

# Aprendizaje híbrido de redes neuronales $q$ -Gaussianas en clasificación binaria

F. Fernández, C. Hervás, P. A. Gutiérrez, M. Cruz, J. Sanchez-Monedero

Dpto. Informática y Análisis Numérico

Universidad de Córdoba

i22fenaf@uco.es, chervas@uco.es, pagutierrez@uco.es, i42crram@uco.es, i02samoj@uco.es

## Resumen

En este trabajo proponemos utilizar como funciones de base diferentes formas de funciones locales de tipo RBF, en concreto funciones de base del tipo  $q$ -Gaussiano, una familia que contiene, entre otras, a las funciones Gaussianas y a las funciones de Cauchy, cambiando los valores del parámetro  $q$ . Para entrenar estos modelos utilizaremos un algoritmo de aprendizaje híbrido (AH) basado en poblaciones, donde la búsqueda local la realizamos utilizando el algoritmo iRprop+ sobre el mejor individuo obtenido en la última generación. Para poder justificar nuestra propuesta y analizar el rendimiento de los modelos entrenados realizamos un estudio experimental considerando cuatro bases de datos de clasificación binaria tomadas del repositorio de la UCI. Comparamos nuestros resultados con los obtenidos con redes neuronales RBF con funciones de base prefijadas. Los resultados muestran que tanto la metodología como el nuevo tipo de funciones de base radial presentan una muy buenas perspectivas para el modelado de clasificadores binarios.

## 1. Introducción

Algunas de las características de las Redes Neuronales las hacen particularmente atractivas para el modelado y control de sistemas no lineales, como el reconocimiento de patrones, la predicción, el modelado de funciones y en general la optimización de sistemas. (i) En primer lugar las funciones de base sig-

moide, producto, o de base radial son aproximados universales. (ii) Presentan claras habilidades para poder realizar un procesamiento paralelo a la hora de entrenarlas. (iii) Tienen buena respuesta cuando se incorporan nuevos datos al sistema, adaptando incrementalmente el aprendizaje mediante una adecuada modificación de los pesos de la red. (iv) Tienen una naturaleza tolerante a fallos asociada a que al añadir o eliminar un nodo o conexión en la capa oculta de una red, el resto de conexiones y/o nodos se adaptan a la nueva situación reajustando sus pesos para obtener una salida lo más adecuada posible. (v) Los mejores modelos de red obtenidos en el proceso de aprendizaje pueden implantarse, con una relativa facilidad, en *hardware*.

Diferentes tipos de redes neuronales, se han utilizado en los últimos 20 años en tareas de clasificación de patrones, incluyendo entre otros modelos: redes neuronales de tipo Perceptron Multicapa (*MultiLayer Perceptron*, MLP) donde las funciones de transferencia son sigmoides o arco tangentes; Redes con Funciones de Base Radial (*Radial Basis Function Neural Networks*, RBFNNs) donde las funciones locales son habitualmente funciones Gaussianas [1]; Redes Neuronales de Unidades Producto (*Product Unit Neural Networks*, PUNNs) [2], donde las funciones de base son productos de funciones de potencias; o Redes Neuronales Híbridas donde las funciones que forman la capa oculta son una mezcla de las funciones descritas anteriormente [3].

En este trabajo nos centraremos en un tipo especial de RBFNNs, llamadas Redes Neu-

ronales de Base Radial  $q$ -Gaussianas, QRBF. Este tipo de modelos permiten modificar la forma de una función de base RBF a través un parámetro adicional  $q$ . Es importante apuntar que las redes QRBF no son una alternativa a los modelos clásicos gaussianos sino una generalización paramétrica de los mismos. Una ventaja de este tipo de redes si las comparamos con las redes MLP es que podemos entrenarlas mediante un procedimiento en dos etapas. En la primera, podemos fijar los parámetros de las funciones de base así como su número; y en una segunda etapa, entrenar una combinación lineal de las funciones de base determinadas en la etapa anterior.

