

CAPITULO 4:

PROCESADO DE SEÑAL

4.1 INTRODUCCIÓN

Una de las piezas angulares en toda nariz electrónica la constituye el conjunto de técnicas que tratan la información obtenida a través de la matriz de sensores químicos. De hecho, algunos fabricantes catalogan como narices electrónicas instrumentos de medida tradicionales, como los espectrómetros de masas, a los que añaden un software de análisis multivariante que funciona sobre un ordenador personal que se acopla al equipo [4.1].

El procesado de señal multivariante que se integra en las narices electrónicas actuales tiene su origen en la química analítica, también denominada “quimiometría”. A las técnicas como el Principal Component Analysis (PCA) o el Partial Least Squares (PLS), consideradas ya tradicionales, se han ido añadiendo nuevos algoritmos que se engloban dentro del campo denominado genéricamente “Reconocimiento de patrones” o “Técnicas de inteligencia artificial”. Dentro de esta última denominación englobaríamos a los diferentes algoritmos que implementan redes neuronales artificiales. Generalmente, estas técnicas se programan en un ordenador que además controla todo el proceso de medida y archiva convenientemente los datos adquiridos en cada experiencia. En el caso de las narices portátiles, los algoritmos inteligentes son implementados en microcontroladores, ya que la potencia de cálculo necesaria en la

mayoría de aplicaciones no es muy elevada debido a que no es necesario dar la respuesta inmediatamente, si no que se pueden tolerar algunos segundos o incluso minutos de retardo.

Todos estos algoritmos pueden ser sometidos a diferentes clasificaciones atendiendo a sus características. Así podemos distinguir entre algoritmos supervisados y no supervisados si nos fijamos en el proceso de aprendizaje, algoritmos de clasificación o de cuantificación en función de la naturaleza (binaria o analógica) de la respuesta que deben proporcionar a cada estímulo de entrada, lineales o no lineales según las operaciones que realicen, paramétricos o no paramétricos en función de si se hacen suposiciones iniciales sobre el proceso a modelar, etc. Debido a los diferentes objetivos que se han perseguido en cada experiencia, en este trabajo de investigación se han probado diferentes técnicas multivariantes. La tabla 4.1 resume las principales características de los algoritmos aplicados.

Técnica	Algoritmo	Aprendizaje	Paramétrica	Aplicación
PCA	Lineal	No supervisado	No	Clasificación
PLS	Lineal	Supervisado	Si	Cuantificación
Feedforward-backpropagation ANN	Neuronal	Supervisado	No	Clasificación/ cuantificación
Fuzzy Art	Neuronal	No supervisado	No	Clasificación
Fuzzy Artmap	Neuronal	Supervisado	No	Clasificación

Tabla 4.1: Principales características de los algoritmos aplicados al los sistemas de olfato diseñados

La evolución de las narices electrónicas va ligada a la de las técnicas que se han aplicado para procesar los datos obtenidos en cada medición. Desde sus comienzos diversos autores han presentado resultados aplicando diferentes técnicas de procesado. Al principio se utilizaron algoritmos lineales ya conocidos en química analítica. Por ejemplo, en [4.2] se describen los resultados obtenidos en la clasificación de diferentes tipos de café utilizando funciones discriminantes, mientras que en [4.3] se describe la clasificación cervezas y whiskies japoneses utilizando análisis de componentes principales.

De todas maneras, la detección de mezclas gaseosas (aromas, por ejemplo) va ligada a la aparición de interacciones no lineales entre los diferentes compuestos que reaccionan con los sensores. Para procesar esta información es conveniente utilizar algoritmos que no requieran linealidad en los datos y que no hagan suposiciones a priori sobre los procesos que rigen dichas interacciones. Como las redes neuronales con tres capas presentaban suficientes grados de libertad para solucionar cualquier problema de clasificación, muchos autores se dedicaron a estudiar su aplicación a los sistemas de olfato electrónico [4.4]. Otra de las ventajas que mostraban era su capacidad para cuantificar la concentración de diferentes compuestos en una mezcla gaseosa. En [4.5] se puede encontrar uno de los trabajos pioneros en los que se explotaba esta capacidad y en [4.6] uno de los primeros trabajos en los que se intenta discernir entre diferentes perfumes utilizando redes neuronales.

Lejos de ser un tema de investigación agotado, el procesado de señal en sistemas de olfato electrónico sigue siendo un activo campo de investigación impulsado por el constante abaratamiento de los sistemas informáticos que los soportan. Cada año se proponen nuevos métodos que pretenden eliminar o reducir las limitaciones que presentan los sensores químicos. Por ejemplo, en [4.7] se presenta uno de los primeros trabajos que aborda el tema de la calibración y en [4.8] el autor de esta tesis presenta un estudio de cómo una red neuronal puede neutralizar la interferencia de la humedad en el funcionamiento de los sensores de óxidos metálicos. Otras redes neuronales como las de resonancia adaptativa [4.9] y técnicas de procesado que explotan los parámetros dinámicos de la respuesta de los sensores son tendencias con gran futuro en el mundo de los sistemas de olfato electrónico. Para mayor información, se recomienda al lector que consulte la referencia [4.10] en la que se presenta un completo repaso de todas las técnicas utilizadas hasta la fecha además de valorar las diferentes líneas de trabajo que deben definir el futuro del procesado de señal en sistemas de olfato electrónico.

En los equipos de olfato electrónico, el objetivo principal del procesado de señal es el de extraer la información relevante para obtener la respuesta deseada, minimizando las interferencias (ruido) que pueda introducir la matriz de sensores. En el caso de la fruta, estas técnicas deben realizar dos funciones claramente diferenciadas:

- Clasificar muestras en función de su estado de maduración. Para ello se pueden aplicar cualquiera de las técnicas que clasifican las medidas a partir de la información multidimensional obtenida con la nariz electrónica. Las técnicas no supervisadas permiten ver la similitud entre medidas sin imponer ningún criterio, mientras que las supervisadas permiten clasificar atendiendo a un criterio previamente establecido en la fase de entrenamiento.
- Estimar el valor de diferentes indicadores de calidad de forma no destructiva a partir de la señal que se obtiene en la matriz de sensores. En este caso es necesario aplicar técnicas de cuantificación multivariantes como el PLS o las redes neuronales feed-forward que aprenden a relacionar señales en los sensores (generadas por los volátiles de la fruta) con condiciones físico-químicas en las piezas de fruta analizadas.

En este capítulo, cada subapartado describe la utilización de un algoritmo diferente, dando especial relevancia a los aspectos novedosos que se hayan podido derivar de la adaptación de técnicas conocidas a la aplicación que nos ocupa. Dependiendo del algoritmo y su utilización, los aspectos novedosos recaen en la modificación del propio algoritmo, en la evaluación de resultados, en el modo de utilización, etc. Así, por ejemplo, el PCA se ha utilizado de forma innovadora en la detección de derivas en los sensores. Otras técnicas, mucho más novedosas (como Fuzzy Artmap), han tenido que ser adaptadas internamente para un funcionamiento adecuado en el campo de las narices electrónicas.

Por otro lado, no hay que olvidar que la aplicación de técnicas de reconocimiento de patrones requieren un preprocesado previo mediante el que se extraiga y normalice la información relevante que suministra la matriz de sensores químicos. Este paso puede ser crucial en el éxito de la aplicación y debe ser tratado con sumo cuidado. Por este motivo se ha incluido un subapartado en el que se explican con detalle todos los pasos previos a la aplicación de técnicas multivariantes, que incluyen desde la extracción de parámetros relevantes hasta la selección de variables.

Todos estos aspectos han llevado a organizar este capítulo en 7 subapartados en los que se intenta cubrir todos los aspectos referentes al tratamiento de la señal que se han implementado en esta tesis. Así, en los dos siguientes subapartados se tratarán las técnicas lineales (PCA y PLS respectivamente). El subapartado 4 inicia el estudio de redes neuronales con la topología feedforward y entrenamiento backpropagation; los apartados 5 y 6 describen las redes Fuzzy Art y Fuzzy Artmap y, finalmente, en el apartado 7 se presentan las diferentes técnicas y estrategias de preprocesado que se han aplicado para optimizar los resultados.

4.2 ANÁLISIS DE COMPONENTES PRINCIPALES (PCA)

4.2.1 Introducción

El análisis de componentes principales es una técnica originalmente propuesta en 1981 [4.11] que surgió como respuesta a la creciente cantidad de datos que podían ser obtenidos en cada medida gracias a los instrumentos de laboratorio de nueva generación [4.12]. Así, por ejemplo, un espectrómetro típico FTIR puede proporcionar datos característicos en 10000 longitudes de onda diferentes para cada medida. Esta nueva situación crea una saturación de información cuya consecuencia más probable es la incorrecta extracción de la información que es realmente relevante para la descripción del experimento.

En definitiva, la necesidad que surge con el nuevo instrumental de laboratorio (incluyendo la nariz electrónica) es doble: es necesario comprimir y es necesario extraer toda información relevante del voluminoso conjunto de datos obtenido ya que en muchas ocasiones la información esencial no depende de variables aisladas sino de la interrelación entre las mismas. El algoritmo PCA aborda estos problemas y por ese motivo es una de las técnicas más utilizadas por los químicos analíticos y, por extensión, por todos aquellos investigadores que trabajan con narices electrónicas ya

que permite reducir, representar y extraer información relevante al mismo tiempo [4.13].

PCA es un método que asume colinearidad entre las variables que intervienen. En otras palabras, se trata de un algoritmo lineal que puede funcionar incorrectamente en procesos altamente no lineales como pueden ser las interacciones químicas entre sensores y compuestos volátiles. De todas formas, funciona sorprendentemente bien en muchas aplicaciones con narices electrónicas, en las que las concentraciones de volátiles no son muy elevadas y el comportamiento de los sensores no es excesivamente alineal.

La compresión de datos y extracción de información relevante se hace más necesaria en aquellas situaciones en las que existe una falta de selectividad en cada uno de los sensores que componen la matriz. Por ese motivo el análisis de componentes principales es un método idóneo para explotar el concepto de sensibilidades solapadas que se aplica en la mayoría de las narices electrónicas.

El algoritmo PCA suele ser clasificado como un método no supervisado de reconocimiento de patrones, ya que su uso más extendido con las narices electrónicas se limita a representar bidimensionalmente un conjunto de medidas para ver si se pueden determinar agrupaciones (“clusters”) espontáneas entre las diferentes medidas realizadas previamente, aunque, como veremos posteriormente, también existen modificaciones que permiten aplicar el algoritmo en modo supervisado.

4.2.2 Algoritmo

Para aplicar el análisis de componentes principales se parte de un conjunto de datos dispuestos en forma matricial con una estructura de m filas por n columnas, suponiendo que se han realizado m experiencias y cada una se ha descrito con n variables. Así, cada fila corresponde a una descripción completa de un experimento y cada columna a una variable concreta que se utiliza como indicador en n experimentos. Para poder aplicar el algoritmo PCA, las columnas de esta matriz (las variables) deben estar centradas (“mean centered”).

Es lícito considerar, por lo tanto, que cada experimento o fila es un vector que pertenece a un espacio vectorial de dimensión n expresado en base canónica (en la que cada una de las coordenadas está asociada directamente a una de las variables descriptivas que proporciona el instrumento de medida). El análisis de componentes principales es un endomorfismo o cambio de base, pasando de la base canónica a una nueva base formada por los autovectores de la matriz de covarianza de los datos. Matemáticamente:

$$E_n(X) \xrightarrow{f} E_n(Y)$$
$$X = t_1 p_1^T + t_2 p_2^T + t_3 p_3^T + t_4 p_4^T \dots + t_k p_k^T \quad (4.1)$$
$$Y = [t_1 \ t_2 \ t_3 \ t_4 \ \dots \ t_k]$$

donde:

- X es la matriz de datos original (dimensión $m \times n$)
- t_k es el vector de coordenadas de las medidas sobre la componente principal k -ésima (denominado “score”)
- p_k es el vector de la componente principal k (denominado “load”), que es autovector de la matriz de covarianzas de X y ortogonal al resto de autovectores
- Y es la matriz de coordenadas de las n medidas referida a la nueva base de componentes principales (dimensión $m \times n$)

En definitiva, los “scores” son las coordenadas de cada una de las medidas respecto a la nueva base, por lo que de alguna manera especifican las relaciones existentes entre experimentos. Por el contrario, los “loads” definen la contribución de cada variable original sobre las componentes principales, aportando información sobre la interrelación entre las variables originales.

