

Tecno Lógicas

ISSN: 0123-7799

tecnologicas@itm.edu.co

Instituto Tecnológico Metropolitano

Colombia

Botero Valencia, Juan Sebastián; Sánchez Giraldo, Luis Gonzalo; Delgado Trejos, Edilson
Clasificador no lineal basado en redes neuronales con funciones de base radial para implementación
en sistemas de punto fijo

Tecno Lógicas, núm. 22, julio, 2009, pp. 11-28

Instituto Tecnológico Metropolitano
Medellín, Colombia

Disponible en: <http://www.redalyc.org/articulo.oa?id=344234314002>

- ▶ Cómo citar el artículo
- ▶ Número completo
- ▶ Más información del artículo
- ▶ Página de la revista en redalyc.org

CLASIFICADOR NO LINEAL BASADO EN REDES NEURONALES CON FUNCIONES DE BASE RADIAL PARA IMPLEMENTACIÓN EN SISTEMAS DE PUNTO FIJO

JUAN SEBASTIÁN BOTERO VALENCIA¹

LUIS GONZALO SÁNCHEZ GIRALDO²

EDILSON DELGADO TREJOS³

Resumen:

Para implementar máquinas inteligentes, es común requerir de sistemas de clasificación que sean eficientes y realizables en plataformas con bajo nivel de procesamiento. En este trabajo se presenta un método de diseño para estimar los parámetros de un clasificador basado en redes neuronales con funciones de base radial para ser implementado en sistemas de procesamiento digital con punto fijo. En principio, usando métricas estadísticas se obtiene el número de centroides necesarios para llevar las clases a un espacio que las haga linealmente separables, posteriormente aplicando el algoritmo k -medias se estima la ubicación de los centroides. Se determina la distancia de los puntos de entrenamiento a cada centroide, y usando aproximación por mínimos cuadrados se calculan los pesos de la función de salida. De esta manera, se obtiene un clasificador con una complejidad computacional reducida que permita ser usado en sistemas con requerimientos de bajo nivel de procesamiento como los de tiempo real.

-
- 1 Ingeniero Electrónico. Estudiante de Maestría en Automatización y Control Industrial. Auxiliar de investigación. INSTITUTO TECNOLÓGICO METROPOLITANO. Email: jsboterov@unal.edu.co
 - 2 Ingeniero Electrónico. M. Sc. en Automatización Industrial, PhD Student Electrical and Computer Engineering Department University of Florida, Gainesville.
 - 3 Ingeniero Electrónico. M. Sc. en Automatización Industrial. Ph. D. en Ingeniería LI Automática. Académico Investigador del Centro de Investigación, INSTITUTO TECNOLÓGICO METROPOLITANO. Email: edilsondelgado@itm.edu.co

Fecha de recepción: 27 de marzo de 2009

Fecha de aceptación: 11 de mayo de 2009

Palabras clave

Redes neuronales, clasificador, centroide, base radial, punto fijo.

Abstract

Implementation of intelligent machines requires of efficient classification systems under limited computational resources. This study introduces a method for estimating the parameters of Radial Basis Function Neural Network (RBF-NN) that can be implemented on a fixed point processor. First, the number of hidden nodes is chosen based on statistics of the mapped data points. A k -means search is then carried out to determine the location of each node. The hidden units mapping corresponds to the Euclidean distance of their centers to each data point, the weights of the output sum are obtained by solving a linear least squares problem. With this procedure, a low computational cost classifier can be readily implemented on a low capacity platform for real time applications.

Keywords

Neural network, classifier, centroid, radial basis, fixed point.

INTRODUCCIÓN

Los sistemas de reconocimiento de patrones, cada vez son más necesarios en la vida cotidiana y la automatización ha hecho de la clasificación un problema fundamental del área de control. Es común que se necesiten implementar clasificadores en sistemas de baja complejidad de procesamiento sin que se tengan que esperar resultados deficientes.

Las redes neuronales *feed-forward* multicapa han sido usadas como estimadores universales con diferentes funciones de activación (Hush & Horne, 1998). Las redes neuronales con función de base radial (RBF) han sido desarrolladas con el fin de superar limitaciones relacionadas con las frecuentemente manifestadas en las *feed-forward*. Las redes RBF son caracterizadas por una estructura simple, de baja complejidad computacional y un desempeño adaptativo superior. Las redes RBF linealmente parametrizadas son más sencillas de inicializar, por lo que su entrenamiento exige menos tiempo que las redes multicapa, las cuales capturan la representación distribuida de las funciones (Schilling, Carroll, & Al-Ajlouni, 2001).

