*Rev.R.Acad.Cien.Exact.Fis.Nat. (Esp)*
*Vol. 92, N.º 4, pp 363-369, 1998*
*Monográfico: Problemas complejos de decisión*

# DETERMINACIÓN DE REGLAS DE DECISIÓN MEDIANTE REDES DE NEURONAS ARTIFICIALES

**(reconocimiento de patrones/regla de Bayes/mixtura de distribuciones/aprendizaje supervisado/redes neuronales)**

RAFAEL INFANTE MACÍAS* Y JOSÉ MUÑOZ PÉREZ**

* *Departamento de Estadística e I.O. Facultad de Matemáticas.* Universidad de Sevilla. 41012 Sevilla.
** *Departamento de L. y Ciencias de la Computación.* Universidad de Málaga.

## 1. INTRODUCCIÓN

Hay muchos problemas que se presentan en las diversas ramas de la ciencia y la tecnología, donde es preciso asignar de forma apropiada cada objeto o patrón a la clase o categoría a la que pertenece; como, por ejemplo, en el campo de las tecnologías de la información, para el reconocimiento de caracteres manuscritos, cada símbolo (objeto o patrón) puede corresponder a una letra o a un número; en el reconocimiento del habla los fonemas pueden ser los patrones; en medicina, los patrones pueden ser las radiografías digitalizadas, las imágenes obtenidas por resonancia magnética o las ecografías, y se trata de clasificarlas según el tipo de patología que presentan; en biología, los patrones pueden ser fotografías digitalizadas de plancton que hay que clasificar, etc.

El problema que vamos a tratar aquí consiste en la determinación (estimación) de la regla de Bayes en los problemas de clasificación mediante redes neuronales. Se trata de clasificar una observación, que procede de una mixtura de m distribuciones, utilizando aprendizaje supervisado basado en un conjunto de patrones (observaciones) de entrenamiento.

Consideremos un vector aleatorio X cuyo dominio es el espacio muestral $\Omega$, su rango es $R^N$ y su distribución de probabilidad es una mixtura de m distribuciones de probabilidad correspondientes a las clases $C_1$, $C_2$,...,$C_m$, que son subconjuntos de $R^N$ que pueden solaparse entre sí. La probabilidad *a priori* de la clase $C_i$ es $\pi_i$. Cada observación $x \in R^N$ procede de una de estas poblaciones o clases. Si x pertenece a la i-ésima población (la clase $C_i$) entonces ocurre según la función de densidad condicionada, $f_i(x) = f(x/C_i)$, de X sobre $C_i$, llamada *verosimilitud.* Sea $a_i$ la acción que corresponde a asignar la observación x a la clase $C_i$. Cuando la asignación es incorrecta obtenemos un error (o pérdida) igual a 1, mientras que dicho error vale 0 cuando la asignación es correcta. La *función de pérdida* que obtenemos es

$$L(C_i, a_j) = \begin{cases} 1 & si\ i \neq j \\ 2 & si\ i = j \end{cases}$$

La *función de riesgo* asociada a una regla de decisión múltiple $\phi \equiv (\phi_1(x),...,\phi_m(x))$, donde $\phi_i(x)$ es la probabilidad de tomar la acción $a_i$ cuando se ha observado x [ver Ferguson (1967)], es

$$R(C_i, \phi) = \sum_{j=1}^{m} L(C_i, a_j) E_{C_i}(\phi_j(X)) = 1 - E_{C_i}(\phi_i(X))$$

Asimismo, el *riesgo de Bayes* con respecto a la distribución de probabilidad *a priori* $\pi \equiv (\pi_1,...,\pi_m)$, donde $\pi_i > 0$, i = 1, 2,..., m, y $\pi_1 + ... + \pi_m = 1$, viene dado por la expresión:

$$r(\pi, \phi) = 1 - \sum_{i=1}^{m} \pi_i E_{C_i}(\phi_i(X)) \qquad (1.1)$$

que se interpreta como la probabilidad de incurrir en una clasificación incorrecta utilizando la regla de decisión $\phi$, habiendo ocurrido x según la distribución a priori $\pi$.

Cualquier regla de decisión para la que

$$\phi_i(x) = 0 \quad \text{siempre que} \quad \pi_i f_i(x) < \max_{ik} \pi_k f_k(x)$$

i = 1, 2,..., m, es de Bayes con respecto a $\pi$, es decir, la regla que minimiza el riesgo de Bayes (Hoel y Peterson, 1949).

