

Detección de esqueletos de caracteres mediante una red neuronal competitiva basada en segmentos

J.A. Gómez Ruiz, J. Muñoz Pérez, M.A. García Bernal, E. López Rubio

Departamento de Lenguajes y Ciencias de la Computación
Universidad de Málaga.
Campus de Teatinos s/n.
29071 Málaga.
{janto,munozp,magb,ezeqlr}@lcc.uma.es

Resumen

La *esqueletización* (palabra técnica procedente del vocablo inglés “skeletonization”) es un proceso mediante el cual se transforma una determinada forma u objeto de una imagen digital, compuesta de una determinada cantidad de píxeles, en un objeto basado en líneas, de forma que las propiedades topológicas del objeto se preserven. Este objeto resultante constituido por líneas se denomina *esqueleto*. Los esqueletos son muy útiles a la hora de reconocer, dentro de una imagen, objetos o patrones que sean alargados o con una determinada forma como por ejemplo caracteres, polígonos, patrones cromosómicos, etc. Los esqueletos proporcionan una abstracción de las características topológicas y geométricas del objeto, de forma que la esqueletización puede verse también como un proceso de compresión de datos. Con el auge de las nuevas tecnologías y los formatos electrónicos para los documentos, mejoran y avanzan también las técnicas de reconocimiento óptico de caracteres (OCR), incluyéndose entre ellas las basadas en el reconocimiento de esqueletos para los caracteres que componen el documento en cuestión. En este artículo presentamos un nuevo modelo de red neuronal competitiva basada en segmentos con fase de expansión, que permite obtener los esqueletos de los caracteres suministrados de una forma totalmente no supervisada.

Palabras clave: Esqueletización, Redes Neuronales Competitivas, Aprendizaje No Supervisado, Reconocimiento de Caracteres.

1. Introducción

El reconocimiento patrones es una disciplina en continuo desarrollo en los últimos años, sobre todo en el campo del análisis de imágenes y la visión por computador. Las redes neuronales artificiales son una herramienta muy potente dentro de este campo, particularmente los mapas autoorganizados de Kohonen (SOFM) (1997), ya que son capaces de detectar la topología y la distribución de probabilidad de los patrones de entrada.

Son muchos los autores que se han basado en estos mapas autoorganizados para realizar reconocimiento invariante de patrones, tales como Pham y Bayro-Corrochano (1994), Corridoni et al. (1996), Wang y Lin (1996), Subba Reddy y Nagabhushan (1998) y Gómez Ruiz, López Rubio y Muñoz Pérez (2001). En el último trabajo indicado, se ha propuesto un nuevo método basado en los mapas autoorganizados que permite detectar objetos de una forma invariante a escalados, traslaciones y rotaciones.

Sin embargo, con el auge de las nuevas tecnologías

y los formatos electrónicos para los documentos, mejoran y avanzan también las técnicas de reconocimiento óptico de caracteres (OCR). Desde mediados de los años cincuenta son muchos los autores que se han centrado en este campo, pudiendo citar entre otros: Glauberman (1956), Ramesh (1989), Taxt et al. (1990), Gader et al. (1991), Suen et al. (1993) y Westall y Narasimha (1993). Dentro de estas técnicas de reconocimiento de caracteres son destacables las basadas en esqueletos (Trier, Jain y Taxt, 1996). La *esqueletización* (palabra técnica procedente del vocablo inglés “skeletonization”) es un proceso mediante el cual se transforma una determinada forma u objeto de una imagen digital, compuesto de una determinada cantidad de píxeles, en un objeto basado en líneas, de forma que las propiedades topológicas del objeto se preserven. Este objeto resultante constituido por líneas se denomina *esqueleto*.

Los esqueletos son muy útiles a la hora de reconocer, dentro de una imagen, objetos o patrones que sean alargados o con una determinada forma como por ejemplo caracteres, polígonos, patrones cromosómicos, etc. Los esqueletos proporcionan una abstracción de las características topológicas y geométricas del objeto, de forma que al almacenarse sólo una determinada información estructural de los objetos en estudio, la esqueletización puede verse también como un proceso de compresión de datos.

El concepto de esqueleto fue introducido por primera vez a mediados de los años sesenta por Blum (1964) y desde entonces se han desarrollado un gran número de técnicas para determinarlo. Estas técnicas varían unas de otras básicamente en términos de implementación y eficiencia (Datta, Parui y Chaudhuri, 2001).

En la última década y con el gran avance y desarrollo de nuevas topologías de redes neuronales, se han propuesto diversos algoritmos para la detección de esqueletos, como los propuestos por Amin et al. (1996), Cardoner y Thomas (1997), Datta, Pal y Parui (1997) y Datta, Parui y Chaudhuri (2001).

La mayoría de estas técnicas se basan en las redes autoorganizadas de Kohonen (1997) con diversas y variadas modificaciones muy dependientes del tipo de objeto o carácter en estudio. Datta, Pal y Parui (1997) proponen una red SOFM para la detección de esqueletos a la que necesitan añadirle mecanismos de supervisión dependiendo de los caracteres a analizar, mientras que con la red competitiva basada en segmentos con fase de expansión que proponemos en este trabajo siempre obtenemos el

esqueleto de una forma no supervisada e independiente del carácter u objeto en cuestión.

Este artículo está organizado de la siguiente forma: en la sección 2 presentamos la nueva red neuronal competitiva basada en segmentos con fase de expansión; en la sección 3 presentamos la detección de esqueletos mediante los mapas autoorganizados de Kohonen y los inconvenientes que tienen; en la sección 4 mostramos la detección de esqueletos mediante la nueva red neuronal propuesta; y finalmente, en la sección 5 exponemos las conclusiones.

2. Red neuronal competitiva basada en segmentos con fase de expansión

2.1. Red neuronal basada en segmentos

Recientemente, García Bernal (2001) presenta una red competitiva basada en segmentos donde el potencial sináptico es una función lineal de dos dipolos. Dicha red permite la formación de grupos o categorías, mediante aprendizaje no supervisado, donde las clases o categorías vienen identificadas por segmentos en lugar de centroides. El segmento conduce a una mejor representación de un grupo o clase que un centroide, que sólo nos da la posición del grupo. La red basada en segmentos se aplica a la formación de grupos utilizando el conjunto de datos IRIS creados por Anderson (1939). Los algoritmos de aprendizaje no supervisado como el de las k -medias de McQueen (1967), el LBG (Linde et al., 1980) y los algoritmos competitivos basados en centroides, consiguen entre 12 y 17 clasificaciones incorrectas, mientras que con la red propuesta en el trabajo citado se consiguen un total de 3 clasificaciones incorrectas, lo cual es un resultado sorprendente.

La red está constituida por K unidades de proceso (neuronas) en la capa de salida y N sensores en la capa de entrada, donde cada unidad de proceso está conectada con los sensores de entrada. El *peso sináptico* de la conexión de la neurona i con la entrada j va a venir dado por dos *dipolos* \mathbf{w}_{i1} y \mathbf{w}_{i2} que constituyen el segmento, donde

$$\mathbf{w}_{i1} = (w_{i1}^1, w_{i1}^2, \dots, w_{i1}^N) \text{ y } \mathbf{w}_{i2} = (w_{i2}^1, w_{i2}^2, \dots, w_{i2}^N)$$

El *potencial sináptico* asociado a la neurona i viene dado por la siguiente expresión:

$$h_i = \alpha_i(\mathbf{x})h_{i1} + \bar{\alpha}_i(\mathbf{x})h_{i2} - \frac{1}{2}\alpha_i(\mathbf{x})\bar{\alpha}_i(\mathbf{x})\|\mathbf{w}_{i1} - \mathbf{w}_{i2}\|^2$$

donde

$$\alpha_i(\mathbf{x}) = \frac{(\mathbf{w}_{i1} - \mathbf{w}_{i2})^T (\mathbf{x} - \mathbf{w}_{i2})}{\|\mathbf{w}_{i1} - \mathbf{w}_{i2}\|^2}$$

con $\alpha_i(\mathbf{x}) + \bar{\alpha}_i(\mathbf{x}) = 1$.