Sin embargo, el número de funciones de base radial para redes RBF es a menudo asumido como una constante o de inicialización aleatoria, es por esto que el número de centroides y el cálculo de los pesos son determinados sin contar con limitaciones de implementación, lo cual resulta en una red usualmente grande. En (Chen, Chng, & Alkadhimi, 1996) se propone un procedimiento de entrenamiento alternativo basado en mínimos cuadrados ortogonales donde las funciones de base radial son escogidas una por una hasta que se obtiene un desempeño deseado. Sin embargo, las rutinas de este método pueden tornarse exhaustivas. En (Kondo, Hatanaka, & Uosaki, 2006) se propone el entrenamiento usando optimización multi-objetivo y se muestra que las RBF son un buen método para clasificar patrones, ya que tienen una buena relación complejidad-desempeño, aunque el algoritmo es complejo, su

adaptabilidad puede llegar a verse comprometida y el rendimiento es similar al de los métodos comunes de implementación.

Un problema común se presenta al momento de determinar el número de centroides cuando se desea diseñar la arquitectura de la red neuronal RBF. En (Tinós & Murta Júnior, 2008) se expone una estrategias de clasificación basada en redes neuronales RBF que usa algoritmos genéticos para estimar el número de centroides asociados al número de neuronas de la capa oculta y el radio de la función radial. El problema es que esta técnica no se puede tomar como efectiva ante conjuntos reales de entrenamiento, ya que no se reporta su expresión generalizada y el conjunto sobre el que se hicieron las pruebas era muy limitado. En (Zhang, He, & Mak, 2001), se presenta la comparación de estrategias MLP y RBF para la clasificación de nubes, donde se encuentra que las estrategias RBF tienen mejor desempeño en general, sin embargo, aunque la MLP pueda ser mejor, encontrar la arquitectura adecuada es muy difícil. En (Schwenker, Kestler, & Palm, 2000), se presenta la comparación de varios métodos de entrenamiento de RBF para clasificación, y aplicaciones en 3D, sobre electrocardiogramas de alta resolución y escritura manual, donde se demuestra que los parámetros de la RBF ofrecen facilidad en la interpretación de los resultados.

En cuanto a la implementación en sistemas de punto fijo y punto flotante, en (Hernandez L. & Salazar G., 2006) se presenta una aplicación usando clasificación mediante SVM sobre FPGA's, determinándose que el punto flotante conduce a problemas con conjuntos de clasificación heterogéneos además de adicionar complejidad computacional, lo que lo hace prohibitivo para aplicaciones con FPGA ya que estos sistemas se caracterizan por su baja capacidad de procesamiento.

De acuerdo a esto, en este trabajo se presenta el diseño de un método que permite estimar los parámetros de un clasificador basado en funciones de base radial, ya que presentan alta capacidad de mapeo a hiperplanos en los que las clases son linealmente separables. Una de las ventajas del método de diseño implementado

es la estimación del número de centroides que adapta el tamaño de la red al tipo de separación que necesita ser implementada, además de encontrar la posición de los centroides que separa de mejor manera las clases.

MATERIALES Y MÉTODOS

Conjuntos de prueba

Para construir los conjuntos de prueba, se limita el espacio de características a una ventana fija que se expande [0 1023] para c_1 , y de [0 1023] para c_2 . Los conjuntos de datos c_1 y c_2 corresponden a la representación de dos clases artificiales diferentes. Esto con el fin de implementar los resultados en hardware con conversores analógico digitales de 10 bits.

Conjunto 1

Corresponde a una clase encerrada por encima y por debajo con fronteras lineales, nótese que la clase 2 no es conexa, tal como se muestra en la Figura 1.

1. La primera clase está construida con una distribución normal que varía entre [0 1023] para c_1 , y una distribución uniforme que varía entre [500 700] para c_2 .
2. La segunda clase está construida con una distribución normal que varía entre [0 1023] para c_1 , y una distribución uniforme que varía entre [0 500] U [700 1023] para c_2 .

Conjunto 2

En este conjunto, las clases se encuentran separadas por una frontera cuadrática, de forma que las dos clases son conexas, tal como se muestra en la Figura 2.

1. La primera clase está construida con una distribución normal para c_1 y c_2 además de ser limitada por la parábola 0.001.