Lo que realmente interesa de esta transformación es que los autovectores de la matriz de covarianzas de X marcan los ejes de máxima varianza de las medidas. Así, el primer vector, llamado componente principal 1, contiene la mayor proporción posible de la varianza inicial en las medidas, la segunda componente principal, la mayor varianza residual (con la restricción de ser ortogonal a la componente principal 1), etc.

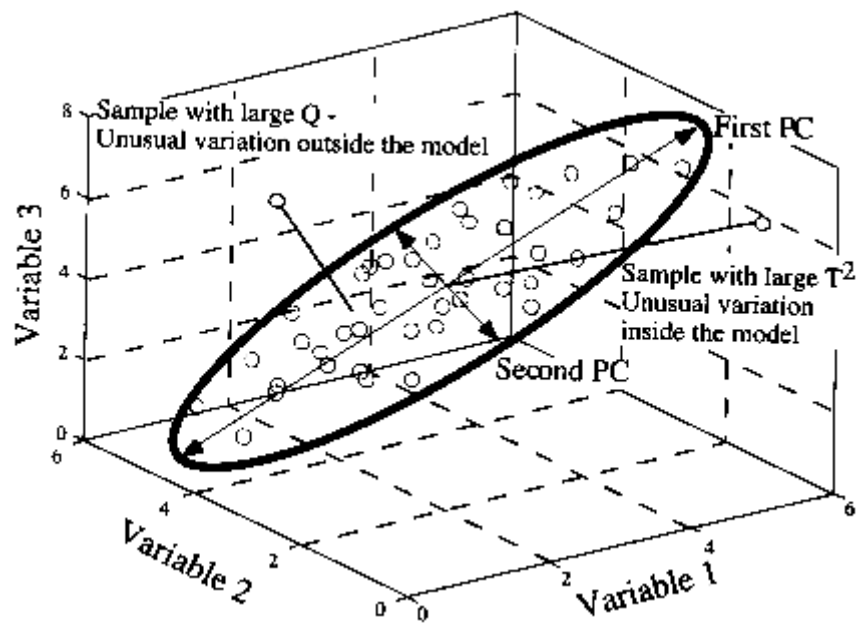


Figura 4.1: Ejemplo gráfico del algoritmo PCA

En principio, la dimensión de la nueva matriz es idéntica a la de la original, pero podemos reducir la dimensionalidad conservando únicamente las coordenadas (“scores”) respecto a las primeras componentes principales, que son las que marcan las direcciones de máxima varianza (y, por tanto, de máxima información). Eso hace posible la representación gráfica en dos dimensiones de datos multidimensionales asegurándonos de que esa representación capturará la máxima varianza posible del conjunto de datos en dos dimensiones. La figura 4.1 ilustra este concepto, en el que se puede observar que, partiendo de un conjunto de datos tridimensionales, se proyectan las medidas sobre un plano bidimensional definido por las dos primeras componentes principales. A la proyección bidimensional resultante se le denomina gráfica de “scores”, mientras que a la proyección de las variables originales sobre la nueva base se le denomina gráfica de “loads”.

En el algoritmo PCA existen una serie de parámetros y variables que permiten determinar el número óptimo de componentes principales para describir correctamente un conjunto de datos, además de algún indicador que permite identificar variaciones inusuales dentro del modelo de alguna medida, lo cual puede indicar la presencia de un “outlier” o experimento erróneo dentro de la rutina de medidas realizadas. Puesto que el

PCA es un método ampliamente conocido, no nos extenderemos mas en su descripción algorítmica.

4.2.3 Implementación

4.2.3.1 Programación

Los algoritmos PCA utilizados en los prototipos de nariz electrónica diseñados en esta tesis doctoral se han implementado en el entorno de programación que ofrece en el paquete informático MATLAB [4.14]. Las subrutinas básicas PCA se encuentran en un paquete multivariante denominado PLS-Toolbox 2.0 [4.15] que se acopla al programa principal. En concreto, existen dos subrutinas, *pca* (versión clásica) y *pcapro* (PCA en modo supervisado).

Sin embargo, estas funciones son excesivamente generales para los objetivos de nuestro trabajo. Ha sido necesario programar una serie de funciones auxiliares para facilitar la integración y evaluación de estos algoritmos en los objetivos planteados para cada variedad de fruta.

En concreto, ha sido necesario remodelar la representación gráfica de resultados para poder asignar códigos numéricos y de color para distinguir correctamente entre multitud de medidas diferentes en cada aplicación. Para ello se han programado algunas funciones gráficas adicionales para representar los datos con mayor claridad, cuyo formato se irá observando en las figuras de resultados en los capítulos 5,6 y 7.

Aunque originalmente catalogado como método no supervisado, el algoritmo PCA puede ser también utilizado como un método supervisado. En todo algoritmo supervisado puede distinguirse una etapa de aprendizaje (entrenamiento o calibración) y una etapa de evaluación o test. En este modo de trabajo, el PCA determina, en la fase de calibración, las componentes principales con un conjunto de medidas de entrenamiento. Las medidas de aprendizaje son proyectadas y se identifican las agrupaciones que corresponden a cada una de las sustancias que se han utilizado para calibrar al equipo. Posteriormente, en la fase de evaluación, las nuevas medidas se proyectan sobre las

componentes principales del modelo PCA generado anteriormente. Dependiendo de la posición de su proyección sobre el diagrama PCA se relacionará la nueva medida con alguna de las sustancias utilizadas en la fase de calibración (previsiblemente sobre la agrupación más cercana a su proyección). Este modo de utilización se ha programado integrando la función *pcapro* a un conjunto de instrucciones más amplio para acoplar su uso a diferentes objetivos del estudio.

4.2.3.2 Mejoras del algoritmo

Uno de los problemas que generalmente se achaca al análisis de componentes principales es la falta de criterio al escoger las direcciones de máxima varianza. Esto es debido a que el método no distingue entre varianza debida a la falta de repetitividad de los sensores (incluyendo posibles derivas) y la debida a información relevante. En este proyecto se ha trabajado sobre una modificación del análisis con el objetivo de reducir esta deficiencia.

La novedad radica en el cálculo de las direcciones de máxima varianza. En vez de utilizarse una serie de medidas de entrenamiento se utiliza una medida representativa de cada clase que se calcula como el valor medio de todas las medidas de entrenamiento que pertenecen a dicha clase. De esta forma el método calcula las direcciones que proporcionarán la máxima separación posible entre centroides de diferentes clases. Si el número de medidas es suficientemente elevado, factores como el ruido serán minimizados y la varianza capturada será debida, fundamentalmente, a diferencias intrínsecas a cada clase de medidas.

Se realizaron diferentes pruebas y, en todas ellas, los resultados solo mejoraron ligeramente. La gran desventaja de esta aproximación radica en que es necesario asignar correctamente las medidas de entrenamiento a una determinada agrupación, algo difícil debido a la gran variabilidad entre muestras de fruta sometidas a unas mismas condiciones. Además, limita el uso del PCA a un modo de aplicación supervisado. Por este motivo, a pesar de haber hecho diversas pruebas, no se llegó a aplicar la modificación a medidas con fruta.

4.2.3.3 Evaluación de resultados

Para valorar la resolución obtenida con la nariz se ha tenido que definir un parámetro que indicase el poder de resolución en cada PCA ateniéndonos a las clases a las que cada medida pertenece realmente. Esta función basa su cálculo en la relación entre varianza intraclase y la varianza interclase. La varianza intraclase puede definirse como la distancia media, en el gráfico de la proyección, entre cada medida de una clase concreta y la posición de la centroide de dicha categoría. Por otro lado, la varianza interclase se calcula como la distancia media entre las centroides de todas las categorías consideradas en el experimento. La figura 4.2 y las ecuaciones 4.2, 4.3 y 4.4 detallan un cálculo de resolución a modo de ejemplo.

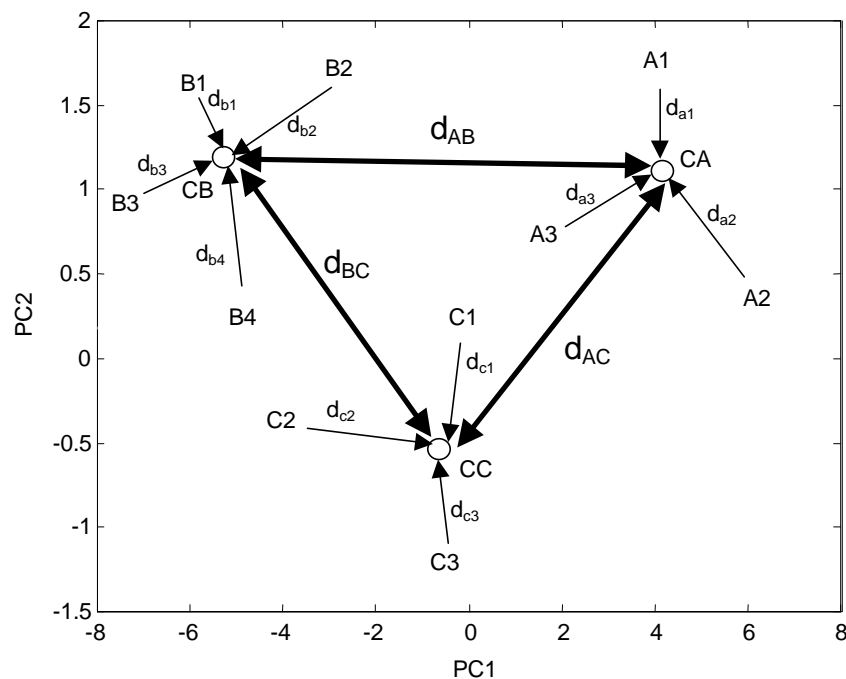


Figura 4.2: PCA con 10 medidas que pertenecen a tres clases diferentes. El siguiente recuadro ilustra el cálculo del poder de resolución

Los pasos detallados de este cálculo son los siguientes:

1. Cálculo de la centroide para cada una de las clases de medidas existentes en la proyección
2. Cálculo de la distancia media entre centroides (varianza interclase)
3. Cálculo, para cada clase, de la distancia media entre todas sus medidas y su centroide
4. Obtención del valor medio del cálculo anterior (varianza intraclase)
5. División de la varianza interclase por la varianza intraclase

Para cada clase, distancia intraclase media (varintraclase):

$$v_A = \frac{1}{3}(da1 + da2 + da3) \quad v_B = \frac{1}{4}(db1 + db2 + db3 + db4) \quad v_C = \frac{1}{3}(dc1 + dc2 + dc3) \quad (4.2)$$

$$\text{Distancia intraclase media: } vm = \frac{1}{3}(v_A + v_B + v_C) \quad (4.3)$$

$$\text{Distancia media interclases: } vmi = \frac{1}{3}(d_{AC} + d_{AB} + d_{BC}) \quad (4.4)$$

$$\text{Poder de resolución: } res = \frac{vmi}{vm} \quad (4.5)$$

4.2.4 Otras aplicaciones

Una de los problemas más acuciantes en las narices electrónicas actuales es la presencia de derivas en los sensores químicos del equipo. Aunque inicialmente pocos investigadores contemplaban su monitorización y control, lo cierto es que es de vital importancia tener en cuenta este efecto en toda aplicación en la que las medidas estén relacionadas con el paso del tiempo. El caso de la fruta es un ejemplo claro de esta situación. Durante su proceso de maduración, a medida que avanzan los días, la producción aromática crece monótonamente. Sin embargo, si los sensores aumentan su conductancia base debido a derivas temporales (envejecimiento, contaminación, etc), es

posible que su conductancia final se modifique de la misma forma, efecto que se puede atribuir erróneamente al proceso de maduración de la fruta.

Aunque existen algunos estudios en los que se intenta compensar este efecto [4.16,17] (alguno de los cuales esta relacionado con los diagramas PCA), lo cierto es que no se han utilizado de forma generalizada. En este trabajo se ha optado por monitorizar el proceso de medida y ver si existen derivas que puedan tener influencia en los resultados. Para ello se han utilizado los diagramas PCA y medidas de calibración a lo largo de las diferentes experiencias realizadas con fruta.

El método consiste en calcular el diagrama PCA con todas las medidas de fruta de una determinada experiencia, identificando las medidas con una etiqueta con el día en que se realizaron. A continuación, utilizando el método PCA en modo supervisado, se proyectan las medidas de calibración sobre las de la fruta, utilizando de esta manera las componentes principales obtenidas anteriormente con las medidas de fruta.

Idealmente, si no hubiesen derivas, las medidas de calibración deberían caer todas en un mismo punto del gráfico, ya que todas ellas se realizan en idénticas condiciones. Sin embargo, la falta de repetitividad de los sensores y las posibles derivas pueden repartir las proyecciones por diferentes zonas del gráfico.

