

# Memoria auto-asociativa y clasificación con redes neuronales recurrentes

Enrique Mérida Casermeiro, Domingo López Rodríguez

Departamento de Matemática Aplicada

E.T.S. Ingeniería Informática

Universidad de Málaga

Campus de Teatinos, s/n 29071 Málaga

{merida,dlopez}@ctima.uma.es

## Resumen

El objetivo de este trabajo es presentar un nuevo modelo general de aprendizaje hebbiano para redes recurrentes discretas y dar una interpretación a la función de energía resultante cuando el número de patrones básicos sobrepasa la capacidad de la red. En este sentido, el modelo aquí presentado sostiene que aunque la capacidad de la red debe ser interpretada como una limitación de la red para aprender patrones individuales, *cuando se sobrepasa dicha capacidad, la red tiende a identificar patrones similares formando grupos en torno a representantes típicos*. Así, lo que en realidad se produce es el aprendizaje no supervisado de clases de patrones. De esta forma mediante la generalización del aprendizaje hebbiano aquí mostrado, pueden aprenderse grupos de patrones (conceptos) en lugar de patrones individuales y esto ocurre generalmente cuando se sobrepasa la capacidad de la red.

Nuestro modelo presenta dos grandes ventajas sobre los algoritmos de clasificación típicos: no es necesario el ajuste de ningún parámetro, e incluso el numero de clases de patrones se aprende automáticamente, de forma que no hace falta ningn tipo de conocimiento *a priori* sobre el número de clases.

## 1. Introducción

En 1949, Hebb [6], introdujo un método de aprendizaje fisiológico basado en el refuerzo de

la interconexión entre neuronas. Lo explicaba con las siguientes palabras:

Cuando el axón de la célula A está suficientemente cerca como para excitar a una célula, B, y repetidamente o persistentemente toma parte en activarla, entonces tiene lugar algún proceso de crecimiento o cambio metabólico, en una o ambas células, de forma que la eficiencia de A, como célula que activa a B, se ve incrementada.

Las redes recurrentes (como la bipolar desarrollada por J. J. Hopfield en 1982 [7], o su versión analógica [8]), utilizan este tipo de método de aprendizaje y permiten recuperar patrones en función de su similitud, pero tienen el problema de su baja capacidad.

El parámetro de capacidad  $\alpha$  se define normalmente como el cociente entre el máximo número de patrones que se necesita memorizar en la red y el número de neuronas que se deben usar para obtener una probabilidad de error aceptable (generalmente  $p_{\text{error}} = 0,05$ ) a la hora de recuperarlos. Se ha determinado que para el modelo bipolar de Hopfield (modelo BH) esta constante tiene un valor aproximado de  $\alpha = 0,15$ .

Este valor significa que, para memorizar  $K$  patrones, serán necesarias más de  $\frac{K}{\alpha}$  neuronas para conseguir una probabilidad de error inferior o igual a  $p_{\text{error}}$ . Equivalentemente, si la red está formada por  $N$  neuronas, el número

máximo de patrones que puede memorizar la red (considerando la restricción ya impuesta sobre el error cometido), es  $K < \alpha N$ .

Recientemente, en [13] se usa una red recurrente para evitar la aparición de patrones espúreos, mediante el uso de patrones aumentados (un método válido también para la red de Hopfield). De esta forma, los patrones se recuperan de forma correcta si están lo suficientemente separados.

Este trabajo nos muestra que, cuando se sobrepasa la capacidad de la red, o si los patrones están muy próximos entre sí, los mínimos de la función de energía tienden a solaparse, esto es, varios patrones fundamentales forman un único mínimo local. Este hecho, que se puede considerar como una limitación de la red como memoria asociativa, puede explicar por otra parte la forma en la que el cerebro humano forma conceptos: se asocian varios patrones similares a un representante típico común (parecido a la cuantificación vectorial), sin hacer distinción entre sus características particulares.

Obviamente, se necesitan suficientes muestras para generalizar y no distinguir sus características diferenciales. Si existen pocas muestras de alguna clase, la red será todavía capaz de recuperarlas de forma individual, es decir, funcionando como una memoria asociativa.