Para determinar una regla de decisión de Bayes, se definen las *funciones discriminantes* siguientes:

$$\delta_k(x) = \pi_k f_k(x), k = 1, 2,..., m. \qquad (1.2)$$

Así, para una muestra o patrón de entrada x, se asigna x a la clase $S_i$ si

$$\delta_i(x) > \delta_k(x), \ \forall k \neq i.$$

Por otra parte, la probabilidad condicionada de la clase $C_i$, dada la muestra $x$, llamada *probabilidad a posteriori,* y que representaremos por $f(C_i/x)$, viene dada por la expresión:

$$f(C_i/x) = \frac{\pi_i f_i(x)}{f(x)} \qquad (1.3)$$

donde $f(x)$ es la función de densidad no condicionada de X. Por lo tanto, se deduce que la regla de Bayes es aquélla que corresponde a los valores máximos de las probabilidades *a posteriori.*

## 2. CONSTRUCCIÓN DE LA FRONTERA DE DECISIÓN MEDIANTE REDES NEURONALES COMPETITIVAS

Los métodos tradicionales para la obtención de la regla de decisión de Bayes tratan primero de estimar (aproximar) las funciones $\delta_1(x),\ldots, \delta_m(x)$. Nosotros vamos a utilizar una filosofía totalmente diferente que consiste en tratar de representar las fronteras de decisión en lugar de estimar las distribuciones.

La regla de Bayes produce una partición en $R^N$ constituida por los conjuntos

$D_i = \{x \in R^N : \phi_i(x) = 1\}, i = 1, 2, \ldots m.$

Por ejemplo, supongamos que tenemos una mixtura de dos distribuciones Gaussianas bidimensionales y simétricas con vectores media (1,1) y (–1,–1), y vectores varianza (4,4) y (1,1), respectivamente. Si la distribución *a priori* es $\pi \equiv$ (1/4, 3/4), entonces la regla de Bayes clasifica en la clase $C_2$ a todas las observaciones (puntos) que están dentro de la circunferencia que se muestra en la figura 2.1. Dicha circunferencia nos determina la **frontera de decisión** de la regla de Bayes. Los puntos que representamos por el signo «+» corresponden a valores de la distribución con vector de medias (1,1), y los puntos representados por el signo «+» corresponden a la distribución con vector de medias (–1,–1). Como son dos distribuciones bastante solapadas, se puede observar el alto porcentaje de puntos mal clasificados utilizando la regla de Bayes.

Ahora vamos a construir la frontera de decisión utilizando un conjunto de vectores de referencia (o prototipos) para cada una de las dos clases. Estos vectores de referencia los vamos a determinar mediante un proceso de aprendizaje supervisado utilizando un conjunto de patrones de entrenamiento. Supongamos que $\{x(1),\ldots,x(k_1)\}$ son $k_1$ patrones de entrenamiento de la clase $C_1$ y $\{x(1),\ldots,x(k_2)\}$ son $k_2$ patrones de entrenamiento de la clase $C_2$. Consideremos una **red neuronal competitiva** [ver Hertz, Krogh y Palmer (1994)] con L unidades de proceso (tantas como vectores de referencia vayamos a utilizar). Cada unidad de proceso tiene asociado un peso sináptico. Sea $\mathbf{w}_i$ el peso de la unidad de proceso i, i = 1, 2,...,L. Para cada entrada $x \in R^N$, el *potencial sináptico* de la unidad de proceso i viene dado por la expresión:

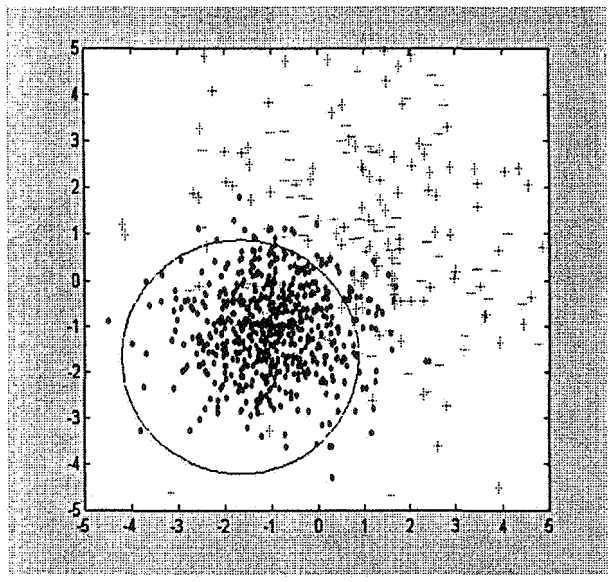


**Figura 2.1.** Frontera de decisión de la regla de Bayes.

$$h_i = \sum_{j=1}^{N} w_{ij} x, -\frac{1}{2}\sum_{j=1}^{N} w_{ij}^2, \quad i = 1,2,\ldots,L$$

Obsérvese que si $h_r > h_s$ entonces el vector $\mathbf{w}_r$ está más cerca del vector de entrada x que el vector $\mathbf{w}_s$ (utilizando la distancia euclídea).

