

Reconocimiento de Dígitos Manuscritos Usando La Transformada Wavelet Continua en 2 Dimensiones y Redes Neuronales

Diego Javier Romero, Leticia Seijas, Ana Ruedin

Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires
Planta Baja, Pabellón I, Ciudad Universitaria, (1428)Buenos Aires, Argentina

e-mail: dromero@dc.uba.ar, lseijas@dc.uba.ar, anita@dc.uba.ar

Abstract

In this paper we present a preprocessing method for handwritten numerals recognition, based on a two dimensional continuous wavelet transform. We use the wavelet transformed digits to input into a multilayer feed forward neural network, trained with backpropagation algorithm. Our preliminary results are promising.

Keywords: Neural Networks, Continuous Wavelet Transform

Resumen

En este trabajo presentamos un método de procesamiento para el reconocimiento de dígitos manuscritos, basado en la aplicación de la transformada wavelet continua en dos dimensiones. Los datos preprocesados son utilizados como entrada de una red neuronal del tipo feed forward multicapa, la cual es entrenada con el algoritmo de backpropagation. Nuestros resultados preliminares son alentadores.

Palabras claves: Redes Neuronales, Transformada Wavelet Continua

1. INTRODUCCIÓN

El reconocimiento de dígitos manuscritos es un problema que fue tratado ampliamente, con técnicas muy diversas. Tiene varias aplicaciones de interés, siendo una de ellas la lectura automática de códigos postales. Entre las metodologías usadas, se encuentran aquellas basadas en redes neuronales.

El desempeño de una red neuronal depende fuertemente de cómo se representa cada dígito, a través de características extraídas en la etapa de preprocesamiento. Se han utilizado diferentes tipos de preprocesamiento de los datos, antes de ingresarlos a una red neuronal. En [9] y [10] se utilizó el análisis de componentes principales, con el objetivo de reducir la dimensionalidad de los datos. En [11] y [3] se utilizaron máscaras de Kirsch, para extraer características direccionales.

La transformada wavelet es una herramienta idónea para reconocimiento de bordes y texturas en imágenes. Sus propiedades la convierte en una herramienta apta para el preprocesamiento de los datos. En [2] se aplica una transformada wavelet diádica ortogonal de una dimensión, sobre los contornos representados con 2 vectores x e y , extraídos previamente de los dígitos. En [1] se utiliza una transformada multiwavelet, de una dimensión, que también se aplica sobre los contornos de los dígitos.

Si a la transformada wavelet continua para datos bidimensionales, se le agrega una rotación con ángulo determinado, se convierte en una transformada anisotrópica que depende de 4 parámetros. Entonces se tienen 2 representaciones posibles de una imagen transformada: la representación posicional, y la representación en escala-ángulo. En [7] y [8] se utilizan ambas representaciones (en distintos pasos) para el reconocimiento de patrones en una imagen. En este trabajo utilizaremos una transformada wavelet continua con la representación posicional, para valores de escala y de ángulo fijos, como método de preprocesamiento sobre imágenes de dígitos manuscritos.

Las imágenes de dígitos fueron obtenidas de la base de datos de números manuscritos de la universidad de Concordia (Canadá). Esta base de datos está compuesta por 6000 dígitos escritos a mano, provistos por el Servicio Postal de los Estados Unidos, y extraídos de códigos postales manuscritos en los sobres de la correspondencia. De la totalidad de los patrones, 4000 fueron utilizados para el entrenamiento (400 patrones por dígito) los cuales forman el llamado conjunto de entrenamiento, y 2000 para el testeo (200 patrones por dígito) que forman el conjunto de testeo. En las figuras 1 y 2 se muestran algunos dígitos elegidos al azar, pertenecientes al conjunto de entrenamiento y de testeo, en ellas se puede observar la característica de “escritura irrestricta” de los mismos. Cabe mencionar que ambos conjuntos incluyen ejemplos que son ambiguos, inclasificables y aún mal clasificados. La base de datos presenta imágenes de patrones de diferentes tamaños, conformadas por ceros y unos. Durante la etapa de preprocesamiento éstos son normalizados a un tamaño de 16x16 píxeles, luego del proceso de normalización se aplica la transformada wavelet continua en dos dimensiones. Los datos preprocesados son la entrada de un sistema de reconocimiento implementado con una red neuronal del tipo feed forward multicapa la cual es entrenada con el algoritmo de backpropagation.

