

UN CLASIFICADOR NEURONAL QUE EXPLICA SUS RESPUESTAS: APLICACIÓN AL RECONOCIMIENTO DE DÍGITOS MANUSCRITOS

Leticia María Seijas – Enrique Carlos Segura

Departamento de Computación – Facultad de Cs. Exactas y Naturales – Universidad de Buenos Aires

Planta Baja - Pabellón I - Ciudad Universitaria - (1428) Buenos Aires – Argentina

Tel. +54-11-4576-3359 - email: lseijas@dc.uba.ar – leticiaseijas@yahoo.com.ar – esegura@dc.uba.ar

Resumen: *En este trabajo se presenta un sistema reconocedor de patrones con un diseño modular, explicativo y altamente paralelizable, basado en una técnica híbrida de aprendizaje, que incluye la utilización de mapas auto-organizativos de Kohonen. Dicho sistema se aplica al tratamiento de un problema real: el reconocimiento de dígitos manuscritos sin restricciones, obteniéndose resultados competitivos con los publicados por otros autores. El sistema propuesto no sólo permite clasificar patrones; también permite explicar las respuestas dadas y detectar patrones ambiguos, indicando a qué otras clases podrían pertenecer. Además, los mapas Kohonen utilizados, muestran gráficamente la distribución de los datos de entrenamiento, contribuyendo también a la explicación de las respuestas. La utilización de varias características representativas extraídas de los datos de entrada durante la etapa previa de preprocesamiento, y el tratamiento de las mismas en forma independiente, contribuye a la modularidad del sistema y a la posibilidad de entrenarlo en forma paralela. El uso de dichas características permite, además, que la presencia de errores en los patrones de entrada no tenga gran influencia en la respuesta del sistema.*

Palabras clave: *reconocimiento de patrones – clasificación – redes neuronales – mapas auto-organizativos.*

1. INTRODUCCIÓN

La utilización de redes neuronales artificiales ha permitido obtener muy buenos resultados en el área de reconocimiento de patrones y, en particular, la de reconocimiento de caracteres. Uno de los primeros intentos en desarrollar una arquitectura de red para tal efecto fue realizado por Fukushima ([FUK/87], [FUK/91], [FRE/92]). El modelo propuesto, denominado *Neocognitron*, presenta una estructura jerárquica multicapa compleja, orientada a la extracción de características de cada patrón, y que permite reconocer números manuscritos aún frente al cambio de posición y de tamaño. Sin embargo, una gran cantidad de trabajos publicados, utilizan los métodos clásicos de reconocimiento de patrones, como las redes feed-forward multicapa, entrenadas con el algoritmo de backpropagation. Esta arquitectura ha sido reconocida como una herramienta poderosa para la resolución del problema de la clasificación de patrones, dado que su potencialidad radica en su poder discriminativo, y en su capacidad de aprender y representar conocimiento implícito. Pero a la hora de resolver problemas reales, este modelo presenta varias dificultades. Por ejemplo, cómo determinar el tamaño y la estructura de la red apropiados y cómo obtener una buena capacidad de generalización. Distintos autores, como [LEC/89],[KRZ/90],[LEE/95], han presentado arquitecturas con métodos que atacan estas dificultades. Es importante destacar también que se han obtenido resultados altamente competitivos utilizando técnicas de aprendizaje no supervisado, como son los mapas auto-organizativos de Kohonen ([CHI/95],[CHO/97]).

Del análisis de la literatura existente surge que la mayoría de los métodos propuestos, frente al problema del reconocimiento de un patrón, dan una respuesta, por ejemplo indicando a qué clase pertenece el patrón, y no brindan información semántica que permita explicar el por qué de dicha respuesta. Por lo tanto, si un patrón de entrada es lo suficientemente confuso como para poder pertenecer a más de una clase, la red se ve obligada a elegir una sola. No coincide esto con la capacidad humana de distinguir entre lo claro y lo confuso, y en el último caso quizás poder responder: el patrón ingresado, ¿a qué patrones se parece más, o con cuáles se podría confundir? El

objetivo de este trabajo es presentar una arquitectura de red y una técnica híbrida de aprendizaje, orientadas a la clasificación de patrones y que permitan recuperar información semántica del sistema. El sistema ha sido testeado en un problema de reconocimiento de dígitos manuscritos sin restricciones. Se analizan ventajas, problemáticas y futuras mejoras al modelo propuesto.

2. MÉTODO PROPUESTO

En esta sección se describe la arquitectura del sistema y el funcionamiento del método propuesto.

2.1. ARQUITECTURA DEL RECONOCEDOR

La Figura 1 presenta la estructura del sistema.

Preprocesamiento de los datos y Entrada al sistema

Una de las características del sistema es que el análisis estructural de cada patrón es previo al proceso de clasificación y separado de él. Esta distinción permite alcanzar un diseño de aplicación más general y estructura menos compleja, sólo orientada al proceso de clasificación. Por lo tanto, durante el diseño del reconocedor se define una etapa de preprocesamiento donde cada patrón es ajustado a un tamaño fijo.

Reconocedor

Como se observa en la Figura 1, el reconocedor está compuesto por dos capas o niveles. El primer nivel está conformado por K mapas auto-organizativos de Kohonen bidimensionales e independientes, cada uno de los cuales está dedicado a una característica diferente, extraída del patrón de entrada. El segundo nivel consiste en un módulo analizador encargado de definir y explicar la salida del sistema, e integrado por los siguientes elementos: la *tabla de confiabilidad*, y dos parámetros ajustables durante la utilización del sistema, denominados *umbral de confiabilidad* y *distancia mínima*. Cada red Kohonen de la primera capa, emite una respuesta frente a un patrón ingresado. Podríamos pensar que cada mapa actúa como un juez que, basándose solamente en el análisis de la característica que le corresponde, decide a qué clase pertenece el patrón de entrada. La conexión entre la primera y la segunda capa del sistema se realiza a través de esta nueva representación del patrón, constituida por las respuestas o votos de los cinco "jueces". Otro elemento que vincula a ambos niveles, es la *tabla de confiabilidad*, la cual refleja cuán confiable es la respuesta dada por cada mapa Kohonen de características. De esta manera, el módulo analizador de la segunda capa, basándose en la nueva representación del patrón, y utilizando la tabla de confiabilidad, deberá decidir cuál es la respuesta final del sistema.