Estudiando la evolución de las proyecciones de las medidas de calibración sobre las medidas de fruta es fácil determinar si las agrupaciones son debidas al proceso de maduración o a la deriva de los sensores de la nariz electrónica. Se remite al lector al capítulo 5 donde el apartado 5.4.3 y las figuras 5.6 y 5.7 muestran gráficamente un claro ejemplo de aplicación de cómo monitorizar derivas con este método. En ese apartado se realiza una discusión más profunda sobre las diferentes posibilidades del mismo.

4.3 PARTIAL LEAST SQUARES

4.3.1 Introducción

En esta tesis se han considerado diferentes algoritmos para correlacionar los datos de los sensores con los indicadores de calidad para la fruta. De entre los lineales se pueden destacar los algoritmos PCR (Principal Component Regression), PLS (Partial Least Squares) y MLS (Minimum Least Squares). PLS, el método finalmente escogido, puede considerarse un término medio entre PCR y MLS.

En el algoritmo PCR se buscan, al igual que en un PCA, las componentes principales que mejor describen la varianza de las variables predictoras. Con estas componentes principales es con las que se monta un modelo de correlación entre las variables predictoras y la predicha (en PCR solo se puede predecir una variable por modelo). En el espectro opuesto, MLR busca un factor simple que maximice la correlación con la variable a predecir.

El algoritmo PLS [4.18-21], tal y como ya se ha comentado, toma una aproximación intermedia. En general se puede afirmar que PLS intenta maximizar la covarianza. Para ello se vale de una serie de variables latentes (el equivalente a componentes principales) que capturan la máxima varianza posible y escoge, para montar el modelo de correlación, aquellas cuya covarianza con las variables a predecir sea máxima. Tomando este criterio PLS suele funcionar mejor que PCR o MLS en la mayoría de situaciones, motivo por el que se decidió utilizar este algoritmo para correlacionar la señal de los sensores con los indicadores de calidad de la fruta.

Como ya se ha comentado, PLS es un método lineal. Sin embargo, aplicando algunas modificaciones puede ser convertido en un algoritmo no lineal. En el PolyPLS [4.22,23], la modificación consiste en incrementar la matriz de datos con sucesivas potencias de las variables predictoras, de manera que lo que en un principio estaba restringido a relaciones lineales queda ahora ampliado a relaciones polinómicas del

orden que se decida. El único problema de este método es que aumenta el número de parámetros a ajustar y en aplicaciones con pocas medidas esto produce un sobreentrenamiento con escasa capacidad de generalización. Tras varias pruebas realizadas con medidas de fruta se comprobó que sistemáticamente se obtenían mejores resultados con el algoritmo PLS sin modificar, a pesar de ser exclusivamente lineal.

En resumen, PLS es un método de correlación multivariante entre una serie de variables de entrada (en nuestro caso las señales que provienen de los sensores de la nariz electrónica) y una serie de variables de salida (en nuestro caso indicadores de calidad de fruta). A diferencia de PCR, puede predecir varios parámetros de salida con un único modelo. Su funcionamiento es supervisado, de manera que en la fase de entrenamiento se crea un modelo que relaciona variables predictoras (variables de entrada) con variables a predecir (variables de salida) maximizando la covarianza entre ambos conjuntos. A partir de ese momento, en la fase de evaluación, PLS se convierte en una herramienta de predicción en la que se utiliza el modelo construido en la fase de entrenamiento junto con nuevas señales de entrada para predecir alguna de las variables de salida para la que ha sido entrenado.

4.3.2 Algoritmo

La manera de implementar el algoritmo de PLS ha sido objeto de numerosos estudios, por lo que no se considera apropiado describirlo con detalle en esta memoria. De todas maneras, a modo de reseña se puede comentar que en su implementación destacan dos algoritmos diferentes, el NIPALS (non-iterative Partial Least Squares), más intuitivo, y SIMPLS, algo más efectivo y fácil de calcular.

Por ese motivo, en este trabajo se ha optado por utilizar SIMPLS como algoritmo de cálculo de PLS. SIMPLS se basa en la ortogonalización de una secuencia de Krylov para calcular los pesos del modelo, estrategia poco intuitiva pero eficaz. Los resultados de este método son idénticos a los del algoritmo NIPALS en predicciones de una única variable y ligeramente mejores en predicción multivariante ya que explota mejor los criterios de covarianza.

4.3.3 Implementación

4.3.3.1 Programación

Al igual que con el algoritmo PCA, los algoritmos PLS utilizados corren bajo el entorno Matlab y son funciones ya programadas incluidas en el paquete de funciones opcional PLS Toolbox 2.0. En concreto, la función que se utiliza se denomina *simpls*. Esta función se utiliza para crear el modelo PLS a través del método SIMPLS. Requiere como entrada la matriz de medidas con las variables de entrada, la matriz con los valores correctos de las variables a predecir y el número máximo de variables latentes a considerar. La salida devuelve, básicamente, una matriz con los coeficientes de predicción del modelo y una serie de valores que informan sobre lo ajustado que ha quedado el entrenamiento.

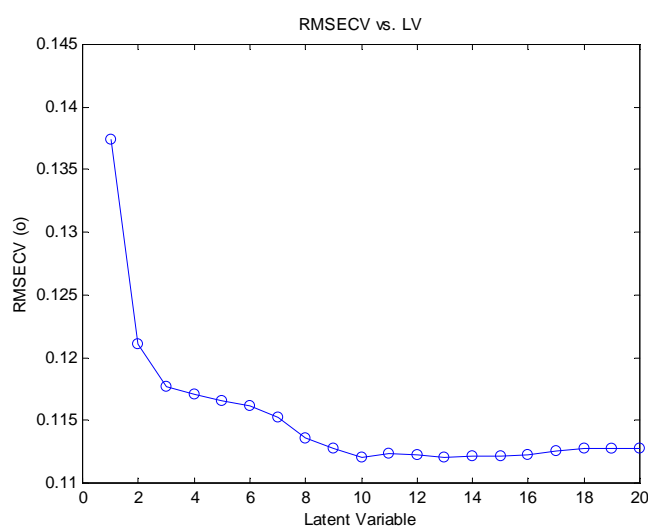


Figura 4.3: Estudio del número óptimo de variables latentes por la función *simpls*

El algoritmo determina el número óptimo de variables latentes realizando una validación cruzada en la que calcula el error cuadrático medio (RMSECV) con un número creciente de variables latentes. La figura 4.3 muestra un gráfico típico de resultados en el que se observa que, para el ejemplo en cuestión, el modelo óptimo queda creado con 10 variables latentes.

La aplicación del modelo a predicciones posteriores, en las que las medidas son diferentes a las utilizadas en entrenamiento y en las que se desconoce el valor de las variables a predecir, es realmente simple. Suponiendo que la matriz de coeficientes del modelo resultante se denomine B , la matriz con nuevos datos de medidas X_N , y la matriz de salida con las correspondientes predicciones Y_N , la forma de obtener las nuevas predicciones es tan simple como indica la ecuación 4.6:

$$Y_N = X_N * B \quad (4.6)$$

4.3.3.2 Aplicación del método a las medidas de fruta

Para determinar la validez de los modelos generados mediante PLS es necesario realizar predicciones de medidas de las que se sabe el valor correcto de salida para poder comparar la estimación PLS con el valor real.

De todas maneras, para que un modelo PLS de buenos resultados es necesario que en la fase de entrenamiento se utilicen suficientes medidas como para tener una buena representación estadística del conjunto de entrada. Al mismo tiempo, resulta interesante poder evaluar un número de medidas elevado para asegurarnos de que el sistema realiza predicciones exactas. Teniendo en cuenta que la obtención de medidas de entrenamiento suele ser una tarea costosa, resulta sumamente interesante optimizar su uso en el proceso de evaluación. Por ese motivo, en todos los estudios de correlación en esta tesis se utiliza una validación cruzada (“crossvalidation”) para determinar la capacidad de predicción de la nariz electrónica a partir de las señales de los sensores.

Suponiendo que disponemos de m medidas, en una validación cruzada de orden n se extraen n medidas del conjunto y se entrena con las $m-n$ medidas restantes. Una vez obtenido el modelo se evalúan las n medidas que se han apartado previamente del conjunto de entrenamiento. Este proceso se repite m/n veces (suponiendo m múltiplo de n), de manera que cada medida se ha utilizado para entrenamiento y para evaluación.

En este trabajo, debido a la escasez de medidas en algunos de los experimentos, se han utilizado, en la mayoría de los casos, validaciones cruzadas de orden 1 (también denominadas “leave-one-out”), lo que supone haber generado tantos modelos PLS como medidas se disponen. Esta filosofía no está pensada para proponer un modelo PLS concreto para la predicción de cada parámetro de calidad, sino para probar la validez de la aproximación.

4.3.3.3 Evaluación de resultados

La evaluación de los resultados requiere de parámetros y algunas herramientas gráficas que permitan esclarecer y comparar con qué variables y métodos se obtienen las mejores predicciones.

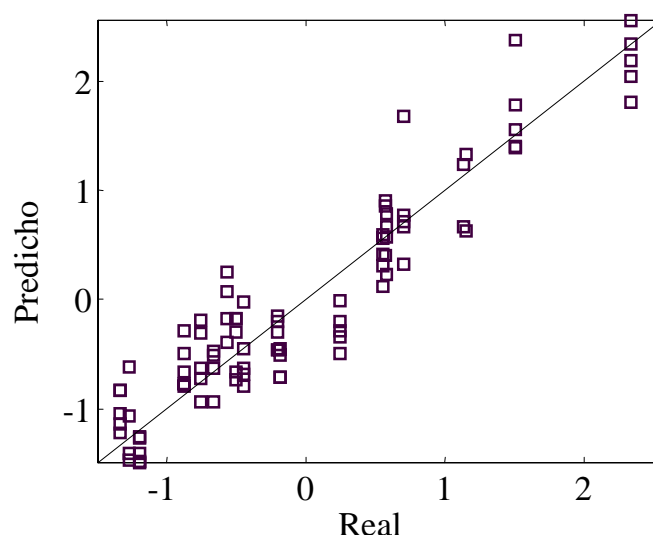


Figura 4.4: Resultado gráfico de predicción con modelos PLS

La figura 4.4 muestra una gráfica típica de resultados, en la que las medidas están representadas por recuadros. El eje horizontal muestra el valor real de la medida y el vertical el valor predicho. Una predicción perfecta pondría todos los puntos sobre la diagonal del gráfico, donde coincide valor real con valor predicho.

Mediante los puntos de la gráfica se puede hacer una regresión lineal, de la cual se puede obtener el punto de corte (parámetro “b”) y su pendiente (simbolizada por una

“m”). El coeficiente de correlación de dicha regresión nos informa de hasta que punto los resultados están alineados formando una recta. Por último, otro parámetro que nos puede dar una buena idea de la calidad de la predicción es el error cuadrático medio (“ssq”) que suma la diferencia entre el valor real y predicho de cada medida elevado al cuadrado. En una predicción perfecta, la pendiente de la recta m valdría 1, el punto de corte valdría 0, el coeficiente de correlación 1 y el error cuadrático medio 0.

4.4 REDES NEURONALES FEEDFORWARD

4.4.1 Introducción

Dentro del campo de la inteligencia artificial hace tiempo que las redes neuronales ocupan un lugar de privilegio [4.24]. Este tipo de algoritmos, cuya topología está inspirada en las redes neuronales biológicas [4.25], ajustan sus modelos de forma experimental, sin necesitar un conocimiento previo de las características de los datos que debe procesar ni de los mecanismos que los producen. Además, su forma matemática de operar no está restringida a relaciones y procesos lineales.

Dentro de los múltiples algoritmos de redes neuronales que se han propuesto hasta la fecha destaca la topología “feed-forward”, ya que fue la estructura que catapultó a las redes neuronales a su situación de privilegio actual. Además, responde perfectamente al paradigma de lo que se entiende hoy en día por una red neuronal artificial. De hecho, está demostrado que las redes neuronales feed-forward con entrenamiento backpropagation y una capa oculta son aproximadores universales [4.26]. Eso quiere decir que, con un entrenamiento adecuado, pueden interpolar cualquier relación entre el vector de entrada y el de salida, lo cual es ideal en situaciones en las que pueden haber procesos no lineales o en los que es imposible inferir una relación analítica.

