

# Comparativa de rendimiento y resultado en el reconocimiento óptico de números escritos a mano usando funciones de base radial y sistema memético diferencial

Comparison of performance and results in optical recognition hand written numbers using radial basis functions and memetic differential system

*Montes Castañeda Bryan\**, *Bello Santos Omar David\*\**, *Piragauta Gómez Oscar Manuel\*\*\**,  
*Orjuela-Cañón Alvaro David\*\*\*\**

## RESUMEN

El problema de reconocimiento óptico de números escritos a mano ha sido abordado mediante diferentes técnicas obteniendo buenos resultados. En el presente trabajo se proponen sistemas difusos con algoritmos genéticos más específicamente meméticos para realizar esta tarea. Los resultados obtenidos con este método son comparados con redes neuronales de aprendizaje semi-supervisado, usando funciones de base radial (RBF). Al realizar la comparación, es posible observar que este tipo de redes neuronales ofrecen ventajas en cuanto a tasas de error y tiempo de obtención del sistema de reconocimiento frente a los métodos basados en sistemas difusos.

**Palabras clave:** Evolución Diferencial, Modelado difuso, Algoritmo memético (MA), reconocimiento óptico de números, Redes neuronales de base radial (RBF).

Fecha Recibido: Enero 21 de 2014

Fecha Aceptado: Abril 11 de 2014

\* Ingeniero Electrónico de la Universidad Distrital Francisco José de Caldas. Correo: [bryanmoncas@hotmail.com](mailto:bryanmoncas@hotmail.com)

\*\* Ingeniero Electrónico de la Universidad Distrital Francisco José de Caldas. Correo: [odbellos@correo.udistrital.edu.co](mailto:odbellos@correo.udistrital.edu.co)

\*\*\* Ingeniero Electrónico de la Universidad Distrital Francisco José de Caldas. Correo: [ompiragautag@correo.udistrital.edu.co](mailto:ompiragautag@correo.udistrital.edu.co)

\*\*\*\* Docente de la Universidad Distrital Francisco José de Caldas; Magíster en ciencias en ingeniería eléctrica, Área: Electrónica, Sub-área: Procesamiento Digital de Señales - COPPE - Universidad Federal de Rio de Janeiro. Correo: [dorjuela@ieee.org](mailto:dorjuela@ieee.org)



## ABSTRACT

The problem optical recognition of handwritten numbers has been approached by different methods, obtaining satisfactory results. In this paper, we propose fuzzy systems with memetic genetic algorithms. Results from this methodology are compared with artificial neuronal networks trained using semi-supervised learning and radial base functions (RBF). It is possible to observe that this kind of neuronal networks offer advantages regarding error rates and time-to-results of the recognition system, compared with methods based in fuzzy systems.

**Keywords:** Differential evolution modeling diffuse memetic algorithm (MA), OCR numbers, radial basis neural networks

## 1. INTRODUCCIÓN

El reconocimiento óptico de caracteres u OCR (por su acrónimo en inglés, Optical Character Recognition) es un proceso que tiene como objetivo lograr digitalizar y reconocer los caracteres de un determinado alfabeto o simbología contenidos en diferentes fuentes de texto. Actualmente es una técnica muy utilizada y estudiada para mejorar la calidad de muchos servicios como el reconocimiento de textos manuscritos para su almacenamiento en forma de datos [1].

Los sistemas OCR son usados para el reconocimiento de caracteres de documentos impresos por un procesador de palabras o en textos escritos a mano. En este último caso, se tiene un nivel de dificultad en el reconocimiento debido a la amplia diversidad que existe en la caligrafía de los individuos. Ejemplo de esto se puede observar en variaciones del tamaño y similitudes entre los símbolos escritos [2].

El algoritmo OCR está basado en una serie de etapas que se deben seguir en orden para lograr diferenciar el texto que se quiere reconocer. la primera etapa tiene que ver con la binarización de la información, donde se digitaliza y umbrales los caracteres, de forma tal que se pueden almacenar en forma de representación binaria (blanco y negro).