Aunque los parámetros de los modelos QRBF pueden ser determinados en dos etapas, también podemos utilizar un procedimiento combinado. Por ejemplo, mediante un algoritmo de aprendizaje supervisado podríamos hacer una estimación simultánea de todos los parámetros mediante técnicas de optimización no lineal incluyendo, entre otros, métodos de gradiente descendente [4] o algoritmos evolutivos [5, 6]

En este artículo consideramos esta última técnica, de forma tal que el algoritmo evolutivo determina el número óptimo de nodos QRBF en la capa oculta, el tipo que viene dado por el valor de  $q$  (el cual es crucial para este tipo de funciones), y los pesos y centros de los modelos.

El trabajo está organizado de forma tal que en la sección 2 se hace una revisión de trabajos relacionados con los modelos RBF y, en concreto, con las funciones  $q$ -Gaussianas. En la sección 3 hacemos una breve introducción a las funciones  $q$ -Gaussianas para, a continuación, presentar nuestro modelo en profundidad. En la sección 4 describimos el algoritmo evolutivo híbrido de estimación de los coeficientes del modelo. En las secciones 5 y 6 mostramos los experimentos realizados y los resultados obtenidos para las bases de datos de prueba y en la sección 7 presentamos las conclusiones y nuevas propuestas en este campo.

## 2. Trabajos Relacionados

La construcción de los modelos de redes RBF depende de tres factores: la elección de la función de kernel (Gaussiana, de Cauchy, polinómica, etc.), la selección de unos centros apropiados y la determinación de los radios y de los coeficientes del modelo lineal (conexiones entre las neuronas de las capas oculta y de salida de la red). Básicamente existen dos categorías de algoritmos de aprendizaje propuestos para redes RBF. En la primera los algoritmos sitúan una función RBF en cada patrón del conjunto de entrenamiento [7]. En la segunda, se tiende a reducir el número de neuronas en capa oculta o lo que es igual el número de RBFs en el modelo [8]. En la primera categoría, está claro que es necesario utilizar un algoritmo de regularización o de eliminación de funciones RBF para poder conseguir modelos que no sobre-entrenen, esto es, que el clasificador pierda eficacia sobre los patrones del conjunto de generalización. En general, el número y la posición de las funciones de base puede ser fijado a priori [9] o puede ser determinado mediante un análisis de agrupamiento sobre los datos del conjunto de entrenamiento (por ejemplo, utilizando un algoritmo del tipo  $k$ -medias,  $k$ -medianas, etc.) para establecer una función radial para cada agrupación o *cluster*. Una alternativa es utilizar Algoritmos Evolutivos para determinar tanto la estructura como los pesos de la red, siendo [10] un buen artículo de referencia para comprender cómo se aplican las técnicas evolutivas con este tipo de modelos.

Existen varios tipos comunes de funciones de base radial utilizados como funciones de transferencia de las redes, donde la más habitual es la Gaussiana (*Standard RBF*, SRBF), aunque también podemos encontrar otras propuestas como la RBF Multicuadrática (*Multiquadratic RBF*, MRBF), la Inversa Multicuadrática (*Inverse MRBF*, IMRBF) y la de Cauchy (*Cauchy RBF*, CRBF). Todas ellas son miembros de una familia de funciones más general llamada  $q$ -Gaussiana, la cual puede reproducir los tipos anteriores cambiando el valor del parámetro  $q$ . Así para entrenar este tipo

de modelos de red RBF utilizaremos un algoritmo híbrido (AH) que estima los parámetros de las funciones de base radial para conseguir modelos con una alta precisión sobre el conjunto de generalización