## 2. Extensiones del modelo de Hopfield y modelos multivaluados

En la literatura, se han propuesto diversas variantes al modelo estándar de Hopfield. Por ejemplo, G. Galán propuso el modelo OCHOM (ver [2, 3, 4]), consistente en  $N \times L$  neuronas binarias dispuestas en una malla rectangular. En cada iteración, sólo una neurona por fila puede estar en el estado 1, siendo esta la que alcance el máximo potencial. De esta forma, existe un mecanismo de competición entre las neuronas de la misma fila a la hora de activarse.

Asimismo, se ha desarrollado algunas potentes generalizaciones del modelo binario a neuronas discretas, donde las posibles salidas de las neuronas pertenecen a un conjunto discreto

de valores. Básicamente, hay cuatro modelos de neuronas multivaluadas tipo Hopfield:

- **Neurona Q-Ising** [20], donde el estado de una neurona viene representado por un número real. La energía de interacción entre dos neuronas viene expresada como una función del producto de estos escalares:

$$E = -\frac{1}{2} \sum_i \sum_j w_{i,j} V_i V_j + \sum_i \theta_i V_i$$

pero ahora los productos  $V_i V_j$  pueden tomar valores diferentes de 1 y -1, como ocurría en el BH.

- **Neurona Phasor** [18, 9], donde los posibles estados de cada neurona son las raíces complejas de la unidad. La interacción entre dos neuronas se escribe en términos de las partes reales del producto de estos números complejos. Así, la función de energía es:

$$E = -\frac{1}{2} \sum_i \sum_j w_{i,j} \operatorname{Re}(V_i V_j)$$

- **Neurona de Potts:** El estado (salida) de una neurona viene dado por un vector  $n$ -dimensional (ver [5, 21]) y la energía de interacción entre dos neuronas es una función del producto escalar de los vectores que representan sus estados:

$$E = -\frac{1}{2} \sum_i \sum_j w_{i,j} \langle V_i, V_j \rangle$$

- **Neurona discreta:** Los modelos MAREN y SOAR utilizan un tipo de neurona cuyo estado es un entero del conjunto  $\{1, 2, \dots, N\}$  (ver [1, 19]) y la interacción entre neuronas viene expresada por la función signo. Por tanto, la función de energía es:

$$E = -\frac{1}{2} \sum_i \sum_j w_{i,j} \operatorname{sign}(V_i - V_j)$$

En este caso, la dinámica de la red sólo

permite transiciones a estados de menor energía, y si una neurona tiene salida  $k$  en el instante  $t$ , sólo podrá tomar uno de los estados  $k - 1$ ,  $k$  o  $k + 1$ , en el instante  $t + 1$ .

Algunos de estos tipos de neuronas multivaluadas se han usado como memoria auto-asociativa ([9, 10, 22, 1]), pero utilizan una representación muy compleja, y no usan apropiadamente la información multivaluada. En este sentido, los resultados obtenidos son similares a los de modelos estándar que utilizan neuronas más simples. El modelo MAREN fue propuesto por Erdem y Ozturk [1] y se usa como memoria asociativa y para resolver problemas de optimización combinatoria. El modelo SOAR fue propuesto por Ozturk y Abut [19], como extensión al modelo MAREN, con el objetivo de realizar clasificación de imágenes.

El modelo de memoria que se expone en este artículo se basa en el modelo neuronal MREM [15, 16], inspirado en la red bipolar de Hopfield. Las neuronas son multivaluadas, y sus salidas son elementos de un conjunto finito  $\mathcal{M} = \{m_1, m_2, \dots, m_L\}$ , que puede ser numérico o cualitativo. La interacción entre neuronas viene expresada por una función real de sus salidas (*función de similitud*), y la dinámica de la red está caracterizada por su función de energía.

Como se menciona en [15, 16], la introducción de esta función de similitud proporciona a la red de un amplio rango de posibilidades a la hora de representar diferentes problemas. Así, utiliza una representación mejor que la de SOAR y MAREN, puesto que en esos modelos, la mayor parte de la información obtenida por la representación multivaluada se pierde con el uso de la función signo, que sólo produce valores en  $\{-1, 0, 1\}$ . Otras redes multivaluadas presentan problemas similares ya que la interacción entre neuronas se expresa por un número perteneciente a un conjunto de bajo cardinal (normalmente  $\{-1, 1\}$  o  $\{0, 1\}$ ) y no se aprovecha de la representación multivaluada.

