

# EL USO DE LAS REDES NEURONALES ARTIFICIALES EN LA DETECCIÓN DE PUNTOS DOMINANTES

Aurora Pons<sup>1</sup>, José M. Iñesta<sup>2</sup>, Reynaldo Gil<sup>1</sup>, José M. Sanchíz<sup>3</sup>, Roxana Danger<sup>1</sup>

<sup>1</sup>Departamento de Computación, Universidad de Oriente, Cuba,

<sup>2</sup>Departamento de Lenguajes y Sistemas Informáticos, Universidad de Alicante, España,

<sup>3</sup>Departamento de Informática, Universidad Jaume I, España,

E-mail: <sup>1</sup>{aurora,gil,roxana}@csd.uo.edu.cu, <sup>2</sup>inesta@dlsi.ua.es, <sup>3</sup>sanchiz@inf.uji.es

## RESUMEN

La búsqueda de aquellos puntos imprescindibles para preservar la morfología de los contornos y que, a su vez, reducen la cantidad de información que los caracteriza es una de las tareas más importantes en el análisis de imágenes. Este proceso se denomina Detección de Puntos Dominantes (DPD). No existe ningún algoritmo de DPD que pueda considerarse el óptimo, por lo que su selección depende del objetivo que se persigue. En el trabajo se estudia la capacidad de las redes neuronales para aprender el funcionamiento de diversos algoritmos de DPD. Se presenta, además, un algoritmo de DPD en cadenas de contornos cerrados, el cual usa una red neuronal dinámica que recorre el contorno dando una medida de la relevancia de cada punto y un simple procesamiento adicional detecta los puntos dominantes. Se utilizaron distintos conjuntos de entrenamiento a partir de los algoritmos de DPD estudiados y se entrenaron diferentes configuraciones de red. Los resultados presentados utilizando imágenes reales de objetos naturales demuestran la validez del algoritmo propuesto, así como la capacidad de las redes neuronales artificiales para aprender algoritmos de DPD previamente desconocidos.

**Palabras claves:** Redes neuronales, Detección de esquinas, Curvatura, Redes neuronales dinámicas, Análisis de imagen, Detección de características.

## 1. INTRODUCCIÓN

Las aproximaciones poligonales son usadas con frecuencia, para reducir los datos asociados a una curva. La Detección de Puntos Dominantes es la operación de búsqueda de aquellos puntos de mayor importancia para el contorno, de forma que preserve su morfología. Estos puntos dominantes pueden ser determinados a través de dos enfoques: por medio de la búsqueda de los puntos de mayor curvatura en su entorno local o a través de la aproximación de una poligonal a la curva. En nuestro trabajo nos dedicaremos al primero de estos enfoques.

En [San96] se demostró la capacidad de las Redes Neuronales Artificiales (RNA) para aprender el funcionamiento de un algoritmo de DPD [Mel93]. En dicho trabajo se comprobó que una red neuronal dinámica del tipo *Time Delay Neural Network* (TDNN) [Her91] era capaz de recorrer un contorno y dar una coincidencia de hasta el 85% en la detección de puntos dominantes con respecto al funcionamiento del algoritmo original, tras una sesión de aprendizaje a la que la red se le informaba de los puntos detectados por dicho algoritmo. Sería de gran importancia demostrar la capacidad de las redes para aprender estos algoritmos y generalizar el concepto que permita emular el comportamiento de otros desconocidos, pues de esta manera, se podrían desarrollar algoritmos adecuados a determinados propósitos, y la red podría aprenderlos,

ejecutándolos en tiempo no dependiente de la complejidad del algoritmo diseñado. Además, con la implementación de la red por hardware la considerable reducción del tiempo permitiría su inclusión en sistemas de procesamiento de imágenes en tiempo real.

El presente trabajo intenta generalizar los resultados anteriores para varios algoritmos de DPD pertenecientes al primer enfoque y demostrar la capacidad de la RNA de aprender dichos algoritmos. Para ello fueron estudiados cinco algoritmos: Rosenfeld-Johnston (R-J) [Ros73], Rosenfeld-Weszka (R-W) [Ros75], Freeman-Davis (F-D) [Fre77], Teh-Chin (T-Ch) [Teh89] y Non-Colinear Dominant Points (NCDP) [Iñe98].

En la sección 2 de este trabajo se expone cómo pueden extenderse los algoritmos de DPD estudiados al caso de contornos abiertos. En la sección 3 se analiza en detalle el algoritmo propuesto por nosotros para la detección neuronal de los puntos dominantes. La sección 4 aborda los resultados obtenidos en el aprendizaje de las RNA de la detección de puntos dominantes y, finalmente, en la sección 5 se dan las conclusiones de nuestro trabajo.

## **2. DETECCIÓN DE PUNTOS DOMINANTES EN CURVAS ABIERTAS**

Una gran cantidad de algoritmos proveen, como salida, curvas abiertas; de manera que no solamente contornos cerrados son susceptibles a utilizar algoritmos de DPD. Los algoritmos de DPD citados anteriormente fueron aplicados solamente en curvas cerradas. Para utilizar ambos tipos de contornos (abiertos y cerrados) en estos algoritmos, hemos diseñado una forma de aplicarlos a curvas abiertas, pero manteniendo invariante su naturaleza.