Salida

El reconocedor siempre da una respuesta que expresa a qué clase pertenece el patrón ingresado y explica las respuestas dadas. La explicación consiste en mostrar a qué clase se pareció más el

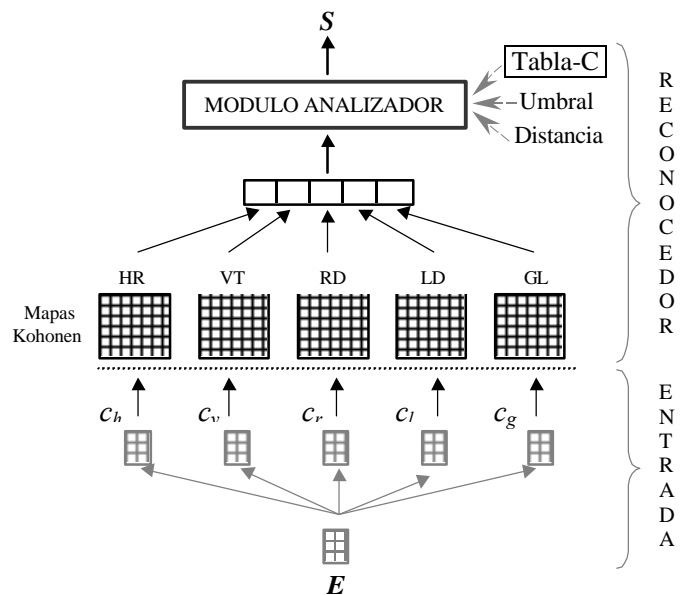


Figura 1: Estructura del sistema reconocedor propuesto; aplicación al clasificador de dígitos manuscritos sin restricciones

Cada mapa Kohonen está asociado a una característica direccional (HR-horizonta; VT-vertical; RD-diag.derecha; LD-diag.izquierda; GL-global). En el módulo analizador: Tabla-C. Tabla de confiabilidad – Parámetros: Umbral de confiabilidad y Distancia mínima

patrón de entrada para cada característica analizada, a través de la observación de los votos de cada red / juez, y el peso asignado a cada uno. Por otro lado, y como parte de la explicación, si un patrón es confuso para el sistema, podremos saber con qué otros dígitos podría confundirse, o lo que es lo mismo, con qué clases comparte la mayor cantidad de características.

Otro elemento que también forma parte de la salida del sistema, es la representación gráfica de la distribución de los patrones de entrada (utilizados en el proceso de entrenamiento del sistema), a través de los mapas topológicos de la primera capa. Este ordenamiento o agrupamiento por similitud se realiza por cada característica e inclusive para el patrón completo, lo que permite realizar una comparación interesante, además de, por ejemplo, evaluar la importancia de cada característica en la definición de los patrones.

2.2. DESCRIPCIÓN DEL ALGORITMO

El entrenamiento total del sistema puede dividirse en dos etapas: el entrenamiento de los mapas de la primera capa, y la generación de la tabla de confiabilidad.

Entrenamiento de la primera capa del sistema

Cada mapa auto-organizativo es entrenado en forma independiente según el algoritmo de Kohonen ([KOH/82],[HER/90],[SCH/92]). Los patrones del conjunto de entrenamiento deben ser preprocesados, extrayéndose las características seleccionadas. De esta manera, la entrada a cada mapa estará compuesta por la característica extraída de cada patrón y asociada a dicho mapa.

Como resultado de esta primera etapa de aprendizaje obtenemos los mapas topológicos que reflejan la distribución de los datos de entrada en un espacio de dos dimensiones. Dichos mapas permitirán, por ejemplo, analizar la relación entre las características observables en cada uno, con el mapa asociado a todas las características combinadas (patrón completo).

Una vez finalizado el entrenamiento se procede al rotulado de las neuronas, recortando de cada mapa las que quedaron asociadas a más de N clases y definiendo el resto de las unidades según la clase a la cual respondieron mayor cantidad de veces.

Generación de la tabla de confiabilidad

Luego del proceso de aprendizaje de los mapas Kohonen, el conjunto de entrenamiento es pasado nuevamente por la primera capa del reconocedor, generando la *tabla de confiabilidad*. El objetivo de esta tabla es expresar con un valor la relevancia que tiene la respuesta de cada red, en la definición de un patrón de entrada. Esto se realiza a través de la asignación de un puntaje o peso a cada respuesta (podríamos pensar en un voto calificado por parte de cada juez / red).

El procedimiento para la confección de la tabla consiste en calcular el porcentaje de equivocaciones cometidas por cada mapa para cada clase. Una vez hecho el cálculo, la elección de un valor mayor y cercano a todos los valores de la tabla permitirá asignar finalmente un puntaje mínimo a las clases menos confiables (con gran número de equivocaciones), y un puntaje mayor a las clases más confiables (con menor número de equívocos). Si por ejemplo, el mapa correspondiente a cierta característica ha asociado una gran cantidad de patrones del conjunto de entrenamiento con una clase en particular, a la que no pertenecían, entonces en el reconocimiento esta respuesta será considerada poco confiable y por consiguiente no será tenida “muy” en cuenta.

Observamos que en esta fase de entrenamiento, para medir el comportamiento de cada mapa y generar la tabla de confiabilidad, es necesario conocer “a priori” a qué clase corresponde cada patrón ingresado. Esto quiere decir que, a diferencia de lo que ocurría en la primera capa, aquí estamos utilizando una técnica de aprendizaje supervisado.

Con la generación de la tabla de confiabilidad, el proceso de entrenamiento del reconocedor queda finalizado.

