología De Capa Gruesa Para Análisis De Medios Complejos.*
Luis Gil Sánchez (2007).
Tesis Doctoral
Universidad Politécnica de Valencia.
- [Gil et al., 2006] *Electronic Tongue for Qualitative Analysis of Aqueous Solutions of Salts Using Thick-film Technology and Metal Electrodes.*
L. Gil, E. Garcia-Breijo, J. Ibanez, R.H. Labrador, E. Llobet, R. Martinez-Manez, J.Soto.
Sensors 6(9) (2006) pp. 1128-1138.
- [Gil et al., 2008] *Fish freshness analysis using metallic potentiometric electrodes.*
L. Gil, J.M. Barat, E. Garcia-Breijo, J. Ibanez, R. Martinez-Manez, J. Soto, E. Llobet, J. Brezmes, M.-C. Aristoy, F. Toldra.
Sensors and Actuators B: Chemical 131(2) (2008) pp. 362-370.
- [Gil et al., 2010] *A potentiometric electronic tongue to monitor meat freshness,*
L. Gil, J.M. Barat, D. Baigts, R. Martinez-Manez, J. Soto, E. Garcia-Breijo, E. Llobet,
Industrial Electronics, IEEE International Symposium on ,
(2010) pp. 390-395.
- [Gil et al., 2011] *Monitoring of physical–chemical and microbiological changes in fresh pork meat under cold storage by means of a potentiometric electronic tongue.*
L. Gil, J. M. Barat, D. Baigts, R. Martinez-Manez, J. Soto, E. Garcia-Breijo, M-C. Aristoy, F.Toldra, E. Llobet.
Food Chemistry, 126 (3) (2011) pp. 1261-1268.
- [Grossberg, 1967] *Nonlinear Difference_Differential Equations in Prediction and Learning Theory*
S. Grossberg, (1967)
Proc. Natl. Acad. Sci. USA 58, p. 132.
- [Grossberg, 1973] *Contour enhancement, short-term memory, and constancies in reverberating neural networks.*
S. Grossberg, (1973)
Studies in Applied Mathematics, 52, pp. 213-257.
cns-web.bu.edu/~steve/
- [Grossberg, 1980] *How does a brain build a cognitive code?*
S. Grossberg, (1980)
Psychological Review, 1 pp. 1-51,
- [Grossberg, 1988] *Nonlinear neural networks: Principles, mechanisms, and architectures*
S. Grossberg, (1988)
Neural Networks, Volume 1, pp. 17-61,

- [Gusmano et al., 1993] *Humidity-sensitive electrical properties of MgAl₂O₄ thin films.*
G. Gusmano, G. Montesperelli, E. Traversa, and A. Bearzotti,
Sens. Actuators B, Chem., vol. 14, pp. 525–527, 1993.
- [Gutierrez, 2002] *Pattern Analysis for Machine Olfaction: A Review*
R. Gutierrez Osuna (2002)
IEEE Sensors Journal, vol. 2, pp. 189-202,
- [Hanley & McNeil, 1982] *Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology, 143, pp. 29-36.*
- [Hand & Till, 2001] *A simple generalization of the area under the ROC curve to multiple class classification problems*
Hand, D. J., & Till, R. J. (2001).. *Machine Learning*, 45 (2), pp. 171-186.
- [Han et al., 1994] *Fuzzy LAPART supervised learning through inferencing for stable category recognition.*
G. Han, F.M. Ham, L.V. Fausett,
Fuzzy Systems IEEE World Congress on Computational Intelligence, Proceedings of the Third IEEE Conference on , 1 (1994) 46-51.
- [Haykin, 1994] *A Comprehensive Foundation,*
S. Haykin (1994)
Macmillan College Publishing Company, Inc.
IEEE Press, p. 2,
- [Haykin, 1999] *Neural Networks, A Comprehensive Foundation*
S. Haykin.
Upper Saddle River, NJ: Prentice Hall, 1999.
- [Heb, 1949] *The Organization of behavior.*
D. Heb (1949)
New York: Wiley,
- [Holler et al., 1989] *An Electrically Trainable Artificial Neural Network (ETANN) with 10240 Floating Gate Synapsys,*
M. Holler, S. Tam, H. Castro, R. Benson,
Proceedings of the IJCNN-89-Washington DC, Vol. II, (1989), pp. 191-196.
- [Holler et al., 1989b] *Neural Network Recognition of Objects Based On Impact Dynamics*
M.Holler, S.Tam, J.Brauch