La etapa posterior al proceso de binarización tiene que ver con la fragmentación o segmentación, donde la imagen es dividida en regiones, donde cada región representa un carácter. De esta forma es posible realizar un análisis individual de los

símbolos que se quieren reconocer. Este proceso es uno de los más complicados dentro del algoritmo OCR, debido a que enfrenta problemas en la obtención de cada uno de los símbolos o caracteres tratados [2].

Cuando se tiene la información almacenada es necesario aplicar técnicas de reducción de dimensionalidad debido a que generalmente hay información redundante [3]. Por esta razón, una etapa de reducción de componentes es necesaria para realizar esta tarea.

Finalmente, existe una etapa encargada de realizar la comparación e identificación de los caracteres. Algoritmos de reconocimiento de patrones son usados aquí para poder interpretar el símbolo o carácter de entrada en el sistema [2]. De esta forma los caracteres son comparados con patrones previamente clasificados y de esta manera agrupados en un determinado grupo que representa un símbolo..

Los sistemas difusos son herramientas computacionales que pueden ser usadas en el campo de reconocimiento de patrones [4]. Aplicaciones de esta herramienta pueden encontrarse en [5] [6] [7], donde la clasificación de símbolos es realizada obteniendo buenos resultados con este tipo de técnicas.

Las redes neuronales artificiales (RNA) son otro tipo de herramientas extraídas de la inteligencia computacional, que también son usadas en tareas de reconocimiento de patrones, clasificación y regresión [5]. Existe un tipo de redes neuronales basadas en funciones de base radial (RBF), que no tienen requerimientos de hardware y soft-

ware tan exigentes como otros tipos de redes. Adicionalmente, el tiempo de entrenamiento es corto, después de determinar la arquitectura óptima para la tarea a desempeñar [8]. Este tipo de redes ha sido utilizada en el problema de reconocimiento de caracteres, obteniendo resultados satisfactorios [4].

En el presente trabajo es analizada la última etapa de un sistema OCR (comparación o identificación), utilizando técnicas de inteligencia computacional como lo son los sistemas difusos y las redes neuronales artificiales. Comparaciones en términos del error producido por cada técnica serán presentadas hacia el final del documento.

## 2. METODOLOGÍA

En esta sección es presentada la base de datos utilizada para implementar el presente estudio. Los métodos de lógica difusa y de redes neuronales son también explicados en detalle sobre su implementación.

### 2.1 Base de datos

La base de datos utilizada para la identificación óptica de números hechos a mano se encuentra en el repositorio de la UCI [3], en ella ya se encuentra hecha la etapa de binarización, fragmentación y reducción de información. Ésta base de datos fue realizada por 43 personas, donde cada una de ellas escribió más de un dígito.

Los números escritos se segmentaron en mapas de 32x32 píxeles. Cada carácter se dividió en ma-

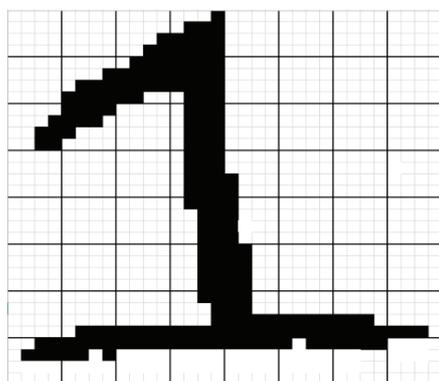


Figura 1. división de la imagen en bloques de 4x4 píxeles

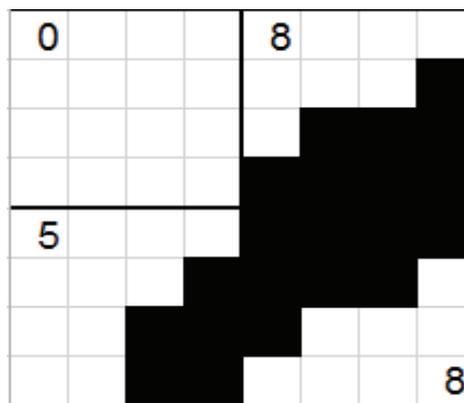


Figura 2. Representación de los números

trices de 4x4 píxeles, como se muestra en la figura 1, esto quiere decir que cada carácter se representa mediante una matriz de celdas o píxeles de 8x8 para un total de 64 datos, de los cuales cada uno tiene la información de un bloque de 4x4.