Los algoritmos estudiados necesitan la determinación de una *región de soporte* para cada punto de la curva. La región de soporte es la zona de la curva que se tiene en cuenta para calcular una medida local de la curvatura discreta, que se utilizará para discriminar los puntos dominantes. Algunas veces se considera una cantidad fija de puntos; en otras, su longitud varía en dependencia de las características locales de cada porción del contorno. Esta última forma de determinación de curvatura es utilizada por los algoritmos Freeman-Davis, Teh-Chin y NCDP. Los algoritmos Rosenfeld-Johnston y Rosenfeld-Weszka, aunque también utilizan regiones de soporte de longitud variable, necesitan un número fijo de puntos a analizar antes y después de cada punto en el contorno.

Si trabajamos con contornos abiertos es imposible satisfacer la existencia de la región de soporte para los primeros y últimos puntos del contorno. Una posible solución es expandir artificialmente la curva manteniendo su suavidad en aquellos puntos para los que no pueda ser determinada la región de soporte. Para esto, proponemos la reflexión de la curva con respecto a los puntos extremos, en la cantidad de puntos que sea necesaria para cada algoritmo. De esta forma, la región de soporte siempre es calculable para todos los puntos del contorno original y los algoritmos trabajarán sin realizar cambio alguno en sus procedimientos. Obviamente, el primer y último punto del contorno original son siempre etiquetados como dominantes.

En la sección 4.1 mostramos los resultados experimentales obtenidos con esta aproximación.

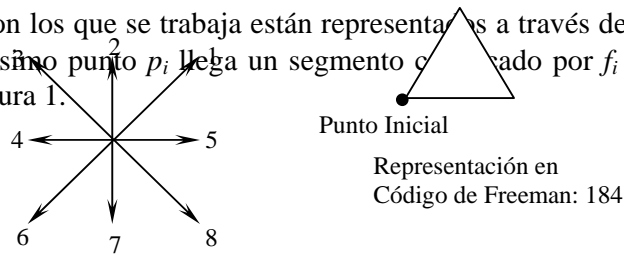
## **3. DETECCIÓN NEURONAL DE PUNTOS DOMINANTES**

Para la detección neuronal de puntos dominantes se diseñó una arquitectura basada en las RNA dinámicas del tipo TDNN [Her91, Can96], para las que se demuestra que en cada instante de tiempo  $t$  se comportan como un perceptron multicapas y, por tanto, son aplicables sus métodos de aprendizaje para el ajuste de los pesos [Sta92, Thi95, Loo96]. En este caso, el aspecto

dinámico de la red es el recorrido de la curva y el parámetro “temporal” es la posición sobre la curva conforme la red se va desplazando a lo largo de ella.

Para cada punto del contorno se toman sus  $s$ -vecinos anteriores y posteriores. Esta porción del contorno constituye un patrón de entrenamiento. La RNA multicapas tiene una capa de entrada de  $2s+1$  neuronas que representa la región de soporte del punto central del patrón de entrenamiento, una capa oculta con un número de neuronas por determinar y una capa de salida con una neurona. La RNA recibe como entrada los códigos de Freeman modificados, que se explicarán más adelante, de los puntos de la región de soporte y será entrenada para obtener una medida de la curvatura del punto central del patrón de entrenamiento. A este valor de la curvatura se le aplica un sencillo proceso adicional para determinar si el punto puede o no ser considerado como dominante, desde el punto de vista de la red neuronal.

Los contornos con los que se trabaja están representados a través de los códigos de Freeman de 8 vecinos (al  $i$ -ésimo punto  $p_i$  llega un segmento codificado por  $f_i \in \{0, 1, \dots, 7\}$ ), tal como se muestra en la figura 1.



**Figura 1. Codificación de Freeman. Ejemplo de uso.**

Por ello, para facilitar el entrenamiento de la red, es necesario un cambio de codificación que permita mantener las entradas en el rango  $[0,1]$  y sea independiente de la orientación del contorno. Se propone una nueva codificación  $a_i$ , calculada como se muestra en la tabla 1.

$d_i = f_{i+1} - f_i$ Si $d_i > 4$ entonces $d_i = 8 - d_i$ Si $d_i < -4$ entonces $d_i = 8 + d_i$			
$d_i$	$a_i$	$d_i$	$a_i$
-4	0	1	0.675
-3	0.125	2	0.75
-2	0.25	3	0.875
-1	0.375	4	1
0	0.5		

**Tabla 1. Transformación del código de Freeman.**

Para el entrenamiento de la red es necesario, además, calcular las salidas deseadas en cada patrón de entrenamiento. Para ello, aplicamos un algoritmo de DPD y obtenemos para cada punto de las curvas si es o no dominante. Supongamos que se obtienen  $M$  puntos dominantes  $\{dp_1, \dots, dp_M\}$ ,  $dp_j \in \{1, \dots, N\}$  indica el orden en el contorno del punto dominante  $j$ , donde  $j=1, \dots, M$  y  $N$  es el tamaño del contorno. Tal como se comprobó en [San96], el aprendizaje de la RNA se facilita esparciendo a su región de vecindad la influencia de cada punto dominante detectado por los algoritmos clásicos de DPD. Para un punto  $i$ , sea  $d_{i,min} = \min \{d_{i,dp_1}, \dots, d_{i,dp_M}\}$  la distancia de  $i$  al punto dominante más próximo. Cada distancia  $d_{i,dp_j}$  puede calcularse como la norma del vector que une las coordenadas  $(x,y)$  del punto  $i$  y del punto dominante  $dp$ , o, simplificando los cálculos, la diferencia de orden en el contorno entre las posiciones  $i$  y  $dp_j$ .