Este trabajo está organizado de la siguiente manera: En la sección 2 se explicarán las nociones básicas de la transformada wavelet continua en dos dimensiones, en la sección 3 se explicará la arquitectura del sistema de reconocimiento, en la sección 4 se mostrarán los resultados obtenidos y en la sección 5 se darán las conclusiones.

0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9

Figura 1: Muestras pertenecientes al conjunto de entrenamiento

0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9

Figura 2: Muestras pertenecientes al conjunto de testeo

2. TRANSFORMADA WAVELET CONTINUA EN 2 DIMENSIONES

La transformada wavelet ha dado excelentes resultados en distintas aplicaciones de procesamiento de imágenes. Su excelente localización espacial y su buena localización frecuencial hacen de ella una herramienta apta para el preprocesamiento de los dígitos. La transformada wavelet diádica discreta, más frecuentemente utilizada, incluye submuestreos y no es invariante a traslaciones. Nuestro objetivo es conseguir un preprocesamiento que arroje los mismos valores para un dígito o para una copia del mismo que ha sido trasladada en uno o varios píxeles. Es por eso que elegimos una transformada wavelet continua, que es invariante frente a traslaciones.

Sea s una función de dos variables a valores reales, de cuadrado integrable, es decir, $s \in L^2(\mathfrak{R}^2)$. Definimos la transformada wavelet de s ([7] y [8]) con respecto a una función wavelet $\psi : \mathfrak{R}^2 \rightarrow \mathfrak{R}$ de la siguiente forma:

$$S(b, a, \theta) = a^{-1} \int_{\mathfrak{R}^2} \psi(a^{-1} r_{-\theta}(b - x)) s(x) dx, \quad (1)$$

donde $b = (b_x, b_y) \in \mathfrak{R}^2$ es un vector de traslación, $a \in \mathfrak{R}$ es la escala ($a > 0$), θ es un ángulo y $r_\theta(x)$ es el operador de rotación con ángulo θ , que actúa sobre un vector $x = (x_1, x_2) \in \mathfrak{R}^2$ de la manera siguiente:

$$r_\theta(x) = (x_1 \cos \theta - x_2 \sin \theta, x_1 \sin \theta + x_2 \cos \theta), 0 \leq \theta \leq 2\pi. \quad (2)$$

En este marco teórico no hay multiresolución, a diferencia de la transformada wavelet discreta. Los requerimientos que se imponen a la función ψ , para asegurar la recuperación de la función a partir de su transformada, son:

- $\psi(x)$ debe ser de cuadrado integrable: $\psi \in L^2(\mathfrak{R}^2)$
- ψ debe ser admisible, es decir, la siguiente integral debe ser finita:

$$c_\psi = \int_{\mathfrak{R}^2} |\widehat{\psi}(\omega)|^2 \frac{d\omega}{|\omega|} < \infty, \quad (3)$$

donde $\widehat{\psi}(\omega)$ es la transformada de Fourier de $\psi(x)$. Esa condición implica que

$$\widehat{\psi}(0) = 0, \quad (4)$$

lo que es equivalente a la condición de *media cero*

$$\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \psi(x) dx = 0. \quad (5)$$

Como podemos ver, la función (1) tiene cuatro variables b_x, b_y, a, θ : esto origina un problema en la visualización del resultado y en el cómputo de la misma. Para obtener una herramienta manejable, se fijan algunas de las variables y se obtiene así una función de una o dos variables. De esto último surgen dos representaciones:

1. Representación posicional

$$S_{a\theta}(b_x, b_y) = S(b_x, b_y, a, \theta) \quad a \text{ y } \theta \text{ fijos} \quad (6)$$

Observar que es una función de dos variables, es decir, tanto a como θ están fijos por lo tanto el resultado es una función en b_x y b_y . Con esta representación se analiza a la señal con una wavelet rotada en un ángulo θ y escalada en un factor a . Dicha wavelet se va desplazando en los ejes x e y acorde al valor de las variables b_x y b_y . Dicho de otra forma cada valor de $S_{a\theta}(b_x, b_y)$ será el cálculo de (1) en b_x, b_y, a y θ .