h_{i1} y h_{i2} son números reales que vienen dados por las expresiones:

$$h_{i1} = \mathbf{w}_{i1}^T \mathbf{x} - \frac{1}{2} \mathbf{w}_{i1}^T \mathbf{w}_{i1} \quad \text{y} \quad h_{i2} = \mathbf{w}_{i2}^T \mathbf{x} - \frac{1}{2} \mathbf{w}_{i2}^T \mathbf{w}_{i2}$$

Finalmente, la regla de aprendizaje para la unidad ganadora (la que tiene mayor potencial sináptico) viene dada por las expresiones:

$$\Delta \mathbf{w}_{r1}(k) = \begin{cases} \eta[\mathbf{x}(k) - \mathbf{w}_{r0}(k)] \alpha_r(\mathbf{x}(k)) & \text{si } 0 < \alpha_r(\mathbf{x}(k)) < 1 \\ \eta[\mathbf{x}(k) - \mathbf{w}_{r1}(k)] & \text{si } \alpha_r(\mathbf{x}(k)) \geq 1 \\ 0 & \text{si } \alpha_r(\mathbf{x}(k)) \leq 0 \end{cases}$$

$$\Delta \mathbf{w}_{r2}(k) = \begin{cases} \eta[\mathbf{x}(k) - \mathbf{w}_{r0}(k)] \bar{\alpha}_r(\mathbf{x}(k)) & \text{si } 0 < \alpha_r(\mathbf{x}(k)) < 1 \\ 0 & \text{si } \alpha_r(\mathbf{x}(k)) \geq 1 \\ \eta[\mathbf{x}(k) - \mathbf{w}_{r2}(k)] & \text{si } \alpha_r(\mathbf{x}(k)) \leq 0 \end{cases}$$

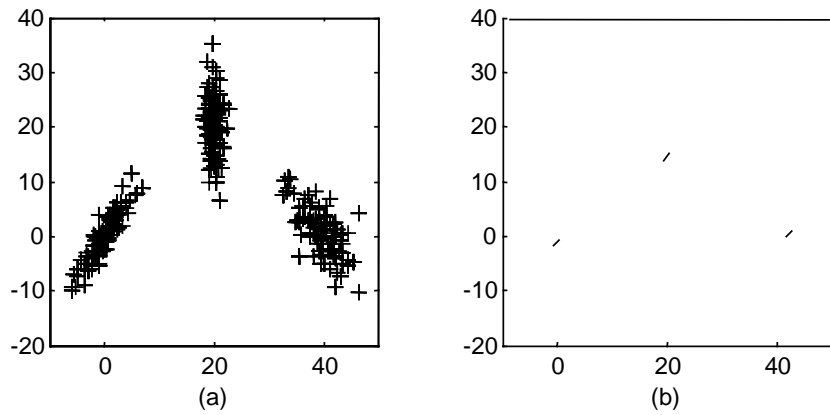


Figura 1. Configuración inicial de la red

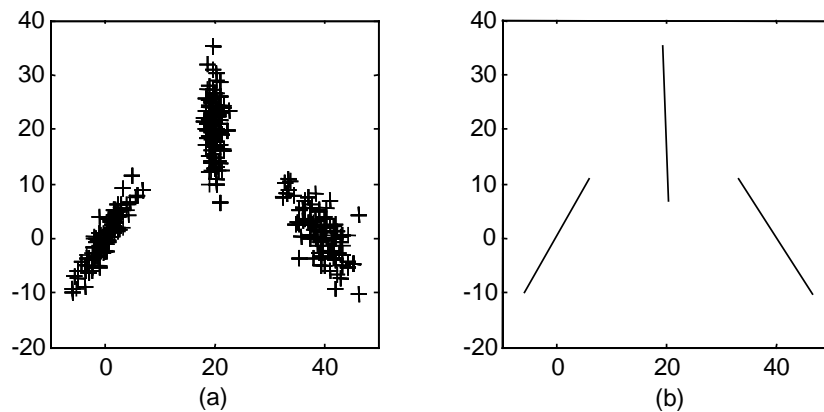


Figura 2. Configuración final de la red

2.2. Inconvenientes de la red neuronal basada en segmentos

Sin embargo, la red citada, que se basa en el método de descenso del gradiente, alcanza soluciones que son mínimos locales (pudiendo no ser globales) y, además, la solución que encuentra la red depende también de los valores iniciales de los pesos sinápticos. Ello conlleva a que los segmentos no representen a un único grupo, teniendo segmentos que representen a dos grupos, mientras que otros grupos vengan representados por más de un segmento. Este inconveniente hace que la red no sea adecuada para la detección, por ejemplo, de esqueletos de imágenes.

Para ilustrar este problema vamos a utilizar un conjunto de datos compuesto por tres grupos linealmente separables, compuestos cada uno de ellos por cien patrones generados con una distribución normal.

Los segmentos sinápticos se seleccionan aleatoriamente entre los patrones del espacio de entrada con un tamaño muy pequeño para que se vayan expandiendo de forma no supervisada. Vamos a ver como evoluciona la red con los patrones de entrada mostrados en la figura 1(a) y los

segmentos sinápticos iniciales en la figura 1(b). Tras la evolución de la red, los segmentos sinápticos van creciendo y adaptándose a los datos de entrada, quedando la configuración final como la mostrada en la figura 2(b).

Podemos observar que el resultado es el deseado, puesto que los segmentos se han posicionado perfectamente encontrando la dirección predominante de los datos. Si nos fijamos en la configuración inicial de la figura 1, los segmentos iniciales escogidos aleatoriamente están uno dentro de cada uno de los tres grupos, de forma que en el proceso de aprendizaje, al introducir un patrón de entrada se activa siempre el segmento de la clase en la que está dicho patrón, puesto que es el que está más cercano.

Vamos ahora a probar otra configuración en la que se posicionen al menos dos segmentos sinápticos en una clase. Esta configuración inicial la podemos observar en la figura 3. Tras la evolución de la red, obtenemos la configuración de la figura 4 en donde podemos apreciar que hay un segmento que abarca dos clases y dos segmentos dentro de una. Ello se debe a que un dipolo (extremo del segmento) se dirige hacia otro grupo mientras que el otro dipolo se mantiene en el grupo donde estaba.

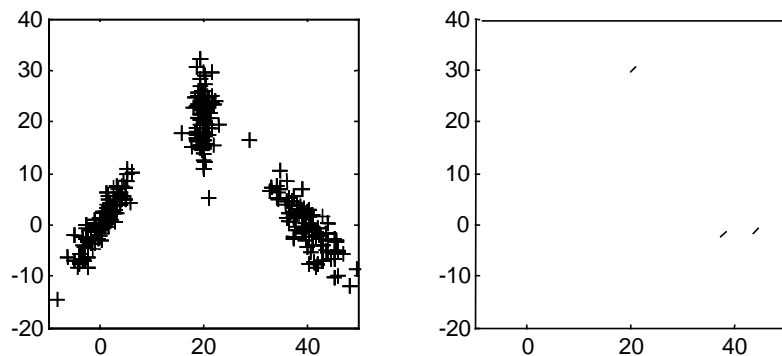


Figura 3. Configuración inicial de la red

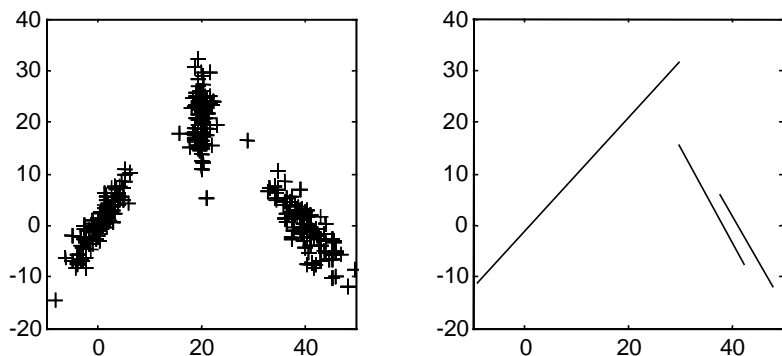


Figura 4. Configuración final de la red

2.3. Incorporación de la fase de expansión

Para solucionar el problema comentado es necesario controlar el tamaño de los segmentos durante la fase de aprendizaje para que no puedan crecer libremente, sino de forma gradual. Por tanto, tenemos que incorporar algún mecanismo que dé libertad a los segmentos para desplazarse y orientarse pero sin crecer demasiado y evitar que abarquen varias zonas del espacio de los patrones de entrada. Dicho mecanismo va a incorporar un nuevo parámetro de control, β , que vamos a denominar parámetro de expansión. La misión de este parámetro es controlar el tamaño que va a tener el segmento durante la fase de aprendizaje: al principio no permitirá que crezca demasiado y por lo tanto se tendrán que desplazar ambos dipolos del segmento hacia el grupo que representará y, posteriormente, una vez posicionados sobre los grupos, dejará que los segmentos crezcan libremente adaptándose a dicho grupo según los patrones que lo constituyan.