De cada celda de las 64 que contiene un carácter, se almacena la información de cuantos píxeles en los bloques de 4x4 tienen un nivel de brillo igual a cero (un nivel de brillo igual a cero equivale a una tonalidad oscura), lo que quiere decir que el máximo valor para cada dato será 16 (figura 2).

La cantidad de datos correspondientes a cada número se muestra en la Tabla 1, en donde se tiene un total de 3823 datos de números del 0 al 9.

Al obtener la base de datos se realizó una normalización dividiendo cada valor de los datos entre el máximo valor, en este caso 16, lo que conlleva a una base de datos más manejable. Otra ventaja en este tipo de normalización es que en sistemas

Tabla 1. Cantidad de datos correspondientes a cada número

Digito	#Datos
0	376
1	389
2	380
3	389
4	387
5	376
6	377
7	387
8	380
9	382

como las redes neuronales se evita trabajar en regiones de saturación después de aplicar la función de cada neurona [5].

De los 3823 datos se tomó el 70% de la base de datos para el entrenamiento del sistema de reconocimiento (2676 datos) y el 30% restante para validación del mismo (1147 datos). De esta forma, es posible medir la generalización del método evitando caer en problemas de sobreentrenamiento [5].

## 2.2 Sistema Difuso Genético a partir de Algoritmo evolutivo de Tercera Generación Memético (MA)

Para el reconocimiento de los números escritos a mano se consideró el uso de una primera metodología basada en la clasificación con un sistema difuso sintonizado con algoritmos genéticos entrenado con aprendizaje supervisado. A continuación se describe las partes que componen el reconocedor a partir de esta metodología.

### 1) Clasificador Difuso

La clasificación de los caracteres está basada en un sistema de inferencia difuso en donde un conjunto de reglas lingüísticas son transformadas en un mapeo no lineal, y de esta forma lograr un aproximador universal [6]. La configuración utilizada en el presente trabajo se conoce como de tipo Mamdani, debido a que presenta en su estructura un fuzzificador, un mecanismo de inferencia difusa con basado en reglas difusas y un defuzzificador, como se muestra en la FIGURA 3. La expresión matemática de un sistema de inferencia difusa tipo Mamdani se muestra a continuación (1), usada para calcular la salida del sistema, donde sus conjuntos difusos son de tipo gaussiano para  $N$  entradas y  $M$  reglas [7].

$$f(x) = \frac{\sum_{l=1}^M \gamma_l \left( \prod_{i=1}^N a_i^l \exp\left(-\frac{(x_i - x_i^l)^2}{(\sigma_i^l)^2}\right) \right)}{\sum_{l=1}^M \prod_{i=1}^N a_i^l \exp\left(-\frac{(x_i - x_i^l)^2}{(\sigma_i^l)^2}\right)}$$

$$\sup_{x \in U} |f(x) - g(x)| < \varepsilon \quad (1)$$

La complejidad de los sistemas difusos depende del número de entradas y de reglas difusas planteadas, de acuerdo a que la cantidad de conjuntos difusos en el antecedente y los centros en el consecuente dependen de estos dos factores. Para el presente estudio se tienen 64 entradas y un total de 10 reglas que están dadas por el total de números a reconocer [7].