Técnica de aprendizaje

El modelo de reconocedor propuesto podría ser encuadrado dentro de un esquema de **aprendizaje híbrido**, pues en una primera etapa, cada mapa es entrenado con el método de Kohonen, que responde a un modelo de aprendizaje no supervisado, mientras que en una segunda etapa, para construir la tabla de confiabilidad, nos vemos obligados a conocer el rótulo de cada patrón, con lo cual, podemos hablar de aprendizaje supervisado. Finalmente, el reconocedor está capacitado para clasificar adecuadamente, luego de transitar por las dos etapas de aprendizaje.

Algoritmo para el reconocimiento y explicación de las respuestas

Los pasos para el reconocimiento de un patrón son los siguientes:

1. Se ingresan las características correspondientes al patrón de entrada a la primera capa del reconocedor.
2. Los “jueces” de la primera capa (mapas Kohonen) emiten su voto, es decir, asocian cada característica del patrón con una clase, generando una nueva representación del patrón.
3. Los votos de los jueces son ingresados a la segunda capa. El módulo analizador, entonces, asigna a cada voto un puntaje, utilizando la tabla de confiabilidad.
4. Los puntajes obtenidos en la tabla se acumulan por clase, quedando la respuesta del reconocedor definida por la clase que ha obtenido el mayor puntaje acumulado.

La respuesta dada por el sistema (paso 4 del algoritmo), puede ser explicada a través de la observación de los votos emitidos por cada juez / red, con el correspondiente puntaje asignado a cada uno (paso 3 del algoritmo). Si además, queremos distinguir entre dígitos confusos para el sistema y no confusos, y en el primer caso, saber con qué clases podrían confundirse, entonces, es necesario definir el *umbral de confiabilidad* y la *distancia mínima*.

El *umbral de confiabilidad* indica a partir de qué valor (puntaje) un patrón se considera confuso. Para esto, se compara el puntaje acumulado para la clase ganadora, obtenido en el paso 4 del algoritmo, con el umbral: si el puntaje lo supera, entonces el patrón analizado está bien definido para el sistema y no se confunde con ninguna otra clase; en caso contrario, dicho patrón no estaría tan definido, con lo cual sí podría confundirse con otras clases.

La idea surge a partir del siguiente razonamiento: si para un patrón de entrada, todos los mapas responden la misma clase, entonces el módulo analizador asignará a dicha clase el puntaje máximo posible (que es la suma de los pesos asignados a todos los mapas para esa característica en la tabla de confiabilidad). Consideramos entonces que el patrón está muy bien definido. A partir de esto podemos pensar que si un patrón obtiene un puntaje cercano al máximo para una clase, entonces está bastante definido, más allá de si luego, el resultado es correcto o no. Por otro lado, la obtención de un puntaje “bajo” para la clase ganadora, estaría reflejando que los mapas Kohonen encendieron gran variedad de clases para el patrón, con lo cual hay confusión en la definición del mismo.

El parámetro que denominamos *distancia mínima* establece un umbral que permite determinar a qué clases podría pertenecer el patrón confuso: estas clases serían las más cercanas en puntaje a la clase ganadora. La elección del umbral de confiabilidad y de la distancia mínima, presupone un ajuste en el comportamiento del sistema, a través de sucesivas pruebas, hasta encontrar los valores apropiados para el conjunto de patrones con el cual se está trabajando.

2.3 IMPLEMENTACIÓN DEL MODELO

2.3.1. Preprocesamiento de los datos de entrada

Base de Datos

Para realizar los experimentos presentados en este trabajo, se ha utilizado la base de datos de números manuscritos sin restricciones, de la Universidad de Concordia, Canadá. Esta base de datos está compuesta por 6000 dígitos escritos a mano sin restricciones, provistos por el Servicio Postal

de los Estados Unidos, y extraídos de códigos postales manuscritos en los sobres de la correspondencia. Cada número de la base fue digitalizado en dos niveles, en una grilla de 64 x 224 elementos cuadrados de 0.153 mm cada uno, obteniéndose una resolución de aproximadamente 166 pixels por pulgada [SUE/90].

De la totalidad de los patrones, 4000 fueron utilizados para el entrenamiento de las distintas redes (400 patrones por dígito), y 2000 para el testeo (200 patrones por dígito). La Figura 2, muestra algunos dígitos elegidos al azar, pertenecientes al conjunto de entrenamiento y de testeo, permitiendo observar la característica de “escritura irrestricta” de los mismos, representada por los diferentes estilos de escritura, inclinaciones y trazos. La base de datos presenta patrones de diferentes tamaños. Éstos serán normalizados durante la etapa de preprocesamiento, para ser ingresados al sistema con un tamaño uniforme. Una característica importante de esta base de datos es que el conjunto de entrenamiento y el de testeo, incluyen ejemplos que son ambiguos, inclasificables y aun mal clasificados.

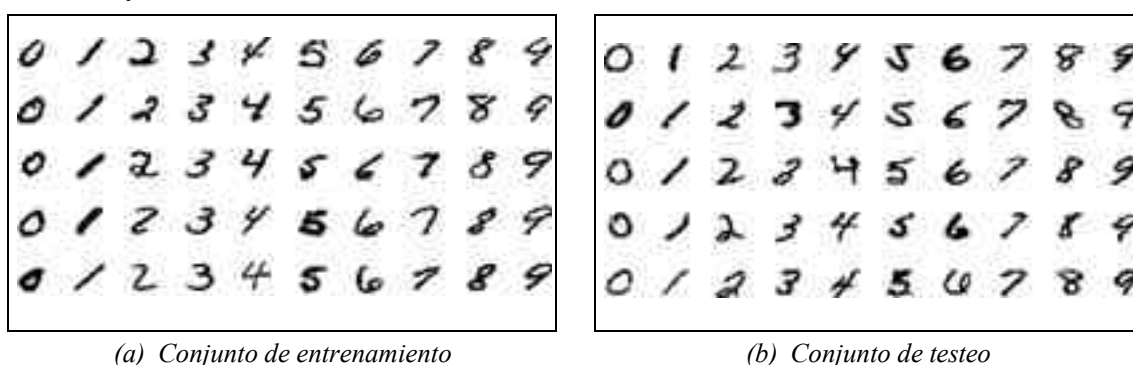


Figura 2: Dígitos manuscritos de ejemplo de la base de datos de la Universidad de Concordia, normalizados en tamaño

Extracción de características

Los números, ya sea manuscritos o impresos, están compuestos esencialmente por dibujos de líneas, es decir, estructuras de una dimensión en un espacio bidimensional. Por lo tanto, la detección localizada de segmentos de línea, parece ser un método de extracción de características adecuado.