Al introducir un patrón \mathbf{x} , tendremos una unidad ganadora, r , que será aquella cuyo segmento esté más cerca al patrón de entrada. Ya sabemos que el segmento se estira siempre que se verifique bien que $\alpha_r(\mathbf{x}) > 1$ ó $0 > \alpha_r(\mathbf{x})$, puesto que cuando $\alpha_r(\mathbf{x}) \in [0,1]$ el segmento sólo se orienta.

Tendremos por tanto que incorporar el mecanismo que controle el tamaño del segmento sólo en los casos en los que se estira, es decir cuando $\alpha_r(\mathbf{x}) > 1$ ó $0 > \alpha_r(\mathbf{x})$.

Vamos a suponer que dado un patrón de entrada \mathbf{x} tenemos que $\alpha_r(\mathbf{x}) > 1$. Según la regla de aprendizaje propuesta en el apartado 2.1, sólo se modifica un extremo del segmento, \mathbf{w}_{r1} , estirándose hacia el patrón de entrada según la expresión:

$$\mathbf{w}_{r1}(k+1) = \mathbf{w}_{r1}(k) + \eta(\mathbf{x} - \mathbf{w}_{r1}(k))$$

donde como ya sabemos, η es la tasa de aprendizaje y k es el número de iteración dentro de la fase de aprendizaje.

Lo que nosotros proponemos ahora es reducir el tamaño del nuevo segmento acercando el otro extremo \mathbf{w}_{r2} . Para acercar el extremo \mathbf{w}_{r2} vamos a utilizar la ecuación paramétrica del nuevo segmento obtenido por la expresión anterior que tiene por extremos $\mathbf{w}_{r1}(k+1)$ y $\mathbf{w}_{r2}(k)$. Es decir,

$$\mathbf{w}_{r2}(k+1) = \mathbf{w}_{r2}(k) + \beta(\mathbf{w}_{r1}(k+1) - \mathbf{w}_{r2}(k))$$

$$0 \leq \beta \leq 1$$

De esta forma, si el valor de β es cercano a uno, el nuevo valor para el extremo \mathbf{w}_{r2} estará muy cercano al extremo \mathbf{w}_{r1} y por tanto el segmento se desplaza sin estirarse demasiado, que era lo que buscamos. Por otra parte, si el valor de β es próximo a cero, el extremo \mathbf{w}_{r2} permanece prácticamente igual, y por tanto ahora el segmento si se va a estirar.

Para el caso de que dado un patrón de entrada \mathbf{x} obtengamos que $0 > \alpha_r(\mathbf{x})$, por las mismas deducciones llegamos a que las modificaciones que se tienen que hacer son:

$$\mathbf{w}_{r2}(k+1) = \mathbf{w}_{r2}(k) + \eta(\mathbf{x} - \mathbf{w}_{r2}(k))$$

$$\mathbf{w}_{r1}(k+1) = \mathbf{w}_{r1}(k) + \beta(\mathbf{w}_{r2}(k+1) - \mathbf{w}_{r1}(k))$$

$$0 \leq \beta \leq 1$$

Este parámetro β , es el parámetro de expansión introducido al principio del subapartado y según hemos visto, cuando toma valores cercanos a uno, los segmentos crecen muy poco y cuando toma valores cercanos a cero los segmentos crecen sin restricciones.

Es conveniente que al principio de la fase de aprendizaje, el parámetro β tome valores cercanos a uno de forma que no permita que los segmentos se estiren demasiado y por tanto se posicionen sobre los grupos de los patrones de entrada. De la misma forma, al final de la fase de aprendizaje, el parámetro β debe tomar valores cercanos a cero permitiendo que los segmentos se estiren libremente pudiéndose ahora adaptar a los patrones que configuran los distintos grupos.

Nosotros vamos a usar la siguiente función para el parámetro de expansión β :

$$\beta(k+1) = \beta(k) \left(1 - \frac{k}{T}\right)^u \quad (1)$$

donde k es la iteración en curso, T es el número total de iteraciones del proceso de aprendizaje y u es una constante, fijada a priori y dependiente del problema. Dicha constante nos va a regular la pendiente de la curva.

Para ver cómo es la función de la ecuación (1), podemos ver dos representaciones de la misma con valores distintos de u . La función con $u = 0.05$ la podemos apreciar en la figura 15, mientras que con $u = 0.03$ la podemos apreciar en la figura 16. En ambos casos, consideramos el valor inicial de $\beta(0) = 1$ y el número total de iteraciones $T = 2400$.

$$\mathbf{w}_{r_1}(k+1) = \begin{cases} \mathbf{w}_{r_1}(k) + \eta(k) \cdot [\mathbf{x}(k) - \mathbf{w}_{r_0}(k)] \cdot \alpha_r(\mathbf{x}(k)) & \text{si } 0 < \alpha_r(\mathbf{x}(k)) < 1 \\ \mathbf{w}_{r_1}(k) + \eta(k) \cdot [\mathbf{x}(k) - \mathbf{w}_{r_1}(k)] & \text{si } \alpha_r(\mathbf{x}(k)) \geq 1 \\ \mathbf{w}_{r_2}(k+1) = \mathbf{w}_{r_2}(k) + \beta(k)(\mathbf{w}_{r_1}(k+1) - \mathbf{w}_{r_2}(k)) & \\ 0 & \text{si } \alpha_r(\mathbf{x}(k)) \leq 0 \end{cases} \quad (2)$$

$$\mathbf{w}_{r_2}(k+1) = \begin{cases} \mathbf{w}_{r_2}(k) + \eta(k) \cdot [\mathbf{x}(k) - \mathbf{w}_{r_0}(k)] \cdot \bar{\alpha}_r(\mathbf{x}(k)) & \text{si } 0 < \alpha_r(\mathbf{x}(k)) < 1 \\ 0 & \text{si } \alpha_r(\mathbf{x}(k)) \geq 1 \\ \mathbf{w}_{r_2}(k) + \eta(k) \cdot [\mathbf{x}(k) - \mathbf{w}_{r_2}(k)] & \text{si } \alpha_r(\mathbf{x}(k)) \leq 0 \\ \mathbf{w}_{r_1}(k+1) = \mathbf{w}_{r_1}(k) + \beta(k)(\mathbf{w}_{r_2}(k+1) - \mathbf{w}_{r_1}(k)) & \end{cases}$$

$$\text{donde } \alpha_r(\mathbf{x}) = \frac{(\mathbf{w}_{r_1} - \mathbf{w}_{r_2})^T (\mathbf{x} - \mathbf{w}_{r_2})}{\|\mathbf{w}_{r_1} - \mathbf{w}_{r_2}\|^2}$$

2.4. Nueva regla de aprendizaje

La deducción de la nueva regla de aprendizaje es similar a la mostrada en el apartado 2.1 y al incorporar la fase de expansión queda de la forma mostrada en la ecuación (2).

La tasa de aprendizaje η tiene que estar comprendida entre cero y uno para garantizar que en cada paso del proceso de aprendizaje, el término de la función de distorsión correspondiente a la unidad ganadora decrece, o por lo menos no crece (Gómez Ruiz, 2002).

El parámetro $\beta(k)$ toma valores entre cero y uno, puesto que dicho parámetro controla la ecuación paramétrica de un segmento constituido por sus dos extremos (dipolos) y toma valores según la expresión mostrada en la ecuación (1).

2.5. Nuevo algoritmo de aprendizaje

La misión del algoritmo de aprendizaje es la siguiente: dado un patrón de entrada, encontrar el segmento más cercano al mismo y modificar sus pesos sinápticos de forma que permitan acercarlo al patrón pero controlando ahora el tamaño del segmento, no dejándolo crecer con libertad en las primeras iteraciones de la fase de aprendizaje.