La salida de la unidad de proceso r es igual a uno siempre y cuando

$$h_r = \max_s h_s,$$

en caso contrario vale cero. Es decir, sólo se activa (valor de salida igual a uno) la unidad de proceso de mayor potencial sináptico, que se considera la ganadora. El resto de las unidades de proceso tienen un valor de salida igual a cero (desactivadas).

Por lo tanto, la unidad de proceso ganadora deberá de identificar la clase a la que pertenece el patrón de entrada.

Para determinar los pesos sinápticos vamos a utilizar la siguiente *regla de aprendizaje supervisado:*

- Si r es la unidad ganadora, entonces

$$w_r(k+1) = \begin{cases} w_r(k) + \alpha(k)[x(k) - w_r(k)] & si\ x \in C_r \\ w_r(k) - \alpha(k)[x(k) - w_r(k)] & si\ x \notin C_r \end{cases}$$

• Si r no es la unidad ganadora, entonces no modifica su peso sináptico.

Por lo tanto, cuando el patrón de entrada x es de la clase que corresponde a la unidad ganadora, entonces el vector de pesos sinápticos de dicha unidad se modifica acercándolo al patrón de entrada x, pero si corresponde a otra clase entonces se modifica alejándolo del vector de entrada.

La *tasa de aprendizaje,* $\alpha(k) \in [0,1]$, controla el grado de acercamiento (o alejamiento) del vector de pesos sinápticos al vector de entrada. Una elección apropiada de la tasa de aprendizaje viene determinada por la siguiente ecuación recurrente [ver Kohonen (1997)]:

$$\alpha(k)=\frac{\alpha(k-1)}{1+s(k)\alpha(k-1)},\ k=1,2,...$$

donde $s(k) = +1$ si la clasificación es correcta y $s(k) = -1$ si es incorrecta. Se puede tomar un valor inicial de $\alpha$ igual a 0.1.

Los patrones de entrenamiento son introducidos en la red varias veces hasta concluir el proceso de aprendizaje y obtener los pesos sinápticos, que no son otra cosa que los vectores de referencia que queríamos determinar.

Sea $\{\mathbf{w}_i \in R^N: i \in I_s\}$ un conjunto de vectores de referencia de la clase $C_s$, s = 1, 2,..., L, y representemos la distancia euclídea entre los vectores x y $\mathbf{w}_i$ por $d(x,\mathbf{w}_i)$. Dado un valor x de la variable aleatoria X, lo clasificaremos en la clase $C_h$ si

$$\min_{i\in I_h}\{d(x,\mathbf{w}_i\}<\min_{j\in I_s}\{d(x,\mathbf{w}_j\},\ \forall s\neq h.$$

Los vectores de referencia determinan polígonos de Voronoi, de manera que la frontera de decisión de la regla obtenida viene configurada por lados de dichos polígonos de Voronoi. Por ejemplo, supongamos que tenemos 25 patrones de entrenamiento, que son puntos generados aleatoriamente de una variable bidimensional gaussiana con vector de medias (1,1) y vector de varianzas (4,4). Además, consideremos 75 patrones de entrenamiento generados aleatoriamente de una distribución gaussiana con vector de medias (−1,−1) y vector de varianzas (1 1). Si utilizamos una red neuronal competitiva con 16 unidades de proceso y realizamos el proceso de entrenamiento de la red con los patrones citados, se obtienen, después de 2.000 iteraciones, los 16 vectores de referencia que se muestran en la figura 2.2 con el símbolo «o»; cuatro de ellos son de la primera clase y los doce restantes de la segunda. La frontera de decisión viene determinada por dichos vectores de referencia y es de tramos lineales (lados de polígonos de Voronoi).

Por lo tanto, este método de determinación de la frontera de decisión, llamado **clasificador de prototipos más cercanos,** realiza una agrupación de los patrones de entrenamiento y sustituye cada grupo obtenido por su correspondiente *centroide.* Los centroides de dichos grupos son precisamente los vectores de referencia que se utilizan para determinar la frontera de decisión. El problema se reduce a diseñar un buen conjunto de vectores de referencia (prototipos) que consiga una tasa de clasificación incorrecta lo menor posible.