2. La segunda clase está construida con una distribución normal para c_1 y una distribución uniforme para c_2 limitada por la parábola 0.001.

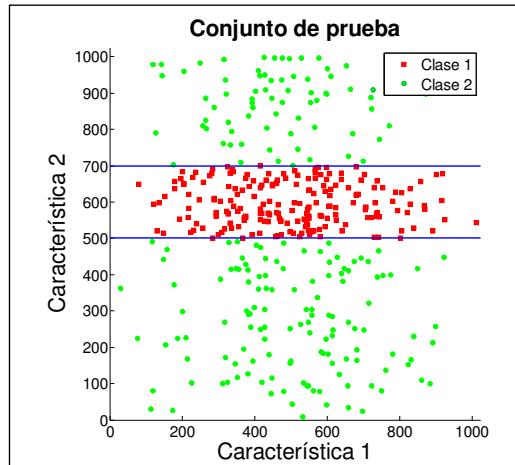


FIGURA 1. CONJUNTO DE PRUEBA 1

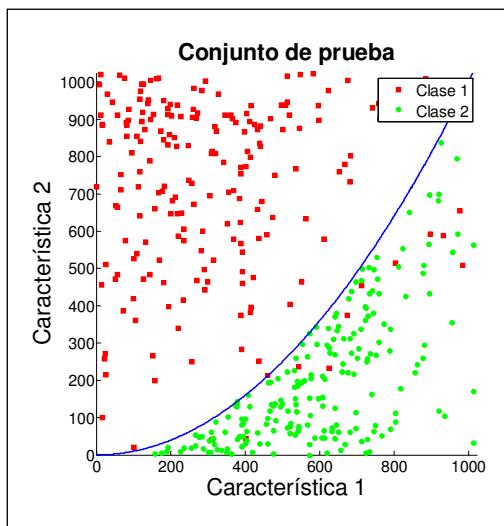


FIGURA 2. CONJUNTO DE PRUEBA 2

Conjunto 3

La clase 1 está encerrada por la clase 2, la frontera de discriminación es un círculo, las clases son conexas tal como se muestra en la Figura 3.

1. La primera clase está construida con una distribución normal para c_1 y c_2 . centrada en (500,500).
2. La segunda clase está construida con una distribución uniforme que varía entre [0 1023] para c_1 y c_2 limitada por $(c_1 - 500)^2 + (c_2 - 500)^2 < 300^2$.

Conjunto 4

La clase 2 está compuesta por una corona circular y el resto del espacio está compuesto por la clase 1. La clase 1 es no conexa, tal como se muestra en la Figura 4.

1. La primera clase está construida con una distribución uniforme que varía entre [0 1023] para c_1 y c_2 limitada por $(c_1 - 500)^2 + (c_2 - 500)^2 < 200^2$ U $(c_1 - 500)^2 + (c_2 - 500)^2 < 400^2$.
2. La segunda clase está construida con una distribución uniforme que varía entre [0 1023] para c_1 y c_2 limitada por $(c_1 - 500)^2 + (c_2 - 500)^2 > 200^2$ U $(c_1 - 500)^2 + (c_2 - 500)^2 < 400^2$.

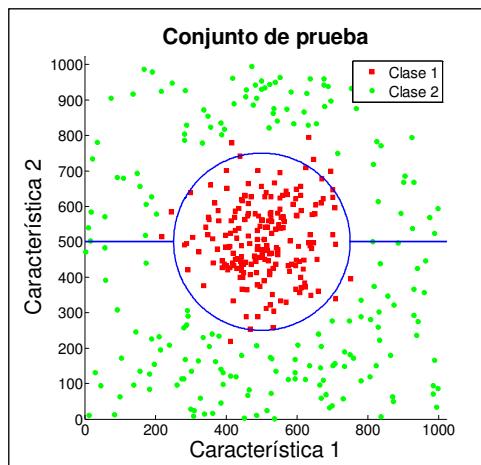


FIGURA 3. CONJUNTO DE PRUEBA 3

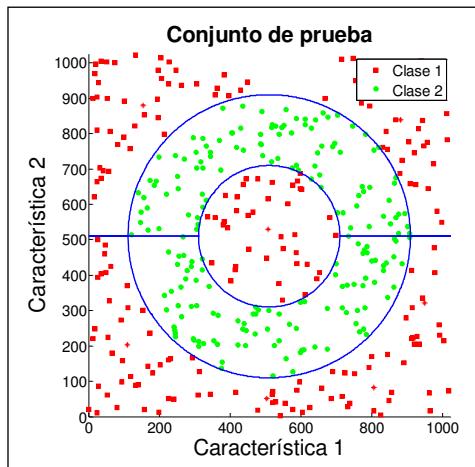


FIGURA 4. CONJUNTO DE PRUEBA 4

Los conjuntos de prueba se construyeron teniendo en cuenta algunas de las superficies que mayor dificultad presentan a los clasificadores comunes.