La nomenclatura utilizada es la siguiente:

- \mathbf{x} vector de entrada, $\mathbf{x} \in \mathfrak{R}^N$
- S_i segmento definido por los pesos sinápticos \mathbf{w}_{i1} y \mathbf{w}_{i2} de la neurona i :

$$\mathbf{w}_{i1} = (w_{i1}^1, w_{i1}^2, \dots, w_{i1}^N), \quad i = 1, 2, \dots, K,$$

$$\mathbf{w}_{i2} = (w_{i2}^1, w_{i2}^2, \dots, w_{i2}^N), \quad i = 1, 2, \dots, K,$$

$\alpha_i(\mathbf{x}) \in \mathfrak{R}$ coeficiente que afecta a los pesos sinápticos

$D(\mathbf{x}, S)$ distancia entre el patrón de entrada \mathbf{x} y el segmento sináptico S , donde

$$D(\mathbf{x}, S) = \|\mathbf{x} - S_{\mathbf{w}_{i1}\mathbf{w}_{i2}}\| = \begin{cases} \|\mathbf{x} - (\alpha_i(\mathbf{x})\mathbf{w}_{i1} + (1 - \alpha_i(\mathbf{x}))\mathbf{w}_{i2})\| & \text{si } \alpha_i(\mathbf{x}) \in (0, 1) \\ \|\mathbf{x} - \mathbf{w}_{i1}\| & \text{si } \alpha_i(\mathbf{x}) \geq 1 \\ \|\mathbf{x} - \mathbf{w}_{i2}\| & \text{si } \alpha_i(\mathbf{x}) \leq 0 \end{cases}$$

$0 < \eta \leq 1$ tasa de aprendizaje.

N dimensión del espacio de entrada.

K número de segmentos de la red

k número de la iteración en curso.

T número total de iteraciones del proceso de aprendizaje

$\beta(k) \in [0, 1]$ parámetro de expansión que toma valores según la expresión mostrada en la ecuación (1)

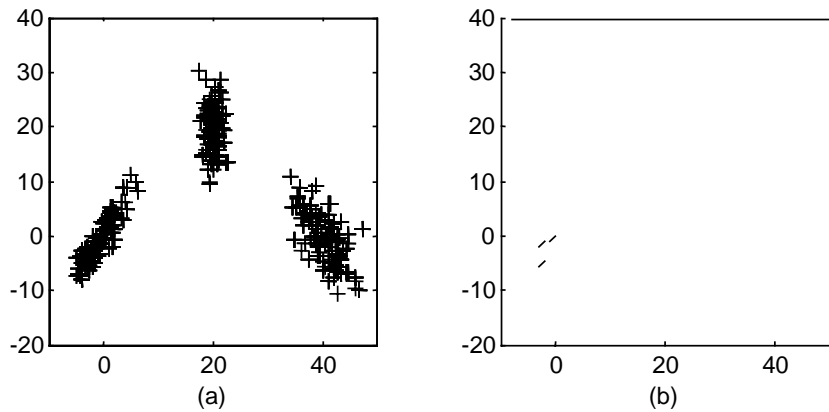


Figura 5. Configuración inicial de la red

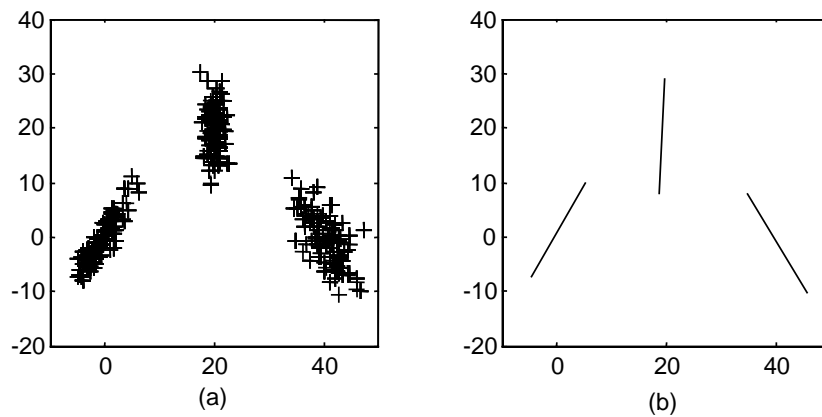


Figura 6. Configuración final de la red

Algoritmo de aprendizaje

Paso 0 Elegir los pesos iniciales que definen los segmentos S_i . Se elegirán aleatoriamente de entre los patrones que constituyen el espacio de entrada, con un tamaño pequeño. Damos valores iniciales a la tasa de aprendizaje y al parámetro de expansión con valores cercanos a uno y fijamos el valor de u .

Paso 1 Mientras la condición de parar (en paso 6) sea falsa, hacer los Pasos 2-5

Paso 2 Por cada entrada \mathbf{x} de entre los p patrones de entrada, hacer los Pasos 3-4

Paso 3 Calcular la unidad r cuyo potencial sináptico sea mayor:

$$h_r = \max_{1 \leq i \leq K} \{h_i\}$$

Paso 4 Actualizar $(\mathbf{w}_{r1}, \mathbf{w}_{r2})$ según la nueva regla de aprendizaje mostrada en la ecuación (2).

Paso 5 Reducir la tasa de aprendizaje y el parámetro de expansión.

Paso 6 Condición de parada:

La condición puede ser un número fijo de iteraciones máximo (es decir, de ejecuciones del Paso 1) o cuando la tasa de aprendizaje tome un valor suficientemente pequeño.

En cualquier caso:

- El segmento inicial ahora no siempre crece y busca, en la primera fase del aprendizaje, los grupos o las zonas donde están concentrados los patrones.
- El segmento estructural final no depende por tanto ahora de la posición inicial que tenga al iniciarse el proceso de aprendizaje.
- Antes de finalizar el proceso de aprendizaje, el parámetro de expansión β vale cero por lo que los segmentos se expanden libremente según las distribuciones de los datos.

2.6. Resultados experimentales

Vamos a comprobar como hemos solucionado el problema planteado en el apartado 2.2 con la nueva red basada en segmentos con fase de expansión. Vamos a utilizar la misma distribución de datos anterior y consideramos una configuración inicial desfavorable en la que los tres segmentos sinápticos estén posicionados en la misma clase. Esta configuración la vemos en la figura 5.

Como en la primera fase del aprendizaje el parámetro de expansión impide que los segmentos crezcan de forma descontrolada, estos se posicionan en las distintas clases y conforme el parámetro de expansión va decreciendo hasta llegar a cero, los segmentos se van estirando, llegando a la configuración final mostrada en la figura 6.

En esta experimentación hemos considerado que el valor para la constante u que gobierna el parámetro de expansión vale $u = 0.05$.

Esta nueva red con fase de expansión permite que los segmentos se adapten a los datos de entrada de forma controlada e independientemente de la configuración inicial de los segmentos y del orden de introducción de los patrones de entrada.

3. Detección de esqueletos con SOFM

3.1. Topología y funcionamiento de la red SOFM

El objetivo principal de los mapas autoorganizados de Kohonen (SOFM) es transformar un espacio de patrones de entrada de dimensión arbitraria en otro espacio discreto uni- o bidimensional de forma que esta transformación se efectúe de forma adaptativa y en un determinado orden topológico. En la figura 7 podemos apreciar un diagrama esquemático de una configuración en donde las neuronas están

dispuestas en una topología bidimensional y los patrones de entrada tienen dimensión cinco. Cada neurona de la rejilla está conectada con todos los sensores de entrada (en la figura sólo hemos conectado una unidad por cuestiones de claridad). En una configuración unidimensional las neuronas estarían dispuestas en una sola fila o columna.

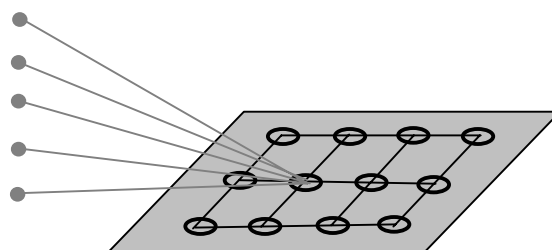


Figura 7. Configuración bidimensional

Cuando ya se tiene establecida la configuración de la red, el siguiente paso es inicializar los pesos sinápticos de la misma. Esto puede hacerse asignando a dichos pesos pequeños valores procedentes de un generador de números aleatorios, de forma que no se imponga ningún orden a priori a la configuración de neuronas.