### 2) Sintonización del sistema difuso a partir del Algoritmo evolutivo de tercera generación Memético (MA)

Como está expuesto en [7], un algoritmo memético es un algoritmo híbrido que incorpora el conocimiento o aprendizaje a un determinado algoritmo evolutivo en cada una de las poblaciones existentes en cada generación. Para implementar este algoritmo se realizó una hibridación entre el algoritmo evolutivo diferencial [8] y un aprendizaje supervisado típico [9], utilizando el algoritmo de retro-propagación (*backpropagation*). En otras palabras, se realizó una sintonización en la inferencia difusa basándonos en un algoritmo de evolución diferencial con búsqueda local, conocido como algoritmo evolutivo memético diferencial (MDE).

La estructura general trabajada para la sintonización de los parámetros del algoritmo es la que se muestra en la figura 4 [8]. Los valores que se encuentran en cada cromosoma de los individuos de la población son los 64 centros en cada atributo, y sus respectivas desviaciones estándar. De esta forma cada individuo está representado por 128 genes.

Para generar la familia de individuos de la primera generación, se tomaron 10 individuos

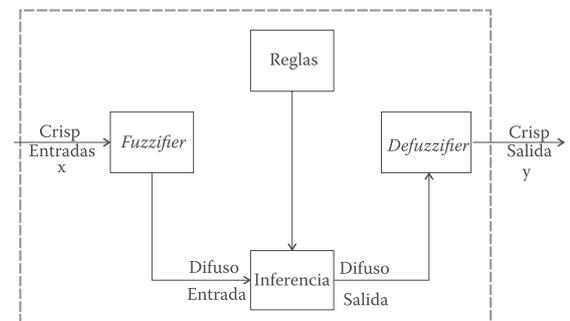
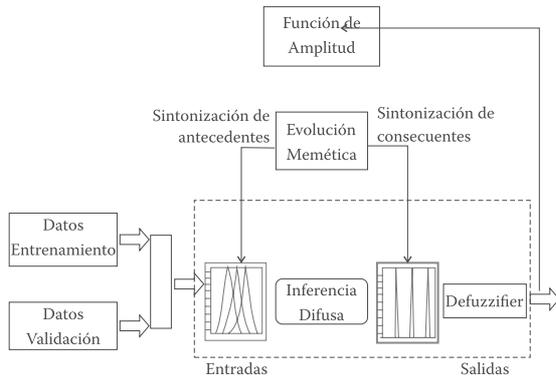


Figura 3. Estructura de un sistema difuso tipo Mamdani



**Figura 4.** Sintonizaciones de los parámetros de un sistema difuso a partir de un algoritmo evolutivo memético diferencial (MDE)

principales (uno por cada número). Con estos 10 individuos se generó aleatoriamente el resto de la población. En general se obtuvo una población de 30 individuos por cada número.

La probabilidad de cruce para todos los parámetros de los principales individuos fue de 0.9. De igual forma, este valor fue utilizado para el cruce del resto de la población, dependiendo de la distancia de cada uno de los individuos con los escogidos como principales en la primera generación (2).

$$PRO = 0.9 * e^{-abs(indi1-indix)} \quad (2)$$

Donde PRO es la probabilidad de un individuo  $x$ ,  $indi1$  es el individuo principal e  $indix$  es el individuo  $x$  en cada generación.

Para generar la mutación en el algoritmo se utilizó la variante del DE (diferencial evolutivo), en donde se tomó como mejor individuo el descrito como individuo principal y el factor de mutación  $F$  se tomó con el valor de 0.8. En general, los valores de probabilidad de cruce, factor de mutación y número de individuos fueron hallados empíricamente, según [7].

### 2.3 Sistema a partir de red neuronal artificial con funciones de base radial (RBF)

La segunda técnica usada en el presente trabajo está basada en una red neuronal artificial con funciones de base radial entrenada con aprendizaje semi-supervisado.