### 3. Funciones $q$ -Gaussianas

Sea  $Y$  una variable aleatoria de Bernoulli asociada a un problema de clasificación en dos clases, de forma tal que codificamos como  $Y = 1$  la pertenencia a la primera clase  $C_1$ , con una probabilidad  $p$  condicionada a los valores de las variables independientes, y como  $Y = 0$  la pertenencia a la segunda clase,  $C_2$ . El conjunto de las  $k$  variables independientes es representado por  $x_1, x_2, \dots, x_k$ . Sea  $D = \{(\mathbf{x}_l, \mathbf{y}_l) : l = 1, 2, \dots, n\}$  un conjunto de patrones de tamaño  $n$  que forman el conjunto de entrenamiento del clasificador. Se dice que el vector aleatorio  $\mathbf{x}$  con  $k$  componentes reales sigue una distribución  $q$ -Gaussiana si su función de densidad de probabilidad es de la forma:

$$\text{Si } 1 < q < \frac{k+2}{k} : \\ f_{\mathbf{x},q}(\mathbf{x}) = A_q (1 - B_q (\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}))_+^{\frac{1}{1-q}} \quad (1)$$

donde  $A_q = \frac{\Gamma(\frac{1}{q-1})}{\Gamma(\frac{1}{q-1} - \frac{k}{2}) |\pi \Lambda|^{1/2}}$ ,  $B_q = \frac{1-q}{(k+4) - q(k+2)}$ ,  $\Sigma$  es la matriz de varianzas-covarianzas de  $\mathbf{x}$ ,  $\Lambda = m\Sigma$  y  $m$  el número de grados de libertad que se define en términos de la dimensión de  $\mathbf{x}$  de la forma  $m = \frac{2}{q-1} - k$ . Hay que hacer notar que cuando  $q \rightarrow 1+$ , la distribución  $q$ -Gaussiana converge a una distribución Normal estándar. Mientras que si  $q < 1$ , la función de densidad quedaría de la forma:

$$f_{\mathbf{x},q}(\mathbf{x}) = A_q (1 - \frac{1-q}{(k+4) - q(k+2)} (\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}))_+^{\frac{1}{1-q}} \quad (2)$$

donde  $A_q = \frac{\Gamma(\frac{2-q+d}{q-1})}{\Gamma(\frac{2-q-d}{1-q} - \frac{d}{2}) |\pi \Omega|^{1/2}}$ ,  $\Omega = d\Sigma$ , el parámetro  $d$  se define como  $d = 2\frac{2-q}{1-q} + k$ , siendo la notación  $(x)_+ = \max(x, 0)$ . Como una consecuencia, la función de probabilidad (2)

tiene un soporte acotado igual al dominio:

$$\{\mathbf{x} \in \mathbb{R}^k \mid (\mathbf{x} - \boldsymbol{\mu})^T \Omega^{-1} (\mathbf{x} - \boldsymbol{\mu}) < 1\} \quad (3)$$

Cuando  $q \rightarrow 1-$ , la distribución  $q$ -Gaussiana converge a la distribución Normal. Las distribuciones (1) y (2) se denominan en la literatura estadística  $t$  de Student y  $r$  de Student multivariantes. Los casos  $q > 1$  y  $q < 1$  se corresponden con dos tipos diferentes de distribuciones de acuerdo con el principio de superestadística de Beck y Cohen [11]. La distribución para  $q > 1$  se corresponde con un sistema Gaussiano sujeto a fluctuaciones que son independientes del estado del sistema y donde la variable  $X$  tiene un soporte no acotado. Para  $q < 1$  se corresponde también a un sistema de fluctuaciones Gaussianas para el cual la amplitud de las fluctuaciones dependen del estado del sistema y donde la variable  $\mathbf{x}$  tiene un soporte asociado acotado [12]. Por otra parte las distribuciones  $q$ -Gaussianas con  $q > 1$  difieren de las distribuciones Gaussianas estándar en que tienen una cola menos suave de forma tal que para la función  $f_{\mathbf{x},q}$  definida en (1) en el caso unidimensional:

$$\Pr(X > x) = \int_x^\infty f_{x,q}(u) du \sim \alpha x^{\frac{3-q}{1-q}}$$

siendo  $\alpha$  una constante. Como una consecuencia esta distribución tiene sólo momentos de orden  $p$  con  $p < \frac{3-q}{1-q}$ , por lo tanto la distribución de Cauchy no tiene momentos finitos, mientras que la distribución Gaussiana estándar si los tiene.