- Nuclear Science Symposium and Medical Imaging Conference, ., Conference Record of the 1992 IEEE, pp. 829 - 831 vol.2*
- [Hopfield, 1982] *Neural network and physical systems with emergent collective computational abilities.*
Hopfield J.J (1982)
Proc. Natl. Acad. Sci. USA 79, pp. 2554-2558.
- [Hung et al.,1996] *Cascade fuzzy ART: a new extensible database for model-based object recognition.*
Hai-Lung Hung, Hong-Yuan M. Liao, Shing-Jong Lin, Wei-Chung Lin and Kuo-Chin Fan.
Proc. SPIE 2727, 187 (1996)
- [Ibáñez, 2010] *Desarrollo de un sistema de medida de la extinción de la luz solar en medios acuáticos.*
J. Ibáñez Civera. (2010)
Tesis Doctoral
Universidad Politécnica de Valencia.
- [Ibáñez et al., 2010] *Nuevo instrumento para medir la extinción de la luz solar en las columnas de agua*
J. Ibáñez, J. González del Río, I. Romero, N. Laguarda, E. García Breijo, L. Gil. (2010)
Int IEEE. Simposio sobre Electrónica Industrial de Bari, Italia.
- [Ibáñez et al., 2011] *Artificial neural network onto eight bit microcontroller for Secchi depth calculation.*
J. Ibáñez Civera, E. Garcia Breijo, N. Laguarda Miró, L. Gil Sánchez, J. Garrigues Baixauli, I. Romero Gil, R. Masot Peris, M. Alcañiz Fillol.
Sens. Actuators B 156 (2011) 132–139.
- [Intel, 1992] *Neural Network Technology & Applications”, Data Sheet , 1992.*
- [Ishihara e al., 1995] *ArboART: ART based hierarchical clustering and its application to questionnaire data analysis,*
S. Ishihara, K. Ishihara, M. Nagamachi, Y. Matsubara,
Neural Networks Proceedings., IEEE International Conference on , 1 (1995) 532-537.
- [Jamu & Piedrahita, 1999] *Relationship between Secchi disk visibility and chlorophyll a in aquaculture ponds. (1999)*
D.M. Jamu, Z. Lu, R.H. Piedrahita
Aquaculture 170, pp. 205–214.

- [Jervis et al., 1999] *Probabilistic simplified fuzzy ARTMAP (PSFAM) [and application to biosignal data],*
B.W. Jervis, T. Garcia, E.P. Giahnakis
Science, Measurement and Technology, IEE Proceedings 146(4)
(1999) 165-169
- [Jervis et al., 2004] *Integrated probabilistic simplified fuzzy ARTMAP,*
B.W. Jervis, S. Djebali, L. Smaglo
Science, Measurement and Technology, IEE Proceedings 151(3)
(2004) 218- 228.
- [Jeyabalan et al., 2008] *FUZZY ARTMAP Classification for motor Imagery based Brain*
Computer Interface.
Vickneswaran Jeyabalan, Andrews Samraj, Loo Chu Kiong,
Arab conference on Information Technology (2008) 1-5.
- [Jin et al. 2010] *Characterization of electrochimiluminiscence of tris (2,2'-*
bipiridine)ruthenium (II) with glyphosate as coreactant in
aqueous solution.
J. Jin, F. Takahashi, T. Kaneko, Toshio Nakamura.
Electrochim. Acta 55 (2010) pp. 5532 – 5537
- [Kasuba, 1993] *Simplified Fuzzy ARTMAP.*
T. Kasuba,
AI Expert 8(11) (1993) 18–25.
- [Jin et al. 2010] *Characterization of electrochimiluminiscence of tris (2,2'-*
bipiridine)ruthenium (II) with glyphosate as coreactant in
aqueous solution.
J. Jin, F. Takahashi, T. Kaneko, Toshio Nakamura.
Electrochim. Acta 55 (2010) 5532 – 5537
- [Khenifi et al. 2009] *Glyphosate and glufosinate detection at electrogenerated NiAl-*
LDH thin films.
A, Khenifi, Z. Derriche, C. Forano, V. Prevot, C. Mousty, E.
Scavetta, B. Ballarin, L. Guadagnini, D. Tonelli.
Anal.Chim. Acta, 654, 2 (2009) pp. 97 – 102
- [Kim et al., 2007] *An Intelligent Wireless Electronic Nose Node for Monitoring Gas*
Mixtures Using Neuro-Fuzzy Networks Implemented on a
Microcontroller.
Young Wung Kim, Jung Hwan Cho, Gi Joon Jeon,
Computational Intelligence for Measurement Systems and
Applications, IEEE International Conference on, (2007)
pp. 100-104.
- [Kohonen, 1982] *Self-organized formation of topologically correct feature maps.*