El espaciado de los puntos se puede realizar mediante distintas aproximaciones. Por ejemplo, mediante una señal triangular con vértice en el punto:  $t_i = \begin{cases} 1 - \frac{d_{i,min}}{v+1} & \text{si } d_{i,min} \leq v+1 \\ 0 & \text{en otro caso} \end{cases}$ , donde  $t_i$

es la salida deseada presentada a la red para el  $i$ -ésimo punto y  $v$  es la distancia máxima permitida en el análisis del punto.

En [Pon98] se explica otra posibilidad para el cálculo de las salidas deseadas mediante la utilización de una función gaussiana centrada en el punto dominante. Sin embargo, no se detectaron diferencias significativas entre uno y otro método y, por razones de eficiencia, en el presente trabajo se emplea la señal triangular como función de espaciado de los puntos.

#### ***Algoritmo de preparación del conjunto de entrenamiento:***

Para conformar el conjunto de entrenamiento se aplica sobre los contornos cerrados que conforman el conjunto de entrada, el algoritmo que deseamos que la red aprenda. Así pues,

Para cada contorno del conjunto de entrada:

1. Aplicar el algoritmo de DPD seleccionado, encontrando los puntos dominantes.
2. Calcular la salida deseada  $t_i$  en cada punto del contorno de acuerdo con la función seleccionada de espaciado de los puntos.
3. Aplicar la traslación del código de Freeman del contorno analizado al código de entrada a la red, obteniéndose la señal  $A = \{a_1, \dots, a_N\}$ .
4. Para cada valor  $a_i$  de la señal obtenida formar un patrón de entrenamiento, tomando como entrada  $\{a_{i-s}, \dots, a_i, \dots, a_{i+s}\}$  y como salida deseada el valor de  $t_i$ .

Una vez preparado el conjunto de entrenamiento se procede a entrenar la RNA utilizando el algoritmo de retropropagación del error (*Backpropagation*) [Tre95], usando como función de error:

$$E_{mc} = \frac{1}{2N_p} \sum_{i=1}^{N_p} (t_i - O_i)^2$$

donde  $N_p$  es el número de patrones del conjunto de entrenamiento,  $t_i$  es la salida deseada para el punto  $p_i$  y  $O_i$  la salida de la neurona de salida de la red para el patrón de entrenamiento centrado sobre el punto  $p_i$ .

La aplicación del algoritmo de DPD mediante la RNA puede describirse de la siguiente forma:

**Algoritmo de DPD neuronal:**

1. Trasladar el código de Freeman de un contorno cerrado  $C = \{ f_1, f_2, \dots, f_N \}$  al código de entrada a la red  $A = \{ a_1, a_2, \dots, a_N \}$  como fue explicado anteriormente, donde  $N$  es el número de puntos del contorno.
2. Activar la red para cada segmento del código de la cadena:  $\{ a_{i-s}, a_{i-s+1}, \dots, a_i, \dots, a_{i+s} \}$ , obteniéndose la señal  $B = \{ b_1, \dots, b_N \}$ , que indica la estimación que la red hace de la curvatura local para cada punto del contorno.
3. Aplicar un umbral  $K_0$  de curvatura a la señal  $B$ , obteniéndose una nueva señal  $C = \{ c_1, \dots, c_N \}$  de la siguiente manera: si  $b_i < K_0$  entonces  $c_i = 0$  sino  $c_i = b_i$ .

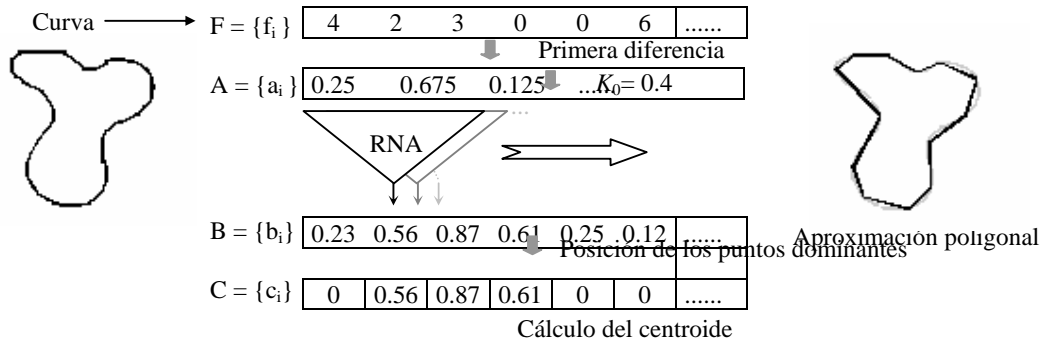
Esta señal es considerada como una cadena formada por segmentos donde  $c_i > 0$ .