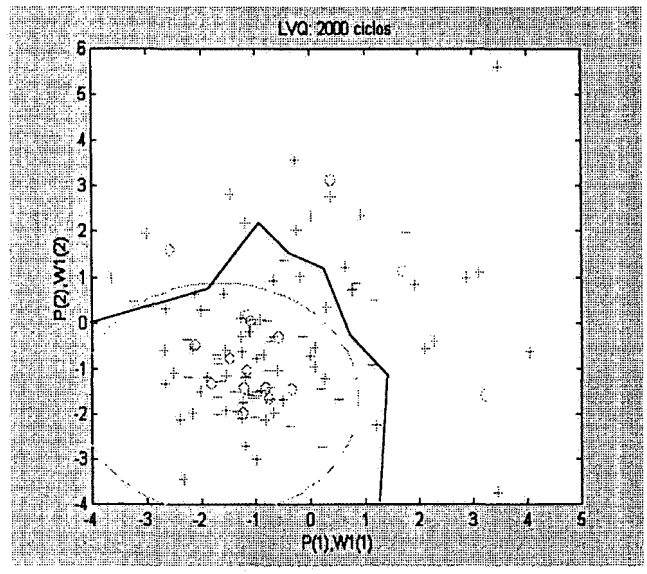


**Figura 2.2.** Frontera de decisión determinada por los vectores de referencia.

Si utilizamos los mismos patrones de entrenamiento como patrones de prueba para analizar la tasa de clasificación incorrecta (resustitución), vemos que sólo clasifica incorrectamente a cuatro, mientras que utilizando la regla de Bayes clasificamos incorrectamente a seis de ellos. Con esta técnica de condensación se pueden conseguir errores de resustitución iguales a cero, eligiendo un conjunto apropiado de vectores de referencia.

## 3. DETERMINACIÓN DE LA REGLA DE DECISIÓN DE BAYES MEDIANTE UN PERCEPTRÓN MULTICAPA

Consideremos un problema de clasificación de patrones de dos categorías $C_1$ y $C_2$ que siguen funciones de densidad Gaussianas N-dimensionales, $f(x/C_i)$, i = 1,2, dadas por la expresión:

$$f(x/C_i)=\frac{1}{\sqrt{(2\pi)^N|\Sigma_i|}}\exp\left\{-\frac{1}{2}(x-\mu_i)'\,\Sigma_i^{-1}(x-\mu_i)\right\}$$

donde $\mu_i$ es el vector de medias de los patrones que proceden de la clase $C_i$ y $\Sigma_i$ es la matriz de covarianza de la clase $C_i$, para i = 1,2.

Consideremos un **Perceptrón Multicapa** [ver Haykin (1994)] con N unidades de entrada, 2N unidades de proce-

so (neuronas) en la capa oculta y una unidad de proceso (neurona) en la capa de salida. Como *función de transferencia* en las unidades de proceso de la capa oculta se va a utilizar la función *logística:*

$$\varphi(u)=\frac{1}{1+\exp(-u)},\ u\in R,$$

y esta misma función en la unidad de salida. Por lo tanto, esta red neuronal implementa la función

$$F(x,c,w)=\varphi\left(\sum_{i=1}^{2N} c_i\varphi\left(\sum_{i=1}^{N} w_{ij}x_j-\theta_i\right)\right)$$

donde $\mathbf{c} = (c_1,...,c_{2N})'$ es el vector de los pesos sinápticos de la capa de salida y $\mathbf{w} = ((w_{ij}))$ es la matriz de los pesos sinápticos de la capa oculta. En la figura 3.1 se muestra la arquitectura de Perceptrón Multicapa con dos sensores de entrada, cuatro neuronas en la capa oculta y una neurona en la capa de salida.