Por cada ubicación en la imagen, la información acerca de la presencia de un segmento de línea en una dirección dada, es almacenada en un mapa de características [KNE/92]. Para la realización de esta tarea se ha utilizado el detector de bordes de Kirsch, seleccionado entre otros detectores de bordes diferenciales de primer orden representativos, como el de Frei-Chen, de Prewitt, de Sobel, etc. Esta elección se basa en que, entre todos ellos, las máscaras de Kirsch, son conocidas por detectar bordes en las cuatro direcciones, en forma rápida, adecuada, y más precisa que otros, dado que los ocho vecinos son considerados en su totalidad [LEE/95].

La base de datos descrita en el punto anterior, presenta los patrones numéricos en matrices de distintos tamaños, por lo tanto, el preprocesamiento incluirá la normalización de estas imágenes conformadas por ceros y unos, a una dimensión determinada. En nuestros experimentos, hemos normalizado los patrones a 16x16, dimensión que creemos adecuada, y que también han utilizado otros autores [LEC/89], [LEE/95], [CHO/97].

Una vez finalizada la etapa de normalización, se procede a la extracción de las cuatro características direccionales, horizontal, vertical, diagonal derecha, diagonal izquierda, de cada patrón, a través de la utilización de las máscaras de Kirsch [PRA/78] mencionadas.

La extracción de características de cada patrón define un nuevo conjunto de datos de entrada al reconocedor, conformado por las cuatro características direccionales y la característica global. En realidad, cada una de ellas ingresará a un mapa Kohonen diferente, en la primera capa del sistema, conformando cinco conjuntos de entrenamiento (uno por característica) que hacen referencia a los mismos patrones. La dimensión de la entrada a la red no es un tema menor, ya que influye

directamente sobre el poder de generalización de la misma [LEC/89]. En nuestro caso, la dimensión de la entrada a cada mapa es de 256 (16x16). Consideramos que este valor es adecuado para obtener redes con buen nivel de generalización y a su vez, patrones claramente definidos, que no pierdan ni su forma ni detalles importantes necesarios para el reconocimiento.

2.3.2. Mapas Kohonen

En la primera capa del reconocedor hemos definido 5 mapas auto-organizativos dedicados, cada uno, a una característica diferente extraída de los datos de entrada. A cada red se ingresa un vector de dimensión 16 x 16, de números reales, en el caso de las cuatro características direccionales, y de ceros y unos, en el caso de la característica global.

Consideramos apropiado definir la dimensión y el tamaño de cada mapa en 30 filas y 30 columnas. De esta manera, cada clase tendría asignado, en promedio, 90 neuronas, cantidad considerable para poder distinguir las diferentes formas de los 400 patrones por dígito, presentados a la red.

Frente al problema de los bordes, que aparece cuando se aplica el concepto de vecindario a las unidades de las filas y columnas que delimitan el mapa Kohonen (a estas unidades les “faltan” vecinos), se ha considerado la estructura de la red como un toroide, donde la columna y la fila 29 son también vecinas de la columna y fila 0, respectivamente. Se ha comprobado que esta consideración mejora los resultados del aprendizaje. Luego del entrenamiento, cada mapa es rotulado, para lo cual los patrones son ingresados nuevamente a la red, para poder asociar cada neurona con la clase del patrón que la hizo ganadora. Como resultado de este proceso, se observa

que en cada mapa un porcentaje de unidades queda asociado con más de una clase. Para obtener un mapa Kohonen con unidades totalmente definidas, es decir, donde cada neurona esté asociada a una única clase, se ha aplicado el siguiente criterio: las neuronas indefinidas asociadas a N ó más clases se “recortan” del mapa. Por ejemplo, si $N = 2$, todas las neuronas indefinidas serán eliminadas del mapa, quedando únicamente las unidades asociadas con una única clase; en cambio, si $N=3$, las neuronas asociadas a 3, 4, 5, etc. clases diferentes, serán eliminadas de la red, permaneciendo las unidades asociadas a una clase y las asociadas a dos clases. Aún así, siguen quedando neuronas indefinidas; ahora éstas serán asociadas con la clase de los patrones que las hicieron ganadoras con mayor frecuencia. El criterio de “recortar” neuronas representa la idea de que si una unidad ha quedado asociada a “varias” clases, entonces no puede definir correctamente ningún dígito, con lo cual provoca equivocaciones en el reconocimiento. Al ser eliminada, permite que el patrón que antes la encendía, pueda encender otra neurona de la clase que realmente lo defina. La Figura 3 presenta uno de los cinco mapas Kohonen luego del entrenamiento y del proceso de definición, según el criterio especificado. Puede observarse