La gran flexibilidad del modelo MREM permite además una buena representación para

la mayoría de los problemas de optimización combinatoria ([12, 13, 14, 17, 11]) sin necesidad de ajustar ningún parámetro, al contrario que los demás modelos. Esta flexibilidad queda de manifiesto cuando se observa que los cuatro tipos de neuronas mencionadas arriba son instancias particulares de MREM, como se demuestra en [15, 16], donde se analizan las posibilidades de MREM como memoria auto-asociativa, y un método para guardar y recuperar un conjunto de patrones mediante una generalización de la regla de Hebb. Se pueden encontrar más aplicaciones en [12, 13, 14, 15, 16].

### 3. El modelo MREM con dinámica semiparalela

Sea  $\mathcal{H}$  una red neuronal recurrente formada por  $N$  neuronas, donde el estado de la neurona  $i$ -ésima viene definido por su salida  $V_i$ ,  $i \in \mathcal{I} = \{1, 2, \dots, N\}$  que puede tomar cualquier valor dentro de un conjunto finito  $\mathcal{M} = \{m_1, m_2, \dots, m_L\}$ . Este conjunto no tiene por qué ser numérico.

El estado de la red, en el instante  $t$ , se puede expresar como un vector  $N$ -dimensional,  $\vec{V}(t) = (V_1(t), V_2(t), \dots, V_N(t)) \in \mathcal{M}^N$ .

Asociado a cada estado de la red, se define una función de energía, dada por la siguiente expresión

$$E(\vec{V}) = -\frac{1}{2} \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{I}} w_{i,j} f(V_i, V_j) + \sum_{i \in \mathcal{I}} \theta_i(V_i) \quad (1)$$

donde  $w_{i,j}$  es el peso de la conexión desde la neurona  $j$ -ésima a la  $i$ -ésima,  $f : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}$  se puede considerar como una medida de la similitud entre las salidas de dos neuronas, verificando normalmente las condiciones de similitud mencionadas en [15, 16]:

1. Para todo  $x \in \mathcal{M}$ ,  $f(x, x) = c \in \mathbb{R}$ .
2.  $f$  es una función simétrica: para todo  $x, y \in \mathcal{M}$ ,  $f(x, y) = f(y, x)$ .
3. Si  $x \neq y$ , entonces  $f(x, y) \leq c$ .

y  $\theta_i : \mathcal{M} \rightarrow \mathbb{R}$  son las funciones de umbral. Puesto que no usaremos los umbrales para

la memoria auto-asociativa, consideraremos  $\theta_i$  como la función constante igual a cero, para todo  $i \in \mathcal{I}$ .

En cada instante, la dinámica de la red nos llevará a un estado de menor energía que el actual.

Para este trabajo, hemos considerado tiempo discreto y dinámica semiparalela, en las que una única neurona es actualizada en cada instante  $t$ . El siguiente estado de la red será aquel que suponga el mayor descenso de la función de energía si cambiamos la salida de una sola neurona.

Consideremos por tanto un orden total en  $\mathcal{M}$ . El incremento de potencial cuando la neurona  $a$  cambia su salida de  $V_a$  a  $l \in \mathcal{M}$  en el instante  $t$ , es

$$\begin{aligned} U_{a,l}(t) &= \frac{1}{2} \sum_{i \in \mathcal{I}} [w_{a,i}f(l, V_i(t)) + w_{i,a}f(V_i(t), l) - \\ &\quad -(w_{a,i}f(V_a(t), V_i(t)) + w_{i,a}f(V_i(t), V_a(t)))] - \\ &\quad - \frac{1}{2}w_{aa}[f(l, l) - f(V_a(t), V_a(t))] \end{aligned} \quad (2)$$

Si  $f$  verifica las condiciones de similitud, se obtiene el incremento de potencial reducido:

$$\begin{aligned} U_{a,l}^*(t) &= -\Delta E = -\frac{1}{2} \sum_{i \in \mathcal{I}} [(w_{a,i} + w_{i,a}) \cdot \\ &\quad \cdot (f(V_i(t), l) - f(V_i(t), V_a(t)))] \end{aligned} \quad (3)$$

Además, usamos la siguiente regla de actualización para la salida de las neuronas:

$$V_a(t+1) = \begin{cases} l, & \text{si } U_{a,l}(t) \geq U_{b,k}(t) \\ & \forall k \in \mathcal{M} \text{ y } \forall b \in \mathcal{I} \\ V_a(t), & \text{en otro caso} \end{cases} \quad (4)$$

En resumen, cada neurona evalúa en paralelo el valor de un vector de potencial de longitud  $L$  (relacionado con el decremento de energía que se produciría si adopta ese nuevo estado), la única neurona que cambia su estado actual es aquella que produce el máximo decremento de la energía.