Estas características hacen que este tipo de redes sean muy utilizadas como algoritmos de procesamiento de datos en las narices electrónicas [4.27]. Hay que tener en cuenta que las interacciones entre mezclas gaseosas complejas (los aromas pueden contener centenares

de componentes) y los sensores químicos son procesos extremadamente difíciles de comprender y modelar. Para mayor dificultad, la mayoría de ellos pueden presentar relaciones fuertemente no lineales, algo que no pueden digerir bien la mayoría de algoritmos clásicos de reconocimiento de patrones.

Sin embargo, para conseguir esa interpolación, el conjunto de medidas de entreno tiene que ser suficientemente representativo del conjunto de posibles vectores de entrada, lo que implica que sin un número considerable de medidas de entrenamiento no se puede realizar una interpolación con los niveles de exactitud deseables. Es este aspecto el que limita el uso de las redes neuronales cuando el número de medidas realizadas no es muy elevado. Por otro lado, el entrenamiento de las redes con el algoritmo backpropagation es computacionalmente muy costoso, por lo que los métodos de cross-validation que se pueden utilizar suelen ser de órdenes elevados, de manera que queden disminuidas el número de iteraciones. En definitiva, las dos grandes limitaciones de este tipo de redes son la necesidad de un amplio conjunto de medidas de entreno y la potencia de cálculo necesaria para ejecutar la fase de entrenamiento.

4.4.2 Algoritmo

4.4.2.1 Estructura

La figura 4.5 muestra el esquema básico de esta estructura. Toda red neuronal puede considerarse como un sistema de computación compuesto por una serie de elementos más simples (neuronas) que se interconectan entre sí. En las redes “feed-forward” las conexiones son de izquierda a derecha (“hacia adelante”). La señal de salida de una neurona sirve de señal de entrada a todas las neuronas de la columna inmediatamente a su derecha, existiendo unos pesos que ponderan esa señal.

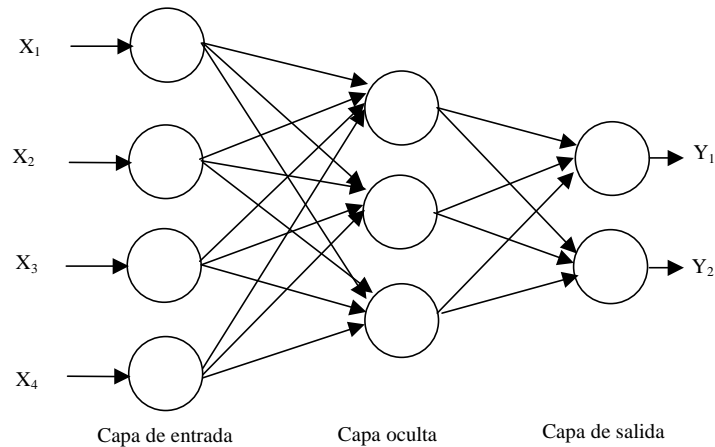


Figura 4.5: Estructura básica de una red feed-forward

Las redes que se han utilizado en este proyecto son de tres capas, considerando las señales de entrada como una capa. Las otras dos son una capa oculta y la capa de salida. La figura 4.6 muestra como trabaja una neurona. La señal total de entrada es una combinación lineal de todas las salidas de las neuronas de la capa anterior más una componente continua o “bías”, estando multiplicada cada una de las salidas por un coeficiente o peso. Una vez se ha calculado la señal de entrada total, se aplica una función de activación, generalmente no lineal, que da como resultado la salida de dicha neurona. Existen múltiples funciones de activación (la única restricción es que deben ser funciones continuamente derivables) y en este proyecto se ha escogido la función sigmoide que se detalla en la figura 4.7.

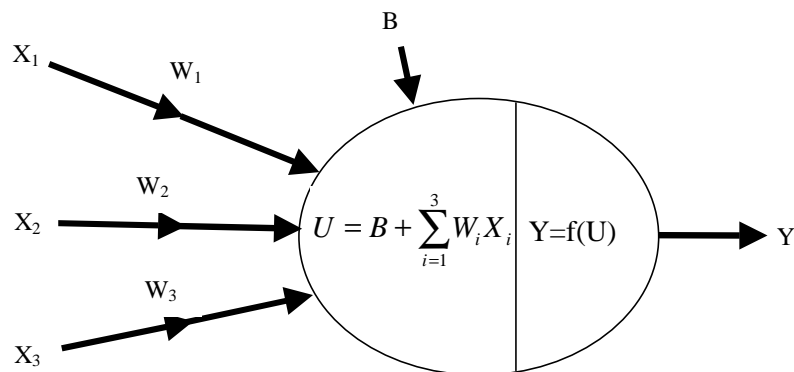


Figura 4.6: Esquema del funcionamiento de una neurona

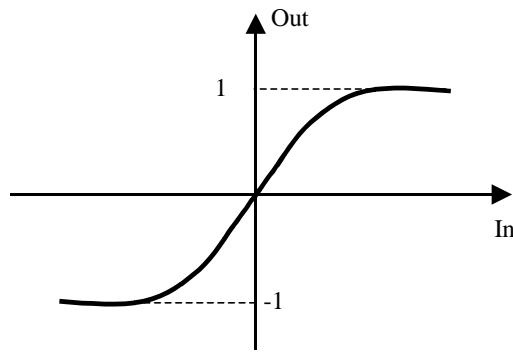


Figura 4.7: Función sigmoide de salida

4.4.2.2 Aprendizaje

El modo de funcionamiento de esta red es supervisado. Es decir, requiere de una fase de entrenamiento para poder ser utilizada posteriormente. En esta fase se le deben presentar parejas de vectores, uno de entrada y otro con la salida correcta para que la red ajuste sus pesos, aprendiendo de esta manera a mapear la relación entre el espacio de entrada y el de salida. El modo de actualizar los pesos se basa en el algoritmo backpropagation (retropropagación del error) [4.28]. Las ecuaciones 4.7, 4.8 y 4.9 muestran las fórmulas de actualización de pesos y propagación de errores:

$$\begin{aligned}\Delta w(i, j) &= lr \cdot \delta(i) \cdot P(j) \\ \Delta B(i) &= lr \cdot \delta(i)\end{aligned}\tag{4.7}$$

Donde

lr : Ritmo de aprendizaje

$B(i)$: Bias

$w(i, j)$: peso de la conexión entre la neurona i y la j

$\delta(i)$: vector de error

$P(j)$: entradas de las neuronas

- Error en las neuronas de salida:

$$\delta_k = (\delta_k - y_k) y_k\tag{4.8}$$

- Propagación del error a las neuronas de la capa oculta:

$$\delta_i = \sum_{k=1}^m \delta_k w_{ik} f_i(Net_i) \quad (4.9)$$

A medida que el ritmo de aprendizaje crece (a través del valor lr), la modificación de los pesos es mayor y el entrenamiento más rápido, aunque por el contrario puede dar lugar a grandes oscilaciones provocando la imposibilidad de llegar al objetivo de error establecido.

Existen diferentes técnicas para mejorar el proceso del entrenamiento. Una de ellas es la del término de momento, pensada específicamente para evitar que el proceso de aprendizaje se detenga en un mínimo local del error. La ecuación (4.10) muestra este término adicional. La técnica consiste en añadir a la expresión $\Delta w(i,j)$ un nuevo término β denominado término de momento. Si la constante β vale 0, el incremento en el peso es el de siempre, en cambio, cuando vale 1, el incremento del peso es idéntico al sufrido en la iteración anterior. Típicamente el valor de β suele ser de 0.95.

$$\Delta w(i, j) = \beta \cdot \Delta w(i, j) + (1 - \beta) \cdot lr \cdot D(i) \cdot P(j) \quad (4.10)$$

Otra técnica que se utiliza es la de utilizar un ritmo de aprendizaje (lr) adaptativo que permitirá converger a la solución en el mínimo tiempo posible. Con esta técnica, si el nuevo error calculado tras la modificación de los pesos supera a la antigua en una cantidad superior definida a través de su cociente (un valor típico es 1.04), los nuevos pesos son desestimados, decrementándose el ritmo de aprendizaje por una constante (0.7 normalmente). En caso de que el error sea menor, el lr se incrementará multiplicándolo por una constante con un valor típico de 1.05.

4.4.3 Implementación

4.4.3.1 Programación

La red se ha implementado, una vez más, bajo el entorno Matlab. El conjunto de funciones utilizadas se incluyen en el Toolbox de redes neuronales “Neuralex” [4.29].

En concreto, en el entrenamiento se han utilizado las funciones *trainbpx* y *logsig* para implementar redes con una capa oculta, funciones de activación sigmoideas y aprendizaje con término de momento y ritmo de aprendizaje adaptativo.

La evaluación es más simple y para ello solo se requieren multiplicar las matrices de pesos de las sucesivas capas por las señales de entrada, activándolas a través de funciones sigmoideas.

4.4.3.2 Aplicación a las medidas con fruta

En el procesado de las medidas realizadas con fruta la red se ha utilizado tanto para clasificación como para evaluación. En el caso de la clasificación se ha utilizado una codificación *n-a-1*, de manera que en la salida se han habilitado tantas neuronas como clases existentes entre las medidas. La identificación ideal de clase se realiza con un vector en el que todos sus valores son cero excepto el de la neurona de la clase correcta, cuyo valor es la unidad.

En el caso de las medidas cuantitativas, se han habilitado tantas neuronas de salida como parámetros a cuantificar simultáneamente. Hay que destacar que en este último caso se han tenido que escalar los datos para que estuviesen comprendidos entre -1 y 1 , tal y como se detallará con mayor profusión en el apartado del preprocesado de los datos.

El proceso de entreno y evaluación se ha realizado, como en el caso del PLS, aplicando técnicas de validación cruzada. A diferencia del PLS, en el caso de las redes neuronales la validación cruzada no siempre ha sido de orden 1 debido a la carga computacional que exige su entrenamiento.

4.4.3.3 Evaluación de resultados

La evaluación de los resultados obtenidos con las redes neuronales se miden en % de aciertos en el caso de que la red se haya utilizado para clasificar. En el caso de la

cuantificación, es decir, la predicción del valor de ciertos parámetros a partir de las señales de los sensores, se han seguido los mismos métodos de evaluación que en el caso del algoritmo PLS.

4.5 REDES FUZZY ART

4.5.1 Introducción

La teoría de la resonancia adaptativa (ART) fue introducida como una teoría que intentaba emular la manera en cómo el cerebro humano procesa la información [4.30,31] . Desde entonces, 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 entrada arbitrarias con un ritmo de aprendizaje rápido o lento. Dentro de estos algoritmos se pueden destacar el ART1 [4.32], ART2 [4.33] y ART3 [4.34].

Fuzzy Art [4.35] es una evolución del algoritmo ART1. Éste último es capaz de categorizar de forma estable entradas arbitrarias binarias. Fuzzy Art, siguiendo el mismo esquema, generaliza esta función a vectores de entrada analógicos con coordenadas comprendidas entre 0 y 1. Para ello substituye los operadores intersección (\cap) y unión (\cup) de ART1 por los operadores MIN (\wedge) y MAX (\vee), respectivamente, de la teoría de lógica difusa [4.36]. 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 figura 4.8 se muestra un esquema del algoritmo.

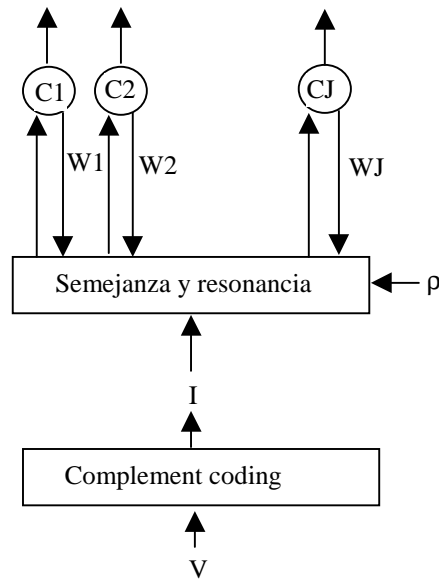


Figura 4.8: Esquema de la red Fuzzy Art

Su modo de funcionamiento es simple: cada vez que la red recibe un nuevo vector de entrada V 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 uno nuevo que representará una nueva clase cuyo primer miembro será este vector.

Desde el punto de vista operativo, este algoritmo cuenta con dos parámetros que controlan su funcionamiento. El parámetro de vigilancia (“vigilance parameter”) 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 nodos de salida, ya que el número de clases diferentes es reducido. Por su parte, el ritmo de aprendizaje queda controlado por el parámetro β , siendo su valor igual a la unidad para un aprendizaje rápido e igual a cero en caso de que no se deba aprender más.