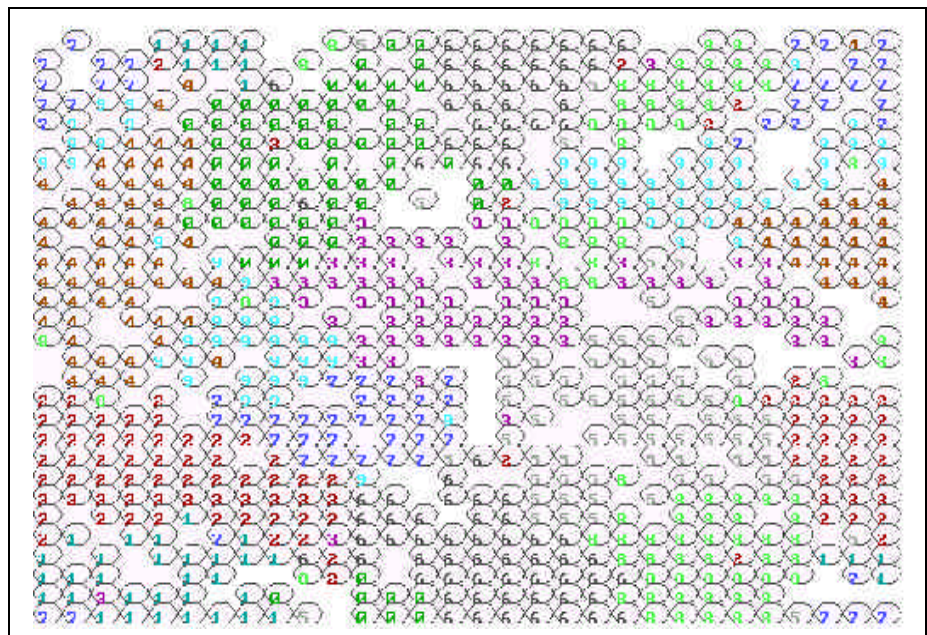


Figura 3: Mapa auto organizativo de la primera capa del reconocedor, asociado a la característica global

que en cada mapa un porcentaje de unidades queda asociado con más de una clase. Para obtener un mapa Kohonen con unidades totalmente definidas, es decir, donde cada neurona esté asociada a una única clase, se ha aplicado el siguiente criterio: las neuronas indefinidas asociadas a N ó más clases se “recortan” del mapa. Por ejemplo, si $N = 2$, todas las neuronas indefinidas serán eliminadas del mapa, quedando únicamente las unidades asociadas con una única clase; en cambio, si $N=3$, las neuronas asociadas a 3, 4, 5, etc. clases diferentes, serán eliminadas de la red, permaneciendo las unidades asociadas a una clase y las asociadas a dos clases. Aún así, siguen quedando neuronas indefinidas; ahora éstas serán asociadas con la clase de los patrones que las hicieron ganadoras con mayor frecuencia. El criterio de “recortar” neuronas representa la idea de que si una unidad ha quedado asociada a “varias” clases, entonces no puede definir correctamente ningún dígito, con lo cual provoca equivocaciones en el reconocimiento. Al ser eliminada, permite que el patrón que antes la encendía, pueda encender otra neurona de la clase que realmente lo defina. La Figura 3 presenta uno de los cinco mapas Kohonen luego del entrenamiento y del proceso de definición, según el criterio especificado. Puede observarse

claramente la distribución de los dígitos, ya que cada clase está asociada a un color. Tener en cuenta que las unidades en los vértices de la red, son vecinas, así como también las que están en los bordes, como consecuencia de haber considerado el mapa como un toroide, durante el entrenamiento.

2.3.3 Módulo analizador

Luego de la construcción de la primera capa del sistema, el conjunto de patrones de entrenamiento es ingresado a dicha capa, con el objetivo de medir el comportamiento de cada red y generar la tabla de confiabilidad.

El procedimiento a seguir ya fue explicado en la sección 2.2. En la Tabla 1 se presentan los valores de confiabilidad generados para el presente sistema reconocedor. Si analizamos el contenido de la tabla, observamos que los pesos varían entre un valor máximo de 18.020 para la característica global del 6, y un valor mínimo de 0.430 para la característica vertical del 7. Evidentemente, el mapa asociado a la característica global ha reconocido adecuadamente los patrones de entrenamiento asociados al 6, y prácticamente no ha confundido con esta clase patrones que representaban otros números. No obtuvo tan buenos resultados el mapa asociado con la característica vertical, cuando respondió “7” frente a los patrones de entrenamiento ingresados. El puntaje mínimo asociado a este mapa y a esta característica refleja que en la mayoría de los casos la respuesta dada por la red “vertical”, fue equivocada.

En el proceso de reconocimiento toda esta información es muy útil: si por ejemplo, frente al ingreso de un patrón, el mapa global responde “6”, y el vertical “7”, podríamos preguntarnos en cuánto la característica global permite definir a un seis y la vertical a un siete.

CLASE	CARACTERISTICA				
	HORIZONTAL	VERTICAL	DIAG.DER.	DIAG.IZQ.	GLOBAL
0	12.670	15.410	14.400	12.400	17.040
1	16.840	13.750	16.610	15.910	15.160
2	12.270	13.600	16.930	13.100	17.460
3	05.680	07.950	06.900	13.230	12.930
4	15.540	07.590	14.850	12.080	16.990
5	11.550	08.470	08.570	15.080	15.150
6	14.090	17.520	14.670	17.950	18.020
7	14.350	00.430	15.200	08.950	14.460
8	07.060	14.460	08.360	12.710	14.820
9	14.750	05.220	10.440	07.620	15.740

Tabla 1: Tabla de confiabilidad para el sistema reconocedor propuesto

La respuesta en este ejemplo es contundente: mucho en el primer caso y nada en el segundo.

Para definir qué número es el dígito ingresado, la suma de puntajes por clase permite manejar situaciones intermedias, donde los pesos no son valores extremos. Los casos de “empate” (dos o más clases con el máximo puntaje acumulado), son muy poco frecuentes, como consecuencia del propio diseño del reconocedor; de hecho, en ninguna de nuestras pruebas, con conjuntos de 2000 y 4000 patrones, se ha presentado esta situación. Sin embargo, si en algún momento ocurriera, el sistema tomaría como respuesta cualquiera de las clases con puntaje máximo, y luego explicaría con qué “otras clases” se estaría confundiendo. La definición de los parámetros *umbral de confiabilidad* y *distancia mínima*, y la presentación de los resultados explicativos por parte del sistema se tratarán en la sección siguiente, donde se describen los resultados experimentales.

3. EXPERIMENTOS REALIZADOS Y ANÁLISIS DE RESULTADOS

3.1. CONJUNTO DE ENTRENAMIENTO

La clasificación de los patrones del conjunto de entrenamiento nos ha permitido calibrar el sistema reconocedor, a través del análisis de los resultados obtenidos (éstos no forman parte de los resultados experimentales). Durante este proceso, el porcentaje de patrones reconocidos ha sido de 97.40 %. En la Tabla 2 se muestran las justificaciones de las respuestas emitidas por el sistema, para los patrones presentados en la Figura 4 (A). En dicha figura podemos observar patrones del

conjunto de entrenamiento, bien clasificados, mal clasificados, y otros, que el sistema ha considerado confusos, luego de definir los valores de umbral de confiabilidad y distancia mínima en 35.00 y 4.00 respectivamente.