En este trabajo proponemos utilizar funciones no probabilísticas del tipo  $q$ -Gaussiano, en donde se sustituye la expresión gaussiana tradicional por su expresión  $q$ -exponencial equivalente, por lo que definiremos la  $i$ -ésima función  $q$ -Gaussiana como:

$$\phi_i(d_i(\mathbf{x})) = (1 - (1-q)d_i(\mathbf{x}))_+^{\frac{1}{1-q}} \quad (4)$$

para los casos en los que  $(1 - (1-q)d_i(\mathbf{x})) \geq 0$  y 0 en los restantes casos, siendo  $d_i(\mathbf{x}) = \frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{\theta_i^2}$ , donde  $\theta_i$  es un escalar que representa el radio de la  $i$ -ésima función de base,

mientras  $\|\cdot\|$  representa la distancia Euclídea y  $\mathbf{c}_i = (c_{i1}, c_{i2}, \dots, c_{ik})$  representa el vector de centros de la  $i$ -ésima función de base radial.

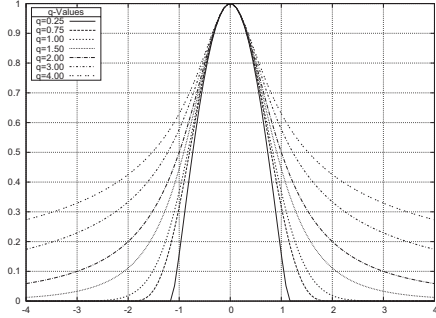


Figura 1: Activación de la función de base  $q$ -Gaussiana RBF con distintos valores del parámetro  $q$

Esta función puede reproducir diferentes tipos de RBF para diferentes valores del parámetro real  $q$ . De esta forma cuando el valor del parámetro es 2 la  $q$ -Gaussiana es la CRBF, para  $q=3$  tenemos una función de activación del tipo IMRBF para  $d_i(\mathbf{x})$  igual a la función de activación de una función  $q$ -Gaussiana con  $d_i(\mathbf{x})/2$ . Por último cuando el valor de  $q$  converge a 1 tenemos la función Gaussiana estándar (SRBF). La Figura 1 representa las funciones de activación  $q$ -Gaussianas para diferentes valores de  $q$ , donde podemos ver que ligeros cambios en el valor de  $q$  representan una modificación alisada de la forma de la RBF. De esta forma definimos una red  $q$ -Gaussiana RBF como una red con conexiones hacia delante con tres capas, la de entrada, la capa oculta y la de salida. Sea  $k$ ,  $m$  y 1 el número de nodos de entrada, ocultos y de salida (vamos a considerar problemas de clasificación binaria). Entonces el modelo de red se puede describir mediante la siguiente ecuación:

$$f(\mathbf{x}, \boldsymbol{\theta}) = \beta_0 + \sum_{i=1}^m \beta_i \phi_i(d_i(\mathbf{x})) \quad (5)$$

donde  $\phi_i(d_i(\mathbf{x}))$  está definida en (4),  $\boldsymbol{\theta}$  es el conjunto de parámetros del modelo,  $\beta_i$  son las conexiones de capa oculta a capa de salida y  $\beta_0$

es el sesgo del modelo. Las salidas de las neuronas se interpretan desde el punto de vista probabilístico a través de la función de activación softmax. De esta manera, la función de probabilidad condicionada es:

$$p(\mathbf{x}; \boldsymbol{\theta}) = \frac{e^{f(\mathbf{x}, \boldsymbol{\theta})}}{1 + e^{f(\mathbf{x}, \boldsymbol{\theta})}} \quad (6)$$

donde  $p(\mathbf{x}; \boldsymbol{\theta})$  es la probabilidad de que el patrón pertenezca a la clase  $Y = 1$ . En este caso los bordes de decisión son funciones no lineales y están definidas por la hipersuperficie  $f(\mathbf{x}, \boldsymbol{\theta}) = 0$  en el espacio  $\mathbb{R}^k$ . Dependiendo del parámetro  $\boldsymbol{\theta}$ , la hipersuperficie puede incluso ser no conexa.