2. Representación Escala - Ángulo

$$S_{b_x b_y}(a, \theta) = S(b_x, b_y, a, \theta) \quad b_x \text{ y } b_y \text{ fijos} \quad (7)$$

Observar que en este caso también tenemos una función de dos variables sólo que ahora están fijos b_x y b_y quedando la función en términos de a y θ . En contraste al caso anterior, en esta representación se analiza a la señal con una wavelet situada en una posición fija (b_x y b_y) pero la misma va sufriendo cambios en la rotación y en la escala acorde al valor de las variables θ y a respectivamente.

En nuestro caso las señales a analizar serán imágenes de dígitos manuscritos binarias de tamaño 16x16 pixeles. El método de preprocesamiento que hemos elegido es la aplicación de (6) a dichas imágenes utilizando una wavelet discretizada que además está escalada en un factor a y rotada en un ángulo θ . Observar que el resultado es otra imagen bidimensional con valores reales, es decir, no necesariamente ceros y unos. Como último paso del método de preprocesamiento se aplica un umbral a la imagen resultado para obtener valores binarios. Como hemos mencionado anteriormente para el cálculo de (6) es necesario saber calcular (1), y para esto se realiza el producto, componente a componente, entre $\psi(a^{-1} r_{-\theta}(b - x))$ y $s(x)$ y luego se suman para estimar la integral. Además, para poder implementar (6) es necesario elegir una función wavelet, una escala, un ángulo y valores discretos para b_x y b_y . En nuestro caso la función wavelet utilizada es la llamada Mexican Hat direccional, cuya fórmula es la siguiente:

$$\psi_{mh}(x, y) = (2 - (x^2 + \frac{y^2}{\epsilon}))e^{-\frac{(x^2 + \frac{y^2}{\epsilon})}{2}} \quad (8)$$

Para b_x se tomaron 16 valores en el intervalo $[-32, 32]$, lo mismo para b_y , esto nos da una imagen cuya dimensión es 16x16. La escala elegida fue de 0.8 y el ángulo de 135° (en el sentido antihorario) ya que esa es la orientación que se observó en la mayoría de los dígitos. En las figuras 3 y 5 se puede ver el dibujo de la wavelet utilizada, junto con sus curvas de nivel, para $a = 1, \theta = 0$ y $\epsilon = 1$. Para obtener una wavelet anisotrópica se da un valor distinto de 1 a ϵ . En las figuras 4 y 6 se muestra el dibujo, junto con sus curvas de nivel, de la wavelet rotada a 135° (en el sentido antihorario), escalada en un factor de 0.8, para $\epsilon = 5$. Por otro lado, en las figuras 7-14 se muestran ejemplos de la aplicación del método de preprocesamiento a varias imágenes de dígitos manuscritos.

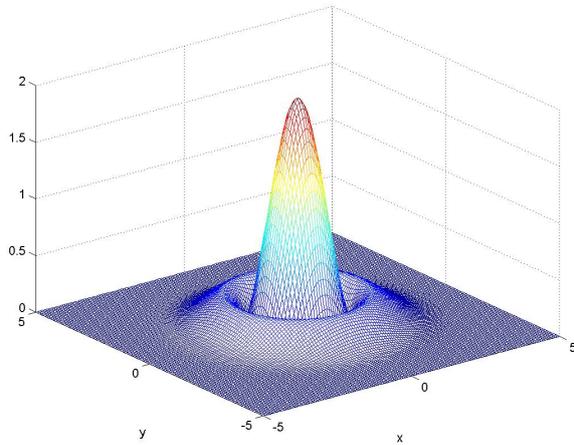


Figura 3: Mexican Hat con $a = 1$, $\theta = 0^\circ$ y $\epsilon = 1$. La misma esta evaluada en el intervalo $[-5, 5]$ con 128 muestras

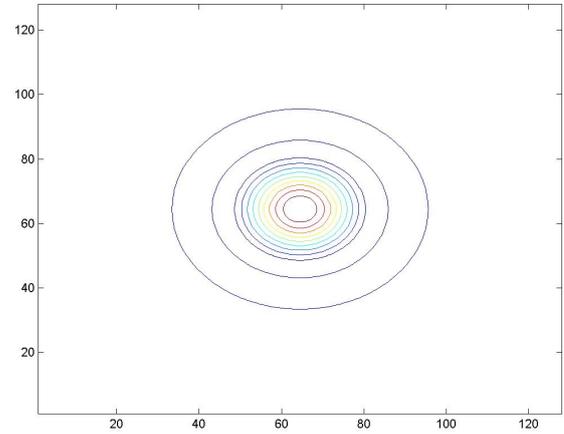


Figura 5: Curvas de nivel de Mexican Hat con $a = 1$, $\theta = 0^\circ$ y $\epsilon = 1$