Una vez que ya se tenga la red propiamente inicializada, se van introduciendo los patrones del espacio de entrada, dando lugar a tres mecanismos esenciales (Haykin, 1999):

- o *Competición*: Por cada patrón introducido en la red, todas las neuronas computan sus respectivos valores de acuerdo a una función discriminante. Esta función proporciona la base de la competición entre las neuronas. Aquella neurona que tenga el mayor valor de acuerdo a la función discriminante es declarada como la *unidad ganadora*.
- o *Cooperación*: La unidad ganadora determina la localización espacial de neuronas que son sus vecinas, de acuerdo a alguna determinada función de vecindad, proporcionando por tanto la base para la cooperación entre tales neuronas.
- o *Adaptación de los pesos sinápticos*: Este último mecanismo posibilita que tanto la unidad ganadora como sus vecinas ajusten sus pesos sinápticos de acuerdo al valor del patrón de entrada.

Vamos ahora a detallar como se llevan a cabo los tres mecanismos enunciados en las SOFM. Las neuronas en un mapa autoorganizado de Kohonen (SOFM) están organizadas en una rejilla m -

dimensional, donde normalmente $m = 1$ ó $m = 2$. En cada instante de tiempo t , se introduce en la red un patrón $\mathbf{x}(t)$ procedente de la distribución de entrada. Estos patrones de entrada pertenecen a un espacio de entrada de dimensión D .

El vector de pesos sinápticos \mathbf{w}_i de la neurona i representa un punto en el espacio de entrada. La neurona i cuyo vector de pesos sinápticos esté más cerca del patrón de entrada $\mathbf{x}(t)$ es la *unidad ganadora*, que viene determinada por la expresión:

$$i = \arg \min_j \|\mathbf{x}(t) - \mathbf{w}_j(t)\|, \quad j = 1, \dots, K$$

siendo K el número total de neuronas.

Tanto el vector de pesos de la unidad ganadora i como el de sus vecinas en la rejilla, se modifican para reflejar las características topológicas de la distribución de entrada. La expresión con la que se modifican dichos pesos es la regla de aprendizaje y viene dada por la expresión:

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \eta(t) \cdot \pi_{ij}(t) \cdot \|\mathbf{x}(t) - \mathbf{w}_j(t)\|$$

$$\pi_{ij}(t) = \exp\left(-\frac{d_{ji}^2}{2\sigma(t)^2}\right), \quad t = 1, 2, \dots, T$$

donde T es el número total de iteraciones del proceso de aprendizaje, $\eta(t)$ es el parámetro de aprendizaje, d_{ji} es la distancia entre la neurona ganadora i y la neurona j en la rejilla, y $\pi_{ij}(t)$ es una función gaussiana de la distancia lateral d_{ji} , llamada *función de vecindad*, con $\sigma(t) \rightarrow 0$ cuando $t \rightarrow \infty$. El valor de $\sigma(t)$ controla el tamaño de la vecindad, es decir dependiendo del valor que tenga $\sigma(t)$, la unidad ganadora i tendrá más o menos unidades vecinas. Por tanto el grado de vecindad entre las neuronas i y j viene dado por la función $\pi_{ij}(t)$.

Una función que gobierne el decaimiento de los parámetros $\sigma(t)$ y $\eta(t)$ usada por muchos autores es el decaimiento exponencial (Ritter et al., 1992), Obermayer et al., 1991) y viene dado por las expresiones:

$$\sigma(t) = \sigma_0 \exp\left(-\frac{t}{\tau_1}\right), \quad t = 1, 2, \dots, T$$

$$\eta(t) = \eta_0 \exp\left(-\frac{t}{\tau_2}\right), \quad t = 1, 2, \dots, T$$

donde η_0, σ_0, τ_1 y τ_2 son constantes a definir en el algoritmo SOFM.

El proceso de aprendizaje se divide en dos fases, la *fase de ordenación* y la *fase de convergencia* (Kohonen, 1997). Durante la fase de ordenación tiene lugar el proceso adaptativo en el cual se produce el ordenamiento topológico de las neuronas, mientras que en la fase de convergencia tiene lugar el afinamiento de dicha topología, proporcionando una cuantificación estadística más exacta del espacio de entrada.

3.2. Aplicación a la detección de esqueletos

La topología de red SOFM que vamos a utilizar es unidimensional con 15 neuronas. El proceso de aprendizaje lo hemos dividido en las fases de ordenación y convergencia, utilizando la regla de aprendizaje y función de vecindad descritas en el apartado 3.1.

El valor inicial para los pesos de la red lo hemos escogido de forma aleatoria del espacio de los patrones de entrada. En todos los ejemplos hemos utilizado 600 patrones de entrada que se introducen de forma aleatoria en el proceso de aprendizaje.

Como primer ejemplo, vamos a ver cual es el esqueleto que nos proporciona la red cuando le suministramos patrones procedentes del número 3.

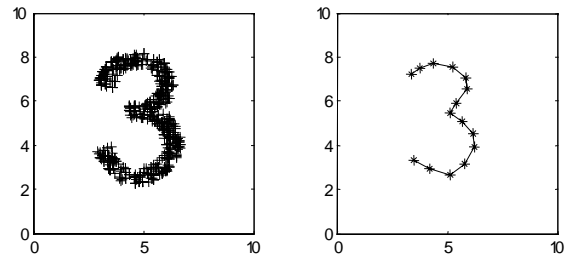


Figura 8. Resultado de la red SOFM para el número 3

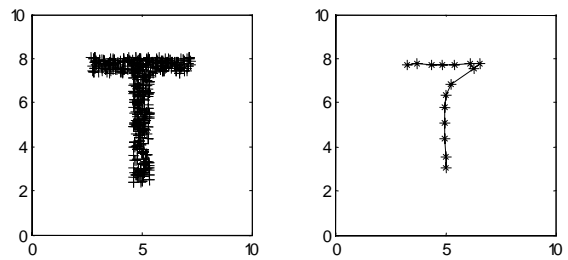


Figura 9. Resultado de la red SOFM para la letra T

En la figura 8 podemos ver tanto la distribución de los patrones de entrada como el resultado de la red después del proceso de aprendizaje. Podemos apreciar que el esqueleto proporcionado por la red es aceptable y adecuado, ya que se preservan perfectamente las características topológicas del conjunto de patrones de entrada.

Vamos a ver ahora cuál es el esqueleto que nos proporciona la red cuando le suministramos patrones procedentes de la letra T. En la figura 9 podemos ver tanto la distribución de los patrones de entrada como el resultado de la red después del proceso de aprendizaje. Podemos apreciar que el esqueleto resultante no mantiene la información topológica correcta que sin embargo en la figura 8 sí era capaz de mantener.

Probemos la red también con patrones procedentes del número 8 y de la letra K. Podemos ver los resultados respectivamente en las figuras 10 y 11 tanto para la distribución de los patrones de entrada como el resultado de la red después del proceso de aprendizaje.

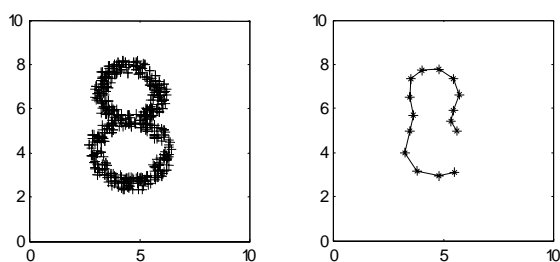


Figura 10. Resultado de la red SOFM para el número 8

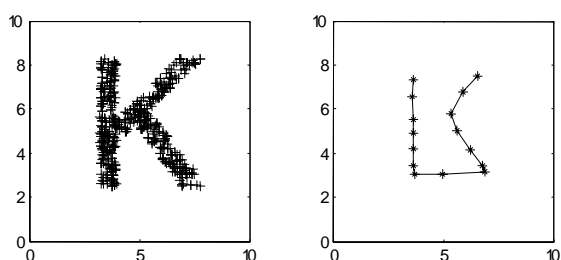


Figura 11. Resultado de la red SOFM para la letra K

Podemos comprobar también que para el número 8 y para la letra K el esqueleto proporcionado no es capaz de mantener la información topológica de los patrones de entrada.