## 1) Redes de Función de Base Radial (RBF)

Las redes RBF están compuestas por tres capas de neuronas, donde la capa intermedia tiene la particularidad de estar conformada por neuronas que utilizan funciones de base radial [5]. La capa de entrada es usada para representar cada una de las variables usadas en el problema de reconocimiento. En el presente trabajo son usadas 64 entradas de acuerdo a lo expuesto anteriormente. La capa de salida está determinada por el número de caracteres a reconocer, como el objetivo es reconocer los caracteres del cero al nueve, la red tiene 10 neuronas en esta capa.

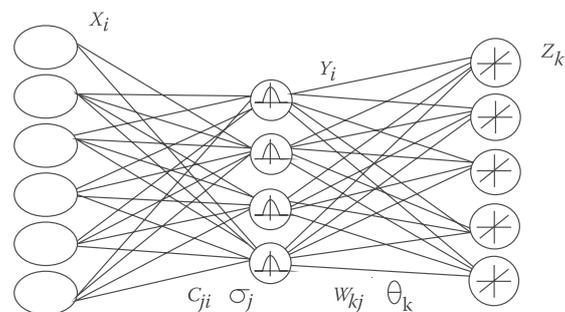
El aprendizaje de este tipo de redes, puede estar determinado por un proceso no supervisado, donde se busca encontrar el número óptimo de funciones de base radial que solucionen el problema. Una etapa supervisada busca encontrar la relación entre la capa intermedia y la capa de salida, ajustando los pesos sinápticos que unen estas dos capas (figura 5).

El paso fundamental del entrenamiento es realizado por las neuronas de la capa oculta, donde se calcula la distancia euclidiana entre el vector de pesos sinápticos (centroides) y la entrada, y sobre esta distancia se aplica una función de tipo radial con forma gaussiana (4).

$$F(x) = \sum_{i=1}^N \omega_i \Phi(\|x - c_i\|) \quad (3)$$

$$\Phi(x) = e^{-(\epsilon r)^2} \quad (4)$$

En la ecuación (3) podemos ver la función general de una red RBF, donde se representa una



**Figura 5.** Arquitectura Típica De Una Red RBF.

relación no lineal entre el vector de entrada y la función de base radial, la cual no es más que el cálculo de la distancia euclidiana de cada neurona y el centro de la función. Cada neurona de la capa intermedia es representada por una función de base radial ( $x$ ) y sus pesos de salida  $\omega_j$ .

## 2) Aprendizaje Semi-supervisado de la Red RBF

El aprendizaje consiste en la determinación de los centros, desviaciones y pesos de las neuronas de la capa oculta a la capa de salida. Como las capas de la red realizan diferentes tareas, se separan los parámetros de la capa oculta de los de la capa de salida para optimizar el proceso. De esta forma, los centros y las desviaciones siguen un proceso guiado por una optimización en el espacio de entrada, mientras que los pesos siguen una optimización basado en las salidas que se desean obtener.

Para el aprendizaje de la capa oculta se usa un aprendizaje no supervisado donde se obtienen el número de funciones, sus centros y su respectiva dispersión o radio. Existen diferentes métodos conocidos, uno de ellos es el algoritmo *k-means* descrito en [10], el cual es un algoritmo no supervisado de agrupamiento de datos, donde  $k$  es el número de grupos que se desea encontrar y corresponde al número de neuronas en la capa oculta.

Una vez fijados los valores de los centros en el paso anterior, se ajusta el ancho de cada neurona. Los anchos son los parámetros de dispersión que aparecen en cada una de las funciones gaussianas. De esta forma es obtenida una medida de cómo la entrada activa una neurona oculta para que produzca una salida significativa. La base de la función gaussiana (dispersión) debe calcularse de manera que se presente el menor solapamiento posible entre sí.

Después de obtener el número de funciones de base radial con sus respectivas características es realizada una fase supervisada con un método como retro-propagación [11]. Aquí son calculados los pesos y umbrales de las neuronas de salida de la red. El objetivo es minimizar las diferencias entre las salidas de la red y las salidas deseadas.