En la Tabla 1 se presentan los momentos estadísticos principales para cada clase de los conjuntos de prueba, donde, \bar{x} es el primer momento estadístico, σ es la desviación estándar y n es el número de muestras.

TABLA 1. MOMENTOS ESTADÍSTICOS DE LOS CONJUNTOS DE DATOS

Conjunto 1		Conjunto 2		Conjunto 3		Conjunto 4	
Clase1	Clase2	Clase1	Clase2	Clase1	Clase2	Clase1	Clase2
c_1	c_2	c_1	c_2	c_1	c_2	c_1	c_2
500	600	500	520	320	720	600	190
490	500	490	512	430	512	510	510
187	56	200	311	230	220	190	170
97	94	97	94	271	343	330	330
216	238	216	238	200	200	200	200
n	200	200	200	200	200	200	200

Arquitectura de las redes RBF

Este tipo de redes neuronales está comúnmente constituido por dos capas. Una capa oculta formada por funciones del tipo radial; y una capa de salida que realiza una suma ponderada a partir de las salidas de la capa oculta. Para esquemas de clasificación es común requerir de una fase final de umbralización como se muestra en la Figura 5.

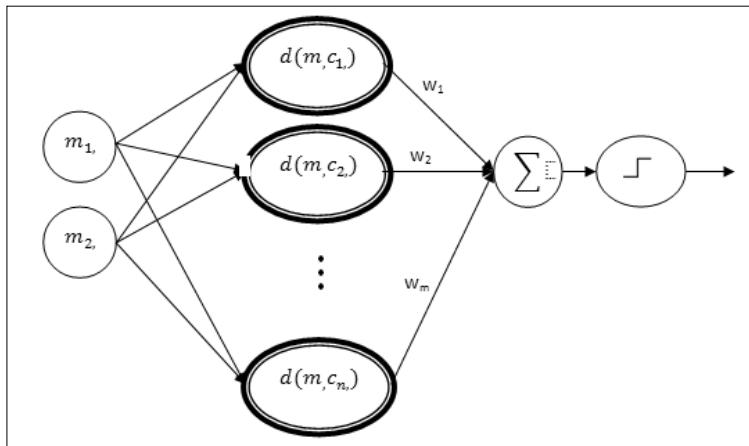


FIGURA 5: ARQUITECTURA RED RBF

La salida de la capa oculta tiene la forma:

$$D = [d(m_i, c_1) \quad d(m_i, c_2) \quad d(m_i, c_3) \quad \cdots \quad d(m_i, c_n)]$$

Donde m_i es el conjunto de variables de entrada, c_1, c_2, \dots, c_n son los centroides estimados en la fase de entrenamiento por el algoritmo de k -medias y denota una función de distancia radial entre la muestra y el centroide. Nótese que D , el vector resultante de la capa oculta, es de dimensión $[1 \times n]$, donde n es el número de centroides como se aprecia en la Figura 5.

La capa de salida se puede expresar:

$$S = D \cdot W = [d(m_i, c_1) \quad d(m_i, c_2) \quad d(m_i, c_3) \quad \cdots \quad d(m_i, c_n)] \cdot \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix}$$

Donde D es el vector de distancias obtenido de la capa oculta, W es el vector de pesos de la capa de salida obtenido en el algoritmo de entrenamiento y S es la única salida de la red $[1 \times 1]$. Como se había expresado anteriormente puede ser necesaria la implementación de un tipo de umbralización que simplifique el proceso de identificación de la salida.