<i>Números Reconocidos</i>				<i>Números Mal Clasificados</i>				<i>Nros. Confusos (para el reconocedor)</i>			
(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)
0	5	7	8	0 - 1	2 - 7	3 - 2	7 - 1	4 9 - 4	5 6 - 5	8 9 - 8	3 3 - 5

(A) Conjunto de entrenamiento

<i>Números Reconocidos</i>				<i>Números Mal Clasificados</i>				<i>Nros. Confusos (para el reconocedor)</i>			
(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)
1	3	6	9	1 - 4	2 - 6	2 - 1	8 - 9	5 3 - 5	5 5 - 3	6 6 - 4	7 1 - 7

(B) Conjunto de testeo

Figura 4: Ejemplos de números reconocidos, mal clasificados, y confusos, luego del proceso de clasificación
(A) Conjunto de entrenamiento - (B) Conjunto de testeo

Números reconocidos: el número debajo de cada letra indica el rótulo del patrón - *Números Mal Clasificados:* $x - y$ indica que el dígito era x pero clasificó y - *Números Confusos:* el número debajo de cada letra indica a qué clase pertenece el patrón. Luego, $x - y$ indica que clasificó x y que puede confundirse con y

Tomemos, por ejemplo, el caso del patrón (b), que ha sido clasificado correctamente como cinco. El sistema explica la respuesta a través de los votos de los jueces, que expresan que todas las características del patrón responden a las asociadas con la clase del cinco, menos la característica diagonal derecha que es más parecida a un nueve. Si observamos el dígito en la Figura 4 (A), veremos que quizás esto sea así, pero el hecho de tener un leve parecido a una clase en alguna característica, no alcanza para decidir que el patrón es de esa clase. En el caso del patrón (c) el sistema reconoce correctamente un siete, pero nos informa que en realidad el patrón no es muy claro, ya que comparte las características vertical y diagonal izquierda con la clase del nueve, y la característica diagonal derecha con la clase del cuatro. A pesar de la aparente indefinición de los votos de los jueces, la respuesta del sistema fue correcta, pues los jueces que han tenido mayor peso, fueron los que decidieron "7". Todo esto, aparte de darnos información acerca de cómo ha sido escrito o representado el dígito, nos hace notar la importancia de analizar las características por separado: si alguna de ellas no está clara o se confunde, igual se puede llegar a un resultado correcto, gracias a la participación de otras características.

Si observamos ahora los patrones mal clasificados, y sus explicaciones, veremos que, el sistema no está lejos de la respuesta considerada correcta. Además, en alguno de los casos presentados, el patrón podría pertenecer tanto a la clase decidida por el reconocedor, como a la definida de antemano. Con respecto a los patrones considerados confusos, el sistema responde según las definiciones del umbral de confiabilidad y la distancia mínima, que deben ser ajustadas según las necesidades. Para el sistema, que un patrón haya sido confuso, implica que, aunque el reconocedor decide una respuesta, ésta no es contundente, o lo que es lo mismo, puede estar sujeta a errores (el patrón es confuso para el sistema). De esta manera, cuando el reconocedor indica con qué otras clases podría haberse confundido el patrón, lo que está diciendo en realidad, es que éste podría perfectamente pertenecer a esas clases. En los tres primeros ejemplos presentados para el conjunto de entrenamiento, el reconocedor no ha clasificado correctamente, pero indica que éstos se

confunden con la clase a la que realmente pertenecen. Si observamos sus formas, veremos que los dígitos comparten la mayor cantidad de características con las clases “equivocadas”, a las que podrían pertenecer, ya sin dudas, con muy pocas modificaciones hechas a la figura, lo que demuestra su cercanía.

		<i>Representación del patrón según el voto de los jueces</i>					<i>El patrón ingresado</i>		
		<i>Hr</i>	<i>Vt</i>	<i>Rd</i>	<i>Ld</i>	<i>Gl</i>	<i>era</i>	<i>reconoció</i>	<i>confunde</i>
RECONOCIDOS	a)	0	0	0	0	0	0	-	
	b)	5	5	9	5	5	5	-	
	c)	7	9	4	9	7	7	-	
	d)	8	8	8	8	8	8	-	
MAL CLASIFICADOS	e)	0	0	1	1	1	0	1	
	f)	7	7	7	2	7	2	7	
	g)	2	2	2	3	2	3	2	
	h)	7	7	9	1	1	7	1	
CONFUSOS	i)	4	9	9	4	9	4	9	
	j)	3	6	6	5	5	5	6	
	k)	8	9	9	9	8	8	9	
	l)	5	3	3	3	5	3	5	

Tabla 2: Explicaciones dadas por el sistema para los resultados de la clasificación presentados en la Figura 4 (A) para el conjunto de entrenamiento

3.2. CONJUNTO DE TESTEO

El conjunto de testeo, formado por 2000 dígitos manuscritos extraídos de la base de datos de la Universidad de Concordia, y ya presentados en la sección 2.3.1, fue ingresado al sistema para su clasificación. El porcentaje de patrones reconocidos ha sido de 91.90 %. Las explicaciones de las respuestas emitidas por el sistema, para los patrones presentados en la Figura 4 (B), se muestran en la Tabla 3. El análisis de los ejemplos presentados para el conjunto de testeo es similar al realizado para el conjunto de entrenamiento. Observamos algunos casos de patrones correctamente definidos, y sin dudas, ya que todos los jueces han votado de la misma manera. Es el caso del ejemplo (d). Otros, también bien clasificados, ya no están tan definidos, con lo cual acá, volvemos a notar la importancia de trabajar con varias características, para que, frente a distorsiones del patrón, esto no afecte demasiado la respuesta final del sistema. No debemos olvidar que cada voto está asociado a un peso, que determina la confiabilidad de la respuesta, o sea cuánto va a influir en el resultado final. Observemos ahora el patrón del ejemplo (e), en la Figura 4 (B), ¿qué número representa? El reconocedor lo asocia con la clase del “4”, aunque en realidad está rotulado como “1”. Si pedimos una explicación (ver Tabla 3), veremos que los jueces lo han identificado con varias clases, aunque finalmente ha predominado la presentada como ganadora. Esto nos demuestra que, por más confuso que sea el patrón, el sistema siempre logra asociarlo a alguna clase, que es con la que comparte la mayor cantidad de características.