3.3. Inconvenientes de la red SOFM para la detección de esqueletos y mejoras propuestas

En las redes autoorganizadas de Kohonen (Kohonen, 1997), el conjunto de neuronas que constituyen la red tienen una topología rígida y fija de vecindad entre ellas que no se puede modificar durante la fase de aprendizaje. Esto puede crear problemas en muchos casos. Por ejemplo, cuando el conjunto de patrones de entrada están mayormente concentrados en una determinada zona del espacio de entrada, puede ocurrir que los vectores sinápticos de algunas unidades queden, después del proceso de aprendizaje, en zonas donde no haya ningún patrón de entrada, ya que han sido atraídos por otros patrones que están en zonas periféricas a la de mayor concentración (Kangas et al., 1990).

El problema descrito lo podemos apreciar en la figura 11, donde vemos como ha quedado una unidad en la parte inferior del esqueleto y no hay, sin embargo, ningún patrón en dicha zona.

Datta, Pal y Parui (1997), proponen una modificación de las redes SOFM de Kohonen para la detección de esqueletos. La modificación radica en que durante la fase de aprendizaje se añaden mecanismos que permitan añadir y suprimir neuronas en la red con un mecanismo *supervisado*, cambiando de esta forma la topología inicial establecida de una forma dinámica según se vayan presentando los patrones de entrada. El método conserva las reglas de actualización de pesos sinápticos propuestas por Kohonen para las unidades ganadoras y sus vecinas topológicas. Hay otros autores que han propuesto modificaciones similares para otros problemas, como el problema de la cuantificación vectorial (Choi y Park, 1994), el de estimación de distribuciones de probabilidad en el plano (Fritzke, 1991) y para clasificación de figuras (Sabourin y Mitiche, 1993).

Para la detección de esqueletos de caracteres, Datta, Pal y Pauri (1997) hacen una distinción de los mismos agrupándolos en tres tipos:

- Patrones con arcos simples: la topología de tales caracteres se puede representar mediante simples estructuras lineales. Dentro de este grupo se encuentran entre otras las letras C, L, N, S y los números 1, 2, 3, 5, y 7. Las redes de Kohonen obtienen por sí mismas esqueletos para este grupo de caracteres, como hemos podido observar en la figura 11.
- Patrones con bifurcaciones: son caracteres que tienen algún tipo de cruce, bifurcación o unión. Dentro de este grupo se encuentran entre otras las letras K, T, X, e Y. En este

tipo de caracteres es obvio que no se puede obtener la topología mediante simples estructuras lineales. Es necesario ir partiendo la estructura de vecindad entre las neuronas, añadiendo nuevas unidades en las bifurcaciones. En la figura 12 podemos apreciar el resultado que obtiene la técnica para la letra X (imagen obtenida del trabajo del autor).

- Patrones con bucles: son caracteres que contienen algún bucle (o cierre). En este tipo de caracteres encontramos entre otras las letras A, B, D, P y los números 4, 6, 8, 9 y 0. Es necesario (además de posiblemente encontrarnos bifurcaciones como el apartado anterior) hacer conexiones entre neuronas ya existentes, para así poder cerrar el bucle. En la figura 13 podemos apreciar el resultado que obtiene la técnica para la letra A (imagen obtenida del trabajo del autor).

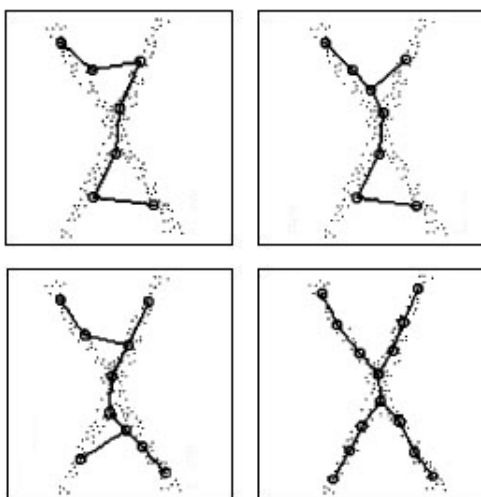


Figura 12. Resultado de la red para la letra X

Como este proceso, una vez hecha la clasificación de los caracteres en tres grupos, se puede ahora obtener un esqueleto que mantenga las características geométricas y topológicas del carácter.

Sin embargo, este método no es genérico. Primero hay que ver en que grupo, de los tres citados, se encuadra el carácter en estudio y hay que ir controlando el aprendizaje de la red, añadiendo nuevas neuronas y enlaces que permitan el cambio de la de la topología original de la misma.

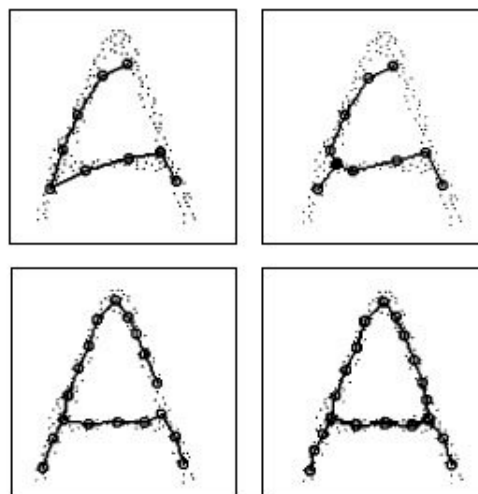


Figura 13. Resultado de la red para la letra A

4. Detección de esqueletos con la red competitiva basada en segmentos con fase de expansión

Vamos a mostrar que, con una adecuada elección de la constante u que gobierna la función del parámetro de expansión β de la ecuación (1), somos capaces de generar segmentos pequeños que se extiendan a lo largo de las figuras de las que suministramos los patrones de entrada.

Al terminar la fase de aprendizaje, la red nos proporcionará los segmentos estructurales obtenidos, es decir, nos aporta los dos pesos sinápticos (dipolos) que determinan los extremos de dichos segmentos. Visualmente este conjunto de segmentos aproxima al esqueleto de la imagen, salvo que los segmentos no están unidos.

Para obtener un trazo continuo que determine el esqueleto de la imagen, basta con ir uniendo los segmentos obtenidos de forma que enlazaremos aquellos extremos que se encuentren más cerca, es decir que tengan una menor distancia euclídea.

Vamos a explicar la importancia que tiene la adecuada elección de la constante u que controla el parámetro de expansión. Las imágenes que vamos a utilizar están constituidas por 600 patrones y como en nuestro proceso de aprendizaje los introducimos 4 veces, vamos a tener un total de $T = 2400$ iteraciones.

Si el parámetro de expansión β alcanzase el valor cero mucho antes de terminar el número total de iteraciones T , permitiríamos a los segmentos crecer

libremente y por tanto veremos que no se alinean a lo largo de la imagen, ya que todos ellos compiten y se estiran libremente. Sin embargo, si no permitimos que el parámetro de expansión β tome el valor cero hasta justamente antes de terminar el proceso de aprendizaje, tendremos controlado el tamaño de los segmentos y se posicionarán a lo largo de la estructura de la imagen suministrada.

Pongamos un ejemplo que nos ayude a ilustrar la afirmación anterior. Vamos a ver como evoluciona una red con 8 unidades (8 segmentos) con dos funciones distintas que gobiernen el parámetro de aprendizaje.

En ambas comparaciones vamos a obtener los patrones de entrada de una imagen que tiene 600 patrones distribuidos uniformemente a lo largo del número 2.

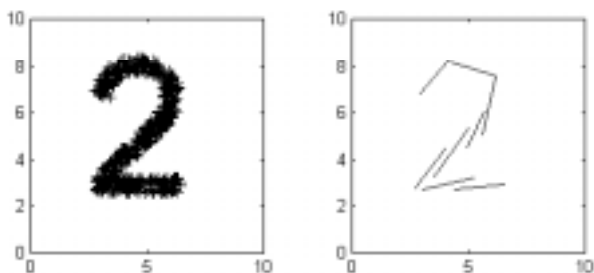


Figura 14. Resultado de la red con $u = 0.05$

En la figura 14 podemos apreciar el resultado de la red cuando usamos la función para el parámetro de expansión mostrada en la figura 15 con un valor de $u = 0.05$.

