

# Reconocimiento facial utilizando Redes de contrapropagación

**Juan Moreno**

jmoreno@inf-cr.uclm.es  
Universidad de Castilla-La Mancha  
13071 Ronda de Calatrava nº5, Ciudad Real  
España

**Miguel A. Fernandez**

miki@info-ab.uclm.es  
Universidad de Castilla-La Mancha  
02071 Avenida de España s/n, Albacete  
España

**Francisco J. Gomez**

fgomez@info-ab.uclm.es  
Universidad de Castilla-La Mancha  
02071 Avenida de España s/n, Albacete  
España

**A. Fernandez Caballero**

caballer@info-ab.uclm.es  
Universidad de Castilla-La Mancha  
02071 Avenida de España s/n, Albacete  
España

## Abstract

Este artículo presenta el funcionamiento de las redes de contrapropagación para el reconocimiento facial. Básicamente el procedimiento que se sigue es convertir la imagen en un vector de números y pasarlas a la capa de entrada de la red. Como vectores de entrada hemos probado vectores de niveles de gris de imágenes y vectores de histogramas por filas de las imágenes. Estas redes obtienen un resultado excelente superando el 80% en ambos casos.

## 1 Introduction

Este artículo pretende comprobar la eficiencia de las redes de contrapropagación (CPN) para el reconocimiento facial. Para comprobar la red se ha implementado un simulador que tiene un proceso de aprendizaje por medio del cual aprende 16 rostros, y posteriormente recibe como entrada variaciones de cada uno de los rostros aprendidos, proporcionando como salida un código que representa a uno de los rostros aprendidos.

Hecht-Nielsen [Hecht-Nielsen, 1990] sintetizó la CPN combinando una estructura conocida como red competitiva con la estructura outstar de Grossberg [Grossberg, 1982; Hecht-Nielsen, 1987a; Hecht-Nielsen, 1987b], obteniendo así lo que se llama Red de Contrapropagación. El funcionamiento de la red se muestra en la figura 1. Dado un conjunto de vectores  $(x_1, y_1), (x_2, y_2), \dots, (x_L, y_L)$ , la red puede aprender a asociar a un vector  $X$  en la capa de entrada con un vector  $Y$  en la capa de salida. Si la relación entre  $X$  e  $Y$  se puede definir mediante una función continua,  $\Omega$ , tal que  $Y = \Omega(X)$ , entonces la red aprenderá a aproximar esa correspondencia para todo valor de  $X$  en el intervalo especificado por el conjunto de vectores de entrenamiento.

Como se ve en la figura 1 la arquitectura consta de tres capas. Las unidades de la capa 1 reciben como entrada un rostro, que es procesado como un vector de niveles de gris, y normaliza dicho vector. Todas las unidades de la capa 2 reciben el mismo vector normalizado de entrada, y calculan su intensidad de salida gracias a un vector de pesos aprendido y al vector que reciben como entrada. La unidad de mayor intensidad de salida envía un 1 a la capa 3, el resto envía un 0. Finalmente, la capa 3 selecciona la imagen de salida de entre las aprendidas.

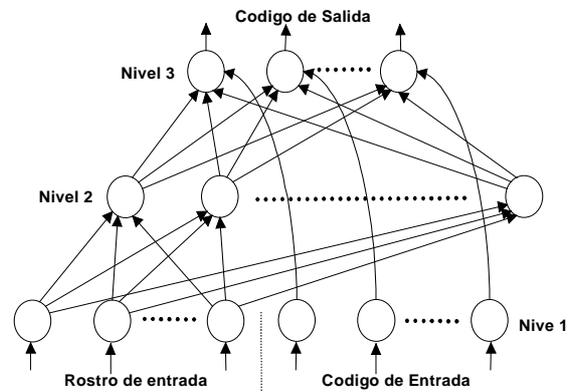


Figura 1 Estructura de la CPN

## 2 Descripción de la CPN

Consideremos la capa 1 de la figura 1. La intensidad total del vector de entrada está gobernada por  $I = \sum_i I_i$ . En correspondencia con cada  $I_i$ , definiremos una magnitud mostrada en la ecuación 2.1. El vector  $(v_1, v_2, \dots, v_n)^t$  es lo que se denomina trama de reflectancia [Freeman and