4. Calcular la posición centroide  $\bar{c}_j$  de cada segmento y marcar el punto del contorno más

cercano a esta posición como punto dominante. Así,  $\bar{c}_j = \frac{\sum_{i=ini_j}^{fin_j} i \cdot c_i}{\sum_{i=ini_j}^{fin_j} c_i}$ , donde:  $ini_j$  y  $fin_j$  son las

posiciones de inicio y fin del segmento  $j$ .

La representación gráfica de todos estos pasos se muestra en la figura 2. En ella se identifican sus tres etapas fundamentales: el pre-procesamiento, el análisis neuronal y el post-procesamiento. Los parámetros que pueden ser fijados son la arquitectura de la red y el valor del umbral  $K_0$ .



**Figura 2. Pasos involucrados en la detección neuronal de puntos dominantes.**

Nosotros hemos encontrado una posible configuración de la red para enseñar cualquier algoritmo de DPD basado en el cálculo de las medidas de relevancia local, que analizaremos en detalle más adelante.

### 3.1. Medidas de Error.

Es importante conocer cuándo ambos: el algoritmo de DPD y la red neuronal, están realizando buenas aproximaciones. Para medir la calidad de los algoritmos de DPD se utilizó el Error de Optimización [Iñe95], pues da una medida de la aproximación a la morfología de la curva con la reducción de información realizada. Esta medida se define como:  $E_0 = \frac{E_{ci} n_d}{N^2}$ , donde:  $N$  es la cantidad de puntos del contorno,  $n_d$  es la cantidad de puntos dominantes detectados por el algoritmo,  $E_{ci} = \sum_{i=1}^N e_i^2$ , donde  $e_i$  es la distancia del punto  $i$ -ésimo del contorno al segmento de la aproximación poligonal que une a los puntos dominantes anterior y posterior a él. Esta medida fue utilizada para seleccionar los mejores parámetros de cada uno de los algoritmos de DPD que serían enseñados a la RNA.

Necesitamos, además, identificar en qué medida las redes aprenden a seleccionar bien los puntos dominantes, o sea, cuán cerca están los puntos dominantes determinados por la red de los puntos correspondientes determinados por el algoritmo. Para medir esto, en [Pon98] se utilizó el error de optimización. Esta selección no es buena, puesto que no se desea encontrar la red que minimice el error de optimización, sino aquella que aprenda mejor el comportamiento de un determinado algoritmo de DPD. En este trabajo, hemos definido una medida de error que controla las diferencias existentes entre los puntos dominantes detectados por los algoritmos clásicos y el algoritmo de detección neuronal. A esta medida de error le llamaremos *error de detección* y la definiremos de la siguiente manera:

$$E = \frac{\sum_{i=1}^m E_i}{m}, E_i = \sum_{j=1}^{nr_i} dra_j + \sum_{j=1}^{na_i} dar_j, \text{ donde:}$$

$m$  es la cantidad de contornos del conjunto de comprobación.

$nr_i$  es la cantidad de puntos dominantes detectados por la red en el contorno  $i$ -ésimo del conjunto de comprobación.

$na_i$  es la cantidad de puntos dominantes detectados por el algoritmo en el contorno  $i$ -ésimo del conjunto de comprobación.

$dra_j$  es la distancia del  $j$ -ésimo punto dominante detectado por la red al más próximo dominante detectado por el algoritmo.

$dar_j$  es la distancia del  $j$ -ésimo punto dominante detectado por el algoritmo al más próximo dominante detectado por la red.

Estas dos distancias se calculan como la diferencia de orden en el contorno entre las posiciones de los puntos analizados.

## 4. RESULTADOS

### 4.1. Análisis de las curvas abiertas.

Para probar la eficiencia de la solución propuesta en la detección de puntos dominantes en curvas abiertas, fueron tomados 315 contornos reales de objetos naturales y cinco puntos contiguos en cada uno de ellos fueron eliminados, abriendo, de esta forma, las curvas. Para cada uno de los contornos, en su versión abierta y cerrada, fue ejecutado el algoritmo de DPD clásico. Los errores de optimización fueron calculados en cada caso, considerando solamente la misma porción de los contornos cerrados que fueron usados en los abiertos. Los resultados son mostrados en la Tabla 2.

Para todos los algoritmos examinados, más del 90% de los puntos dominantes encontrados en las curvas abiertas coinciden exactamente con los encontrados en las cerradas. Los errores de optimización en ambos casos son muy similares, mostrando la validez de la solución propuesta.