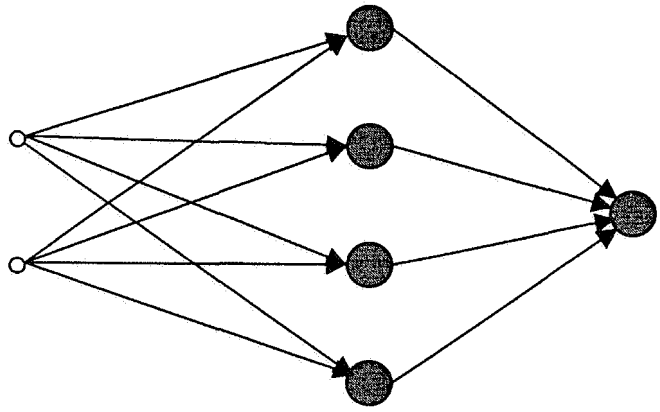

**Figura 3.1.** Perceptrón Multicapa.

Los pesos sinápticos se determinan siguiendo un *proceso de aprendizaje supervisado,* conocido como algoritmo de **retropropagación** (con sus múltiples variantes, como la inclusión del término de momentos o el algoritmo de Leberberg-Marquart), que utiliza dos conjuntos de patrones de entrenamiento (un conjunto $S_1$ de la clase $C_1$ y otro $S_2$ de la clase $C_2$). Con dicho proceso de aprendizaje se pretende minimizar el error cuadrático entre la salida del Perceptrón Multicapa y la salida deseada para el patrón de entrada x, es decir, minimizar la expresión:

$$EC=\sum_{x\in S_1}\left(F(x,\ \mathbf{c},\mathbf{w}-1)\right)^2+\sum_{x\in S_2}F(x,\ \mathbf{c},\mathbf{w})^2,$$

pues se desea que la red tenga una salida $F(x,\mathbf{c},\mathbf{w})$ igual a 1 cuando el patrón x es de la clase $C_1$ y una salida igual a 0 cuando dicho patrón sea de la clase $C_2$. Sin embargo, si todos los patrones fuesen introducidos en la red según las respectivas distribuciones de probabilidad de las categorías $C_1$ y $C_2$, se obtiene el error cuadrático medio (esperado) dado por la expresión

$$ECM_T=\pi_1\int_{R^N}\left(F(x,\mathbf{c},\mathbf{w})-1\right)^2 f(x/C_1)dx+\pi_2\int_{R^N}F(x,\mathbf{c},\mathbf{w})^2 f(x/C_2)dx,$$

y utilizando (1.3) se llega a la expresión

$$=\int_{R^N}\left[F(x,\mathbf{c},\mathbf{w})-f(C_1/x)\right]^2 f(x)dx+\int_{R^N}f(C_1/x)\left(1-f(C_1/x)\right)f(x)dx$$

donde sólo el primer término depende de los pesos sinápticos. Así, el algoritmo de retropropagación modifica los pesos sinápticos iterativamente de forma que decrezca el error cuadrático medio, o lo que es lo mismo, el primer término de la expresión anterior, que conlleva a que la salida de la red, $F(x,\mathbf{c},\mathbf{w})$, tienda a la probabilidad *a posteriori*, $f(C_1/x)$, de que un patrón dado x pertenezca a la clase $C_1$.

Funahashi (1998) ha demostrado que $f(C_1/x)$ se puede aproximar en sentido estadístico por la red neuronal que estamos utilizando [la distancia $L^2$ entre la función de entradas y salidas de la red neuronal, $F(x,\mathbf{c},\mathbf{w})$, y la distribución de probabilidad *a posteriori*, $f(C_1/x)$, ponderada con la función de densidad $f(x)$, tiende a cero].

## 4. FUNCIÓN DISCRIMINANTE DE BAYES: APROXIMACIÓN POR REDES NEURONALES MULTICAPA

Consideremos la función discriminante de Bayes

$$g(x) = g_1(x)-g_2(x),$$

donde $g_i(x) = log(\pi_i f(x/C_i))$, i = 1,2, de manera que se asigna $x$ a $C_i$ si $g(x)>0$ y a $C_2$ si $g(x)<0$. Cuando las distribuciones son Gaussianas la función $g(x)$ es un polinomio cuadrático de N variables y la superficie de decisión, dada por $g(x)=0$, es una forma cuadrática en $R^N$.

Vamos a aproximar la función discriminante $g(x)$ mediante un Perceptrón Multicapa con 2N unidades de proceso en su capa oculta. La salida deseada será igual a 1 si el patrón es de la clase $C_1$ y será igual a –1 si es de la clase $C_2$. Por ello, vamos a utilizar como función transferencia para las unidades de la capa oculta la función tangente hiperbólica, que viene dada por la expresión:

$$\sigma(u)=\tanh\left(\frac{u}{2}\right)=\frac{1-\exp(-u)}{1+\exp(-u)}$$

Podemos tomar, como función de transferencia para la unidad de salida, esta misma función o la función identidad. Por ejemplo, consideremos una mixtura de dos distribuciones Gaussianas bidimensionales con vectores de medias (1,1) y (_1,_1), y vectores de varianzas (4,4) y (1/4 1/4), respectivamente. Supongamos que la distribución *a priori* es $\pi$ _ (1/4, 3/4). Vamos a aproximar la función discriminante de Bayes mediante una red neuronal multicapa con dos sensores de entrada, una capa oculta formada

por cuatro neuronas y una neurona solamente en la capa de salida, como se muestra en la figura 3.1.