Skapura, 1991]. Observese que esta trama está normalizada, es decir que  $\sum \partial_i = 1$ .

$$\vartheta = I_i \left( \sum_i I_i \right)^{-1} \quad (2.1)$$

La trama de reflectancia es independiente de la intensidad total de la trama de entrada correspondiente. Por ejemplo, la trama de reflectancia correspondiente a la imagen de la cara de una persona es independiente de si la imagen de la persona se tuviera más o menos luminosidad. La ecuación 2.2 representa la salida de la capa 1, donde  $0 < x_i(0) < B$ ,  $A, B > 0$ .

$$\dot{x}_i = -Ax_i + (B - x_i)I_i - x_i \sum_{k \neq i} I_k \quad (2.2)$$

Los elementos de proceso alcanzan rápidamente un estado de equilibrio una vez que se aplica el vector de entrada X, ( $\dot{x} = 0$ ) [Freeman and Skapura, 1991]. A la hora de la simulación de la CPN, se puede simplificar el programa normalizando por software los vectores de entrada. La simulación normaliza de acuerdo a la ecuación 2.3.

$$x_i = I_i / \left( \sqrt{\sum_n I_n^2} \right) \quad (2.3)$$

La capa 2 se conoce con el nombre de red competitiva. Consta de una serie de elementos de proceso llamados instar [Grossberg, 1982; Freeman and Skapura, 1991]. El vector de entrada I, y el vector de pesos w se han normalizado. La salida de la instar está gobernada por la ecuación 2.4, donde  $neta = I * w$  y  $a, b > 0$ .

$$\dot{y} = -ay + b * neta \quad (2.4)$$

Las unidades instar alcanzan el valor de equilibrio cuando  $y^{eq} = (b/a) * neta$ . A la hora de la simulación los valores a y b han tomado el mismo valor, luego se eliminan de la formula. La red competitiva clasifica cualquier vector de entrada. La instar de mayor salida es la ganadora de la competición, y será la única que tendrá salida no nula. La ganadora enviará a las outstar el valor 1, y el resto de instar de la red competitiva enviará el valor 0.

$$\dot{y}'_i = -ay'_i + c W_i^{eq} \quad (2.5)$$

La outstar alcanza rápidamente un valor de equilibrio igual al valor del peso que haya en la conexión procedente de la unidad ganadora de la red competitiva. Una forma sencilla de visualizar este procesamiento consiste en darse cuenta de que la salida de equilibrio de la outstar es igual a la ecuación 2.6, donde  $z_j$  es la entrada que se recibe de la correspondiente instar de la Red Competitiva.

$$y_k^{eq} = \sum_j w_{kj} z_j \quad (2.6)$$

Además, dado que  $z_j = 0$  a no ser que  $j=i$ , la ecuación 2.7 es la salida de la outstar.

$$y_k^{eq} = w_{ki} z_i = w_{ki} \quad (2.7)$$

En la simulación la salida de la outstar se obtiene mediante la ecuación 2.7.

Como ya se ha comentado antes, nuestro sistema realiza una fase de aprendizaje, en la cual la CPN se aprende la serie 1 de los 16 rostros de las figuras 2 y 3. A continuación se le pasan todos los rostros de las figuras 2 y 3. Posteriormente el sistema devuelve como resultado un código que representa cada una de las imágenes aprendidas. En nuestra aplicación hemos utilizado imágenes de tamaño 112 filas y 92 columnas, con 256 niveles de gris. Finalmente el esquema de nuestra CPN es el siguiente:

- La Capa 1 consta de 112\*92 nodos en el vector de entrada x, que corresponde a la longitud del vector de entrada. Con respecto a las entradas y, tenemos 4 nodos correspondientes a los 4 valores binarios de que se compone el código de cada uno de los rostros a aprender.
- La capa oculta, la red competitiva, está formada por 16 nodos instar, donde cada instar se aprende un rostro.
- La capa 3 o capa de salida, consta de 4 nodos, cada una de las cuales se aprende un número binario que permite identificar uno de los componentes del código de salida. La salida de esta capa será el rostro seleccionado.