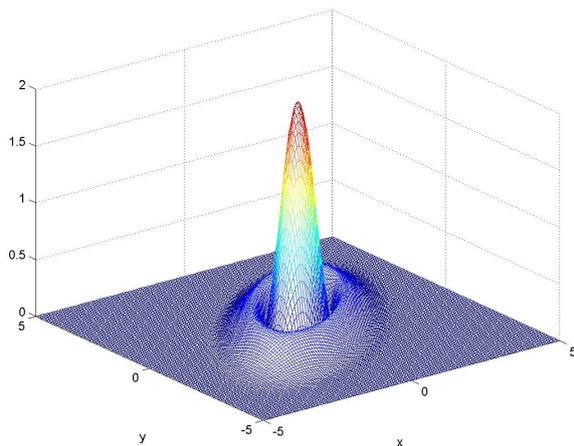


Figura 4: Mexican Hat con $a = 0.8$, $\theta = 135^\circ$ y $\epsilon = 5$. La misma está evaluada en el intervalo $[-5, 5]$ con 128 muestras

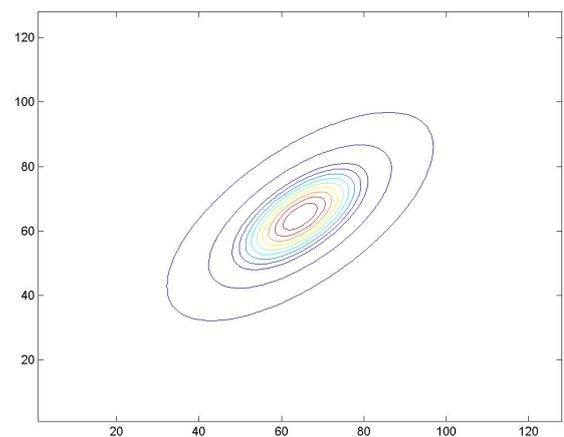


Figura 6: Curvas de nivel de Mexican Hat con $a = 0.8$, $\theta = 135^\circ$ y $\epsilon = 5$