Algoritmo k-medias

El algoritmo de k -medias es empleado para encontrar agrupamientos en un espacio de características. Para ilustrar este método, sea n el número de características en vectores x_1, x_2, \dots, x_n , donde x representa un espacio m dimensional y suponemos que el espacio de características se puede agrupar en k cúmulos. Se define μ_j como la media del j -esimo centroide. Ahora usando una distancia para separarlos, se puede decir que los elementos x_i están en el j -esimo cúmulo si $d(x_i, \mu_j)$ es el mínimo con respecto a los k cúmulos. El algoritmo se puede simplificar entonces:

1. Estimar las k -medias, $\mu_1, \mu_2, \dots, \mu_k$.
2. Mientras no cambia alguna media.
 - a. Usar la media estimada para clasificar los datos en cúmulos, $b(i,j)=1$ si el i -esimo dato es el más cercano a la j -esima media.
 - b. Calcular la nueva media para todos los cúmulos:

$$\mu_j = \frac{\sum_{i=1}^n b(i,j) x_i}{\sum_{i=1}^n b(i,j)}$$

3. El algoritmo se repite sucesivamente hasta que no cambie ninguna media.

Teorema de Cover

Un problema de clasificación de patrones transformado no linealmente a un espacio de dimensión superior tiene mayor probabilidad de ser linealmente separable que en un espacio de dimensión menor. Cuanto mayor es el número de neuronas ocultas

mayor es la probabilidad de separabilidad lineal. En muchos problemas es suficiente con una transformación no lineal sin aumentar la dimensión.

En el caso de las RBF se espera que el espacio oculto sea linealmente separable, ya que la capacidad de la capa de salida es limitada y eficiente solo en esta condición. Esta conclusión está basada en el teorema de Cover descrito en (Haykin, 1999).

Número de centroides

Uno de los problemas comunes al utilizar métodos basados en agrupamiento es el de la escogencia del número ideal de centroides. Para muchas técnicas usadas actualmente (Duda, Hart, & Stork, 2000), esta escogencia se realiza de forma heurística, pero es claro que esto reduce la flexibilidad y adaptabilidad del algoritmo de clasificación.

En este estudio se presenta un método para determinar un número adecuado de núcleos que cubran de manera óptima la superficie perteneciente a una clase. En primer lugar hay que tener en cuenta que para los algoritmos basados en agrupamiento, la solución buscada es una solución geométrica, por tanto, la técnica que más puede ayudar debe estar fundamentada en el análisis de distancias y su disimilitud. De acuerdo a lo que se muestra en la Figura 6, el círculo es una de las figuras geométricas que mejor cubren una superficie no lineal.

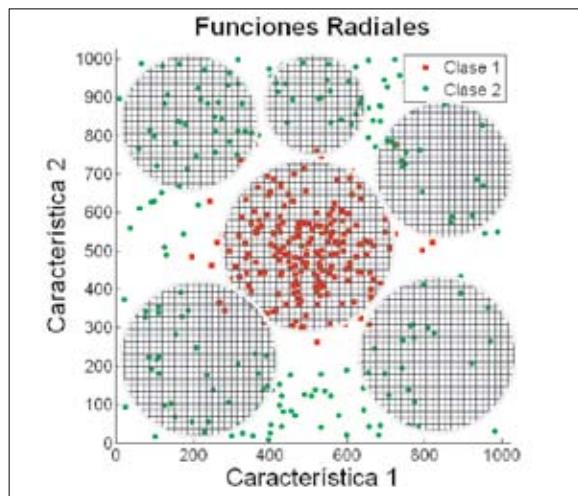


FIGURA 5. DISTRIBUCIÓN GEOMÉTRICA DE FUNCIONES DE BASE RADIAL

En este caso, se comienza por obtener una matriz de distancias euclídeas para cada clase:

$$D_1 = \begin{bmatrix} d(m_1, m_1) & d(m_1, m_2) & d(m_1, m_3) & \cdots & d(m_1, m_n) \\ d(m_2, m_1) & d(m_2, m_2) & d(m_2, m_3) & \cdots & d(m_2, m_n) \\ d(m_3, m_1) & d(m_3, m_2) & d(m_3, m_3) & \cdots & d(m_3, m_n) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d(m_m, m_1) & d(m_m, m_2) & d(m_m, m_3) & \cdots & d(m_m, m_n) \end{bmatrix}$$

Donde:

$$d(m_i, m_j) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

$$m_i = (x_i, y_i)$$

Después de obtener la matriz se calcula la desviación estándar en dos dimensiones. La medida de disimilitud en el agrupamiento viene dada por la desviación que muestra la dispersión de los datos respecto al valor medio.

$$\sigma(D_1) = ak$$

Donde a es un parámetro de escalamiento y k el número de centroides. El factor de escala fue determinado realizando pruebas de compromiso entre complejidad y rendimiento del clasificador,

y fue aplicado el mismo factor a todos los conjuntos de prueba. El número de centroides se estima independientemente para cada clase.