Hemos elegido la red de contrapropagación para implementar este sistema ya que son buenas para la detección de vectores, ya que en [Moreno et al, 1998] se obtenían unos buenos resultados, además hemos de tener en cuenta que cuanto mayor es el número de coordenadas del vector mejor repartidas pueden estar las clases dentro del espacio. Las redes de contrapropagación emplean algoritmos de aprendizaje distintos para cada capa, y que permiten a la red entrenarse muy rápidamente.

Consideramos que una agenda con 16 rostros es una muestra significativa para comprobar la validez de la red de contrapropagación para este sistema, aunque nos consta que con más rostros la CPN tendría un comportamiento peor.

Para terminar este apartado comentar que para cada valor de entrada  $i$  se ha elegido una intensidad de entrada al sistema de  $i+1$ , para que cuando  $i=0$  se proporcione algún valor de entrada distinto de 0.

### 3 Aprendizaje de la CPN

El aprendizaje de la CPN se realiza por capas. La capa 1 no tiene aprendizaje, ya que tan solo normaliza los valores de entrada. Como ya se dijo la capa 2 esta formada por unos elementos llamados instar. La instar se aprende el vector de pesos  $w$ . El aprendizaje [Moreno et al, 1998] se realiza a partir del vector de pesos inicial  $w$ , que va evolucionando según la ecuación diferencial 4.1, en donde  $y$  es la salida, y  $c, d > 0$ .

$$\dot{w} = -cw + dly \quad (4.1)$$

La misión de una instar es aprenderse un vector de entrada  $y$ , proporcionando una mayor intensidad de salida cuando más se parezca el vector de entrada al vector aprendido. A la hora de la simulación es posible simplificar el aprendizaje de la instar, asignando directamente los valores de los pesos del vector  $w$  a partir de vector  $y$ , ya normalizados.

Finalmente, el aprendizaje [Moreno et al, 1998] de cada outstar de la capa 3 evoluciona según la ecuación 4.2, donde los parámetros  $a, b, c > 0$  y el valor de la neta, se calcula de la forma ya descrita.

$$\dot{y}'_i = -ay'_i + by_i + c * neta_i \quad (4.2)$$

Al efectos de la simulación digital, se puede aproximar este aprendizaje asignando directamente los pesos de la outstar a partir de los vectores de entrada  $y$ .

### 4 Resultados

Primeramente damos las gracias por la cortesía de *The ORL Database of Faces* ([www.cam-orl.co.uk/face database.html](http://www.cam.ac.uk/face database.html)) por el uso público de estas imágenes de las figuras 2 y 3 que se han utilizado para nuestras pruebas.

Para la verificación de resultados se ha implementado un simulador de la CPN. El simulador realiza previamente la fase de aprendizaje de las diversas capas. Y a continuación envían a la capa de entrada de ambas redes, uno a uno, todos los rostros a testear (figuras 2 y 3 de las series 1 a 16).

Se han realizado dos tipos de pruebas: (a) pasando las imágenes como un vector de niveles de gris, (b) pasando la imagen como un vector de histogramas por filas de la imagen [Moreno et al, 1999]. Hemos elegido histogramas por filas para solucionar el problema que tiene la CPN cuando el rostro no aparece exactamente en la misma posición de la imagen que en la imagen aprendida.

Los resultados obtenidos para los casos (a) y (b) son muy buenos, superando en ambos casos el 80%.