- T. Kohonen. (1982)*
Biological Cybernetics.N° 43, pp. 59-69,
- [Kohonen, 1984] *Self-organization and associative memory.*
T. Kohonen. (1984)
Springer. Berlín
- [Kohonen, 1988] *Learning Vector Quantization*
T. Kohonen (1988)
Abstracts of the 1st Annual INNS Meeting, Boston, MA, 303,
- [Laconte et al., 2003]
Capacitive humidity Sensor Using 3-Layer Patterned Polyimide Sensing Film
J. Laconte, V. Wilmart, D. Flandre, and J. P. Raskin,
in Proc. IEEE Sens. Conf., (2003), pp. 372–377.
- [Laundgren, 2002] *Investigation of Paperbased RFID With Integrated Sensors. Demonstration and Analysis of Printed Sensors.*
J. Laundgren,
Examensarbete.
- [Lapiedra, 2004] *Desarrollo de algoritmos de selección de variables para clasificadores neuronales: métodos no estocásticos.*
Ismael Lapiedra Michel
Tesis Doctoral
Universitat Rovira i Virgili, (2004)
- [Legin, 1996] *Cross-sensitivity of chalcogenide glass sensors in solutions of heavy metal ions.*
Andrey V. Legin, Yuri G. Vlasov, Alisa M. Rudnitskaya, Evgeni A. Bychkov
Sensors and Actuators B, 34 (1996), 456-461
- [Legin et al., 1997] *Tasting of beverages using an electronic tongue.*
A. Legin, A. Rudnitskaya, Y. Vlasov, C. Di Natale, F. Davide, A. D'Amico.
Sensors and Actuators B, 44 (1997) pp. 291-296,
- [Legin et al., 2002] *“Electronic Tongues: Sensors, Systems, Applications”.*
Legin, A.; Rudnitskaya, A; Vlasov, Y.
Sensors Update, 10, (2002) pp. 143–188.
- [Lerner & Vigdor, 2004]
An empirical study of fuzzy ARTMAP applied to cytogenetics.
B. Lerner, B. Vigdor,
Electrical and Electronics Engineers in Israel, Proceedings 23rd IEEE Convention of, (2004) pp. 301- 304.

- [Lim et al., 1997] *Modified Fuzzy ARTMAP Approaches Bayes Optimal Classification Rates: An Empirical Demonstration.*
Lim, Chee Peng and Harris, Robert F.(1997)
Neural Networks pp. 755–774.
- [Lin &Soo, 1997] *Pruning fuzzy ARTMAP using the minimum description length principle in learning from clinical databases.*
Ten-Ho Lin; Von-Wun Soo.
Tools with Artificial Intelligence Proceedings, Ninth IEEE International Conference on, 3-(8) (1997) 396-403.
- [Llobet et al., 1999a] *Non-destructive banana ripeness determination using a neural network-based electronic nose*
E. Llobet, E. Hines J.W. Gardner, S. Franco.
Meas. Sci. Technol. 10 (1999a) pp 538.
- [Llobet et al., 1999b] *Fuzzy ARTMAP based electronic nose data analysis*
Eduard Llobet, Evor L. Hines, Julian W. Gardner, Philip N. Bartlett, Toby T. Mottram,
Sensors and Actuators B, vol. 61 (1999b) pp. 183–190.
- [Llobet et al., 2001] *Gardner, Multicomponent gas mixture analysis using a single tin oxide sensor and dynamic pattern recognition,*
E. Llobet, R. Ionescu, S. Al-Khalifa, J. Brezmes, X. Vilanova, X. Correig, N. Barsan, J.W.
Sensors Journal IEEE 1(3)(2001) 207-213.
- [Llobet et al., 2002] *Wavelet transform and fuzzy ARTMAP-based pattern recognition for fast gas identification using a micro-hotplate gas sensor,*
E. Llobet, J. Brezmes, R. Ionescu, X. Vilanova, S. Al-Khalifa, J.W. Gardner, N. Barsan, X. Correig,
Sensors and Actuators B: Chemical 83(1–3) (2002) 238-244.
- [Llobet et al., 2004] *Building parsimonious fuzzy ARTMAP models by variable selection with a cascaded genetic algorithm: application to multisensor systems for gas analysis,*
E. Llobet, J. Brezmes, O. Gualdron, X. Vilanova, X. Correig,
Sensors and Actuators B: Chemical 99(2–3) (2004) 267-272.
- [Lopes et al., 2005] *Electric load forecasting using a fuzzy ART&ARTMAP neural network.*
M.L. Lopes, C.R. Minussi, A. Diva,
Applied Soft Computing 5(2) (2005) 235-244.
- [Lvova et al., 2006] *Electronic tongue based on an array of metallic potentiometric sensors*