Pesos de la capa de salida

Se trata de encontrar la solución que mejor se ajuste al sistema:

$$Gx = w$$

Donde:

$$G = \begin{bmatrix} d(m_1, c_1) & d(m_1, c_2) & d(m_1, c_3) & \cdots & d(m_1, c_n) & 1 \\ d(m_2, c_1) & d(m_2, c_2) & d(m_2, c_3) & \cdots & d(m_2, c_n) & 1 \\ d(m_3, c_1) & d(m_3, c_2) & d(m_3, c_3) & \cdots & d(m_3, c_n) & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & 1 \\ d(m_m, c_1) & d(m_m, c_2) & d(m_m, c_3) & \cdots & d(m_m, c_n) & 1 \end{bmatrix}$$

Las filas de G están construidas con la salida de la capa oculta. Nótese que se agrega una columna de 1's a la derecha de la matriz de entrenamiento, este parámetro es conocido como *bias* y es usado para ajustar el umbral de la salida de la red. El número de centroides n , resulta de la suma del número de centroides estimados para cada clase ($k_1 + k_2$), y el número de muestras m es el número total de muestras de ambas clases en el conjunto de entrenamiento.

El vector de salida w deseado se construye a partir de:

$$w = \begin{bmatrix} a \\ a \\ \vdots \\ b \\ b \end{bmatrix}$$

En el vector de salida los valores de y se escogen de manera que sean fácilmente interpretables por el sistema en el que se implementará el clasificador.

Para resolver este sistema debemos hacer uso de la pseudoinversa. Premultiplicando se tiene:

$$G^T Gx = G^T w$$

Despejando x , se obtiene la proyección del vector w en el espacio columna $R(G)$:

$$\bar{x} = p = (G^T G)^{-1} G^T w$$

Ya que la matriz es simétrica el sistema es invertible y por lo tanto siempre tiene solución.

RESULTADOS Y DISCUSIÓN

Número de grupos

Las Figuras 6 y 7 muestran la matriz ordenada de distancias para el conjunto de prueba 1 y la clase 1 y 2, respectivamente. Si se compara esta gráfica con los resultados obtenidos en la Tabla 2 acerca del número de centroides estimados, se puede observar que la variabilidad es mucho mayor en la clase 2 y por tanto se requieren más centroides para mapear adecuadamente la clase.

FIGURA 6. DISTANCIAS CLASE 1

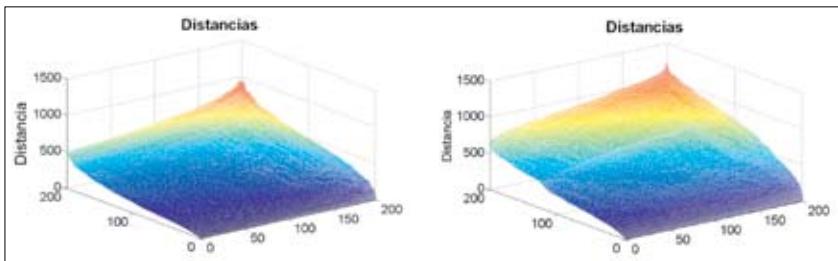
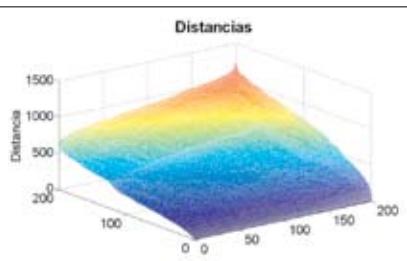
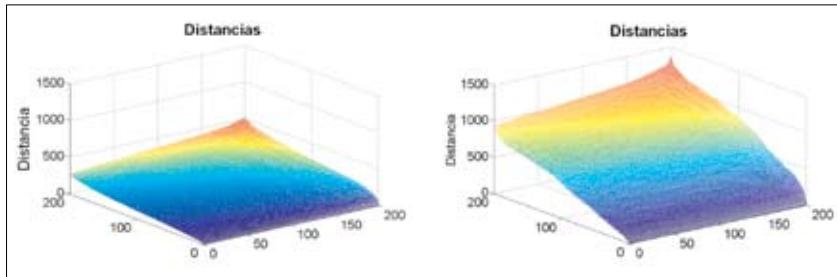