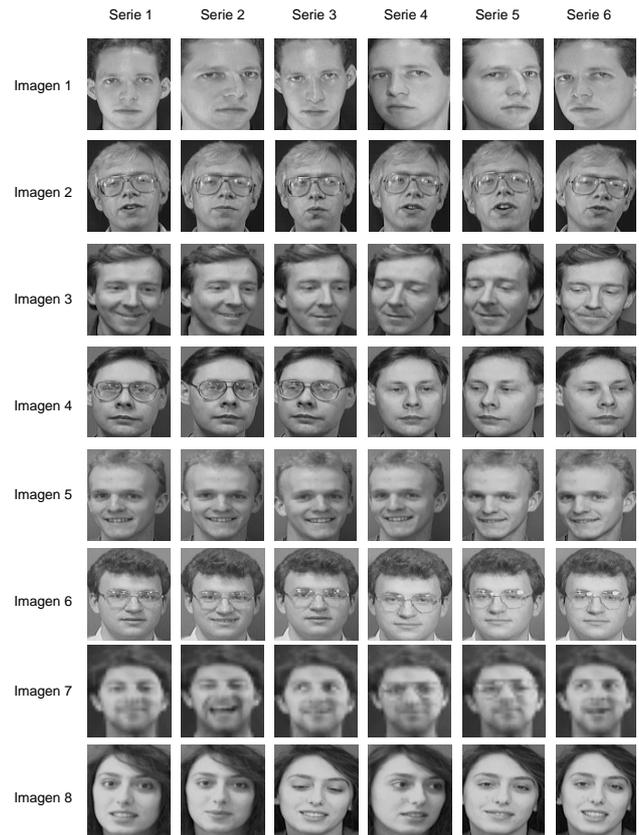


Figura 2 Series de rostros (1-8)

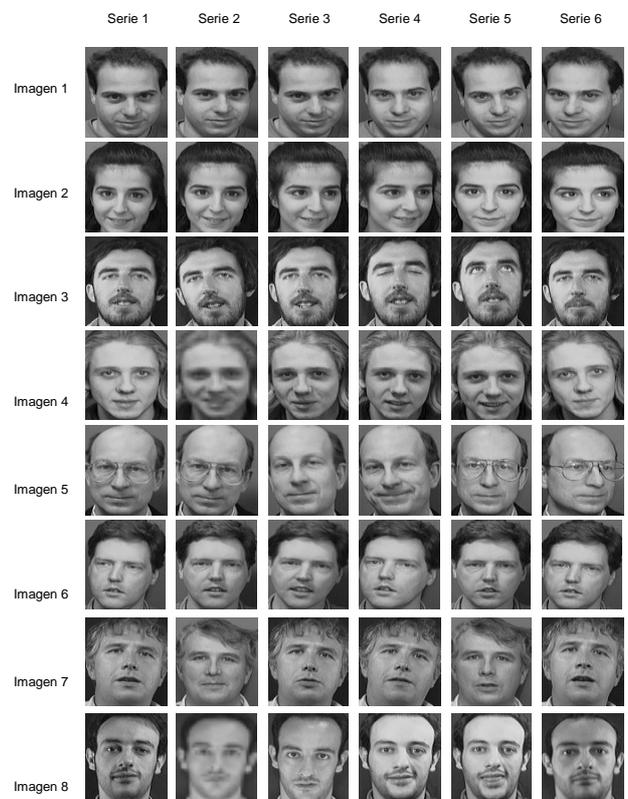


Figura 3 Series de rostros (series 9-16)

La tabla 1 muestra las imágenes acertadas para el caso (a). El acierto global es del 83.3%. Uno de los problemas detectados, pese a que la normalización que realiza la capa 1, es cuando la imagen aprendida tiene una luminosidad distinta a la imagen aprendida, ejemplos de esto son las imágenes 1 (series 2 y 4), 15 (series 2 y 5) y 16 (series 2 y 3). Otro problema, pero menos importante, es la presencia o ausencia de gafas, por ejemplo en la imagen 4 (series 5 y 6), aunque en esta misma imagen, para la serie 4 la ausencia de gafas no supone problema, por otro lado las series 7 y 13 esto no supone ningún problema.

	Series 1	Series 2	Series 3	Series 4	Series 5	Series 6
I-1	✓		✓		✓	✓
I-2	✓	✓	✓	✓	✓	
I-3	✓	✓	✓	✓		✓
I-4	✓	✓	✓	✓		
I-5	✓	✓	✓	✓	✓	✓
I-6	✓	✓	✓	✓	✓	✓
I-7	✓	✓	✓	✓	✓	✓
I-8	✓	✓	✓	✓	✓	✓
I-9	✓	✓	✓	✓	✓	
I-10	✓	✓				✓
I-11	✓	✓	✓			✓
I-12	✓	✓	✓	✓	✓	✓
I-13	✓	✓	✓	✓	✓	✓
I-14	✓	✓	✓	✓	✓	✓
I-15	✓		✓	✓		✓
I-16	✓			✓	✓	✓