Algoritmo	$E_o$ (Abierto)	$E_o$ (Cerrado)	% coincidencia
R-J	0.1167	0.1337	91.5
R-W	0.1133	0.1259	93.5
F-D	0.1032	0.0822	95.5
T-Ch	0.0532	0.0603	97.1
NCDP	0.0532	0.0603	97.2

**Tabla 2. Error de optimización medio para contornos abiertos y cerrados y el porcentaje de coincidencia.**

### 4.2. DPD neuronal.

Para la experimentación se usó un conjunto de 30 imágenes reales de objetos naturales, que agrupan 490 contornos y 39354 puntos. Esta muestra fue dividida en dos grupos: uno para el entrenamiento y otro para la comprobación de manera que el primero estuviera formado por, aproximadamente, el doble de la información del segundo y en ambos hubiera representatividad en cuanto a cantidad de puntos por contorno. De esta forma, quedó conformada una muestra de entrenamiento con 315 contornos y 26435 puntos (patrones de entrenamiento) y una de comprobación con 175 contornos y 12919 puntos (patrones de comprobación).

Cabe recordar aquí, que se persigue el objetivo de estudiar qué arquitectura de red se comporta mejor en el aprendizaje de estos algoritmos. Para cada uno de ellos, la red adoptará un conjunto de pesos diferente, y el experimento de generalización que se plantea no es sobre el comportamiento de la red, sino sobre los parámetros que se fijan heurísticamente en su arquitectura (número de neuronas en la capa oculta, etc.) para fijar la configuración que mejor se comporte en este problema.

La mayoría de los algoritmos de DPD necesitan parámetros de entrada para su ajuste adecuado; así pues, antes de entrenar la red fue realizado un estudio para seleccionar los parámetros de cada algoritmo de DPD con los cuales se obtenía un menor *Error de Optimización medio* en la muestra de entrenamiento, resultando escogidos los siguientes parámetros para los algoritmos seleccionados:

- Rosenfeld-Johnston:  $m = N/10$
- Rosenfeld-Weszka:  $m = N/10$
- Freeman-Davis:  $m = 1, s = 5$
- Teh-Chin: con  $k$ -curvatura
- NCDP: con  $k$ -curvatura,  $\delta_u = 0.15$

El número de neuronas de la capa de entrada está normalmente determinado por la naturaleza de los datos del problema. En este caso, este número coincide con el tamaño de la región de soporte, que, en principio, depende del algoritmo. Por consiguiente, este número de neuronas también será un parámetro a introducir en el estudio. Para analizar el comportamiento con distintos tamaños de la región de soporte, se realizó, además, un estudio para cada uno de los algoritmos de DPD variando sus parámetros, obteniéndose que los valores aceptables del número de puntos considerados en la región de soporte varían de 5 a 11. Los resultados obtenidos se muestran en la tabla 3.

Algoritmo	Región de Soporte Promedio	Error de Optimización
R-J	10.2	0.2806
R-W	10.2	0.3034
F-D	5.0	0.1008
T-Ch	5.9	0.0611
NCDP	5.9	0.0625

**Tabla 3. Estudio de la anchura de las regiones de soporte para los algoritmos.**

Por otro lado, decidimos comenzar la investigación sobre el tamaño de la capa oculta teniendo en cuenta las experiencias previas [San96]. La cantidad de neuronas para la capa oculta se varió de 3 a 11. Por todo ello, fueron tomadas en consideración las 15 topologías mostradas en la tabla 4 (I: número de neuronas en la capa de entrada; H: en la capa oculta; O: en la de salida).