Se ha probado [12] que el modelo MREM, con esta dinámica, siempre converge a un estado minimal. Este resultado es particularmente interesante cuando se trata con problemas de

optimización combinatoria, campo en el que la aplicación de MREM ha sido muy fructífera [12, 13, 14, 17, 11].

#### 4. MREM como memoria auto-asociativa

Consideremos  $X = \{\vec{X}_k : k \in K\}$ , un conjunto de patrones que queremos cargar en la red neuronal. Entonces, para memorizar un patrón  $\vec{X} = (x_i)$ ,  $i = 1, 2, \dots, N$ , debemos modificar las componentes de la matriz  $W$  para hacer que el vector  $\vec{X}$  sea el estado de la red con energía mínima.

Como se señalaba en [15, 16], ya que la función de energía viene dada por la expresión (1), calculamos  $\frac{\partial E}{\partial w_{i,j}} = -\frac{1}{2}f(V_i, V_j)$  y modificamos las componentes de la matriz  $W$  para reducir la energía del estado  $\vec{V} = \vec{X}$ , según la regla:

$$\Delta w_{i,j} = -\alpha \frac{\partial E}{\partial w_{i,j}} = \frac{\alpha}{2}f(x_i, x_j), \quad (\alpha > 0)$$

En particular, para  $\alpha = 2$  (todos los  $\alpha > 0$  producen redes funcionalmente equivalentes, ver [15, 16]), resulta:

$$\Delta w_{i,j} = f(x_i, x_j) \quad (5)$$

y, considerando que en un principio,  $W = 0$ , es decir, todos los estados de la red tienen la misma energía, y realizando la suma sobre todos los patrones, se obtiene la siguiente expresión:

$$w_{i,j} = \sum_{k \in K} f(x_{ki}, x_{kj}) \quad (6)$$

Esta expresión es una generalización del postulado de aprendizaje de Hebb, ya que el peso  $w_{i,j}$  entre dos neuronas se incrementa en relación con su similitud.

Hay que señalar que, cuando se usan neuronas bipolares y la función producto,  $f(x, y) = xy$ , se obtiene la conocida regla de aprendizaje de patrones de la red de Hopfield.

Para poder recuperar un patrón, la red se inicializa con la parte conocida del patrón. La dinámica de la red convergerá a un estado estable (debido al decrecimiento de la función de energía) que se corresponderá con un mínimo

de la función de energía y será la respuesta de la red. Normalmente, este estado estable se encuentra próximo al inicial.

Pero, como se explica en [15, 16] en términos de la similitud entre los vectores asociados a distintos estados, cuando la red bipolar de Hopfield carga el patrón  $\vec{X}$ , según la regla (5) con  $f(V_i, V_j) = V_i V_j$ , se reduce también la energía de otros estados, y se memoriza también el estado opuesto. A los estados con mínimo local de energía, y no asociados con ningún patrón de entrada, se les llama estados espúreos, y la carga de estados espúreos generalmente se considera como un efecto no deseable.

## 5. Cómo evitar patrones espúreos

En [15, 16] se presenta un ‘truco’ para evitar la aparición de estados espúreos cuando se carga en la red bipolar de Hopfield el patrón  $\vec{X} = (x_i)$ .

**Definición 5.1** Dado un estado  $\vec{V}$  de la red, y una función de similitud  $f$ , se define su **matriz asociada** ( $G_{\vec{V}}$ ) como la matriz de orden  $N \times N$  cuyos elementos son  $G_{i,j} = f(V_i, V_j)$ .

**Definición 5.2** Además, se puede definir su vector asociado ( $\vec{G}_{\vec{V}}$ ) como el vector de  $N^2$  componentes obtenido al expandir la matriz  $G_{\vec{V}} = (G_{i,j})$  como vector  $\vec{G}_{\vec{V}} = (G_k)$ , verificando  $G_{j+N(i-1)} = G_{i,j}$ .