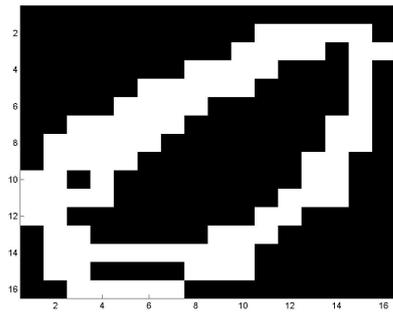


Figura 7: Un dígito 0 sin procesar

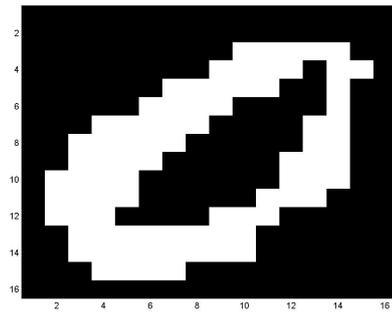


Figura 11: El dígito 0 preprocesado

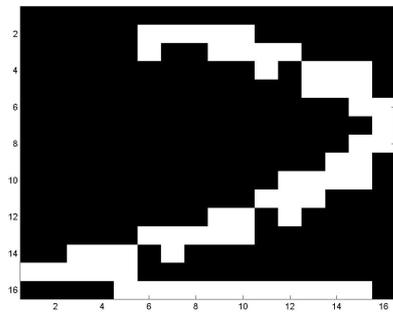


Figura 8: Un dígito 2 sin procesar

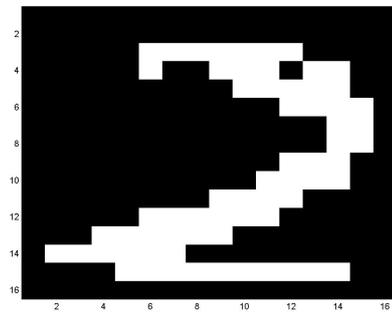


Figura 12: El dígito 2 preprocesado

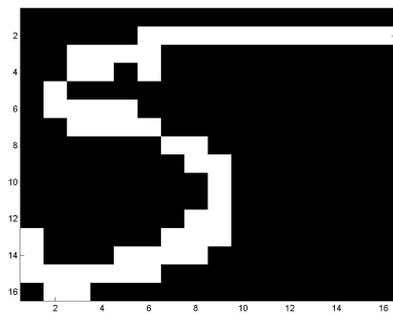


Figura 9: Un dígito 5 sin procesar

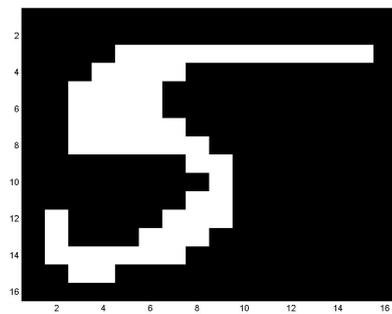


Figura 13: El dígito 5 preprocesado

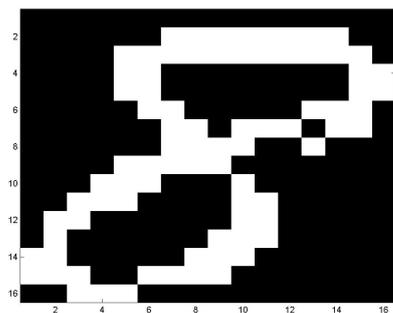


Figura 10: Un dígito 8 sin procesar

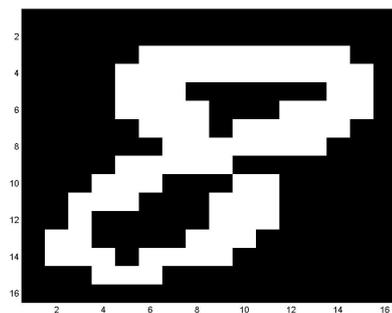


Figura 14: El dígito 8 preprocesado

3. SISTEMA DE RECONOCIMIENTO

El sistema de reconocimiento ha sido implementado utilizando una red feedforward multicapa. Un ejemplo de arquitectura de este tipo de redes puede observarse en la figura 15.

La red implementada presenta una capa de entrada de 256 neuronas, este número se corresponde con la dimensión de las imágenes preprocesadas (16x16). Luego presenta una capa oculta de 160 neuronas. Como no existe un método exacto para establecer la cantidad de neuronas en la capa oculta, la misma tiene que ser determinada en base a pruebas, en nuestro caso el mejor resultado fue obtenido con una cantidad de 160 para una capa oculta. Esta arquitectura fue seleccionada entre varias, a través de la utilización de un conjunto de validación en la etapa de entrenamiento. Se han realizado pruebas con redes de 1 y 2 capas ocultas, y distintas cantidades de unidades para cada capa. Por último está la capa de salida con 10 neuronas, una por cada dígito a reconocer. Cada unidad de la red está totalmente conectada con las unidades (neuronas) de la capa siguiente, a través de los pesos sinápticos w . Por cada patrón ingresado se calculan las salidas de cada unidad hasta obtener la salida real de la red. Definimos para una iteración o tiempo t :

- w_{ij} peso que conecta a la unidad i de la capa m con la unidad j de la capa $m - 1$
- $h_i = \sum_{j \in J} w_{ij} V_j$ entrada neta a la unidad i , J incluye a todas las neuronas de la capa anterior a la de la unidad i
- $V_i = g(h_i)$ salida de la unidad i , donde g es la función de activación asociada a dicha unidad. Si V_i está en la capa de entrada de la red, entonces toma los valores del patrón ingresado.
- ς_i salida esperada en la unidad i , forma parte del rótulo de cada patrón
- O_i salida real de la red para la unidad i
- $E(t) = \frac{1}{2} \sum_{i \in C} (\varsigma_i - O_i)^2$ donde C incluye a todas las neuronas de la capa de salida de la red, es el error en la iteración t

Definimos la función de costo de la siguiente manera

- $E(w) = \frac{1}{N} \sum_{\mu \in P, i \in C} (\varsigma_i^\mu - O_i^\mu)^2$ donde C incluye a todas las neuronas de la capa de salida de la red y P incluye a todos los patrones de entrenamiento.