El algoritmo de retropropagación clásico suele requerir muchos ciclos de aprendizaje (introducir muchas veces los patrones de entrenamiento) hasta alcanzar unas tasas de error satisfactorias. Por ello vamos a determinar los pesos de la red neuronal siguiendo la regla de aprendizaje de Lebenberg-Marquart, que aunque realiza una mayor cantidad de cómputo en cada iteración, sin embargo requiere una cantidad mucho menor de ciclos de entrenamiento. En nuestro ejemplo hemos realizado sólo 20 barridos o ciclos de entrenamiento, es decir, hemos introducido en la red 20 veces el conjunto de patrones de entrenamiento. Los pesos sinápticos de las neuronas de la primera capa, que se obtienen utilizando como conjunto de entrenamiento 400 patrones extraídos aleatoriamente de la mixtura de distribuciones Gaussianas citada anteriormente, vienen dados por la siguiente matriz:

$$\mathbf{W}_i = \begin{pmatrix} -1.3126 & -7.5687 & 11.4839 & 3.7210 \\ 4.4507 & -2.8397 & 16.1994 & -2.2595 \end{pmatrix};$$

el sesgo de cada una de las cuatro neuronas de la primera capa, viene dado por

$\mathbf{b}_1$ = (7.9758 −3.8061 −3.78404.1173);

los pesos sinápticos de la capa de salida son:

$\mathbf{W}_2$ = (−35.3992 −33.25631.7358 −33.5957),

y el sesgo de la neurona de salida es

$\mathbf{b}_2$ = 37.5996.

La curva de nivel de la función implementada por esta red neuronal que corresponde al valor cero viene trazada en la figura 4.1. Esta red clasifica incorrectamente sólo a cinco patrones.

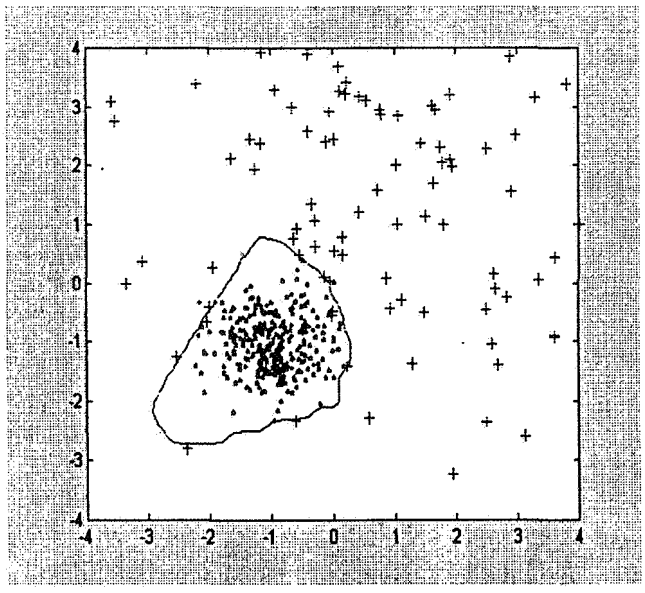


**Figura 4.1.** Frontera de decisión de la función discriminante de Bayes *g(x)* aproximada por un Perceptrón Multicapa.

Sin embargo, esta aproximación no está bien fundamentada. Si en lugar de utilizar la función *g(x)* utilizamos una transformación monótona creciente de la misma, concretamente la función $\varphi(g(x))$, donde $\varphi$ es la función logística, la regla de decisión de Bayes se puede establecer de la forma siguiente:

- asignamos $x$ a la clase $C_1$ si $\varphi(g(x))>1/2$, y
- asignamos $x$ a la clase $C_2$ si $\varphi(g(x))<1/2$.