**Definición 5.3** Supongamos que  $\vec{X} = (x_1, x_2, \dots, x_N)$  es un patrón que debe cargarse en la red y  $\mathcal{M} = \{m_1, m_2, \dots, m_L\}$ . Llamamos **patrón aumentado** de  $\vec{X}$  al vector  $\tilde{\vec{X}} = (x_1, x_2, \dots, x_N, m_1, m_2, \dots, m_L)$  de  $\mathcal{M}^{N+L}$  cuyas componentes son:

$$\hat{x}_i = \begin{cases} x_i & i \leq N \\ m_{i-N} & i > N \end{cases}$$

Hay que notar que, para cargar patrones aumentados en la red, sólo son necesarias  $N$  neuronas, ya que las últimas  $L$  componentes tienen un valor fijo. Sólo será necesario considerar el peso  $w_{i,j}$  asociado a estas componentes.

**Teorema 5.1** La función  $\Psi : \tilde{\vec{X}} \rightarrow \vec{G}_{\tilde{\vec{X}}}$ , que hace corresponder a un patrón aumentado su vector asociado, es inyectiva.

### Demostración.

Sean  $\tilde{\vec{X}} = (x_1, x_2, \dots, x_N, m_1, m_2, \dots, m_L)$  y  $\tilde{\vec{Y}} = (y_1, y_2, \dots, y_N, m_1, m_2, \dots, m_L)$  los vectores aumentados correspondientes a  $\vec{X} = (x_1, \dots, x_N)$  y  $\vec{Y} = (y_1, \dots, y_N)$  que verifican  $\Psi(\tilde{\vec{X}}) = \Psi(\tilde{\vec{Y}})$ , o equivalentemente  $\vec{G}_{\tilde{\vec{X}}} = \vec{G}_{\tilde{\vec{Y}}}$ .

Si consideramos las componentes  $(N+1)$ -ésima a  $(N+L)$ -ésima de  $\vec{G}_{\tilde{\vec{X}}}$ , obtenemos  $f(x_1, m_1), f(x_1, m_2), \dots, f(x_1, m_L)$ .

Si  $x_k = a$  para algún  $1 \leq k \leq N$ , (con  $a$  cualquier valor en  $\mathcal{M} = \{m_1, m_2, \dots, m_L\}$ ), entonces  $f(x_k, a) = f(y_k, a) = 1 \Rightarrow y_k = a = x_k$ .

Luego, para todo  $k$  ( $1 \leq k \leq N$ ) se tiene que  $x_k = y_k$  y por tanto  $\tilde{\vec{X}} = \tilde{\vec{Y}}$   $\square$

De esta forma, para cargar un patrón  $\vec{X}$  con  $x_i \in \mathcal{M} = \{m_1, m_2, \dots, m_L\}$ , podemos cargar en la red su patrón aumentado  $\tilde{\vec{X}}$ . El estado  $\tilde{\vec{X}}$  será el único que maximice el decremento de energía. Las nuevas  $L$  neuronas se pueden considerar como ‘falsas’ o no implementadas, ya que siempre mantienen el mismo valor.

## 6. Refuerzo del aprendizaje

En la expresión de la regla de aprendizaje (5), generalización de la hebbiana, podemos observar que en la actualización de la matriz de pesos sólo interviene el patrón que se desea memorizar en dicho instante. Por tanto, es una información muy ‘local’, sin tener en cuenta las posibles relaciones que ese patrón tenga con los ya almacenados. Es conveniente, pues, introducir un mecanismo suplementario en la fase de aprendizaje, de forma que en la actualización de la matriz de pesos se haga patente la información que nos proporciona el patrón en relación a los demás.

Consideremos que la función de similitud es  $f(x, y) = 2\delta_{x,y} - 1$ , es decir, valdrá 1 si  $x = y$  y -1 en caso contrario. Supongamos que hemos memorizado en la red el patrón (aumentado)  $\vec{X}_1$ . Tenemos por tanto la matriz de pesos

$W = (w_{i,j})$ . Si ahora queremos cargar en la red el patrón  $\vec{X}_2$ , aplicando la expresión antes mencionada (5) obtenemos las componentes de la matriz  $\Delta W$ .