4.5.2 Algoritmo

Incluimos, a continuación, una descripción esquemática del algoritmo:

- Vector de entrada: Cada uno de los vectores de entrada V es un vector M -dimensional donde cada una de sus componentes tiene coordenadas incluidas en el intervalo $[0,1]$.
- Codificación complementaria: A partir del vector de entrada V , se crea un nuevo vector normalizado I de dimensión $2M$ en el que la componente $I_{J+M} = 1-I_J$.
- Vector de pesos del nodo de salida j (categoría j): W_j . Inicialmente, $W_{J1}=W_{J2}=W_{J2M}=1$
- Velocidad de aprendizaje: (“learning rate”), β entre $[0 \ 1]$. Aprendizaje rápido, $\beta =1$; Aprendizaje lento, $\beta \ll 1$; Sin aprendizaje, $\beta =0$
- Parámetro de vigilancia: ρ entre $[0 \ 1]$. ρ cercano a cero implica menos categorías al agrupar con criterios de semejanza poco exigentes, ρ cercano a uno implica muchas clases, cada una con pocos miembros pero muy parecidos entre sí.
- 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 vector de entrada V y cada categoría j se calcula la función de selección $T_j(V)$ como indica la ecuación 4.11:

$$T_j = \frac{|I \wedge W_j|}{\alpha + |W_j|} \quad (4.11)$$

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

$$A \wedge B = \min(A,B) \quad (4.12)$$

Y la norma $||$ se define como:

$$|I| = \sum_{i=1}^{2M} I_i \quad (4.13)$$

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

- Resonancia o reset: Se dice que aparece resonancia si se cumple la desigualdad 4.14:

$$\frac{|I\Lambda W_j|}{|I|} \geq \rho \quad (4.14)$$

En ese caso, se activa la categoría j como respuesta al vector de entrada V , lo que quiere decir que la red clasifica al vector 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 (ecuación 4.11). Si ninguna categoría “resuena”, se crea un nuevo nodo para el vector de entrada V .

- Aprendizaje: Una vez activada la categoría j debido al vector V , sus pesos son actualizados según la ecuación 4.15:

$$W_j^{NUEVO} = \beta(I\Lambda W_j^{ANTERIOR}) + (1 - \beta)W_j^{ANTERIOR} \quad (4.15)$$

Si se quiere un aprendizaje rápido, se utiliza una $\beta = 1$, 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 $\beta=1$.

4.5.3 Implementación

4.5.3.1 Programación

El paquete matemático Matlab no contiene ninguna función que implemente este tipo de red, por lo que se ha tenido que programarla bajo dicho entorno. La programación se ha

hecho adaptando el uso de la red a las aplicaciones de la nariz electrónica. Desde un principio se ha implementado la codificación complementaria.

La programación se ha realizado de forma modular, con funciones que implementan cada uno de los aspectos con los que componen el algoritmo. Así, hay funciones que implementan los operadores de lógica difusa (módulo, AND, OR), una función que implementa la codificación complementaria y otra que implementa la iteración para una medida del núcleo del algoritmo. Para poder clasificar un número concreto de medidas existe una última función que invoca a todas las anteriores para formar un bucle que se repite tantas veces como medidas deban ser procesadas.

4.5.3.2 Evaluación de resultados

Para poder evaluar los resultados, se ha implementado un parámetro que sirve de figura de mérito para valorar las diferentes clasificaciones no supervisadas comparándolos con la agrupación ideal deseada. Dicho parámetro se ha denominado “separación”.

Este parámetro valora las diferencias (atendiéndonos a la categorización deseada), entre todas las medidas que activan un mismo nodo de la red. Un valor igual a cero implica que todas las medidas que activan dicho nodo pertenecen a la misma categoría real, por lo que la agrupación que realiza la red coincide con la clasificación deseada. En caso contrario, el valor del parámetro aumenta, de forma que valores de “separación” elevados indican una clasificación espontánea diferente a la deseada a priori. Hay que destacar que este parámetro penaliza la agrupación en un mismo nodo de medidas cuya categorización correcta es muy distante. Eso es debido a que cada categoría correcta es identificada por un ordinal de manera que categorías contiguas contienen medidas parecidas y categorías alejadas ordinalmente contienen medidas muy diferentes.

Inicialmente, el algoritmo calcula una “centroide” para cada una de las clases que se han formado a la salida de la red. Esta centroide se obtiene como la media aritmética de los ordinales que designan la categoría correcta de cada medida. A continuación se calcula

la distancia media absoluta entre cada una de las medidas y la centroide. Finalmente, se hace la media de estos valores y eso nos da el parámetro de separación.

Para poder realizar comparaciones adecuadas, se debe tener en cuenta que éstas solo se pueden hacer cuando el número de clases generadas es idéntico, ya que, por ejemplo, si se obtienen tantas clases como medidas (con un parámetro de vigilancia igual a la unidad) la separación media es cero, a pesar de que la clasificación realizada no aporta ninguna información.

4.6 RED FUZZY ARTMAP

4.6.1 Introducción

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.

La red Fuzzy Artmap [4.37] proviene de la red Artmap [4.38] 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.

La red Fuzzy Artmap presenta múltiples ventajas que la hacen muy interesante para las aplicaciones con narices electrónicas [4.39,40]. De entre todas ellas destacaremos las siguientes:

- Aprendizaje rápido (con muy poca carga computacional) de las medidas que se presentan en entrenamiento, lo que permite programar el algoritmo en dispositivos programables de bajo coste, aplicar validaciones cruzadas de orden 1 y probar con diferentes combinaciones de parámetros.

- Aprendizaje con un conjunto reducido de medidas de entrenamiento, algo muy interesante en cualquier aplicación experimental en la que sea costoso la obtención de conjuntos de medida extensos. 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 funcione correctamente.
- Aprendizaje continuo de nuevas características sin olvidar lo aprendido con anterioridad, algo muy útil para adaptarse a derivas producidas por 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 como influyen en la categorización de resultados.

Sin embargo, su implementación práctica 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 “outliers” o medidas erróneas en el conjunto de entrenamiento puede requerir un incremento del valor de vigilancia excesivo, lo cual perjudicará seriamente la capacidad de generalizar de la red. Este problema es una de las razones por las que este tipo de algoritmo se ha utilizado poco en narices electrónicas, ya que en este tipo de aplicaciones es sumamente difícil identificar medidas erróneas dada la baja repetitividad de las señales de los sensores. En este trabajo se han estudiado diferentes modificaciones para eliminar este problema.

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.

4.6.2 Algoritmo

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 A) le llegan los vectores de entrada (V). A la red B le llegan, en la fase de entrenamiento, los vectores que codifican la categoría correcta de cada medida del conjunto de entrenamiento (C). La figura 4.9 esquematiza este concepto.

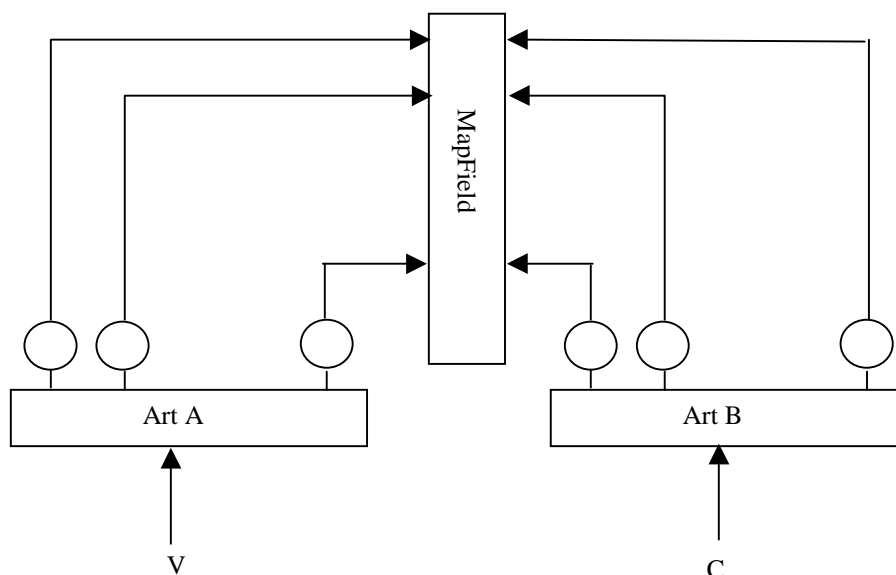


Figura 4.9: Esquema general de una red Fuzzy Artmap

Inicialmente, en la red A el vector de vigilancia es cero. En la red B se suele dar un valor igual a la unidad, ya que medidas que deban ser clasificadas conjuntamente enviarán a la red B 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 B.

Cada vez que se suministra una medida de entrenamiento, la red A activa un nodo y la red B 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 A se le asocia un nodo en B. Cabe destacar que los nodos B normalmente serán imagen de varios nodos A (cada categoría contiene varias medidas), mientras que cada una de las categorías creadas en A sólo tendrá una imagen en B (cada medida solo puede pertenecer a una categoría).

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

En definitiva, el valor de vigilancia en A solo se incrementa lo estrictamente necesario para que la red A 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 de mapeado que relaciona nodos de salida de ARTa con 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 b se denominará W_{BJ} . El mapfield tiene un solo vector de longitud igual al número de nodos activados en A. 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, ρ_a , 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 ecuación 4.16, lo que forzará la activación de un nodo diferente en ARTa:

$$\rho_a = \frac{|I \Delta W_J^A|}{|I|} \quad (4.16)$$

- 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.

4.6.3 Implementación

4.6.3.1 Programación

Aunque existen varios paquetes comerciales que implementan algoritmos basados en Fuzzy Artmap no existe ninguno que implemente la red en su definición estricta. Todos estos paquetes utilizan algoritmos propietarios que no son desvelados por motivos comerciales. Los motivos de estas modificaciones ya han sido explicados con anterioridad. El problema básico reside en la posible existencia de medidas erróneas (“outliers”) en el conjunto de entrenamiento.

Por razones de flexibilidad, al igual que la red Fuzzy Art, esta red se ha programado bajo el entorno Matlab para poderla adaptar a las medidas realizadas con nariz electrónica. Al igual que al implementar Fuzzy Art, Fuzzy Artmap se ha programado a partir de funciones más simples que son anidadas en una función con un bucle principal. Como la red Fuzzy Artmap está formada por dos redes Fuzzy Art, sus funciones principales incluyen llamadas a la función Fuzzy Art implementada anteriormente.

En la programación del algoritmo Fuzzy Artmap se ha distinguido entre una función de entreno (en la que a partir de unas medidas de entrada y sus salidas correspondientes se calculan los pesos) y una de evaluación (en la que a partir de unas medidas de entrada y los pesos se devuelven las clases a las que pertenecen según la red entrenada previamente).

Para evitar los problemas anteriormente comentados y conseguir que la red Fuzzy Artmap funcione correctamente en aplicaciones con nariz electrónica, se han implementado diferentes versiones que modifican ligeramente el algoritmo original con el objetivo de inmunizar a la red ante la presencia de outliers en el conjunto de entrenamiento. Explicar las modificaciones en cada una de estas versiones alargaría innecesariamente el capítulo, por lo que se indicará la filosofía de cada una de ellas y se indicará la versión finalmente escogida para procesar las medidas realizadas con fruta.

- Una primera versión imponía un tope al parámetro de vigilancia. En el caso de que una medida exija un aumento de dicho parámetro superior al estipulado, se concede momentáneamente dicho valor para regresar al anterior en la siguiente medida. La ventaja de esta versión radica en que se puede preservar la capacidad de generalización aprendiendo medidas extrañas que pueden ser “outliers”. Su desventaja radica en saber encontrar el valor óptimo para ese tope que permita un aprendizaje adecuado sin incurrir en un sobreentrenamiento.
- En una segunda versión se asignaba un parámetro de vigilancia particular para cada nodo de la red ARTa. Esta estrategia permite tener diferentes niveles de generalización para cada categoría, algo muy útil si la varianza de medidas de diferentes categorías es muy dispar. El inconveniente más importante de esta versión radica en la falta de convergencia en la fase de entrenamiento en multitud de situaciones prácticas.
- Finalmente, la solución que mejor resultados prácticos ha propiciado modifica el algoritmo original sólo en la etapa de evaluación. En dicha fase el algoritmo modificado ignora el factor de vigilancia alcanzado en la fase de entrenamiento

y busca, de forma iterativa, el valor más elevado que permita clasificar la nueva medida en alguna de las categorías existentes.