Tabla 1. Resultados usando escala de niveles de gris

	Series 1	Series 2	Series 3	Series 4	Series 5	Series 6
I-1	✓	✓	✓	✓	✓	✓
I-2	✓	✓	✓	✓	✓	✓
I-3	✓	✓	✓	✓	✓	
I-4	✓	✓	✓	✓	✓	✓
I-5	✓	✓	✓	✓	✓	✓
I-6	✓	✓	✓	✓	✓	✓
I-7	✓	✓	✓	✓	✓	✓
I-8	✓	✓	✓	✓	✓	✓
I-9	✓	✓	✓			
I-10	✓	✓	✓	✓	✓	✓
I-11	✓	✓	✓	✓	✓	✓
I-12	✓					✓
I-13	✓	✓	✓	✓	✓	✓
I-14	✓	✓	✓	✓	✓	✓
I-15	✓		✓	✓		✓
I-16	✓					✓

Tabla 2. Resultados usando histogramas por fila.

La tabla 2 muestra las imágenes acertadas para el caso (b). El acierto global en este caso es del 86.6%. En esta serie el problema de la presencia o ausencia de gafas está solucionado por completo. Mientras que el cambio de luminosidad entre la imagen aprendida y la de entrada sigue

provocando confusiones en la detección de las imágenes, como se muestra en las imágenes 3 (serie 6), 9 (series 4,5,y 6) y 16 (series 2, 3, 4 y 5).

Finalmente, destacar que estos resultados superan ampliamente otros estudios realizados por nuestro grupo, en concreto los explicados en [Moreno et al, 1999]. En dichos estudios, para el caso (a) se obtenían unos malos resultados, mientras que para el caso (b) se conseguía un 70%, muy bajo en comparación con el 86.6% de la CPN, más aún si tenemos en cuenta que nuestro actual algoritmo de aprendizaje es muy sencillo, y que las imágenes no sufren ningún tipo de preprocesamiento.

## 4 Conclusiones

La CPN es una buena herramienta para el reconocimiento facial, como demuestran los resultados obtenidos, tanto en el caso de tratar la imagen como un vector de niveles de gris, como en el caso de tratar la imagen como un vector de histogramas por filas.

La diferencia de luminosidad entre la imagen aprendida y la testeada puede provocar en ocasiones confusiones, si bien la normalización de la capa 1 evita el problema en muchos casos.

Tratando las imágenes como un vector de histogramas por fila se superan los resultados de caso anterior. Esto se debe a que la entrada es el histograma por fila, y la posición relativa del rostro en la imagen no altera el resultado.

## References

[Hecht-Nielsen, 1990] R. Hecht-Nielsen. *Neurocomputing*. Addison-Wesley, Reading MA, 1990.

[Grossberg, 1982] S. Grossberg. *Studies of Mind and Brain*. Boston Studies in the Philosophy of Science, vol. 70, D. Reidel Publishing Company, Boston, 1982.

[Hecht-Nielsen, 1987a] R. Hecht-Nielsen. *Counterpropagation Networks*. Applied Optics, vol. 26, no. 23, pp. 4979-4984, 1987.

[Hecht-Nielsen, 1987b] R. Hecht-Nielsen. *Counterpropagation Networks*. In Proceedings of the IEEE First International Conference on Neural Networks, Piscataway, NJ, pp. II-19 - II-32, IEEE, 1987.

[Freeman and Skapura, 1991] J.A. Freeman and D.M. Skapura. *Neural Networks, Algorithms, Applications, and Programming Techniques*. Addison-Wesley, Reading, MA, 1991.

[Moreno et al, 1998] Juan Moreno, Gabriel Sebastian, Miguel A. Fernandez and A. Fernandez Caballero. *A Telephone Number Corrector using a Counterpropagation Network*. Proceedings of the Fifth International Conference on Neural Information Processing ICONIP'98, Kitakyushu, Japón, October, 1998.

[Moreno et al, 1999] Juan Moreno, Francisco J. Gomez, Miguel A. Fernandez, A. Fernandez Caballero. *Reconocimiento de Rostros utilizando Secuencias de Histogramas como tramas Espacio-Temporales*. Actas del IV Simposio Iberoamericano de Reconocimiento de Patrones SIARP'99, La Habana, Cuba, Marzo, 1999.