Utilizando la función softmax definida mediante la ecuación (6), la clase predicha por la red QRBF se corresponde con el nodo de la capa de salida que tiene mayor valor de probabilidad condicionada asociado. De esta manera, en el caso de clasificación binaria, donde sólo definimos un nodo en la capa de salida, la regla de clasificación óptima es de la forma  $C(\mathbf{x}) = l$ , donde  $l = \arg \max [p(\mathbf{x}, \boldsymbol{\theta}); (1 - p(\mathbf{x}, \boldsymbol{\theta}))]$ .

La mejor red QRBF la determinaremos mediante un algoritmo híbrido que definiremos más adelante y que optimizará el logaritmo de la función de verosimilitud, función que consideraremos como función de error y que para  $n$  observaciones se define como:

$$l(\boldsymbol{\theta}) = \sum_{l=1}^n \left\{ y_l f(\mathbf{x}_l, \boldsymbol{\theta}) - \log(1 + e^{f(\mathbf{x}_l, \boldsymbol{\theta})}) \right\} \quad (7)$$

La naturaleza y las propiedades de la función  $f(\mathbf{x}, \boldsymbol{\theta})$  implican que la matriz Hessiana asociada a la maximización de (7) es en general indefinida y que puede tener máximos locales, y por tanto, es probable que el algoritmo se quede atrapado en ellos. Para salvar este problema utilizamos un algoritmo evolutivo como parte del proceso de estimación de los coeficientes del modelo y también como un método para determinar el número  $m$  de unidades  $q$ -Gaussianas que resulte más apropiado.

#### 4. Aprendizaje Híbrido de los parámetros

La dinámica de trabajo del algoritmo híbrido implementado es la siguiente: inicializamos la

- 1: **Algoritmo Híbrido:**
- 2: Generar aleatoriamente una población de tamaño  $N$
- 3: **repetir**
- 4: Calcular la aptitud de cada individuo de la población
- 5: Ordenar los individuos con respecto a su aptitud
- 6: Copiar el mejor individuo en la siguiente población
- 7: Copiar el mejor 10 % de los individuos de la población y sustituirlos por el peor 10 % de los individuos
- 8: Aplicar mutación paramétrica a los  $(p_m)$  % mejores individuos de la población
- 9: Aplicar mutación estructural a los restantes  $(100 - p_m)$  % individuos de la población
- 10: **hasta** alcanzar el criterio de parada
- 11: Aplicar el algoritmo *iRprop+* a la mejor solución obtenida por el EA en la última generación.

Figura 2: Estructura del Algoritmo Híbrido (AH)

búsqueda de los mejores modelos de red construyendo una población inicial de redes neuronales  $q$ -Gaussianas, y en cada iteración se utiliza un algoritmo de búsqueda basado en poblaciones que cambia la población inicial mediante cambios en la estructura y en los pesos de los modelos de red. La población esta sujeta por tanto a operaciones de replicación y de mutación. La Figura 2 muestra el procedimiento para seleccionar los parámetros de las unidades radiales QRBF. Las principales características son las siguientes:

- Representación de los individuos. El algoritmo evoluciona conjuntamente tanto la estructura de las redes (número de nodos y número de conexiones) como los valores de dichas conexiones o pesos. Cada individuo de la población será un modelo completo de red QRBF. Las redes neuronales se representan utilizando una aproximación orientada a objetos de forma tal que el algoritmo actúa directamente sobre el fenotipo del individuo. Dentro de esta representación, cada conexión está especificada mediante un valor binario que indica si la conexión ex-

iste o no y un valor real que representa su peso.