Junto a las funciones necesarias para implementar las diferentes versiones de la red fuzzy artmap se han diseñado funciones de validación cruzada de orden 1 (“leave-one-out”) que permiten evaluar el método dando como resultado un porcentaje de aciertos en la clasificación de las medidas, tal y como se hizo con los algoritmos PLS y de redes neuronales backpropagation.

4.7 PREPROCESADO

4.7.1 Introducción

El paso previo a la aplicación de técnicas de reconocimiento de patrones es lo que se denomina preprocesado. Su importancia es capital para el éxito de cada aplicación. Es en esta fase en la que se extrae la información relevante de los sensores para procesar luego su respuesta. Dentro del preprocesado se pueden distinguir tres etapas claramente diferenciadas, cada una de las cuales repercute de una manera esencial sobre los resultados que se obtendrán posteriormente con los algoritmos de reconocimiento de patrones.

En una primera etapa se extraen los parámetros que aportan información relevante del conjunto de cambios físico-químicos que experimentan los sensores. Posteriormente, se acondicionan estos resultados en función de la algoritmos que se vayan a aplicar. Finalmente, se pueden aplicar diferentes tipos de criterios para seleccionar las variables que aportan más información y eliminar aquellas que solo aportan ruido.

4.7.2 Extracción de parámetros

Los sensores de estaño basan su funcionamiento en el cambio de conductividad que experimenta su capa activa ante la presencia de gases reductores. Este aspecto ya ha

sido tratado con mayor detenimiento en el Capítulo 3, por lo que se aconseja al lector el volver a dicho capítulo si requiere revisar su funcionamiento.

Lo que sí es relevante es el hecho de que ese cambio de conductividad conlleva unos transitorios que llevan a la capa activa del sensor desde una situación de reposo a una conductancia final que depende del tipo de volátil y de su concentración.

La información que se puede extraer del comportamiento del sensor es muy variada. Si se obtienen los valores de conductancia inicial y/o final se dice que se están utilizando parámetros estáticos. Con ellos se pueden hacer varias combinaciones que se detallan en la tabla 1:

G_i	Conductancia inicial
G_f	Conductancia final
Δg	Incremento de conductancia ($G_f - G_i$)
Δg_n	Incremento de conductancia normalizada $\Delta g / G_i$

Tabla 4.2: Parámetros estáticos extraíbles en los sensores de óxidos metálicos

Una nueva forma de obtener información adicional es modulando alguno de los factores que afectan a la respuesta del sensor. Los parámetros que se obtienen al aplicar este tipo de aproximación se clasifican como dinámicos y existen de diferentes tipos.

La modulación de concentración es una de las maneras para obtener información adicional de los sensores. En [4.41] se demuestra como la utilización de parámetros asociados a la respuesta transitoria a un aumento brusco de concentración incrementa notablemente el poder de resolución ante diferentes gases de los sensores de óxido de estaño. Esto es debido a que el tiempo de respuesta del sensor es característico del gas o mezcla gaseosa cuya concentración se modula [4.42].

De esta modulación, el parámetro que más se utiliza es el tiempo de subida, medido como el tiempo que tarda el sensor en recorrer un determinado porcentaje del

incremento de conductancia (normalmente del 10 al 90% de ΔG). De hecho, este tipo de modulación se utiliza en los dos prototipos de nariz electrónica desarrollados, ya que la llegada de volátiles de la fruta es un fenómeno rápido en comparación con la inercia de los sensores. En el primer prototipo la concentración aumenta bruscamente a través de varias inyecciones sucesivas, algo que se ha estudiado y aporta una buena repetitividad ante medidas con gases simples [4.43]. En el segundo equipo, la concentración previa de volátiles alcanza de forma rápida la cámara de sensores ante una conmutación de las electroválvulas pertinentes. La figura 9 muestra el transitorio típico de un sensor ante la rápida llegada de aromas provenientes de la fruta en el segundo prototipo. Se puede observar como el transitorio tarda unos minutos en estabilizarse y como de él se pueden extraer parámetros estáticos y parámetros dinámicos.

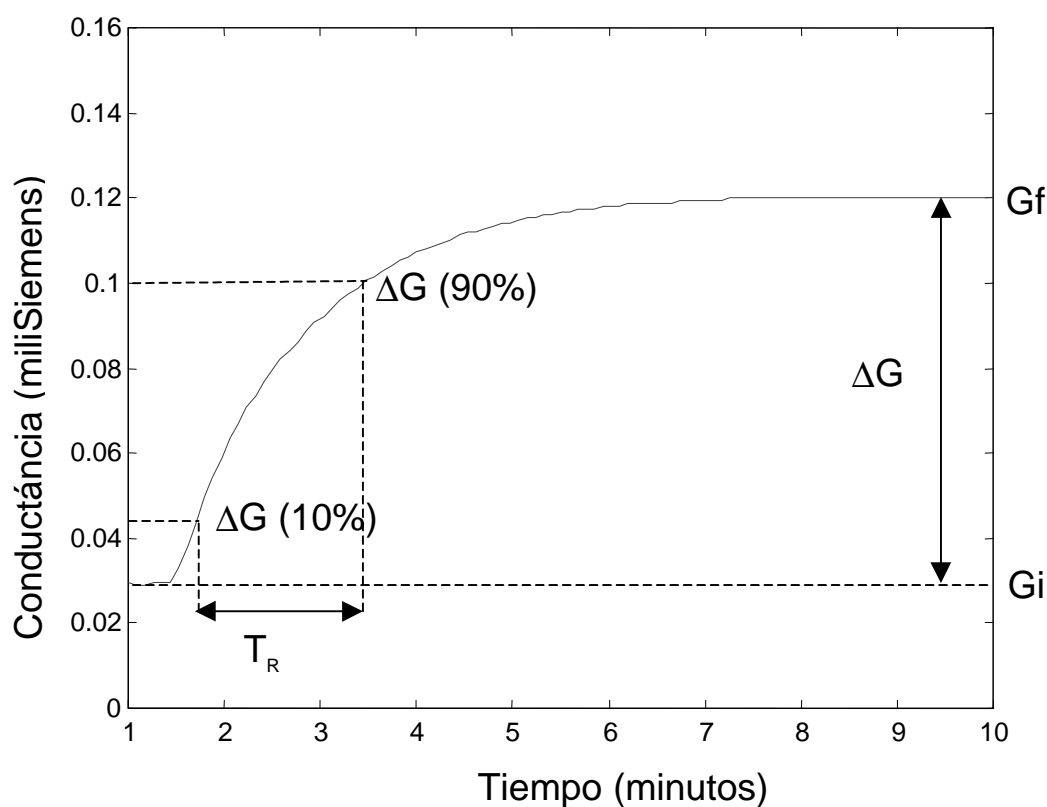


Figura 4.10: Extracción de parámetros temporales ante un típico transitorio típico de los sensores de óxido de estaño ante un brusco cambio de la concentración de volátiles emitidos por la fruta.

Otra forma de obtener parámetros dinámicos de los sensores es mediante la modulación de la temperatura de su capa activa. Aunque ya se han realizado varios trabajos al respecto, en alguno de los cuales el autor de esta tesis ha tenido una contribución

significativa [4.44,45], no se ha aplicado el concepto al prototipo de nariz diseñada para monitorizar la fruta dado que los resultados obtenidos con sensores comerciales no han sido alentadores.

4.7.3 Normalización de los datos

Una vez decididos los parámetros que deben ser extraídos de cada sensor es necesario escalarlos para que puedan ser utilizados por el sistema de reconocimiento de patrones que se considere adecuado para cada aplicación. Existen multitud de escalados diferentes: escalados lineales o logarítmicos, de variables o de medidas, etc. En este proyecto se han considerado cuatro tipos diferentes que pasaremos a detallar a continuación. Para ello cabe recordar que el conjunto de m medidas se resume en una matriz de m filas por n columnas, donde cada una de las m filas describe una medida a través de n variables (una por columna).

- Centrado de datos (“mean centring”): En esta normalización a cada una de las columnas (variables) se le resta su valor medio. Es decir, toda medida es descrita por variables de media nula. Este tipo de centrado es fundamental para los métodos lineales aplicados en este estudio (PCA, PLS), ya que sin él no funcionan adecuadamente debido a la matriz de covarianzas. De todas formas, como veremos a continuación, este tipo de normalización no suele ser suficiente si se pretenden obtener buenos resultados.
- Autoescalado (“Autoscaling”): En este método de normalización, en primer lugar se aplica un “mean centring” a las columnas, para que todas las variables tengan media cero. A continuación, se calcula la varianza de cada una de las variables (columnas) y se divide cada uno de los m valores de cada variable por dicho resultado, de forma que al finalizar la normalización cada una de las n variables que describen las m experiencias tiene media cero y varianza unidad.

Este tipo de escalado es muy útil cuando los parámetros que describen cada experiencia son de naturaleza (y por tanto de unidades) diferentes. Incluso en el

caso de que cada variable represente el mismo parámetro para un sensor hay que recordar que diferentes sensores trabajan en diferentes valores de resistencia, por lo que es conveniente que todos ellos trabajen dentro del mismo rango de valores. El éxito de este escalado radica en que, a priori, asigna la misma importancia a cada una de las variables que describen una medida, independientemente de su naturaleza o sensor del que provengan. De todas formas, hay que tener cuidado ya que este proceso puede incrementar notablemente el ruido existente en señales “pequeñas”.

- Escalado por el valor máximo: La filosofía detrás de este tipo de escalado es similar a la del autoescalado en cuanto a que su objetivo es dar la misma importancia numérica a todas las variables. En este método se divide cada columna de la matriz de experimentos por el valor máximo de cada una de las variables, de manera que todas ellas tienen un valor absoluto comprendido entre 0 y 1. A diferencia del método anterior, en este caso no se resta el valor medio de cada variable.

Este tipo de escalado es fundamental al aplicar redes neuronales de tipo backpropagation y fuzzy artmap. En el primer caso, si se utilizan neuronas con funciones de activación tipo sigmoide, su salida siempre estará en el margen comprendido entre -1 y 1 , por lo que es fundamental escalar las variables de salida entre esos dos valores. Por otro lado, la red fuzzy artmap trabaja con vectores de lógica difusa (fuzzy logic), en los que todos los valores deben estar comprendidos entre 0 y 1 para funcionar correctamente.

- Escalado de “cantidad”: Un último tipo de escalado es el que normaliza el módulo del vector que describe cada medida. Su objetivo es eliminar la información de cantidad (concentración) para facilitar la tarea de identificación y clasificación de las diferentes experiencias que se realicen con la nariz. La ecuación 4.17 muestra la fórmula que permite normalizar cada una de las filas siguiendo este criterio, suponiendo que cada medida queda descrita por N variables.

$$S'_{IM} = \frac{S_{IM}}{\sum_{I=1}^N S_{IM}} \quad (4.17)$$

donde : S'_{IM} : Componente i-ésima de la medida m normalizada

S_{IM} : Componente i-ésima de la medida m sin normalizar

La figura 4.11 intenta esquematizar, de forma gráfica, las diferencias conceptuales entre estas dos últimas normalizaciones partiendo de un conjunto de 4 medidas (dos de cada tipo de sustancias pero con concentraciones diferentes) representadas mediante gráficos polares. Como se puede observar, en la normalización de valor máximo (gráfico superior) existe alguna medida para toda variable en la que el valor máximo es la unidad. Sin embargo, la forma de las medidas de la misma sustancia es siempre la misma, aunque cambiando el tamaño.

En la normalización por medida (gráfico inferior) se puede observar como la forma de las medidas no varía respecto al gráfico original pero se pierde la información del tamaño, por lo que medidas de la misma sustancia se superponen.