## 3. RESULTADOS

Los resultados son presentados en el orden de acuerdo al modelo mostrado en la metodología. Después es realizada una comparación entre las técnicas y finalmente se discuten los resultados del presente trabajo con otros encontrados en la literatura.

### 1) Resultados del Sistema Memético Diferencial

Los resultados arrojados por este algoritmo se estudiaron después de realizar alrededor de 80 pruebas para obtener representación estadística. El error de clasificación fue observado en función del número de generaciones requeridas. Con esto, el algoritmo memético diferencial logró disminuir el error de clasificación hasta un valor del 6.3%. Además, fue disminuido considerablemente el número de generaciones para llegar a dicho valor en comparación a otros algoritmos evolutivos. En la tabla 2 se puede observar los resultados obtenidos por el algoritmo memético, en donde el número de generaciones fue modificado para observar si los resultados eran alterados por este parámetro.

Como es posible observar en la Tabla 2, los resultados obtenidos reportan bajas tasas de error. El porcentaje de error mínimo que se logró obtener

**Tabla 2.** Resultados arrojados por el sistema difusos sintonizado con un algoritmo genético memético.

Número de generaciones	% Error	Desviación estándar
10	10,3	3,12
50	8,94	3,1
100	8,28	3,12
500	7,9	2,92
1000	7,51	2,94
1500	7,2	2,86
2000	7	2,61
2500	6,3	2,34
3000	6,3	2,23

fue de 6,3%, necesitando 2500 generaciones. De la misma forma, la desviación estándar muestra que los resultados tienen una coherencia alta debido a su baja dispersión.

## 2) Resultados obtenidos de la red RBF

Para la implementación de la red neuronal de funciones de base radial, fue modificado el número de neuronas en la capa oculta y también la dispersión para la función gaussiana (4) utilizada. Los resultados obtenidos son mostrados en la tabla 3.

A partir de los resultados de la tabla 3, se hicieron pruebas con los mejores resultados obtenidos para cada varianza y, así, poder sacar una media de error y desviación estándar a partir de varias pruebas con la misma configuración de número de neuronas y varianza, obteniendo los resultados de la tabla 4.

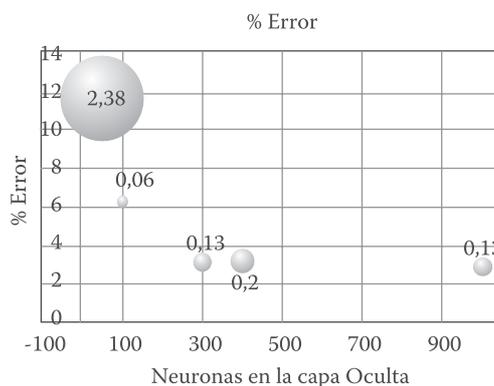
En la figura 6 podemos observar los resultados de forma gráfica. Es posible ver el número de neuronas en la capa oculta, el error y la desviación estándar. En los resultados con mayor dispersión se puede ver un círculo de mayor radio.

**Tabla 3.** Resultados obtenidos con la red rbf, variando el número de neuronas y la varianza.

Varianza Gaussiana	Neuronas			
	50	100	300	400
5	10,28%	5,58%	3,22%	2,87%
10	11,68%	6,62%	3,13%	3,22%
25	12,81%	6,36%	3,40%	3,40%
100	16,39%	6,53%	3,22%	3,40%

**Tabla 4.** Resultados obtenidos con la red RBF.

Neuronas capa oculta	Varianza gaussiana	% Error	Desviación estándar
50	5	11,682	2,38
100	25	6,277	0,06
300	10	3,138	0,13
400	10	3,225	0,2



**Figura 6.** Resultados obtenidos con la red RBF.

Los mejores resultados que relacionan la dimensión de la red y el porcentaje de error fueron obtenidos para una red RBF con 300 neuronas en la capa oculta. El porcentaje de error para este caso fue de 3,138%, con una desviación estándar baja. Esto indica que la red neuronal tiene resultados consistentes, haciendo el método estable frente al problema de reconocimiento.