- Funciones de Error y de Aptitud. Consideraremos la función  $l(\theta)$  definida en (7) como la función de error asociada a un modelo  $g$  de red. Por tanto definimos la medida de aptitud de un individuo mediante una función estrictamente decreciente de la función de error  $l(\theta)$  en la forma  $A(g) = \frac{1}{1+l(g,\theta)}$ , donde  $0 < A(g) \leq 1$ .
- Inicialización de la Población. La población inicial se genera intentando obtener redes QRBF que tengan una buena aptitud inicial. De esta forma generamos al azar 5000 redes. A continuación definimos los centros mediante un algoritmo de agrupamiento del tipo  $k$ -medias, utilizando para ello de forma empírica diferentes valores de  $k$  entre un valor mínimo,  $M_{min}$ , y un valor máximo,  $M_{max}$ , del número de nodos en capa oculta. El radio de las unidades QRBF se inicializa utilizando la media geométrica de la distancia del centro de la QRBF a los dos vecinos más cercanos y el valor de  $q$ , asociado a la forma de la función y muy sensible frente a pequeños cambios, se inicializa a valores cercanos a 1 puesto que ya comentamos que una función QRBF converge a una función SRBF cuando  $q \rightarrow 1$ . Por otra parte asignamos a los pesos de las conexiones entre las capas oculta y de salida de cada modelo de red de la población inicial un valor aleatorio comprendido entre  $[-5, 5]$ . Los individuos obtenidos son evaluados utilizando la función de aptitud y por último obtenemos la población inicial seleccionando los 500 mejores modelos.
- Mutación estructural. Existen en el algoritmo cuatro tipos de mutaciones estructurales: añadir y eliminar nodos en la capa oculta y añadir y eliminar conexiones. Estos cuatro tipos de mutadores se aplican secuencialmente a cada individuo (modelos de red QRBF) de la población, cada uno con una determinada probabilidad. Si el mutador estructural añade

un nuevo nodo a un individuo se asigna al parámetro  $q$  un valor comprendido entre  $[0,75, 1,25]$ , puesto que es un entorno simétrico centrado en el valor del parámetro de una SRBF ( $q \rightarrow 1$ ).

- **Mutación Paramétrica.** Dadas las características de cada uno de los parámetros que conforman un modelo de red QRBF, se definen diferentes mutadores para cada uno de ellos: centros, radios, parámetros que definen la forma de las funciones de base y coeficientes de la partes lineal del modelo. En el caso de los centros su valor se modifica añadiéndole una variable aleatoria Normal de media 0 y de desviación típica la mitad del radio de la función de base radial, en la forma:

$$c_i(t+1) = c_i(t) + N(0, r_i)$$

En el caso de los radios, cada valor se modifica añadiéndole una variable aleatoria Normal de media 0 y de desviación típica  $d_i$ , según la ecuación:

$$r_i(t+1) = r_i(t) + N(0, d_i)$$

donde  $d$  es la mitad del dominio de las variables normalizadas.

Con respecto a la mutación del parámetro  $q$ , esta se realiza añadiendo un valor aleatorio asociado a una distribución uniforme definida en el intervalo  $[-0,25, 0,25]$ , dado que las modificaciones de este valor varían de forma substancial las características y la forma de la función  $q$ -Gaussiana, como hemos visto en la sección de teoría y en la Figura 1. Las mutaciones de las conexiones (coeficientes del modelo lineal) entre la capa oculta y la capa de salida se realizan añadiendo una variable aleatoria Normal de media 0 y de varianza asociada a una función de enfriamiento simulado, de forma tal que su valor vaya disminuyendo conforme avanza el algoritmo evolutivo:

$$w_j(t+1) = w_j(t) + N(0, T(g)),$$

siendo  $T(g) = 1 - A(g)$ .

- **Optimizador local iRprop+.** El algoritmo de optimización local utilizado en nuestro procedimiento es el método iRprop+ [33] puesto que es uno de los más rápidos y robustos. Este algoritmo aplica una estrategia de retrotrazado, de forma tal que decide si volver o no a la dirección anterior del vector de pesos por medio de una heurística. En la metodología híbrida propuesta en el trabajo se ejecuta el Algoritmo Evolutivo hasta obtener el mejor individuo en la última generación, sobre el que aplicaremos a continuación el algoritmo iRprop+ de búsqueda local.