La red fue entrenada con el algoritmo de backpropagation estocástico con momentum ([5] y [6]). El objetivo del algoritmo es obtener un conjunto de pesos, asociados al conexionado, que minimice la función costo. Por cada patrón de entrenamiento ingresado, se calcula la salida de la red y luego el error cometido por ésta. Este error se propaga hacia atrás, capa por capa, lo que permite calcular las derivadas parciales de la función de error con respecto a los pesos (estimación del vector gradiente). De esta manera, el ajuste de los pesos para el patrón ingresado en la iteración t se calcula como,

- $w_{ij}(t + 1) = w_{ij}(t) + \Delta w_{ij}(t)$
- $\Delta w_{ij}(t) = -\eta \frac{\partial E(t)}{\partial w_{ij}(t)} + \alpha \Delta w_{ij}(t - 1)$

Donde α es el parámetro de momentum o inercia (el mismo hace que el cambio realizado en los pesos sea en la dirección de descenso promedio, acelerando la convergencia) y η es el coeficiente de velocidad de aprendizaje, en nuestro caso esos valores fueron de 0.9 y 0.01 respectivamente. El aprendizaje de la red se desarrolla durante una cierta cantidad de épocas (en cada época se ingresa

todo el conjunto de entrenamiento), que en nuestro caso fue de 3500, hasta lograr minimizar la función de costo. En la capa oculta y en la de salida la función de activación utilizada fue la llamada Logística Sigmoidea definida por

$$g(h) = \frac{1}{1 + e^{-h}} \quad (9)$$

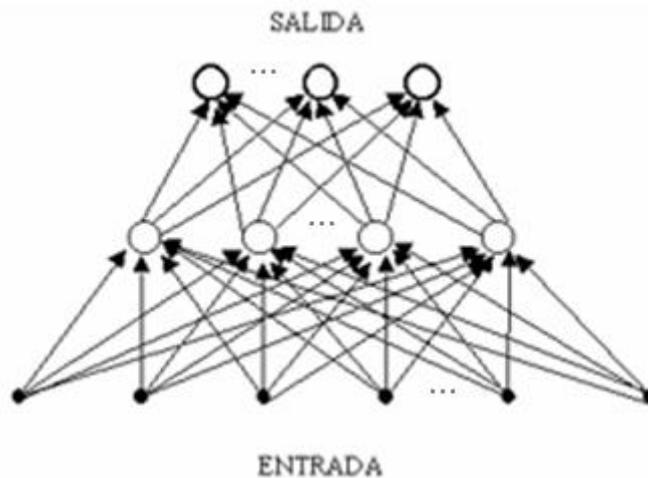


Figura 15: Ejemplo de arquitectura de red feedforward multicapa

4. RESULTADOS

En esta sección se muestran los resultados obtenidos, luego de clasificar el conjunto de patrones preprocesados utilizando la red neuronal descrita en la sección anterior. Dichos resultados se presentan en las tablas 1 y 2 con el siguiente formato: en la primer columna se especifica el dígito, en la segunda la cantidad de muestras de ese dígito que fueron mal clasificadas, en la tercer columna la cantidad de muestras reconocidas para ese dígito y en la última columna se muestra el porcentaje correspondiente a la cantidad de muestras reconocidas para ese dígito. En la última fila se muestran los porcentajes totales por dígito y sobre todo el conjunto de patrones.

El porcentaje de patrones correctamente clasificados fue de 99.17 % y 90.20 % para el conjunto de entrenamiento y testeo respectivamente. Este resultado es alentador, ya que mejora el porcentaje obtenido por la misma arquitectura de red entrenada con el mismo conjunto de patrones pero sin preprocesar (87.1 % de patrones de testeo reconocidos). Asimismo, el resultado obtenido para el conjunto de testeo preprocesado es comparable al presentado por otros autores [1] [2] que utilizan técnicas basadas en la transformada wavelet para extraer características de los patrones de entrada, y luego entrenan una red feedforward multicapa como clasificador.