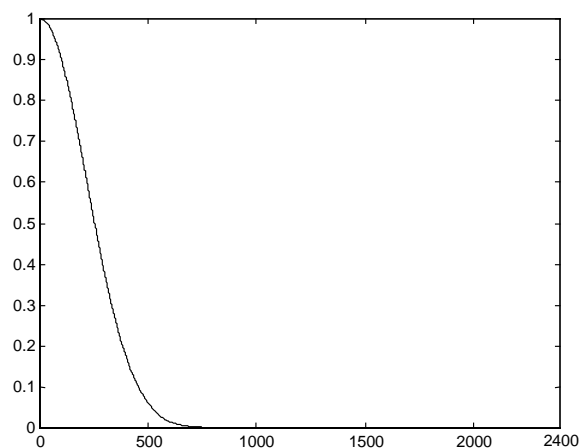


Figura 15. Parámetro de expansión con $u = 0.05$

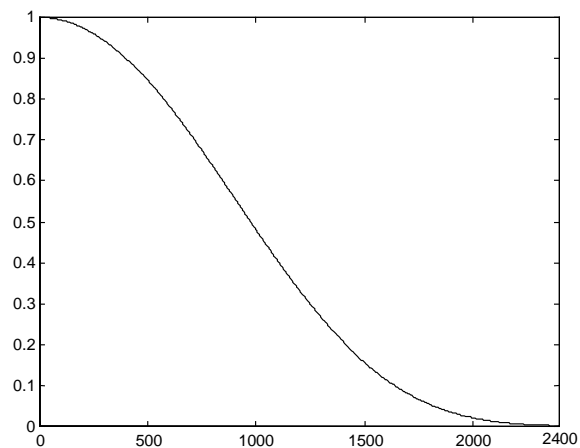


Figura 16. Parámetro de expansión con $u = 0.003$

En la figura 17 podemos apreciar el resultado de la red cuando usamos la función para el parámetro de expansión mostrada en la figura 16 con un valor de $u = 0.003$.

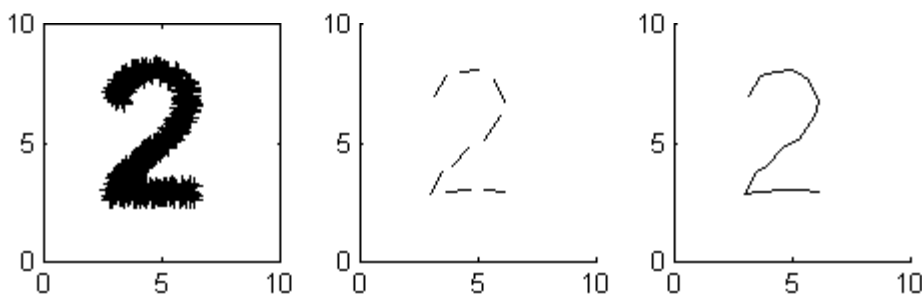


Figura 17. Resultado de la red con $u = 0.003$ y 8 segmentos

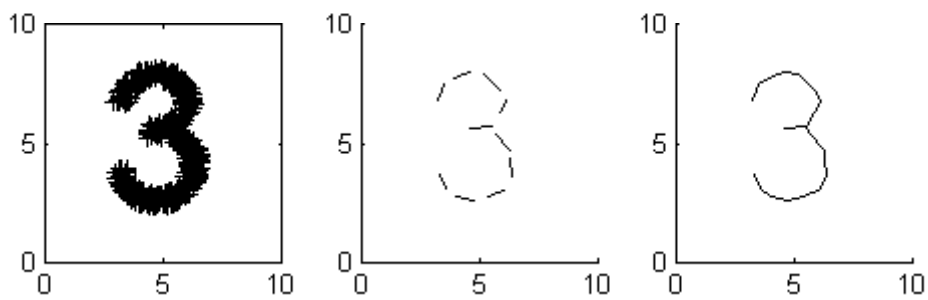


Figura 18. Resultado de la red para el número 3 con 10 segmentos

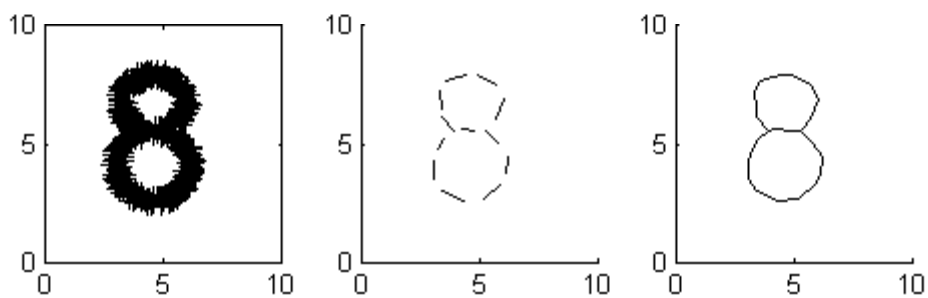


Figura 19. Resultado de la red para el número 8 con 12 segmentos

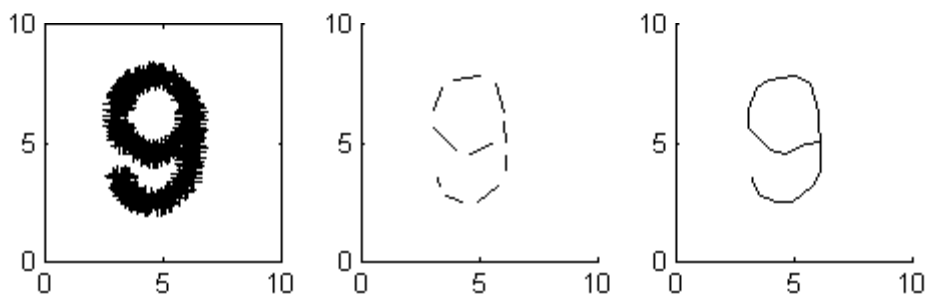


Figura 20. Resultado de la red para el número 9 con 10 segmentos

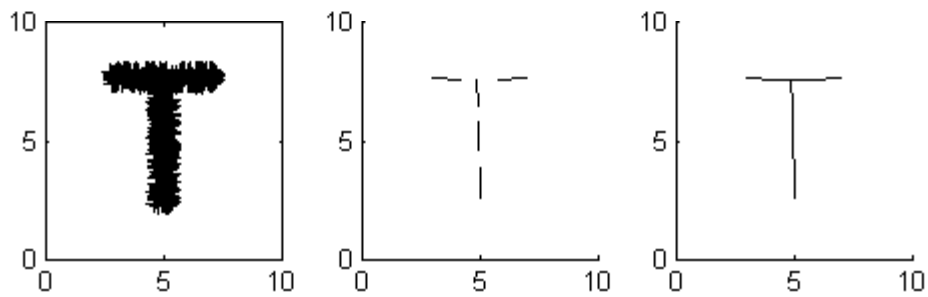


Figura 21. Resultado de la red para la letra T con 5 segmentos

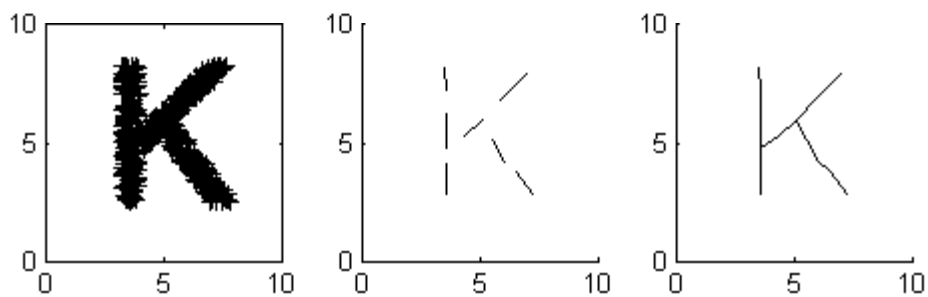


Figura 22. Resultado de la red para la letra K con 7 segmentos

Para obtener el esqueleto de las figuras, basta con unir los segmentos obtenidos por la red cuyos extremos estén a una distancia euclídea menor o igual que una dada, obteniéndose de esta forma el esqueleto de la imagen. Por ejemplo el esqueleto de la figura 17 se obtiene uniendo los segmentos que se encuentren a una distancia menor o igual a 0.9.