- L. Lvova, E. Martinelli, E. Mazzone, A. Pede, R. Paolesse, C. Di Natale, A. D'Amico.*
Talanta, 70 (2006) pp. 833-839
- [Martin, 2006] *Redes Neuronales y Sistemas Borrosos.*
B. Martin del Rio.
Ed. Rama, Madrid, 2006.
- [Martin & Medrano, 2000] “Microcontroller makes linear temperatura measurement simple,”
B. Martín-del-Brío and N. Medrano-Marqués,
Electron. Design, vol. 48, no. 22, (2000) pp. 133–134,
- [Martínez, 1999] *Herramienta Informática para reconocimiento de números manuscritos mediante redes neuronales Fuzzy-ARTMAP.*
Jorge Martínez Escribano. (1999)
Proyecto Fin de Carrera. UPCO.
- [Martínez et al., 2005a] *A multisensor in thick-film technology for water quality control.*
Ramón Martínez-Máñez, Juan Soto, Eduardo García-Breijo, Luis Gil, Javier Ibáñez y Elena Gadea.
Sensors and Actuators A 120(2005) pp.589-595
- [Martinez et al., 2005b] *An electronic tongue design for the qualitative analysis of natural waters.*
R. Martinez-Manez, J. Soto, E. Garcia-Breijo, L. Gil, J. Ibañez, E. Llobet,
Sensors and Actuators B: Chemical 104(2) (2005) 302-307.
- [Massicotte et al., 1998] “Neural network based method of calibration and measure reconstruction for a high-pressure measuring system,”
D. Massicotte, S. Legendre, and A. Barwicz,
IEEE Trans. Instrum. Meas., vol. 47, no. 2, Apr. 1998
pp. 362–370,
- [Masot et al. 2010] *Design of a low-cost non-destructive system for punctual measurements of salt levels in food products using impedance spectroscopy.*
R. Masot, M. Alcañiz, A. Fuentes, F.C. Schmidt, L. Gil, D. Baigts, R. Martínez-Máñez, J. Soto, J.M. Barat
Sens. Actuators A 158 (2010) pp. 217–223.
- [MathWorks] *MATLAB, marca registrada por The MathWorks, Inc.*
- [McGaughey et al., 2006] “Development of a fluorescence lifetime-based solgel humidity sensor,”

- O. McGaughey, J. V. Ros-Lis, A. Guckian, A. K. McEvoy, C. McDonagh, and B. D. MacCraith,*
Anal. Chimica Acta, vol. 570, no. 1, pp. 15–20, Jun. 2006.
- [Méndez et al. 2007] *Electrochemical properties and electro-aggregation of silvercarbonate sol on polycrystalline platinum electrode and its electrocatalytic activity towards glyphosate oxidation.*
M.A. Méndez, M.F. Suárez, M.T. Cortés, V.M. Sarria.
Electrochem. Commun 9 (2007) pp. 2585 – 2590.
- [McCulloch & Pitts, 1943]
A logical calculus of ideas immanent in nervous activity.
W.S. McCulloch; W.H. Pitts. (1943)
Bull. Math. Biophys. vol 5. pp. 115-133,
- [Medrano et al., 2001]
“A thermocouple model based on neural networks,”
N. J. Medrano Marqués, R. Del Hoyo, and B. Martín del Brío,
in Proc. IWANN, vol. 2085, Lecture Notes in Computer Science,
Jun. (2001), pp. 531–538.
- [Medrano & Martin, 2005]
Implementing neural networks onto standard low-cost microcontroller for sensor signal processing,
N. Medrano-Marqués, B. Martin-del-Brío,
Emerging Technologies and Factory Automation, 2005. ETFA 2005. 10th IEEE Conference, September, 2005,
pp. 972-978.
- [Megard & Bergman 1989]
Efectos de las algas en la transparencia de Secchi del sureste del Mar Mediterráneo.
R.O., Megard, T. Bergman. (1989)
Limnol. Oceanográfico. 34 (8), pp. 1640 - 1655.
- [Minsky & Papert, 1969].
Perceptrons: An introduction to Computational Geometry
M. Minsky, S. Papert. (1969)
Massachusetts: Cambridge University; p. 226,
- [Mohamed et al., 2006]
Multi-class Protein Sequence Classification Using Fuzzy ARTMAP,
S. Mohamed, D. Rubin, T. Marwala,
Systems, Man and Cybernetics, IEEE International Conference on, 2 (2006) 1676-1681.
- [Moreno et al., 2006]
Multi-sensor array used as an “electronic tongue” for mineral water analysis
L. Moreno, A. Merlos, N. Abramova, C. Jiménez, A. Bratov
Sensors and Actuators B: Chemical, 116 (2006) 130-134