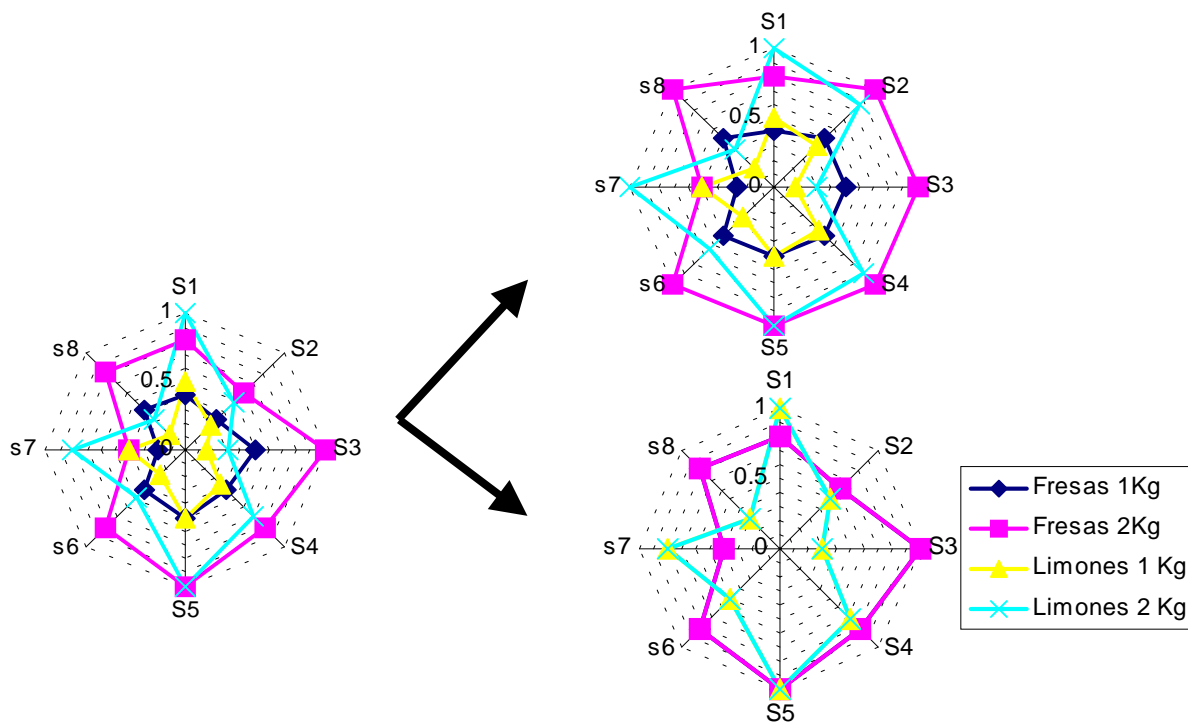


Figura 4.11: Normalizaciones por sensor (arriba) y por medida (abajo)

4.7.4 Selección de variables

En todo estudio de viabilidad en el que se pretende determinar si una nariz electrónica puede realizar satisfactoriamente una determinada función se debe realizar una cuidadosa selección de los sensores que deben formar la matriz química del instrumento. Una práctica común es empezar realizando medidas con un prototipo que contenga una matriz con muchos sensores, de manera que se obtenga la máxima información posible de cada medida. Esta aproximación suele ser muy útil ya que los recursos dedicados al realizar la medida son prácticamente los mismos con pocos o muchos sensores (solo es necesario aumentar la capacidad de adquisición de datos del ordenador). Esta práctica se ha seguido en este trabajo, ya que la realización de medidas con fruta es una práctica muy laboriosa y conviene extraer tanta información como sea posible de cada medida.

De todas formas, el aumentar la cantidad de información que se recoge no es garantía de obtener mejores resultados. Hay variables que aportan información útil y otras que solo aportan ruido. Por ese motivo es necesario escoger cuidadosamente las variables que los diferentes algoritmos de reconocimiento de patrones van a utilizar.

Si tenemos en cuenta que, además, de cada sensor se pueden extraer varios parámetros (estáticos y dinámicos) podemos encontrarnos en una situación en la que incluso haya más variables que medidas. Por ese motivo resulta interesante el utilizar algún criterio de selección de variables que permita reducir la dimensionalidad de los datos sin eliminar información útil y minimizando, a la vez, las interferencias que aporten las variables con ruido.

Dentro de los diferentes métodos existentes podemos destacar tres filosofías diferentes que se han considerado en este proyecto: Criterios de sensibilidad, de correlación entre variables y de optimización mediante la utilización de un algoritmos genéticos.

Cabe aclarar que estas estrategias no siempre han sido aplicadas, ya que en algunos casos se ha optado por incluir familias enteras de parámetros dada la dificultad para obtener criterios y resultados fiables sobre que variables escoger para obtener resultados óptimos. Por lo tanto, en algunos casos se han programado funciones iterativas de prueba y error en las que las combinaciones testadas se han escogido por intuición, sentido común o análisis gráfico de resultados.

4.7.4.1 Criterio de sensibilidad

Este método se ha aplicado, fundamentalmente, para seleccionar sensores más que para escoger variables. Su objetivo es seleccionar los sensores que presenten una mayor sensibilidad media a las sustancias que deben ser analizadas. El parámetro que mejor describe este concepto es el incremento de conductancia normalizada (Δgn), ya que nos da información sobre el cambio de conductancia relativo al valor de reposo. Otros parámetros, como la conductancia inicial o el incremento de conductancia, pueden ser engañosos ya que sensores menos sensibles pueden tener valores más elevados si su conductancia base es superior. La figura 4.12 muestra esta idea con una respuesta real al aroma de la fruta producida por dos sensores diferentes. En ella se puede observar como el sensor 1 tiene valores de ΔG y G_f mayores que el sensor 2, y, sin embargo, este último tiene una mayor sensibilidad ya que triplica su conductancia inicial mientras que el sensor 1 solo la incrementa en un 50%.

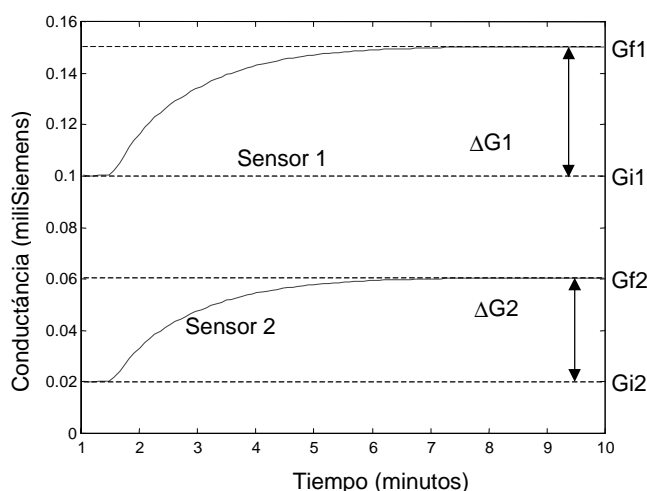


Figura 4.12: Comparación de sensibilidades entre dos sensores

Por otro lado, el escoger los sensores con mayor sensibilidad lo único que garantiza es una capacidad de detección superior, pero no garantiza un mayor poder de resolución que identifique y distinga diferentes compuestos volátiles, ya que, a lo mejor, los sensores con mayor señal reaccionan de manera muy parecida y no cubren correctamente el espectro de aromas a medir. Por lo tanto, este criterio se debe utilizar simplemente para asegurarnos de que la señal que producen los sensores es útil y no es única y exclusivamente ruido.

4.7.4.2 Por covarianza entre variables

Tanto PCA como PLS son algoritmos lineales que basan su funcionamiento en la correlación entre variables. Ambos análisis buscan unas componentes principales (denominadas variables latentes en PLS) sobre las que se pueden proyectar las contribuciones de cada una de las variables (los “Loads” en el argot quimiométrico).

Proyectando las variables sobre las dos primeras componentes principales se puede obtener información sobre la relación entre las mismas. La cercanía entre variables suele presuponer una buena correlación entre ellas. En el caso de que estén en situación completamente opuesta indica que están fuertemente anticorreladas. Estas situaciones tienen diferente interpretación según se esté trabajando con PCA o PLS.

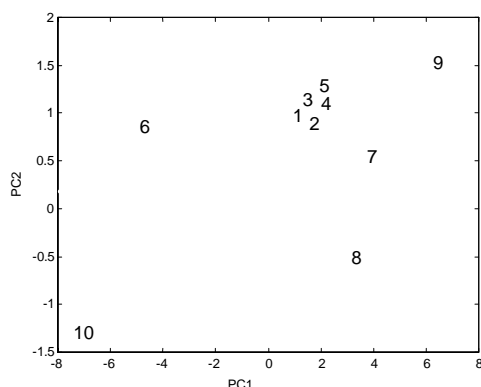


Figura 4.13: Ejemplo de interpretación de loads en un diagrama PCA

Si se está trabajando con componentes principales, variables que estén fuertemente correladas o anticorreladas aportan información similar, lo que indica que son redundantes. El eliminar todas menos una puede permitir reducir la dimensionalidad del vector que define la medida sin perder información importante. Sin embargo, cuando las variables son bastante ruidosas puede ser interesante conservar más de una para intentar promediar su efecto. La figura 4.13 muestra un ejemplo simulado en el que se observa como las variables 1 a 5 aportan información similar (están fuertemente correladas), mientras que la variable 10 está claramente anticorrelada a la variable 9. Así pues, una posible elección para reducir el número de variables podría incluir la 1, 6, 7, 8, 9, representando la variable 1 a todas las de su agrupación y la 9 a su variable anticorrelada numero 10.

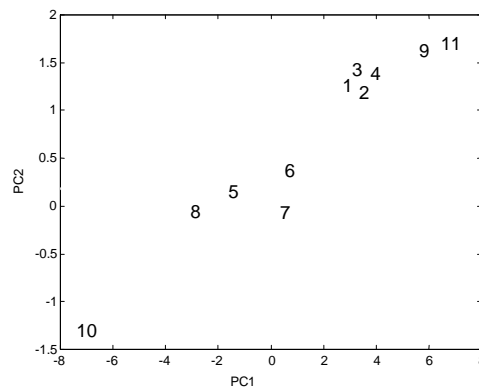


Figura 4.14: Ejemplo de interpretación de loads en un diagrama PLS

En el caso del algoritmo PLS, la cercanía entre variables predictoras y a predecir indica una buena correlación entre ellas, por lo que para construir el modelo PLS interesa incluir las variables cercanas y aquellas en posición opuesta a la variable a predecir. La figura 4.14 muestra un ejemplo ficticio para explicar este concepto, donde las variables 1 a 10 son las predictoras y la 11 la variable a predecir. A partir de la proyección bidimensional el conjunto mínimo de variables que se podrían escoger serían la 1 (como “representante” de la agrupación 1-4), la 8 y la 9. La agrupación 5-7 no está cerca de la variable a predecir y, además, su aportación a las dos primeras componentes principales es prácticamente nula.

De todas maneras, este tipo de selección de variables, a nivel práctico, no tiene mucho éxito. B.Wise, experto en el campo del análisis multivariante, cree que la existencia de variables perturbadoras puede anular por completo la eficacia de este método. Además, es un método lineal que no funciona adecuadamente en procesos no lineales.

4.7.4.3 Algoritmos genéticos

Los algoritmos genéticos son procesos de búsqueda basados en los principios de la selección y la evolución natural [4.46-48]. Las posibles soluciones a un problema son codificadas en forma de cadenas binarias y la búsqueda se inicia con una población aleatoria de posibles soluciones.

En el problema de la selección de variables, cada posible combinación es codificada con una cadena binaria tan larga como parámetros se consideren para encontrar la combinación óptima de variables. En dicha cadena, cada variable tiene asignada una posición o bit, de manera que una posible solución vendrá descrita por una sucesión de unos y ceros indicando la presencia (con un 1) o la ausencia (con un cero) de cada una de las variables en esa combinación particular.

En este tipo de algoritmos, cada miembro de la población, que representa una posible solución, es testada con algún criterio objetivo, de manera que cada uno de los miembros de la población se valora en función de su “fitness” (valor del criterio).

A las soluciones mejor valoradas se les permite sobrevivir y pasar a la siguiente iteración (“generación”), mientras que las soluciones de peor “fitness” desaparecen en las sucesivas generaciones. El algoritmo genético mantiene las iteraciones hasta que iguala o supera el “fitness” establecido como meta o hasta que existe una convergencia en la población, de manera que un determinado porcentaje de sus miembros acaben siendo idénticos.

La creación de los miembros de la población de una nueva iteración se realiza a partir de combinaciones y mutaciones entre los miembros supervivientes de la anterior

población. La combinación consiste en cruzar dos miembros de la antigua población, creando dos nuevos individuos en los que los primeros N bits son de uno de los “padres” y el resto del otro. N, denominado “crossover point” o punto de cruce es un valor aleatorio. La figura 4.15 muestra este concepto. Por otro lado, la mutación de un miembro consiste en el cambio aleatorio de algún bit de su cadena.