¿Qué es $\varphi(g(x))$? Sustituyendo, se obtiene que

$$\varphi(g(x)) = \frac{\pi_1 f(x/C_1)}{\pi_1 f(x/C_1) + \pi_2 f(x/C_2)} = f(C_1/x),$$

es decir, $\varphi(g(x))$ es la probabilidad *a posteriori* de la categoría $C_1$. Por lo tanto, hemos obtenido una clara interpretación de esta función discriminante. Así podemos aproximar la función discriminante $\varphi(g(x))$ mediante un Perceptrón Multicapa con 2N unidades de proceso en su capa oculta y tomando como función de transferencia para la unidad de salida la función logística $\varphi(u)$, pues toma valores en el intervalo [0,1]. La salida deseada será igual a 1 si el patrón es de la clase $C_1$ e igual a 0 si es de la clase $C_2$. De esta forma la red neuronal implementa la función dada en (3.3) como aproximación de la función discriminante $\varphi(g(x))$, que no es otra cosa que la probabilidad *a posteriori* de la clase $C_1$.

Por ejemplo, consideramos una mixtura de dos distribuciones Gaussianas bidimensionales con vectores de medias (1,1) y (−1,−1), vectores de varianzas (4,4) y (1, 1/4), respectivamente, y supongamos que la distribución *a priori* es $\pi \equiv$ (1/4, 3/4). Para estimar la función discriminante $\varphi(g(x))$ vamos a utilizar un Perceptrón Multicapa con cuatro unidades de proceso en la capa oculta y una unidad de proceso en la capa de salida, que tienen la función de transferencia logística $\varphi$. Si determinamos los pesos sinápticos utilizando el proceso de aprendizaje de Lebenberg-Marquart, con sólo 20 ciclos o épocas de entrenamiento para 400 patrones de entrenamiento generados de dicha mixtura, obtenemos la aproximación de la función discriminante $\varphi(g(x))$ cuya curva de nivel para el valor 1/2 viene representada en la figura 4.2. Como las distribuciones están más solapadas se obtiene un error de sustitución de 12 patrones, mayor que en el ejemplo anterior.

Si ahora consideramos una mixtura de dos distribuciones Gaussianas bidimensionales con vectores de medias (1,1) y (−1,−1), vectores de varianzas (1,1) y (1, 1), entonces el Perceptrón Multicapa implementa la función discriminante cuya curva de nivel en el valor 1/2 viene representada en la figura 4.3, y como las distribuciones están muy poco solapadas se consigue un error de sustitución igual a cero.

Finalmente, si los vectores de varianzas fueran (1,1/4) y (1/2,2), respectivamente, entonces la función discriminante que proporciona el Perceptrón Multicapa tiene las

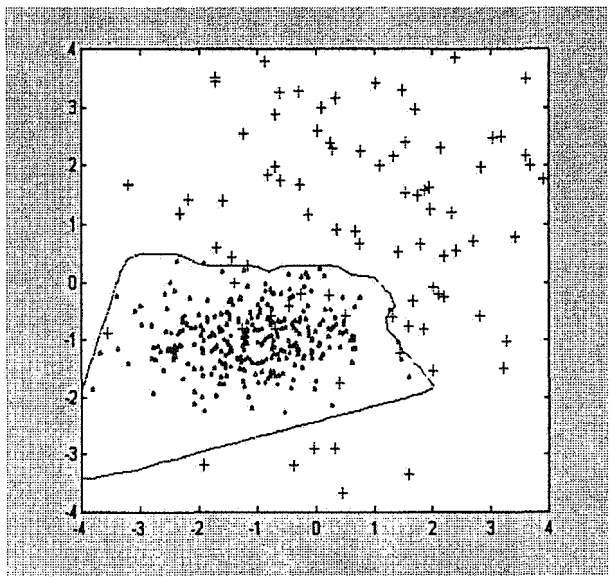

**Figura 4.2.** Frontera de decisión de la función *j(g(x))* aproximada por un Perceptrón Multicapa con cuatro neuronas en la capa oculta.