- [Muñoz, 1996] *Aplicación de técnicas de redes neuronales artificiales al diagnóstico de procesos industriales.*
A. Muñoz San Roque. (1996)
Tesis Doctoral.
Universidad Pontificia Comillas. Madrid,
- [Nachev, 2007] *Data mining with Fuzzy ARTMAP neural networks: prediction of profiles of potential customers,*
A. Nachev,
International Conference Knowledge-Dialogue-Solutions (2007)
1-8.
- [Nafisi et al., 2009] *Using Fuzzy ARTmap neural network for determination of partial discharge location in power transformers.*
H. Nafisi, M. Davari, M. Abedi, G.B. Gharehpetian,
PowerTech, (2009).
- [Nigrin,1993] *Neural Networks for Pattern Recognition.*
A. Nigrin, (1993)
A Bradford Book; 1St Edition edition (September 16, 1993)
ISBN: 0262140543
- [Nirascou, 2006] *Les pesticides dans les eaux*
F. Nirascou.
Dossier 5 IFEN (2006) pp 36.
- [Oprea et al., 2007] *“Integrated temperature, humidity and gas sensors on flexible substrates for low-power applications,”*
A. Oprea, N. Barsan, U. Weimar, J. Courbat, D. Briand, and N. F. de Rooij,
inProc. IEEE Sens. Conf., 2007, pp. 158–161.
- [Pallás, 2003] *Sensores y acondicionadores de señal*
Ramón Pallás Areny
Marcombo S.A. 4ª Edición (2003)
- [Pallás & Webster, 1991] *Sensors and Signal Conditioning.*
R. Pallás-Areny and J. G. Webster,
New York: Wiley, 1991.
- [Papkovsky et al., 1994] *Luminescence lifetime-based sensor for relative air humidity*
D. B. Papkovsky, G. V. Ponomarev, S. F. Chernov, A. N. Ovchinnikov, and I. N. Kurochkin,
Sens. Actuators B, Chem., vol. 22, no. 1, pp. 57–61,
Oct. 1994.

- [Pearce, 1998] *Strategies for Mimicking Olfaction: The Next Generation of Electronic Noses*
T.C. Pearce, J.W. Gardner, W. Göpel
Sensors Update, Vol. 3, Wiley-VCH, Weinheim, (1998), 61-130
- [Perera,2003] *Smart Instrumentation based on Gas Sensor Arrays: Novel Hardware Concepts and Algorithmic Issues*
Alexandre Perera i Lluna.(2003)
Tesis Doctoral. Universidad de Barcelona.
- [Pineda, 1987] *Generalization of backpropagation to recurrent neural networks*
F.J. Pineda. (1987)
Physical Review Letters 59, pp.2229-2232.
- [Pineda, 1988] *Generalization of backpropagation to recurrent and higher order neural networks*
F.J. Pineda. (1988)
Neural Information Processing Systems (D.Z. Anderson, ed.) pp. 602-611. New York: American Institute of Physics,
- [Piazena et al., 2002] *La penetración de la radiación solar en la columna de agua de las propiedades centrales subtropicales del Océano Atlántico-ópticos y las posibles consecuencias biológicas*
H. Piazena, E. Pérez-Rodríguez, D.-P. Haeder, F, López-Figueroa. (2002)
Deep-Sea Res.. (Top II. Stud. Oceanográfico.) 49 (17), pp. 3513-3528.
- [Pioggia, et al., 2007] *A composite sensor array impedentiometric electronic tongue: Part I.*
G. Pioggia, F. Di Francesco, A. Marchetti, M. Ferro, A. Ahluwalia.
Characterization. Biosensors and Bioelectronics, 22 (2007) pp. 2618-2623
- [Pyle, 1999] *Data Preparation for Data Mining*
Dorian Pyle
Morgan Kaufmann, Mar 15, (1999) pp . 88
- [Qasim et al., 2008] *Agent Based Fuzzy ARTMAP Neural Network for Classifying the Power Plant Performance.*
M. Qasim, Doos, Zouhair Al-Daoud, M. Suhair, Al-Thraa,

Jordan Journal of Mechanical and Industrial Engineering 2(3)
(2008) 123-129.

[Raahemi et al., 2008]

Classification of Peer-to-Peer traffic using incremental neural networks (Fuzzy ARTMAP).