Luego de los resultados anteriormente citados se decidió calcular el porcentaje de error para cada número en particular, usando el subconjunto de validación. La TABLA 5 muestra la tasa de clasificación para cada número, el porcentaje de valores no clasificados y el porcentaje de los errores cometidos por la red.

De los resultados expuestos (TABLA 6), podemos observar que los números que mejor clasifica la red RBF son el cero y el cuatro donde el porcentaje de error es ligeramente superior al 0,5%, donde la red no pudo clasificar correctamente a todos los datos presentados. También podemos observar que el número que tiene un mayor error de clasificación es el ocho, con un error total de 4,06%, este carácter es confundido con otros caracteres debido a su forma, lo que hace difícil su reconocimiento.

En general, es posible notar que los resultados en cuanto al error para todo el sistema es de 1,51% (suma de los porcentajes de números sin clasificar y el error), partiendo del hecho de que un 1,17% son de números que la red no clasificó erradamente sino que consideró que no podía clasificarlos.

- Comparativa de rendimiento y resultado en el reconocimiento óptico de números escritos a mano •

**Tabla 5.** resultados obtenidos con la red RBF para cada número usando toda la base de datos.

Número	Acierto (%)	Sin clasificar (%)	Error (%)
Cero	99.4	0.53	0
Uno	98.7	1.30	0
Dos	99.2	0.78	0
Tres	97.4	1.79	0.77
Cuatro	99.4	0	0.52
Cinco	98.9	0.53	0.53
Seis	99.2	0.26	0.79
Siete	98.9	0.77	0.25
Ocho	96.05	3.28	0.78
Nueve	97.38	2.35	0.26
Total	98.48	1.17	0.34

### 3) Comparación de los resultados para las metodologías propuestas

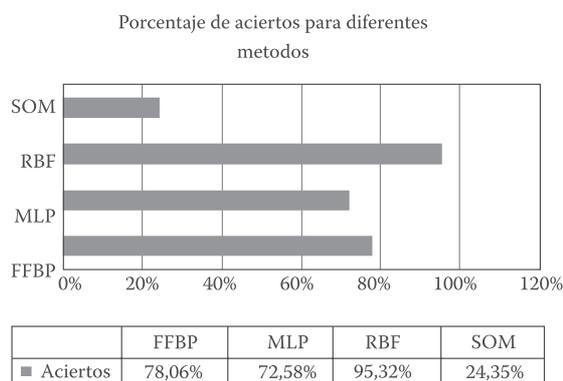
Los resultados expuestos anteriormente muestran que para el modelo de reconocimiento basado en la red neuronal el porcentaje de error es más bajo, como se muestra en la Tabla 6, y que el costo computacional demandado por la red RBF es menor que el obtenido a través del sistema memético diferencial. Mientras la red RBF toma unos tres minutos en entrenar la red neuronal y proporcionar resultados, el algoritmo memético puede tomar horas en dar un resultado satisfactorio. Es claro que el sistema difuso igualmente resolvió el problema disminuyendo el porcentaje de error considerablemente y lo hizo mucho mejor que con otros algoritmos probados.

### 4) Discusión de los resultados

Analizando los resultados obtenidos en [12], podemos concluir que entre los diferentes tipos

**Tabla 6.** comparación de resultados

Algoritmo	%Error	Costo Computacional
Memético Diferencial	6,3	Más de 60 Minutos
Red rbf	3,138	2 A 4 minutos



**Figura 7.** tasa de aciertos para diferentes tipos de redes

de redes neuronales los mejores resultados se obtuvieron con RBF y los peores con SOM, lo cual hizo más coherente desarrollar nuestra red neuronal bajo este modelo.

Adicionalmente, bajo la misma base de datos y en la misma máquina, el tiempo de entrenamiento y validación del RBF es de una cuarta parte del MLP y la mitad del SOM. Esto nos hace afirmar aún más la idea de trabajar con redes de tipo RBF.