Dígito	Mal Clasificados	Bien Clasificados	% Reconocidos
0	5/400	395/400	98,75
1	1/400	399/400	99,75
2	4/400	396/400	99,00
3	4/400	396/400	99,00
4	4/400	396/400	99,00
5	1/400	399/400	99,75
6	4/400	396/400	99,00
7	6/400	394/400	98,50
8	2/400	398/400	99,50
9	2/400	398/400	99,50
TOTAL	33/4000	3967/4000	99,17

Cuadro 1: Resultados obtenidos sobre el conjunto de entrenamiento.

Dígito	Mal Clasificados	Bien Clasificados	% Reconocidos
0	20/200	180/200	90,00
1	6/200	194/200	97,00
2	18/200	182/200	91,00
3	32/200	168/200	84,00
4	7/200	193/200	96,50
5	30/200	170/200	85,00
6	17/200	183/200	91,50
7	15/200	185/200	92,50
8	36/200	164/200	82,00
9	15/200	185/200	92,50
TOTAL	196/2000	1804/2000	90,20

Cuadro 2: Resultados obtenidos sobre el conjunto de testeo.

5. CONCLUSIONES

Hemos presentado un método de preprocesamiento basado en la transformada wavelet continua en dos dimensiones que nos ha permitido entrenar una red neuronal feedforward multicapa como clasificador de dígitos manuscritos, utilizando la base de datos de la Universidad de Concordia, Canadá. Los resultados obtenidos son comparables a los de otros autores [1] [2], que utilizan técnicas de preprocesamiento basadas en la transformada wavelet y también entrenan una red neuronal feed forward como clasificador. Los trabajos citados utilizan wavelets o multiwavelets de una dimensión, y requieren la identificación del contorno del dígito como paso previo, lo cual no es necesario en nuestro caso.

También hemos logrado una representación de los dígitos que es invariante frente a traslaciones, con resultados alentadores. Como trabajo futuro pensamos extraer características de los dígitos que también sean invariantes frente a cambios de escala y de orientación, utilizando propiedades de la misma transformada, con la idea de mejorar el desempeño de nuestro clasificador.

REFERENCIAS

- [1] G. Y. Chen, T. D. Bui and A. Krzyzak: Contour-Based Handwritten Numeral Recognition using Multiwavelets and Neural Networks, *Pattern Recognition*, Vol.36, No.7, pp.1597-1604, 2003.
- [2] P. Wunsch, A.F. Laine: Wavelet Descriptors for multiresolution recognition of handprinted characters, *Pattern Recognition*, vol. 28, no. 8, pp. 1237-1249, 1995.
- [3] L. M. Seijas y E. C. Segura: Un clasificador neuronal que explica sus respuestas: aplicación al reconocimiento de dígitos manuscritos; Proc. IX Congreso Argentino de Ciencias de la Computación (CACIC 2003), La Plata, Argentina, 2003.
- [4] S.-W. Lee, C.-H. Kim, H. Ma y Y. Y. Tang.: Multiresolution Recognition of Unconstrained Handwritten Numerals with Wavelet Transform and Multilayer Cluster Neural-Network, *Pattern Recognition*, Vol. 12, pp. 1953-1961, 1996.
- [5] Hertz J., A. Krogh, R. Palmer: *Introduction to the Theory of Neural Computation*, Santa Fe Institute Editorial Board, 1990.
- [6] Haykin S.: *Neural Networks A Comprehensive Foundation*. Prentice Hall. 1999.
- [7] Antoine, J.-P.; P. Vandergheynst; K. Bouyoucef; R. Murenzi: Target detection and recognition using two-dimensional isotropic and anisotropic wavelets, *Automatic Object Recognition V, SPIE Proc.*, 2485, pp. 20-31, 1995.
- [8] Antoine, J.-P.; R. Murenzi: Two dimensional directional wavelets and the scale-angle representation, *Signal Process.* 53, p. 259-281, 1996.
- [9] E. López-Rubio, J. Muñoz-Pérez, , J. Gómez-Ruiz: A principal components analysis self-organizing map. *Neural Networks* 17(2), 261-270, 2004.
- [10] Zhang, B., Fu, M., Yan, H.: A nonlinear neural network model of mixture of local principal component analysis: application to handwritten digits recognition. *Pattern Recognition* 34(2), pp. 203-214, 2001.
- [11] Cho, S.B.: Self-Organizing Map with Dynamical Node Splitting: Application to Handwritten Digit Recognition, *Neural Computation* 9, 1345-1355, 1997.