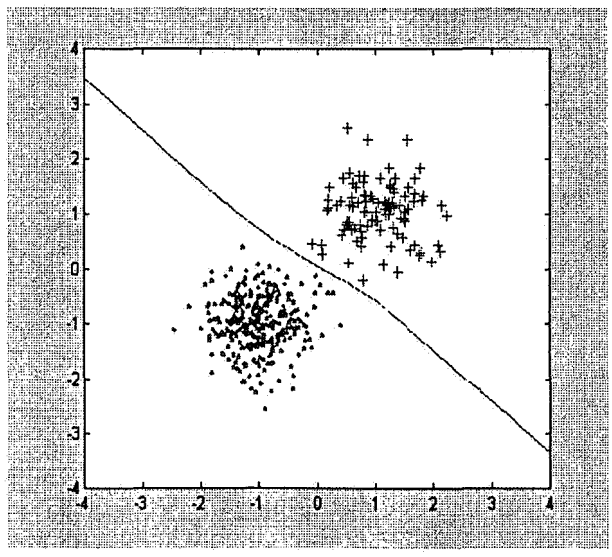

**Figura 4.3.** Frontera de decisión con error de resustitución igual a cero.

líneas de contorno (para el valor 1/2) según se representan en la figura 4.4. Como puede verse, ya no es tan simple como en el caso anterior, al estar más solapadas en ciertas direcciones y tratar de buscar una buena separación de las dos clases. Es difícil de conseguir un error por sustitución igual a cero en este caso.

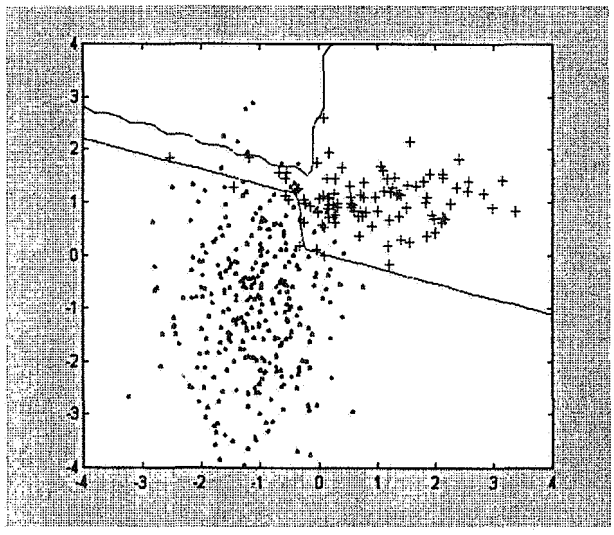

**Figura 4.4.** Frontera de decisión de la función *j(g(x))* para distribuciones con varianzas desiguales.

Por lo tanto, la adaptabilidad y la flexibilidad de las redes neuronales las hacen apropiadas para aproximar la frontera de decisión de la regla de Bayes cuando no se conoce la distribución de procedencia y sólo se dispone de un conjunto de patrones entrenamiento.

## 5. CONCLUSIONES

Las redes de neuronas artificiales son especialmente útiles para la extracción de características y para la clasificación. En este trabajo se ha puesto de manifiesto que podemos conseguir una estimación de las probabilidades *a posteriori* y la implementación de la *función discriminante de Bayes* mediante un Perceptrón Multicapa. Asimismo, utilizando una red neuronal competitiva, se consigue una aproximación de la *frontera de decisión de Bayes* basada en vectores de referencia. Por lo tanto, las redes neuronales son una herramienta apropiada para el modelado de problemas en teoría de la decisión cuando no se conocen las distribuciones de procedencia. Por otro lado, se ha dado una interpretación clara de la relación funcional de las entradas con las salidas, en el Perceptrón Multicapa, como las probabilidades *a posteriori*, $p(C_i/x)$. Así, la teoría de la decisión es también fuente de ideas para la neurocomputación.

## REFERENCIAS


1. Duda, R.O. & Hart, P.E. (1973) *Pattern Classification and scene analysis.* John Wiley and Sons, New York.
2. Ferguson, T.S. (1967) *Mathematical Statistics,* Academic Press.
3. Funahashi, K. (1998) Multilayer neural networks and Bayes decision theory. *Neural Networks,* Vol. **11,** pp. 209-213.
4. Haykin, S. (1994) *Neural Networks: A Comprehensive Foundation.* MacMillan.
5. Hertz, J., A. Krogh & R.G. Palmer (1994) *Introduction to the Theory of Neural Computation.* Addison Wesley.

**6.** Hoel, P.G. & Peterson, R.P. (1949) «A solution to the problem of optimum classification». *Ann. Math. Statist.* Vol. **20,** pp. 433-438.

**7.** Holmstrom, L. & Koistinen, P. (1997) Neural and Statistical Classifiers-Taxonomy and Two Case Studies. *IEEE Trans. on Neural Networks,* Vol. **8,** N.° **1,** pp. 5-17.

**8.** Kohonen, T. (1997) *Self-Organizing Maps.* Springer-Verlag.

**9.** Richard, M.D. & Lippmann, R.P. (1991) Naural Networks Classifiers Estimate Bayesian a Posteriori Probabilities. *Neural Computation,* Vol. **3,** pp. 461-483.