FIGURA 7. DISTANCIAS CLASE 2

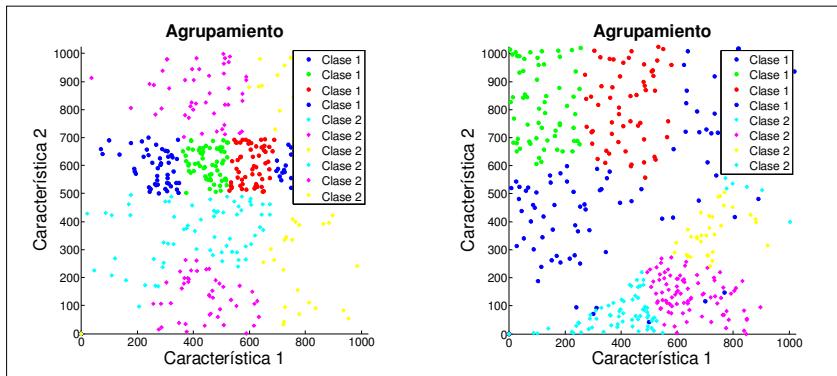


La Figura 8 y 9 muestran la matriz ordenada de distancias para el conjunto de prueba 3 y la clase 1 y 2 respectivamente. En esta gráfica el efecto de variabilidad es más notorio. La clase 1 varía poco y el algoritmo empleado para la estimación de centroides determinó un solo grupo, mientras que para la clase 2 que presenta mayor variabilidad el algoritmo estimó 4 centroides.

**FIGURA 8.** DISTANCIAS CLASE 1**FIGURA 9.** DISTANCIAS CLASE 2

Agrupamiento

En las Figuras 8 y 9 se puede observar la distribución de los agrupamientos para los conjuntos de prueba 1 y 2, respectivamente. Es claro que los subconjuntos generados por el agrupamiento corresponden a una forma de separación uniforme de la clase.

**FIGURA 8.** AGRUPAMIENTO CONJUNTO 1**FIGURA 9.** AGRUPAMIENTO CONJUNTO 2

La Figura 10 muestra la eficiencia del algoritmo de escogencia de centroides, en este caso la clase 1 puede ser mapeada con un solo subconjunto, y dada la distancia de este centroide a los centroides de la segunda clase, la segunda clase puede ser mapeada con tan sólo 4 subconjuntos.

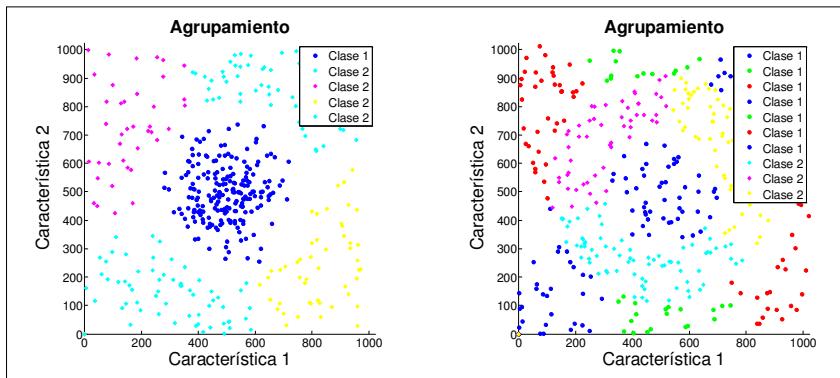


FIGURA 10. AGRUPAMIENTO CONJUNTO 3

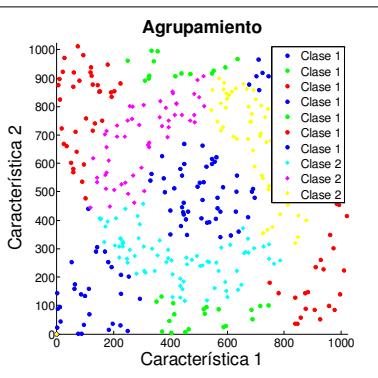


FIGURA 11. AGRUPAMIENTO CONJUNTO 4

Cuando se trata del problema de la corona circular (Figura 11), los clasificadores suelen tener dificultad, primero por tratarse de una superficie no conexa para la clase 2, y segundo por la forma de la función discriminante necesaria para separar las clases.