B. Raahemi, A. Kouznetsov, A. Hayajneh, P. Rabinovitch,
Electrical and Computer Engineering.

Canadian Conference (2008) 719-724.

[Rajasekaran & Vijayalakshmi, 2000a]

Simplified Fuzzy ARTMAP as Pattern Recognizer.

S. Rajasekaran, G.A. Vijayalakshmi Pai,

J. Comput. Civ. Eng. 14(92) (2000) 92-100.

[Rajasekaran & Vijayalakshmi, 2000b]

Image recognition using simplified fuzzy ARTMAP augmented with a moment based feature extractor,

S. Rajasekaran, G.A. Vijayalakshmi Pai

International Journal of Pattern Recognition and Artificial Intelligence 14(8) (2000) 1081-1095.

[Rajasekaran, Vijayalakshmi, 2005]

Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis and Applications.

S. Rajasekaran, G. A. Vijayalakshmi Pai,

Ed. Prentice Hall. (2004)

[Ramón y Cajal, 1911]

Histología del sistema nervioso del hombre y de los vertebrados

Santiago Ramón y Cajal. (1911)

Consejo Superior de Investigaciones Científicas

Madrid,

[Ramuhalli et al., 1999]

Use of reliability measures to improve the performance of fuzzy ARTMAP networks,.

P. Ramuhalli, L. Udpa, S.S. Udpa,

Neural Networks International Joint Conference on, 6 (1999)

pp. 4015-4020.

[Ramuhalli, et al., 2000]

Fuzzy ARTMAP network with evolutionary learning, Acoustics.

P. Ramuhalli, R. Polikar, L. Udpa, S.S. Udpa,

Speech and Signal Processing, ICASSP '00. Proceedings IEEE

International Conference on, 6 (2000) pp. 3466-3469.

[Reverter & Casas, 2008]

Direct interface circuit for capacitive humidity sensors,”

F. Reverter and O. Casas, “

ens. Actuators A, Phys., vol. 143, no. 2, pp. 315–322,

May 2008.

- [Reverter & Pallás, 2005] *Direct Sensor-to-Microcontroller Interface Circuits.*
F. Reverter and R. Pallás-Areny,
Barcelona, Spain: Marcombo S.A., Sep. 2005.
- [RFID, 2005] “RFID white paper: Technology, systems and applications,”
GermanAssoc. Inf. Technol., Commun. New Media, Berlin,
Germany, 2005.
- [Rich. & Knight, 1991] *Artificial Intelligence,*
E. Rich and K. Knight. (1001)
2nd ed. U. S. of America: McGraw-Hill, Inc.,
- [Rida et al., 2007] *Paper-based ultra-low-cost integrated RFID tags
for sensing and tracking applications*
A. Rida, R. Vyas, S. Basat, A. Ferrer-Vidal, L. Yang, S. K.
Bhattacharya, and M. M. Tentzeris,
- [Roman et al., 1995] *A capacitive-type humidity sensor using crosslinked
poly(methyl methacrylate-co-(2 hydroxypropyl)-methacrylate),”*
C. Roman, O. Bodea, N. Prodan, A. Levi, E. Cordos, and
I. Manovicu, “
Sens. Actuators B, Chem., vol. 25, no. 1–3, pp. 710–713,
Apr. 1995.
- [Roseblatt, 1958] *The Perceptron: A probabilistic model for información
storage and organization in the brain.*
F. Roseblatt. (1958)
Psychol. Rev 656, pp. 386-408,
- [Rosenblatt, 1962] *Principles of Neurodynamics.*
F. Rosenblatt. (1962)
Washington, DC: Spartan Books.
- [Rumelhart & Hinton, 1986] *In Parallel Distributed Processing: Explorations in the
Microstructure of Cognition.*
D. E. Rumelhart, G. E. Hinton, G. E. & R. J. Williams. (1986)
Vol. 1: Foundations (eds Rumelhart, D. E. & McClelland,
J. L., p. 318–362
MIT, Cambridge,.
- [Russell & Norvig, 2003] *Artificial Intelligence: A Modern Approach*
S. Russell and P. Norvig, (2003)
2nd ed. U. S. of America: Pearson Education, Inc.