## 5. Experimentación

Los experimentos han sido planteados con el objetivo de contrastar los resultados de nuestro modelo de red QRBF con los modelos estándar de redes SRBF evaluando el rendimiento sobre cuatro bases de datos de prueba de clasificación binaria y que han sido tomadas del repositorio de la UCI. La tipología QRBF de modelo de red de base radial la hemos comparado con el mismo algoritmo híbrido utilizando otros tipos de redes de base radial donde las funciones son fijas en todos los nodos de la capa oculta; así, lo comparamos con: redes de base radial estándar (SRBF) donde las funciones de transferencias son Gaussianas; con las funciones de base radial de Cauchy (CRBF) y con las funciones de base radial Multicuadrática Inversa (IMRBF).

El diseño experimental para los problemas propuestos fue realizado utilizando un procedimiento de validación cruzada del tipo *10-fold* con 10 ejecuciones para cada *fold*, dado que los algoritmos evolutivos son estocásticos. Los parámetros utilizados en el algoritmo son comunes para los 4 problemas abordados en este trabajo a excepción del número máximo de generaciones, 100 para Heart y Heart-C y 40 para Breast-C y Liver. Los valores asociados a los valores de  $M_{min}$  y  $M_{max}$  son 2 y 5 respectivamente; las conexiones entre los nodos de la capa oculta y la de salida se inicializan en el intervalo  $[-5, 5]$ , el tamaño de la población es 500. El número de nodos que pueden ser

Cuadro 1: Características de las bases de datos: Tamaño muestral, número de variables reales (R), binarias (B) y nominales (N); número total de variables (T), distribución de los patrones por clase (Dist.)

Datos	Casos	R	B	N	T	Dist.
Heart	270	13	-	-	13	(150,120)
Breast-C	286	4	3	2	15	(201,85)
Heart-C	302	6	3	4	26	(164,138)
Liver	345	6	-	-	6	(145,200)

añadidos o eliminados mediante una mutación estructural es 1 o 2; mientras que el número de conexiones que se pueden añadir o eliminar en una mutación estructural es un número entero del intervalo [1, 7]. La condición propuesta para el criterio de parada es que el algoritmo alcance el número máximo de generaciones.

## 6. Resultados

En el Cuadro 1 describimos brevemente las cuatro bases de datos de clasificación binaria, con un número diferente de patrones, que van desde 270 hasta 345, diferentes tipos de variables de entrada, en escala nominal, binaria o dicotómica y continua. Para validar el rendimiento del modelo utilizamos la proporción de patrones correctamente clasificados para el conjunto de generalización ( $C_G$  %) y que se define como el porcentaje de patrones de los datos de generalización correctamente clasificados. En la Tabla 2 se incluyen los resultados obtenidos en las bases de datos evaluadas.

Para estas cuatro bases de datos, en tres de ellas los mejores resultados se obtienen utilizando las funciones QRBF mientras que este tipo presenta el segundo mejor resultado en la otra base de datos. Aún así no existen diferencias significativas entre las medias de los resultados obtenidos con el tipo de función QRBF y el siguiente tipo de función de base radial; de esta forma aunque los resultados son prometedores no son concluyentes por lo que será necesario ampliar el diseño experimental para un buen número de bases de datos de clasi-

ficación binaria y multiclase, realizando contrastes de hipótesis de igualdad de resultados medios tanto paramétricos como no paramétricos, para poder concluir que este tipo de funciones de base radial que conllevan la estimación del parámetro adicional  $q$  tienen sentido como alternativa a las funciones de activación estándar de los modelos de red neuronal de base radial.

## 7. Conclusiones

En este trabajo, nos hemos centrado en los problemas de clasificación binaria mediante modelos de redes neuronales de funciones de base  $q$ -Gaussianas una familia de funciones de transferencia, donde la más habitual es la Gaussiana (SRBF), aunque también forman parte de esta familia la Multicuadrática (MRBF), la Inversa Multicuadrática (IMRBF) y la de Cauchy (CRBF); las cuales se obtienen para determinados valores del parámetro  $q$ . Para el entrenamiento de este tipo de modelo de red neuronal hemos utilizado un algoritmo híbrido formado por un algoritmo de programación evolutiva que busca las mejores regiones del espacio de modelos de red junto con un algoritmo de búsqueda local basado en gradiente, en concreto una versión mejorada del iRprop. El modelo de red  $q$ -Gaussiana propuesto lo hemos aplicado a cuatro bases de datos de prueba mejorando en tres de ellas los resultados obtenidos con la misma metodología pero utilizando nodos en capa oculta con funciones prefijadas de base radial. Por último, hay que destacar que nuestros resultados son competitivos con los resultados que habitualmente se citan para estas bases de datos.