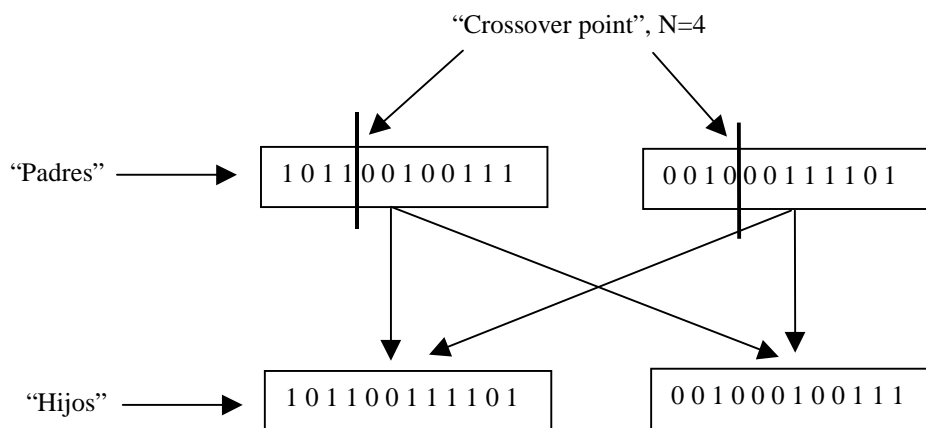


Figura 4.15: Esquema explicativo del cruce entre “padres” para la generación de “hijos”

Existe un teorema [4.49] en el que se demuestra que la iteración sucesiva permitiendo sobrevivir a las combinaciones que mejor cumplen con el criterio preestablecido tras aplicarles combinaciones y mutaciones, permite llegar a una serie de patrones (denominados “schematas”) que convergen a la solución óptima del problema.

El algoritmo genético que se ha aplicado en esta tesis tiene por objetivo encontrar la mejor combinación de variables que debe utilizar un modelo PLS para predecir cada uno de los indicadores de calidad medidos en las piezas de fruta que a su vez fueron utilizadas para obtener resultados con la nariz electrónica. Una vez más, dicha función pertenece al paquete de análisis multivariante “PLS Toolbox” [4.50] que corre bajo el entorno Matlab.

Entre las características de la implementación particular que hace su autor sobre los algoritmos genéticos destacaremos las siguientes:

- Implementa técnicas de combinación (“crossovers”) simples o dobles
- En cada iteración descarta la mitad de la población y permite sobrevivir a la mitad de los miembros con un mejor “fitness”
- El criterio o “fitness” que se utiliza se basa en una validación cruzada PLS, escogiendo de forma aleatoria grupos de medidas de las que obtiene un error cuadrático medio entre la predicción y el valor real. Por su puesto, a menor error, mejor “fitness”.

Los parámetros que utiliza dicha función son los siguientes:

- Número de miembros de la población: Indica el número de individuos que deben considerarse y evaluarse en cada iteración
- Número de términos iniciales: Número de miembros en la población inicial
- Número máximo de generaciones: Número máximo de iteraciones antes de parar el proceso
- Criterio de convergencia: Porcentaje de miembros de la población idénticos para considerar que se ha convergido a una solución y que por lo tanto se deben finalizar las iteraciones
- Probabilidad de mutación: Fracción de bits que deben ser cambiados en cada generación
- Tipo de combinación: “Crossover simple o doble”
- Número de subconjuntos en validación cruzada PLS: Como su nombre indica, numero de subconjuntos que serán iterativamente utilizados para entrenar y evaluar el “fitness” de cada miembro de la población
- Numero de iteraciones: Numero de veces que se deben formar los subconjuntos para obtener un “fitness” medio con menor “rizado” debido a la aleatoriedad.

REFERENCIAS

- [4.1] Agilent Technologies, <http://www.chem.agilent.com>
- [4.2] Gardner J.W., Shurmer H.V., Tan T.T., *Application of an electronic nose to the discrimination of coffees*. Sensors and Actuators B, (1992) vol 6, 71
- [4.3] Abe H., Yoshimura T.Y., Kanaya S., Takahashi Y., Miyashita Y., Sasaki S., *Automated odor-sensing system based on plural semiconductor gas sensors and computerised pattern recognition techniques*. Anal. Chim. Acta, (1987) 194, 1-9.
- [4.4] Bartlett P.N., Blair N., Gardner, J.W., *Electronic noses, principles, applications and outlook*. in 15e Colloque Assoc. Sci. Int. Cafe. 1993. Montpellier, France.
- [4.5] Sundgren, H., Winqvist, F., Lukkari, I., Lundström, I., *Artificial neural networks and gas sensor arrays: quantification of individual components in a gas mixture*. Meas. Sci. Technol.(1991) vol 2, 464-469.
- [4.6] Nakamoto T., Fukuda A., Moriizumi T., *Perfume and flavor identification by odor sensing system using quartz resonator sensor array and neural network pattern recognition*. Sensors and Actuators B, (1993) 10, 85-90.
- [4.7] Carey W.P., Yee S.S., *Calibration of nonlinear solid-state sensor arrays using multivariate regression techniques*. Sensors and Actuators B, (1992). 9, 113-122.
- [4.8] J.Brezmes, E.Llobet, X.Vilanova, X.Correig, *Vapour identification and quantification under variable moisture conditions using gas sensor arrays and artificial neural networks*, Proceedings of the 11th European Conference on Solid-State Transducers, Eurosensors XI, Warsaw (Poland) 1997, 1317-1320
- [4.9] Gardner J.W., Hines E.L., Pang C., *Detection of vapours and odours from a multisensor array using pattern recognition: self-organising adaptive resonant techniques*. Measurement + Control (1996) vol 29
- [4.10] E.L.Hines, E.Llobet, J.W.Gardner, *Electronic noses: A review of signal processing techniques*, IEE Proceedings, Circuits Devices and Systems, Vol. 146, No. 6, (1999) 297-310
- [4.11] Jackson J.E., *Principal component and factor analysis: Part 1-Principal components*, J.Qual. Tech (1981) vol 13,1

- [4.12] Kresta J.V., MacGregor J.F, Marlin T.E., *Multivariate statistical monitoring of process operating performance*, Can. J. of Chem. Eng., (1991) vol 69, 35-47
- [4.13] Gardner, J.W., *Detection of vapours and odours from a multisensor array using pattern recognition. Part 1: principal components and cluster analyses*. Sensors and Actuators B (1991) vol 4, 108-116.
- [4.14] The Mathworks, inc, Matlab (version 5.3), The Mathworks, inc. <http://www.mathworks.com>
- [4.15] B.Wise, N.B. Gallager, *PLS ToolBox 2.0*, Eigenvector research , <http://www.eigenvector.com>
- [4.16] Artursson T., Sjoström M., Holmberg M., *A comparison between different drift counteraction methods*, proceedings of ISOEN 99, Tübingen, Germany september 1999, 241-242
- [4.17] Holmberg M., Winquist F., Lundström I., Davide F., Di Natale C., D'amico A., *Drift counteraction for an electronic nose*, Sensors and actuators B 35-36 (1996), 528-535
- [4.18] Lorber A., Wangen L.E., Kowalski A., *A theoretical foundation for the PLS algorithm*. J. Chemometrics, 1987. Vol 1, 19.
- [4.19] Geladi P., Kowalski B.R., *PLS tutorial*, Anal. Chim. Acta, 185 (1), 1986
- [4.20] Martens H., Naes T., *Multivariate calibration*, John Wiley & Sons, New York, 1989
- [4.21] Höskuldsson A., *Prediction methods in Science and Technology*, Thor publishing, Denmark 1996.
- [4.22] Wold S., Esbensen K., Geladi P., *Principal component analysis*, Chemo. And Intell. Lab. Sys., (1989) vol 7, 53-65
- [4.23] Wise B.M., *Adapting multivariate analysis for monitoring and modeling dynamic systems*, Ph D. Dissertation, University of Washington, Seattle, 1991
- [4.24] Lippman R.P., *An introduction to computing with neural nets*, IEEE ASSP Magazine , abril 1997, 4-22
- [4.25] Grossberg S., *Studies of the mind and brain*, Dordrecht, Holland: Reidel Press, 1982
- [4.26] Hornik K. et al., *Multilayer Feedforward networks are universal approximators*, Neural Networks, (1989), vol 2, 359-366

- [4.27] Gardner J.W., Hines E.L., Wilkinson M., *The application of artificial neural networks in an electronic nose*. Meas. Sci. Technol., (1990). Vol 1, 446-451
- [4.28] Rummelhart D., McClelland J. and the PDP research group, *Parallel distributed processing*, Vol 1, chapter 8, the M.I.T. press, 1986
- [4.29] Demuth H., Beale M., *Neural Network Toolbox v1.0* , The Mathworks Inc., <http://www.mathworks.com>
- [4.30] Grossberg S., *Adaptive pattern classification and universal recoding, II: Feedback, expectation, olfaction and illusions*, Biological cybernetics (1976) vol 23, 187-202
- [4.31] Grossberg S., *How does a brain build a cognitive code?*, Psychological Review, (1980) vol 1, 1-51
- [4.32] Carpenter G.A., Grossberg S., *A massively parallel architecture for a self-organizing neural pattern recognition machine*, Computer vision, graphics, and image processing, (1987) vol 37, 54-115
- [4.33] Carpenter G.A., Grossberg S., *ART2: Stable self-organization of pattern recognition codes for analog input patterns*, Applied Optics (1987), vol 26 4919-4930
- [4.34] Carpenter G.A., Grossberg S., *ART 3 hierarchical search: Chemical transmitters in self-organizing pattern recognition architectures*, International joint conference on neural networks (washington DC), 30-33 Hillsdale, NJ: Erlbaum Associates
- [4.35] Carpenter G.A., Grossberg S., Rosen D., *Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system*, Neural Networks, (1991) vol 4, 759-771
- [4.36] Zadeh L., *Fuzzy sets*, Information and control (1965), vol 8, 338-353
- [4.37] Carpenter G.A., Grossberg S., Markuzon N., Reynolds J., Rosen D., *Fuzzy Artmap: A Neural Network architecture for incremental supervised learning of analog multidimensional maps*, IEEE Transactions on neural networks, (1992) vol 3, No 5, 698-713
- [4.38] Carpenter G.A., Grossberg S., Reynolds J., *Artmap: Supervised realtime learning and classification of nonstationary data by a self-organizing neural network*, Neural Networks, (1991) vol 4, 565-588

- [4.39] Gardner J.W., Hines E.L., Pang C., *Detection of vapours and odours from a multisensor array using pattern recognition: self-organising adaptive resonant techniques*. Measurement + Control, (1996) vol 29, 1996.
- [4.40] E. Llobet, E. L. Hines, J.W. Gardner, S. Franco, *Non-destructive banana ripeness determination using a neural network-based electronic nose*, Meas. Sci. Technol. 10 (1999), 538–548
- [4.41] Llobet E., Brezmes J., Vilanova X., Fondevila Ll., Correig X., *Quantitative vapor analysis using the transient response of non-selective thick-film tin oxide gas sensors*, Transducers'97, Chicago, Digest of Technical Papers, Vol 1, ISBN: 0-7803-3829-4, 971-974
- [4.42] Vilanova X., Llobet E., Alcubilla R., Sueiras J.E., Correig, X., *Analysis of the conductance transient in thick-film tin oxide gas sensors*, Sensors and Actuators B31 (1996), 175-180
- [4.43] Llobet E., Vilanova X., Brezmes J., Alcubilla R., Calderer J., Sueiras J.E., Correig, X., *Analisis of the conductance transient of thick-film tin dioxide gas sensors under successive gas injection steps*, Meas. Sci. Technol., (1997) Vol. 8, Iss 10, 1133-38
- [4.44] Nakata S., Akakabe S., Nakasuji M., Yoshikawa K., *Gas sensing based on a nonlinear response: Discrimination between Hydrocarbons and quantification of individual components in a Gas mixture*, Anal. Chem., (1996) vol 68, 2067-2072
- [4.45] Llobet E., Brezmes J., Ionescu R., Vilanova X., Al-Khalifa S., Gardner J.W., Bârsan N., Correig X., *Wavelet Transform and Fuzzy ARTMAP Based Pattern Recognition for Fast Gas Identification Using a Micro-Hotplate Gas Sensor*, Sensors and Actuators B *en prensa* (2001)
- [4.46] Holland J.H., *Genetic algorithms*, Scientific american (1992), 267(1),66
- [4.47] Lucasius C.B., Kateman G., *Understanding and using genetic algorithms Part 1. Concepts, properties and context*, Chemom. Intell. Lab. Syst.,(1993) 19, 1-33
- [4.48] Lucasius C.B., Kateman G., *Understanding and using genetic algorithms Part 2. Representation, configuration and hybridization*, Chemom. Intell. Lab. Syst.,(1994) 25, 99-145
- [4.49] Holland J., *Adaptation in natural and artificial systems*, (1975), University of Michigan Press