- [Sakai et al., 1996] *Humidity sensors based on polymer thin films,*”
Y. Sakai, Y. Sadaoka, and M. Matsuguchi,
Sens. Actuators B, Chem., vol. 35, no. 1–3, pp. 85–90,
Sep. 1996
- [Sakai et al., 2000] *Evaluation of water quality and pollution using
multichanel sensors.*
H., S. Liyama, K. Toko.
Sensors and Actuators B, 66 (2000), pp. 251-255
- [Salah &Trinder, 2001]
*Fuzzy ARTMAP Neural Networks for Automatic Feature
Extraction from Aerial Images and Lidar Data,*
M. Salah, J C Trinder,
Sciencemcmasterca (2001) pp. 1-10.
- [Sánchez et al., 2010]
*An amperometric method for the detection of amitrole, glyphosate
and its aminomethyl-phosphonic acid metabolite in environmental
waters using passive samplers.*
F. Sánchez-Bayo, R.V: Hyne, K.L. Desseille.
Anal.Chim.Acta 675 (2010) pp. 125 – 131.
- [Sankir, 2005] *Flexible Electronics: Materials and Device Fabrication.*
N. D. Sankir
Blacksburg, VA: Virginia Polytech. Inst. State Univ.,
Dec. 2005.
Proc. IEEE-ECTC Symp., San Diego, CA, (2007), pp. 1977–1980.
- [Sawchuk et al., 2006]
*Influence of a Range of Dosages of MCPA, Glyphosate, and
Thifensulfuron:Tribenuron (2:1) on Conventional Canola
(Brassica napus) and White Bean(Phaseolus vulgaris) Growth
and Yield.*
J.W. Sawchuk, R.C.V Van Acker, L.F. Friesen.
Weed Technol. 20 (2006) pp. 184-197.
- [Serra & Ventura, 2003]
*“Flavour index and aroma profiles of fresh and processed
honeys”.*
Serra-Bonvehí, J., & Ventura-Coll, F.
Journal of the Science of Food Agriculture, 83, (2003)
pp. 275–282.
- [Serrano et al., 1998]
*Adaptive Resonance Theory Microchips, Circuit Design
Techniques.*
T. Serrano-Gotarredona, B. Linares-Barranco, A.G. Andreou,

- The Springer International Series in Engineering and Computer Science 456 (1998).*
- [Sierra et al., 2008] *Electrooxidation of glyphosate on nickel and copper electrodes.*
E.V. Sierra, M.A. Méndez, V.M. Sarria, M.T. Cortés.
Quimica Nova 31 (2008).
- [Songa et al., 2009] *Electrochemical detection of glyphosate herbicide using horseradish peroxidase immobilized on sulfonated polymer matrix.*
E.A. Songa, O.A. Arotiva, J.H.O. Owino, N. Jahed, P.G.L. Baker, E.I. Iwuoha.
Bioelectrochemistry 75 (2009) pp. 117 – 123.
- [Soria et al., 2005] *“Estimation of the honey dew ratio in honey samples from their physicochemical data and from their volatile composition obtained by SPME and GC–MS”.*
Soria, A. C., Gonzalez, M., De Lorenzo, C., Martínez-Castro, I. & Sanz, J.,
Journal of Science and Food Agriculture 85, (2005) 817-824.
- [Soto et al., 2008] *A model for the assessment of interfering processes in Faradic electrodes.*
Juan Soto, Roberto H. Labrador, M.D. Marcos, Ramón Martínez-Máñez, Carmen Coll, Eduardo García-Breijo, Luis Gil.
- [Subhajini, & Santhanam, 2011]
Fuzzy ARTMAPNEURAL network achitecture for weather forecasting.
A.C. Subhajini, T. Santhanam
Journal of Theoretical and Applied Information Technology 34 (1) (2011) pp. 022 – 028.
- [Sumithradevi et al., 2007]
Evaluation of Fuzzy ARTMAP with DBSCAN in VLSI Application,
K. A. Sumithradevi, M.N. Vijayalakshmi, A. Annamma, Vasanta
World Academy of Science, Engineering and Technology 36(2007) pp. 261-264.
- [Swets, 1986] *Indices of Discrimination or Diagnostic Accuracy: Their ROCs and Implied Models.*
Swets, J. A. (1986)
Psychological Bulletin, 99(1), pp. 100-117.
- [Swets, 1988]. *Measuring the accuracy of diagnostic systems.*
Swets, J. A. (1988):
Science, 240, pp. 1285-1293.
- [Swets et al., 2000] *Better decisions through science*
Swets, J. A., Dawes, R. M., & Monahan, J. (2000).

Scientific American, 283, 82-87.
www.psychologicalscience.org/newsresearch/publications/journals/asiam.pdf

[Tetsuhiro & Naonobu, 1999]

Relación entre observados y estimados concentraciones de clorofila a partir de la profundidad de Secchi en el Pacífico Central subártico.

T. Kobari, Tetsuhiro A., S. Naonobu, (1999)

Toro. Fac.. Pescado. Universidad de Hokkaido. 50 (3), pp.171 - 174.