## Agradecimientos

Este trabajo ha sido financiado en parte por el proyecto TIN 2008-06681-C06-03 de la comisión interministerial de Ciencia y Tecnología (MICYT), y con fondos FEDER y con el proyecto P08-TIC-3745 de Excelencia de la Junta de Andalucía. La investigación de Francisco Fernández-Navarro ha sido financiada por el programa predoctoral de la Junta

Cuadro 2: Comparación de las funciones de base radial propuestas con otras funciones de base en 100 ejecuciones: media±desviación típica de la precisión medida en  $C_G$  (%)

Datos	SRBF	CRBF	IMRBF	QRBF
Heart	81.85±8.97	83.70±8.76	<b>84.81±8.45</b>	84.07±7.20
Breast-C	72.04±6.39	71.35±8.00	73.10±6.39	<b>73.16±6.77</b>
Heart-C	85.44±3.83	85.45±5.59	85.77±3.05	<b>85.79±5.20</b>
Liver	68.41±5.15	65.23±8.23	65.52±6.31	<b>71.30±6.50</b>

de Andalucía con referencia 390015-P08-TIC-3745.

### Referencias

- [1] M. Bishop, *Neural Networks for Pattern Recognition*: Oxford University Press, 1995.
- [2] Hervás, C. Martínez, F.J. y Gutiérrez, P.A. Classification by means of Evolutionary Product-Unit Neural Networks. *IEEE World Congress of Computational Intelligence. International Joint Conference on Neural Networks*. 2006. Vancouver, Canada.
- [3] Duch, W. and Jankowsky, N., Transfer functions: hidden possibilities for better neural networks, 9th European Symposium on Artificial Neural Networks, (ESANN), Brugge, pp. 81-94, 2001.
- [4] McLoone, S., Brown, M. D., Irwin, G., & Lightbody, G. (1998). A hybrid linear/nonlinear training algorithm for feed-forward neural networks. *IEEE Transactions on Neural Networks*, 9 (4), 669–684.
- [5] Gonzalez, J., Rojas, I., Ortega, J., Pomares, H., Fernández, J., & Diaz, A. F. (2003). Multiobjective evolutionary optimization of the size, shape, and position parameters of radial basis function networks for function approximation. *IEEE Transactions on Neural Networks*, 14 (6), 1478–1495.
- [6] Whitehead, B. A., & Choate, T. D. (1996). Cooperative-competitive genetic evolution of radial basis function centers and widths for time series prediction. *IEEE Transactions on Neural Networks*, 7 (4), 869–880.
- [7] M. J. Orr, “Optimising the widths of radial basis functions,” in *Proceedings of the 5th Brazilian Symposium Neural Networks*, 1998, pp. 26–29.
- [8] S. Chen, C. F. N. Cowan, and P. M. Grant, “Orthogonal least squares learning algorithm for radial basis functions,” *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 302–309, 1991.
- [9] K. Hunt, D. Sbarbaro, R. Zbikowski, and P. Gawthrop, “Neural networks for control system - a survey,” *Automatica*, vol. 28, pp. 1083–1112, 1992.
- [10] O. Buchtala, M. Klimek, and B. Sick, “Evolutionary optimization of radial basis function classifiers for data mining applications,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 35, no. 5, pp. 928–947, 2005.
- [11] C. Beck, E.G.D. Cohen, *Physica A* 344 (2004) 393.
- [12] C. Vignat, A. Plastino, *Phys. Lett. A* 343 (2005) 411.
- [13] Igel, C., Husken, M., 2003. Empirical evaluation of the improved rprop learning algorithms. *Neurocomputing* 50 (6), 105–123.