En la Tabla 2 se muestra la estimación del número de centroides óptimo entregado por el algoritmo desarrollado.

TABLA 2. NÚMERO DE CENTROIDES POR CLASE

	Conjunto 1		Conjunto 2		Conjunto 3		Conjunto 4	
	Clase1	Clase2	Clase1	Clase2	Clase1	Clase2	Clase1	Clase2
k	4	6	4	4	1	4	7	3

Error de clasificación

En la Tabla 3 se muestra el error de clasificación de los conjuntos de prueba, nótese que aunque el error es tolerable, la clase tipo corona circular, es la que presenta un mayor error. ε_1 es el error usando el conjunto de entrenamiento y ε_2 el error del clasificador promediado sobre 10 conjuntos de prueba generados con las mismas características.

TABLA 3. ERROR DE CLASIFICACIÓN

Conjunto 1		Conjunto 2		Conjunto 3		Conjunto 4	
Clase1	Clase2	Clase1	Clase2	Clase1	Clase2	Clase1	Clase2
ε_1 (%)	0.05	0.06	10.53	4.2	6.5	5.4	12.1
ε_2 (%)	0.06	0.04	9.52	5.5	7.2	6.4	11.2
							7.8

En la Tabla 4 se observa el porcentaje de memoria ROM de programa ocupada por el clasificador en un microcontrolador de la familia Microchip® 18F2550. Los resultados de clasificación del microcontrolador son iguales a los obtenidos en la simulación hecha en Matlab® relacionados en la Tabla 3, esto se debe a que se siguió el mismo procedimiento matemático sobre variables enteras.

TABLA 4. OCUPACIÓN DE MEMORÍA ROM

Conjunto 1		Conjunto 2		Conjunto 3		Conjunto 4	
ROM (%)	55	40	32	60			

CONCLUSIONES

1. La clasificación de patrones con funciones discriminantes no lineales o no conexas, se puede realizar sin utilizar métodos complejos y costosos computacionalmente, ya que se pueden lograr un mapeo a la mínima dimensión superior donde las clases presentan separabilidad lineal. Así, el método propuesto expone ventajas en cuanto a la estimación del número de centroides que adapta el tamaño de la red al tipo de separación que necesita ser implementada, además de encontrar la posición de los centroides que separa de mejor manera las clases.
2. Dado que la arquitectura de la red es dinámica es posible lograr el buen desempeño del clasificador sin comprometer el costo computacional asociado a las rutinas del procedimiento.
3. El clasificador puede ser implementado en aplicaciones reales, sobre plataformas de baja exigencia de procesamiento.

REFERENCIAS

- Chen, S., Chng, E. S., & Alkadhimi, K. (1996). Regularized orthogonal least squares algorithm for constructing radial basis function networks. *International Journal in Control* , 829–837.
- Duda, R. O., Hart, P. E., & Stork, D. G. (2000). *Pattern Classification*. New York: Wiley.
- Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation* . New York: Prentice-Hall.
- Hernandez L., J., & Salazar G., S. (2006). Implementación de una Maquina de Vectores de Soporte empleando FPGA. *Scientia et Technica Año XII, No 31* , 47-52.
- Hush, D. R., & Horne, B. (1998). Efficient algorithms for function approximation with piecewise linear sigmoidal networks. *IEEE Transaction on Neural Networks* , 1129–1141.
- Kondo, N., Hatanaka, T., & Uosaki, K. (2006). Pattern Classification via Multi-objective Evolutionary RBF Networks. *SICE-ICASE International Joint Conference* , 137-142.
- Schilling, R. J., Carroll, J. J., & Al-Ajlouni, A. F. (2001). Approximation of nonlinear systems with radial basis function neural network. *IEEE Transactions on Neural Networks* , 1-15.
- Schwenker, F., Kestler, H. A., & Palm, G. (2000). Radial-Basis-Function Networks: Learning and Applications. *Fourth International Conference on knowledge-Based Intelligent Engineering Systems & Allied Technologies* , 33-43.
- Tinós, R., & Murta Júnior, L. O. (2008). Selection of Radial Basis Functions via Genetic Algorithms in Pattern. *10th Brazilian Symposium on Neural Networks* , 171-176.
- Witten, I. H., & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. San Francisco: Morgan Kaufman.
- Zhang, W., He, M., & Mak, M. W. (2001). Application of MLP and RBF Networks to cloud detection. *Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing* , 60-63.