En los ejemplos de patrones mal clasificados de la Tabla 3, claramente podemos identificar en ellos varios números combinados. Es el caso del ejemplo (h) donde observamos un ocho y un nueve, y así lo expresa el sistema en su explicación. Los ejemplos (f) y (g), son realmente confusos; en el primer caso, en el que el dígito es rotulado de antemano como “2”, observamos que ninguno de los jueces ha votado por esta clase, pues para cada mapa, la figura del patrón se acerca más a las características de otros números.

		<i>Representación del patrón según el voto de los jueces</i>					<i>El patrón ingresado</i>		
		<i>Hr</i>	<i>Vt</i>	<i>Rd</i>	<i>Ld</i>	<i>Gl</i>	<i>era</i>	<i>reconoció</i>	<i>confunde</i>
RECONOCIDOS	a)	1	1	0	4	1	1	-	
	b)	8	3	3	3	3	3	-	
	c)	6	0	5	6	6	6	-	
	d)	9	9	9	9	9	9	-	
MAL CLASIFICADOS	e)	4	2	9	4	1	4	-	
	f)	8	6	6	8	2	6	-	
	g)	0	3	6	2	1	1	-	
	h)	9	8	9	8	9	9	-	
CONFUSOS	i)	3	5	3	5	3	3	5	
	j)	0	3	5	3	5	5	3	
	k)	0	6	6	4	4	6	4	
	l)	7	7	7	1	1	1	7	

Tabla 3: Explicaciones dadas por el sistema para los resultados de la clasificación presentados en la Figura 4 (B), para el conjunto de testeo

Para que el sistema explique qué patrones considera confusos y con qué clases podrían confundirse, hemos utilizado los mismos valores de umbral de confiabilidad (35.00) y de distancia mínima (4.00), que para el conjunto de entrenamiento. Si observamos las figuras de los dígitos en la Figura 4 (B) – *Números confusos*, veremos que las respuestas del sistema, en este caso, se acercan a lo que podría concluir un humano.

3.3. COMPARACIÓN DE RESULTADOS CON OTROS TRABAJOS

En la Tabla 4 se presentan los resultados del reconocimiento del conjunto de entrenamiento y de testeo, por dígito, utilizando el clasificador propuesto.

<i>CONJUNTO DE ENTRENAMIENTO</i>				<i>CONJUNTO DE TESTEO</i>			
<i>Clase</i>	<i>Cantidad patrones mal clasificados</i>	<i>% Error</i>	<i>% Reconocidos</i>	<i>Clase</i>	<i>Cantidad patrones mal clasificados</i>	<i>% Error</i>	<i>% Reconocidos</i>
0	01 / 400	0.25	99.75	0	05 / 200	2.50	97.50
1	-	0.00	100.00	1	02 / 200	1.00	99.00
2	07 / 400	1.75	98.25	2	18 / 200	9.00	91.00
3	14 / 400	3.50	96.50	3	29 / 200	14.50	85.50
4	05 / 400	1.25	98.75	4	06 / 200	3.00	97.00
5	21 / 400	5.25	94.75	5	20 / 200	10.00	90.00
6	03 / 400	0.75	99.25	6	10 / 200	5.00	95.00
7	18 / 400	4.50	95.50	7	20 / 200	10.00	90.00
8	24 / 400	6.00	94.00	8	35 / 200	17.50	82.50
9	11 / 400	2.75	97.25	9	17 / 200	8.50	91.50
TOTAL		2.60	97.40	TOTAL		8.10	91.90

Tabla 4: Resultados del reconocimiento de dígitos manuscritos extraídos de la base de datos de la Universidad de Concordia, utilizando el reconocedor propuesto.

En la Tabla 5 pueden observarse los resultados obtenidos por otros autores en el reconocimiento de dígitos manuscritos, a través del uso de diferentes métodos basados en redes neuronales. Algunos, indicados en la tabla con “*”, han utilizado para el entrenamiento y testeo, la base de datos de la Universidad de Concordia.

Luego del análisis de los valores presentados, podemos apreciar que el resultado obtenido por el clasificador propuesto es muy bueno, teniendo en cuenta las ventajas que presenta su arquitectura, como por ejemplo: simplicidad, modularidad, facilidad de diseño e implementación, permitiendo,

además, obtener la explicación de las respuestas así como también la detección de patrones considerados confusos para el sistema, y la representación gráfica de la distribución de los datos de entrada a través de los mapas auto-organizativos de Kohonen. El tamaño del conjunto de entrenamiento es otra variable a tener en cuenta, ya que algunos de los mejores resultados obtenidos por otros modelos, han necesitado de un conjunto con gran cantidad de patrones. Sin embargo, creemos que el rendimiento del sistema propuesto es posible de mejorar.