[Vakil & Pavešić, 2003]

A Fast Simplified Fuzzy ARTMAP Network.

M. Vakil-Baghmisheh, N. Pavešić,

Neural Processing Letters 17 (3). (2003) 273-316.

[Verrelli et al., 2007]

Development of silicon-based potentiometric sensors: Towards a miniaturized electronic tongue.

G. Verrelli, L. Francioso, R. Paolesse, P. Siciliano, C. Di Natale, A. D'Amico, A. Logrieco.

Sensors and Actuators B: 123 (2007) pp. 191-19

[Vilakazi & Marwala, 2006]

Application of Feature Selection and Fuzzy ARTMAP to Intrusion Detection, Systems, Man and Cybernetics.

C.C. Vilakazi, T. Marwala,

IEEE International Conference on ,6 (2006) 4880-4885.

[Vinaixa et al., 2005]

A fuzzy ARTMAP- and PLS-based MS e-nose for the qualitative and quantitative assessment of rancidity in crisps,

M. Vinaixa, E. Llobet, J. Brezmes, X. Vilanova, X. Correig,

Sensors and Actuators B: Chemical 106(2) (2005) 677-686

[Vlasov, 2005]

Nonspecific sensor arrays ("electronic tongue") for chemical analysis of liquids (IUPAC technical report)

Y. Vlasov, A. Legin, A. Rudnitskaya, C. Di Natale, A. D'amico

Pure and applied chemistry, 77, (2005), 1966-1983

[Vuskovic & Sijiang, 2002]

Classification of Prehensile EMG Patterns With Simplified Fuzzy ARTMAP Networks,

M. Vuskovic, D. Sijiang,

Proceedings of the 2002 International Joint Conference on Neural Networks (2002) 2539-2544

[Wei et al., 2009] *Technique potential for classification of honey by electronic tongue.*
Wei, Z.; Wang, J.; & Liao, W.
Journal of Food Engineering, 94(3-4), (2009) pp. 260–266.

[Wei & Wang, 2011]
Classification of monofloral honeys by voltammetric electronic tongue with chemometrics method.
Zhenbo Wei, Jun Wang.
Electrochimica Acta 56 (2011) 4907–4915

[Widrow & Hoff, 1960]
Adaptive switching circuits
B. Widrow, M.E. Hoff
Neurocomputing, pp.126-134. J. Anderson and E. Rosenfield (Eds.)
(1960)
MIT Press, Cambridge, MA.

[Widrow & Winter, 1988]
Neural Nets for adaptive filtering and adaptive pattern recognition
B. Widrow, R.G. Winter (1988)
IEEE Computer, pp.25-39, Mar.

[Williamson, 1996]
Gaussian ARTMAP: A neural network for fast incremental learning of noisy multidimensional maps.
R. Williamson.
Neural Networks 9(5)(1996) 881-897.

[Winqvist et al., 2002]
Electronic Tongues and Combinations of Artificial Senses.
F. Winqvist, C. Krantz-Rülker, I. Lunström I.
Sensors Update, 11 (2002) pp. 279-306 Ed. Wiley

[Xu et al., 2009] *Pattern recognition for sensor array signals using Fuzzy ARTMAP,*
Zhe Xu, Xiajing Shi, Lingyan Wang, Jin Luo, Chuan-Jian Zhong,
Susan Lu,
Sensors and Actuators B: Chemical, 141(2) (2009) 458-464.

[Yaffe et al., 2002] *Fuzzy ARTMAP and Back-Propagation Neural Networks Based Quantitative Structure–Property Relationships (QSPRs) for Octanol–Water Partition Coefficient of Organic Compounds,*
D. Yaffe, Y. Cohen, G. Espinosa, A. Arenas, F. Giralt,
Journal of Chemical Information and Computer Sciences 42 (2)
(2002) 162-183.

- [Zampolli et al., 2004] *An electronic nose based on solid state sensor arrays for low-cost indoor air quality monitoring applications*
S. Zampolli, I. Elmi, F. Ahmed, M. Passini, G. C. Cardinali, S. Nicoletti, L. Dori.
Sensors and Actuators B, 102 (2004) 117-125
- [Zhe, 2008] *Fuzzy ARTMAP Based Neurocomputational Spatial Uncertainty Measures*,
L. Zhe,
Photogrammetric engineering & remote sensing (2008) 1573-1584.
- [Zou, 2002], *Receiver operating characteristic (ROC) literature research*
Zou, K. H. (2002).
<http://splweb.bwh.harvard.edu:8000/pages/ppl/zou/roc.html>.
- [Zurada, 1992] *Introduction To Artificial Neural Systems*,
J.M. Zurada, (1992)