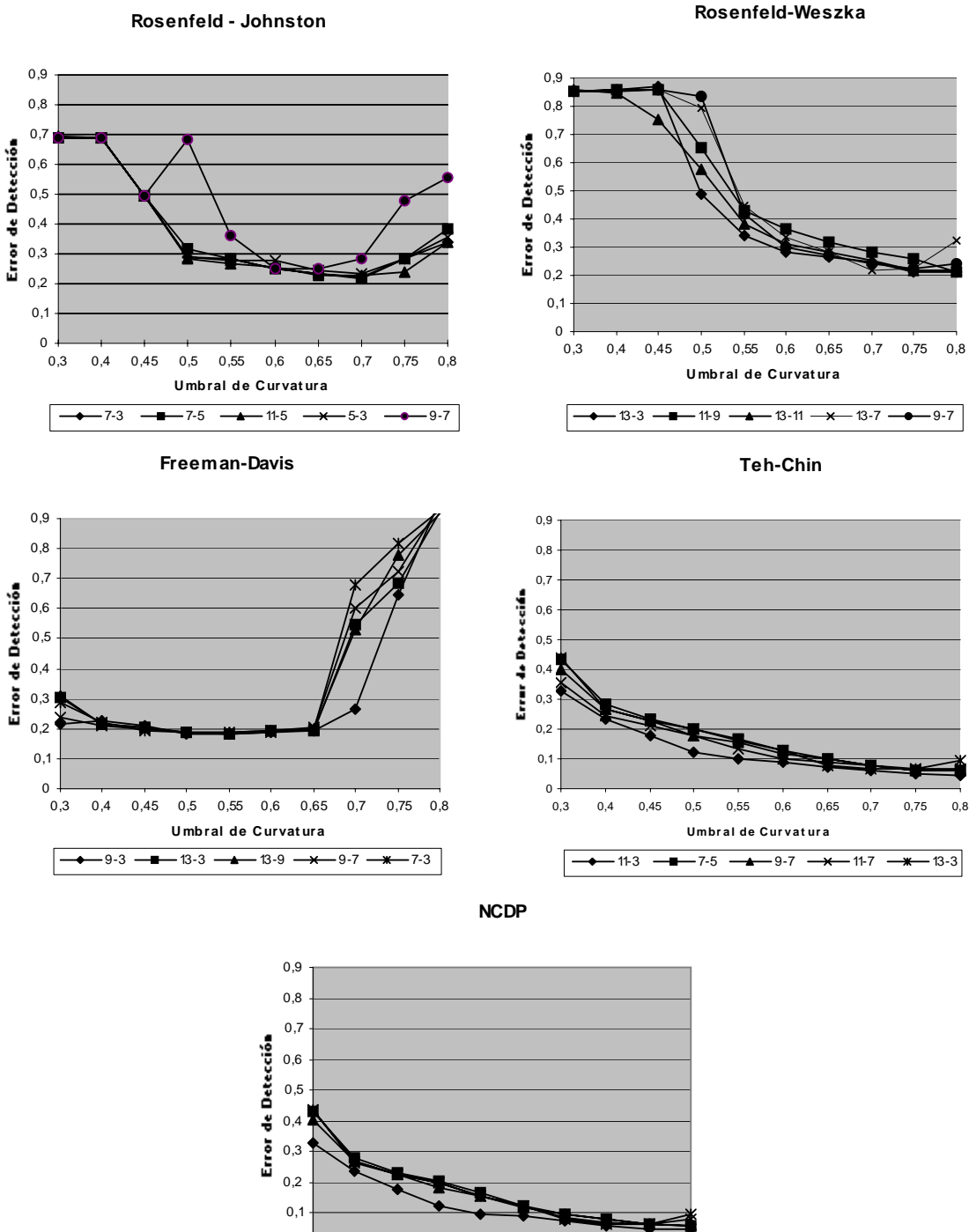
I-H-O	13-11-1	13-9-1	13-7-1	13-5-1
	13-3-1	11-9-1	11-7-1	11-5-1
	11-3-1	9-7-1	9-5-1	9-3-1
	7-5-1	7-3-1	5-3-1	

**Tabla 4. Configuraciones de RNA utilizadas.**

Con los 5 algoritmos de DPD seleccionados y las 15 configuraciones de redes, fueron preparados los 75 conjuntos de entrenamiento. Usando el algoritmo *BackPropagation* clásico con factor de aprendizaje 0.35, las redes convergieron en un promedio de 75 épocas y un error



cuadrático medio entre 0.01 y 0.06. Para valorar la capacidad de generalización de las redes estudiadas fue aplicada la RNA entrenada con cada algoritmo al conjunto de comprobación, y comparados los resultados con los proporcionados por la aplicación directa del algoritmo de DPD correspondiente sobre estos mismos contornos, utilizando el error de detección anteriormente mencionado.



**Figura 3. Comportamiento del error de detección por algoritmo.**

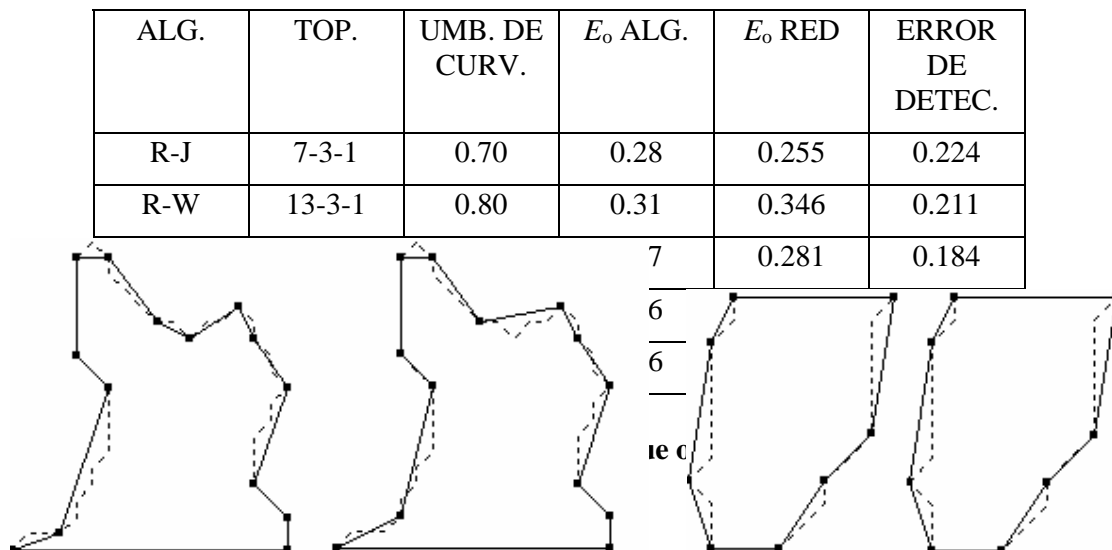
#### 4.3. Aprendizaje de los algoritmos.