Cabe notar que la base de datos usada en [12] es diferente a la usada en el presente trabajo, pero tiene características interesantes como estilos muy poco elegantes en cuanto a caligrafía, lo que hace particularmente difícil clasificar el dígito en alguna categoría. Encontraron especial dificultad los dígitos cero (0), tres (3), cuatro (4) y siete (7). [4].

## 4. CONCLUSIONES

Los métodos propuestos y analizados en este artículo se obtuvieron resultados satisfactorios. Fueron obtenidos valores cercanos al 99% de clasificación exitosa de toda la base de datos, y un promedio de 99.1% en el subconjunto de validación como es el caso de la red RBF. El sistema memético diferencial obtuvo resultados cercanos al 95% en la tasa de clasificación usando toda la base de datos y 93% en el subconjunto de validación, mostrando generalización en el problema.

Como conclusión final es bueno resaltar que para este caso en particular (el reconocimiento óptico de números escritos a mano) es mejor usar una red RBF, que un sistema memético diferencial, con los resultados expuestos se pudo entrever que el porcentaje de clasificación de la red RBF es ligeramente más efectivo que el sistema difuso genético, agregando como parte fundamental los costos computacionales que son requeridos para el desarrollo e implementación de cada uno de ellos.

## 5. REFERENCIAS

- [1] J. A. Navarro, "OCR Sistemas de Reconocimiento óptico de caracteres" *Revista del instituto tecnológico de informática*, No 13, pp 9-11, Nov 2007.
- [2] C. Sánchez, V. Sandonís. *Reconocimiento Óptico de Caracteres (OCR)*. Universidad Carlos III, Madrid España. Available: <http://www.it.uc3m.es/jvillena/irc/practicas/08-09/09.pdf>. [Jan 20 2014]
- [3] K. Bache, M. Lichman, UCI Machine Learning Repository. University of California, School of Information and Computer Science, 2013. Available: <http://archive.ics.uci.edu/ml> [Jan 20 214]
- [4] H. Young-Sup, B. Sung-Yang. "Recognition of unconstrained handwritten numerals by a radial basis function neural network classifier", *Pattern Recognition Letters*, Vol 18, No 7, pp 657-664, 1997
- [5] S. Haykin. *Neural Networks: A Comprehensive Foundation*, New York: McMillan College Publishing Company, 1994.
- [6] L.X, Wang. *A course in fuzzy systems and control*, Estados Unidos: Prentice-Hall, 1997.
- [7] M. Melgarejo. *Sintonización de sistemas difusos mediante un algoritmo memético adaptativo* [diapositivas]. Universidad Distrital Francisco José de Caldas, Bogotá Colombia, 2013.
- [8] M. Melgarejo. *Sintonización de sistemas difusos utilizando evolución diferencial* [diapositivas]. Universidad Distrital Francisco José de Caldas, Bogotá Colombia 2013.
- [9] Wang, L.-X.; Mendel, J.M., "Back-propagation fuzzy system as nonlinear dynamic system identifiers," *Fuzzy Systems*, 1992., IEEE International Conference on , vol., no., pp.1409,1418, 8-12 Mar 1992
- [10] L. Hongmin, Y. Zhixi, "Applying Multiple Agents to Fuzzy Collaborative Filtering," *E-business and Information System Security*, 2009. EBISS '09. International Conference on pp.1,5, 23-24 May 2009
- [11] H. Boughrara, M. Chtourou, C.B. Amar, "MLP neural network based face recognition system using constructive training algorithm," *Multimedia Computing and Systems (ICMCS)*, 2012 International Conference on, pp.233,238, 10-12 May 2012
- [12] H. A. Al Hamad, "Use an efficient neural network to improve the Arabic handwriting recognition," *Signal and Image Processing Applications (ICSIPA)*, 2013 IEEE International Conference on, pp.269,274, 8-10 Oct. 2013