Puesto que todas las imágenes que vamos a utilizar tienen las mismas características que la anterior y el mismo número de patrones, utilizaremos para todas ellas la función para el parámetro de expansión mostrada en la figura 16 con $u = 0.003$.

Vamos a suministrarle a la red caracteres que se recojan en los tres grupos que hacen Datta, Pal y Pauri (1997) en su trabajo: los números 3, 8 y 9 y las letras T y K. En resultado de la red para dichos caracteres los podemos ver respectivamente en las figuras 18, 19, 20, 21 y 22.

Para obtener los esqueletos de los caracteres 3, 8, 9, T y K, unimos respectivamente los segmentos obtenidos por la red que se encuentren a una distancia euclídea menor igual a 0.9.

5. Conclusiones

En este artículo se ha propuesto una red neuronal competitiva no supervisada basada en segmentos con fase de expansión. Hemos visto cómo la red expuesta sin la fase de expansión es capaz de proporcionar segmentos que nos indican la dirección predominante de los datos. Sin embargo es muy sensible a la inicialización de los pesos sinápticos de las neuronas y al orden de introducción de los patrones.

Al añadir un nuevo parámetro que gobierna la fase de expansión de los segmentos, se proporciona un nuevo mecanismo que permite que los segmentos crezcan con la rapidez deseada y de manera controlada, haciendo que los resultados de la misma no dependan ni de la inicialización de los pesos sinápticos ni del orden de introducción de los patrones, posibilitando que los segmentos se concentren en los distintos grupos donde están situados los patrones del espacio de entrada.

Se ha conseguido una regla de aprendizaje no supervisado que utiliza un potencial sináptico que es una función lineal de las entradas de la red y que permite la formación de clases o categorías y determina los segmentos que mejor las representan.

Hemos mostrado como las redes autoorganizadas de Kohonen (SOFM) no son adecuadas para la

obtención de esqueletos de caracteres, debido a que el conjunto de neuronas que constituyen la red tienen una topología fija de vecindad entre ellas que no se puede modificar durante la fase de aprendizaje.

No obstante, hemos estudiado una modificación de las redes SOFM hecha por los autores Datta, Pal y Pauri (1997) que permiten obtener los esqueletos de cualquier carácter. Estos autores clasifican los caracteres en tres grupos según su topología. La red que proponen va cambiando dicha topología durante la fase de aprendizaje dependiendo del grupo en el que se encuentre el carácter en cuestión. Sin embargo, esta topología va cambiando de una forma supervisada mediante la adición o sustracción de nuevas neuronas y nuevos enlaces.

Con la nueva red competitiva no supervisada basada en segmentos con fase de expansión desarrollada en este trabajo, hemos comprobado que, con una adecuada elección de la constante u que gobierna la función del parámetro de expansión β de la red, somos capaces de obtener, de una forma totalmente no supervisada, el esqueleto de cualquier carácter sin hacer ningún tipo de distinción ni agrupación de los mismos según su topología.

Referencias

- Anderson, E.: The IRISes of the gaspe Peninsula. Bulletin of the American IRIS society, 59:2-5, 1939.
- Amin, A., Al-sadoun, H., Fischer, S.: Hand-printed arabic carácter recognition system using an artificial network. Pattern Recognition, 29(4):663-675, 1996.
- Bayro-Corrochano, E.J., Pahn, D.T.: Self-organizing neural-network-based pattern clustering method with fuzzy outputs. Pattern Recognition, 27:1103-1110, 1994.
- Blum, H.: A transformation for extracting new descriptions of shape. IEEE Proceedings of the Symposium on Models for the Speech and Vision Form. pp. 362-380, Boston, 1964.
- Cardoner, R., Thomas, F.: Residuals + directional gaps = skeletons. Pattern Recognition Letters, 18:343-353, 1997.
- Choi, D., Park, S.: Self-creating and organizing neural networks. IEEE Neural Networks, 5:561-575, 1994.
- Corridoni, J.M., Bimbo, A., Landi, L.: 3D Object classification using multi-object Kohonen networks. Pattern Recognition, 29:919-935, 1996.
- Datta, A., Pal, T., Parui, S.K.: A modified self-organizing neural net for shape extraction. Neurocomputing, 14:3-14, 1997.

- Datta, A., Parui, S.K., Chaudhuri, B.B.: Skeletonization by a topology-adaptive self-organizing neural network. *Pattern Recognition*, 34:617-629, 2001.
- Fritzke, B.: Self-organizing feature maps with problem dependent cell structure. *Artificial Neural Networks*, 1:403-408, 1991.
- Gader, P., Forester, B., Ganzberger, M., Gillies, A., Mitchell, B., Whalen, M., Yocum, T.: Recognition of handwritten digits using template and model matching. *Pattern Recognition*, 24(5):421-431, 1991.
- Glaubergerman, M.H.: Character recognition for business machines. *Electronics*, 132-136, 1956.
- García Bernal, M.A: Modelos de Neurocomputación Competitiva para el Descubrimiento de Estructuras. Tesis doctoral. Universidad de Málaga, 2001.
- Gómez Ruiz, J.A: Descubriendo Estructuras en los Datos Mediante Aprendizaje Competitivo y Reproductivo. Tesis doctoral. Universidad de Málaga, 2002.
- Gómez Ruiz, J.A., López Rubio, E., Muñoz Pérez, J.: Invariant pattern identification by self-organising networks. *Pattern Recognition Letters*, 22:983-990, 2001.
- Haykin S.: *Neural Networks: A Comprehensive Foundation*. Prentice Hall. New Jersey. 1999.
- Kangas, J.A., Kohonen, T., Laaksonen, J.: Variants of self-organizing maps. *IEEE Neural Networks*, 1:93-99, 1990.
- Kohonen, T.: *Self-organizing Maps*. Springer. Berlin. 1997.
- Linde, Y., Buzo, A., Gray, R.M.: An algorithm for vector quantizer design. *IEEE Trans. On Communication*, 28(1):84-95, 1980.
- McQueen, J.: Some methods for classification and analysis of multivariable observations. V Berkeley Symp. On Mathematical Statistics and Probability, pp. 281-297, Berkeley, 1967.
- Obermayer, K., Ritter, H., Schulten, K.: Development and spatial structure of cortical feature maps: A model study. *Advances in Neural Information Processing Systems*, 3:11-17, 1991.
- Ramesh, S.R.: A generalized character recognition algorithm: a graphical approach. *Pattern Recognition*, 22(4):347-350, 1989.
- Ritter, H., Martinez, T., Schulten, K.: *Neural Computacion and Self-Organizing Maps: An Introduction*, Reading, MA: Addison-Wesley, 1992.
- Sabourin, M., Mitiche, A.: Modeling and classification of shape using a Kohonen associative memory with selective multiresolution. *Neural Networks*, 6:275-283, 1993.
- Subba Reddy, N.V., Nagabhushan, P.: A three-dimensional neural network model for unconstrained handwritten numeral recognition: a new approach. *Pattern Recognition*, 31:511-516, 1998.
- Suen, C.Y., Legault, R., Nadal, C., Cheriet, M., Lam, L.: Building a new generation of handwriting recognition systems. *Pattern Recognition Letters*, 14:303-315, 1993.
- Taxt, T., Olafsdóttir, J.B., Daehlen, M.: Recognition of handwritten symbols. *Pattern Recognition*, 23(11):1155-1166, 1990.
- Trier, O.D., Jain, A.K., Taxt, T.: Feature extraction methods for character recognition: a survey. *Pattern Recognition*, 29(4):641-662, 1996.
- Wang, s.s., Lin, W.G.: A new self-organizing neural model for invariant pattern recognition. *Pattern Recognition*, 29:677-687, 1996.
- Westall, J.M., Narasimha, M.S.: Vertex directed segmentation of handwritten numerals. *Pattern Recognition*, 26:1473-1486, 1993.