Un parámetro que influye en el aprendizaje de la red es el umbral de curvatura  $K_0$  escogido en el paso 3 del algoritmo de detección neuronal. Por ello, fueron seleccionados 9 valores de umbrales (0.3,0.4,0.45,0.5,0.6,0.65,0.7,0.75,0.8), con los cuales fueron hechos todos los análisis.

En todos los casos a partir de un valor del umbral, el error de detección comienza a disminuir más lentamente, sin importar la topología hasta que a partir de otro valor vuelve a aumentar.

En la figura 3 puede verse el comportamiento del error de detección en las 5 mejores topologías de redes para cada algoritmo.

La mejor topología para cada algoritmo se muestra en la Tabla 5. Puede verse además que, a excepción de los algoritmos Teh-Chin y NCDP, las mejores topologías seleccionadas fueron diferentes, lo cual indica que no existe a priori una arquitectura privilegiada de cara a la segunda parte del estudio. La excepción anteriormente citada puede ser perfectamente admisible teniendo en cuenta la similitud entre ambos algoritmos.



**Figura 4. Ejemplo del comportamiento de la red después del aprendizaje del algoritmo (las imágenes de la izquierda corresponden al algoritmo y las de la derecha, a la red).**

En la figura 4 se muestra el comportamiento de la red 11-3-1 después del aprendizaje del algoritmo Teh-Chin, con umbral 0.75 para dos contornos de la muestra de comprobación. Como puede verse la red detecta unos puntos dominantes muy cercanos a los obtenidos con el algoritmo.

#### 4.4 Búsqueda de una arquitectura generalizada.

Queremos demostrar que si una topología aprende  $n$  algoritmos es capaz de aprender uno más ( $n+1$ ). Para ello, aplicamos la siguiente prueba de validación cruzada:

Se tomaron cada vez 4 de los 5 algoritmos analizados, dejando uno fuera, y en cada caso se realizaron los siguientes pasos:

1. Se entrenó cada una de las 15 topologías de red con los 4 algoritmos seleccionados.
2. Se aplicó el algoritmo de detección neuronal para cada una de esas  $15 \times 4$  redes neuronales al conjunto de comprobación variando el valor del umbral  $K_0$  en los 9 valores seleccionados.
3. En cada una de las variantes topología-umbral anteriores se halló el *error de detección* de la muestra de comprobación para cada uno de los 4 algoritmos (en total son  $15 \times 9 \times 4$ ) y, luego, se calculó el *error de detección medio* por cada una de las 15 variantes de topología en su mejor valor de umbral.
4. Se entrenó la red neuronal, cuyo *error de detección medio* fue el mejor, con el algoritmo que quedó fuera, y se aplicó el algoritmo de detección neuronal para el conjunto de comprobación usando el mejor umbral.

Los resultados obtenidos en esta prueba se muestran en la tabla 6. Las últimas tres columnas representan el valor del error de detección y los errores de optimización obtenidos por el algoritmo de DPD y por la red neuronal para el algoritmo de validación, respectivamente.

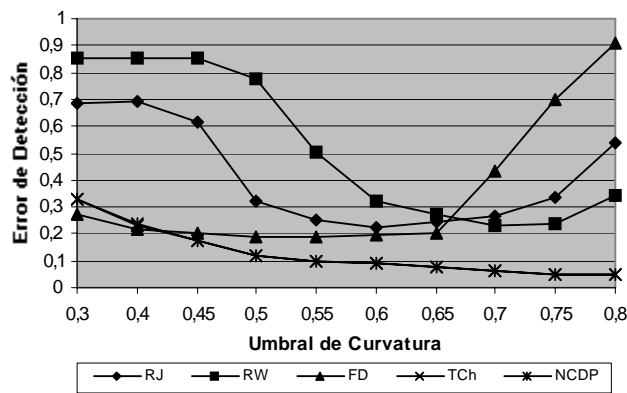
Algoritmos usados para seleccionar la mejor arquitectura	Mejor Topología	Algo. de validación	Error de detección	$E_o$ Algoritmo	$E_o$ Red
RW, FD, TCh, NCDP	11-3-1	RJ	0.13	0.28	0.35

RJ, FD, TCh, NCDP	11-3-1	RW	0.13	0.31	0.23
RJ, RW, TCh, NCDP	11-3-1	FD	0.14	0.17	0.32
RJ, RW, FD, NCDP	13-3-1	TCh	0.17	0.06	0.07
RJ, RW, FD, TCh	13-3-1	NCDP	0.17	0.06	0.07

**Tabla 6. Resultados de la validación cruzada.**

Como se muestra en esta tabla en todos los casos la red fue capaz de aprender el algoritmo que quedó fuera, pues el *error de detección* obtenido para el algoritmo de validación en la mejor topología para los cuatro restantes es bajo. Además los errores de optimización obtenidos por el algoritmo de DPD y por el algoritmo de detección neuronal son aceptables.