Métodos	% Reconocidos	% Error	% Rechazados	Patrones en el conjunto de entrenamiento	Patrones en el conjunto de testeo
* Nadal (1988)	86.05	2.25	11.70	4000	2000
* Krzyzak (1990)	86.40	1.00	12.60	4000	2000
* Krzyzak (1990)	94.85	5.15	0.00	4000	2000
Le Cun (1990)	92.00	2.00	6.00	7291	2007
Le Cun (1990)	96.40	3.40	0.20	7291	2007
* Mai (1990)	92.95	2.15	4.90	4000	2000
* Suen (1990)	93.05	0.00	6.95	4000	2000
Knerr (1992)	90.30	1.00	8.70	7200	1800
Lemarie (1993)	97.97	2.03	0.00	8783	7394
* Kim & Lee (1994)	95.40	4.60	0.00	4000	2000
* Lee (1995)	97.80	2.20	0.00	4000	2000
* Cho (1997)	96.05	3.95	0.00	4000	2000

Tabla 5: Algunos de los mejores resultados obtenidos por otros autores en el reconocimiento de dígitos manuscritos, utilizando distintos reconocedores basados en redes neuronales [LEE/95] [CHO/97]

* Utilización de la base de datos de la Universidad de Concordia

4. CONCLUSIONES Y TRABAJOS FUTUROS

En este trabajo se ha presentado un sistema reconocedor de patrones con un diseño modular, explicativo y altamente paralelizable, basado en una técnica híbrida de aprendizaje, que incluye la utilización de mapas auto-organizativos de Kohonen. Dicho sistema ha sido aplicado a la resolución de un problema real: el reconocimiento de dígitos manuscritos sin restricciones. Consideramos que los resultados obtenidos en la clasificación son muy buenos y pueden compararse ventajosamente con los publicados en la literatura preexistente. Más específicamente, nuestra propuesta es:

General: el hecho de que la información a priori sobre el problema a resolver haya sido utilizada para preprocesar la entrada, en lugar de incorporarla al reconocedor, hace posible un diseño de propósito general, aplicable a otros problemas de reconocimiento de patrones.

Modular: el ordenamiento topológico de los mapas auto-organizativos permite la explicación gráfica de ciertos comportamientos, ayudando a decidir si es necesario incorporar / eliminar mapas y cuáles. Esto no implica el re-entrenamiento de los otros mapas: sólo habrá que redefinir la tabla de confiabilidad. Además, al incorporar nuevos mapas (o módulos de cualquier otro tipo), los parámetros de los mismos pueden diferir de los utilizados para los módulos existentes. En cualquier caso, el analizador deberá simplemente tener en cuenta más votos para decidir el resultado.

Paralelizable: en la etapa de implementación, el entrenamiento de las redes puede realizarse en paralelo, con el consiguiente ahorro de tiempo.

Explicativo: en la clasificación, el sistema reconocedor propuesto siempre da una respuesta asociando el patrón de entrada con la clase con la que éste comparte la mayor cantidad de características. Además, brinda información acerca de la distribución de las entradas, lo que permite explicar las respuestas dadas. Esta característica resulta fundamental en ciertos tipos de problemas, como los relacionados con la toma de decisiones. La descomposición de los patrones de entrada en características permite aumentar el nivel de detalle en el análisis de cada patrón con la consecuente

mejora en la respuesta del sistema. Es también para destacar la detección de patrones confusos para el sistema, y el hecho de explicitar con qué clases se confunden estos patrones una vez clasificados.

Como trabajos futuros, consideramos interesante la posibilidad de aplicar la arquitectura propuesta a la resolución de otros problemas de reconocimiento de patrones, para observar la adaptabilidad del clasificador. Por otro lado sería interesante ver qué ocurre con el comportamiento del sistema si se agregan redes asociadas a otras características tales como cantidad de vértices, curvas, aberturas, entre otras presentes en cada patrón a tratar. Asociado a esto último, podría investigarse la técnica que utiliza el módulo analizador para decidir las respuestas, más aun si éste se ve obligado a manejar un gran número de votos.

REFERENCIAS

- [CHI/95] Chi Z. y H. Yan. Handwritten Numeral Recognition Using a Small Number of Fuzzy Rules With Optimized Defuzzification Parameters. *Neural Networks*, Volumen 8, número 5, 821-827. 1995.
- [CHO/97] Cho, S. B. Self-Organizing Map with Dynamical Node Splitting: Application to Handwritten Digit Recognition. *Neural Computation* 9, 1345-1355. 1997.
- [FRE/92] Freeman J. A., D.M. Skapura. *Neural Networks: Algorithms, Applications and Programming Techniques*. Addison-Wesley Publishing Company. 1992.
- [FUK/87] Fukushima K. Neural network model for selective attention in visual pattern recognition and associative recall. *Applied Optics* Volumen 26, número 23, 4985-4992. Optical Society of America. 1987.
- [FUK/91] Fukushima K., N. Wake. Handwritten Alphanumeric Character Recognition by the Neocognitron. *IEEE Transactions on Neural Networks* Volumen 2, número 3, 355-365. 1991.
- [HER/90] Hertz J., A. Krogh, R. Palmer. *Introduction to the Theory of Neural Computation*. Santa Fe Institute Editorial Board. 1990.
- [KNE/92] Knerr S., L. Personnaz y G. Dreyfus. Handwritten digit recognition by neural networks with single layer training. *IEEE Transactions on Neural Networks*, Volumen 3, número 6, 962-968. 1992. Citado en [LEE/95].
- [KOH/82] Kohonen T. Self-organized formation of topologically correct feature maps. *Biological Cybernetics* 43, 59-69. Springer Verlag, 1982.
- [KRZ/90] Krzyzak A., W. Dai y C.Y. Suen. Unconstrained handwritten character classification using modified backpropagation model. *Proceedings of 1st. International Workshop on Frontiers in Handwritten Recognition*, Montreal, Canadá, 155-166. 1990. Citado en [LEE/95].
- [LEC/89] Le Cun Y., B. Boser, J. S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation* 1, 541-551. 1989.
- [LEE/95] Lee, S. W. Multilayer Cluster Neural Network for Totally Unconstrained Handwritten Numeral Recognition. *Neural Networks*, Vol. 8, Nro. 5, 783-792. 1995.
- [PRA/78] Pratt W. K. *Digital image processing*, New York: Wiley. 1978. Citado en [LEE/95].
- [SCH/92] Schalkoff, R. *Pattern Recognition: Statistical, Structural and Neural Approaches*. Singapur, John Wiley & Sons. 1992.
- [SUE/90] Suen C.Y., C. Nadal, T. Mai, R. Legault y L. Lam. Recognition of handwritten numerals based on the concept of multiple experts. *Proceedings of 1st. International Workshop on Frontiers in Handwritten Recognition*, Montreal, Canadá, 131-144. 1990. Citado en [CHO/97].