Finalmente, se aplicaron los mismos pasos de la prueba de validación cruzada pero, ahora, para encontrar la topología que fuera capaz de aprender mejor los 5 algoritmos estudiados, resultando la topología 11-3-1. Las curvas de error obtenidas para todos los algoritmos con esta configuración de red se muestran en la figura 5.



**Figura 5. Curvas de error de detección en la topología 11-3-1.**

## 5. CONCLUSIONES

En este trabajo se ha presentado un estudio de generalización acerca de la capacidad de las redes neuronales artificiales multicapas para aprender el comportamiento de los algoritmos de detección de puntos dominantes en curvas digitales por búsqueda de máximos locales de la curvatura local.

Del análisis de los resultados del experimento de validación cruzada se puede concluir que un perceptrón multicapas con la topología 11-3-1, entrenado mediante el algoritmo de retropropagación del error, tiene capacidad para aprender cualquier algoritmo de DPD, al menos si éste se basa en el enfoque de detección de puntos de alta curvatura; demostrando, de esta

forma, la capacidad de generalización de la RNA para la tarea de detección de puntos dominantes.

En el futuro se prevee incorporar al estudio otros algoritmos de DPD, así como entrenar otros modelos de RNA para la Detección de Puntos Dominantes, en la búsqueda de arquitecturas más potentes o de algoritmos de aprendizaje más rápidos. La consecución de una arquitectura fiable en el aprendizaje de estos algoritmos permitirá la detección de puntos dominantes mediante cualquier algoritmo (el que se adapte a las necesidades de cada problema) en tiempo no dependiente de la complejidad del mismo y abre la puerta a la paralelización o a la implementación *hardware* de cualquier procedimiento de detección, sea cual sea su complejidad.

## 6. AGRADECIMIENTOS

Este trabajo ha sido financiado en parte por el proyecto de investigación de la Fundación Bancaja P1A96-18.

## 7. REFERENCIAS

- [Can96] Cancellé, R. & Gemello, R. "Efficient training of Time Delay Neural Networks for sequential patterns". *Neurocomputing*, vol. 10 (1), pp. 33-42, 1996.
- [Fre77] Freeman, H. & Davis, L.S. "A corner-finding algorithm for chain-coded curves". *IEEE Trans. Comput.*, vol. C-26, pp. 297-303, 1977.
- [Her91] Hertz, J., Krog, A. & Palmer, R.G. "Introduction to the Theory of Neural Computation". Addison-Wesley, 1991.
- [Iñe98] Iñesta, J.M., Buendía, M. & Sart, M.A. "Reliable Polygonal Approximations of Imaged Real Objects Through Dominant Point Detection". *Pattern Recognition*, vol. 31(6), pp. 685-699, 1998.
- [Loo96] Looney, G. "Stabilization and Speedup of Convergence in Training Feedforward Neural Networks". *Neurocomputing*, vol. 10 (1), pp. 7-31, 1996.
- [Mel93] Melen T & T. Ozanian. "A fast algorithm for dominant point detection on chain-coded contours". In *Proc. of the 5<sup>th</sup> Int. Conf. on the Computer Analysis of Image and Patterns*. Budapest, Sept. 1993. Lecture Notes on Computer Science, Springer-Verlag.
- [Pon98] Pons, A., Gil, R., Danger, R., Sanchíz, J.M., Iñesta, J.M. "A generalisation study on neural corner detection", In *Progress in Artificial Intelligence, Lecture Notes series in Artificial Intelligence, 1484*, Helder Coelho (ed.) Springer Verlag, Octubre de 1998, pp. 385-396.
- [Ros73] Rosenfeld, A. & Johnston, E. "Angle detection on digital curves", *IEEE Trans. Comput.*, vol. C-22: 875-878, 1973.
- [Ros75] Rosenfeld, A. & Wezcka, J.S. "An improved method of angle detection on digital curves". *IEEE Trans. Comput.*, vol. C-24: 940-941, 1975.
- [San96] Sanchíz, J.M., Iñesta, J.M. & Pla, F. "A Neural Network Algorithm to Detect Dominant Points from the Chain-code of a Closed Contour". In *Proc. of 13<sup>th</sup> International Conference on Pattern Recognition, ICPR'96*, Vienna, Austria. IEEE Computer Society Press, vol. IV, pp. 325-329, 1996.

- [Sta92] Starita, A. & Sperduti, A. "Speedup Learning and Network Optimization with Extended Back Propagation". Technical Report. Università Degli Studi Di Pisa. Dipartimento di Informatica, October 1992.
- [Teh89] Teh, C. & Chin, R.T. "On the Detection of Dominant Points on Digital Curves". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. vol. 2 (8), pp. 859-872, 1989.
- [Thi95] Thimm, G. & Fiesler, E. "Neural Network Initialization". *Proceedings of International Workshop on Artificial Neural Networks*. Málaga-Torremolinos, Spain, June 1995, pp. 535-542.
- [Tre95] Trejo, L.A. & Sandoval, C. "Improving Back-Propagation: Epsilon-Back-Propagation". *Proceedings of International Workshop on Artificial Neural Networks*. Málaga-Torremolinos, Spain, June 1995. pp. 427-432.