Si  $w_{i,j}$  y  $\Delta W_{i,j}$  tienen signo positivo (ambas valen 1), significa que  $X_{1i} = X_{1j}$  y que  $X_{2i} = X_{2j}$ , lo cual indica la relación que existe entre  $X_{1i}$ ,  $X_{1j}$ ,  $X_{2i}$  y  $X_{2j}$ . Si ambos son de signo negativo, ocurre algo parecido, pero con desigualdades.

Por tanto, el hecho de que  $w_{i,j}$  y  $\Delta W_{i,j}$  tengan el mismo signo es señal de una relación que se repite entre las componentes  $i$  y  $j$  de los patrones  $\vec{X}_1$  y  $\vec{X}_2$ . Para reforzar el aprendizaje de esta relación, multiplicamos las componentes donde se verifique la igualdad de signo por una constante,  $\beta > 1$ . Así, la matriz de pesos que aprende la red será, después de cargar el patrón  $\vec{X}_2$ :

$$w'_{i,j} = \begin{cases} w_{i,j} + \Delta W_{i,j} & \text{si } w_{i,j} \cdot \Delta W_{i,j} < 0 \\ \beta[w_{i,j} + \Delta W_{i,j}] & \text{si } w_{i,j} \cdot \Delta W_{i,j} > 0 \end{cases} \quad (7)$$

Análogamente a este caso, si la red tiene almacenados una serie de patrones, mediante la matriz  $W$ , y deseamos cargar en ella el patrón  $\vec{X}$ , hemos de calcular  $\Delta W$  y después aplicar la nueva regla de aprendizaje (7).

## 7. Sobrecargando la red

Este trabajo trata de explicar lo que puede ocurrir psicológicamente en el cerebro humano. Cuando se tiene que memorizar un número reducido de patrones, el cerebro es capaz de recordarlos todos cuando sea necesario. De la misma forma, cuando no se sobrepasa la capacidad de la red, ésta es capaz de recuperar exactamente los mismos patrones que fueron cargados en ella. Pero cuando el cerebro recibe una gran cantidad de datos que reconocer, es capaz de distinguir entre algunos grupos de datos (de forma no supervisada). Este tipo de comportamiento es también simulado por las redes neuronales, como mostraremos a continuación, probando el potencial (y la adecuación) del modelo aquí propuesto.

Por tanto, consideramos que reglas de aprendizaje como la de Hebb (o incluso la más

general dada por la expresión (5) y también la dada por (7)), donde la conexión entre neuronas se ve reforzada por la similitud de sus salidas esperadas, puede producir clasificadores que extraigan algún tipo de conocimiento de los patrones de entrada, como el número real de grupos en los que los datos están divididos. Así, se produce automáticamente un agrupamiento no supervisado del espacio de datos de entrada.

Si un patrón, digamos  $\vec{X}$ , va a ser cargado en la red, mediante la regla (5), se crea un mínimo local de la función de energía  $E$  en  $\vec{V} = \vec{X}$ . Si otro patrón  $\vec{X}'$  está distante de  $\vec{X}$ , su carga creará otro mínimo local. Pero, si  $\vec{X}$  y  $\vec{X}'$  están cercanos uno al otro, los mínimos locales creados por la regla de aprendizaje se mezclarán y solaparán, formando un único mínimo local en su lugar.

Por tanto, si se carga un grupo de patrones en la red, sobre pasando su capacidad, y todos se encuentran cerca unos de otros, sólo se formará un mínimo local, y, en el momento de recuperar estos datos, el único patrón que se recuperará será el que esté asociado al estado de mínima energía. De esta forma, los patrones se podrán clasificar atendiendo al estado estable de la red al cual convergen.

## 8. Simulaciones

Para demostrar el hecho explicado en la sección anterior, hemos realizado diversas simulaciones cuyo objetivo es el de clasificar datos discretos.

Por tanto, hemos considerado una red con  $N = 40$  neuronas que pueden tomar valores en el conjunto  $\mathcal{M} = \{1, \dots, 10\}$ . Como parámetro de refuerzo del aprendizaje se ha usado el valor  $\beta = 1.2$ .

Se han generado varios conjuntos de datos, cada uno de ellos formado por  $K$  patrones generados aleatoriamente alrededor de  $n$  centroides, esto es, se han generado aleatoriamente los  $n$  centroides y los patrones de entrada se forman a partir de ellos mediante la introducción de ruido que modifica una componente del mismo con probabilidad 0.0225. Por tanto la distancia de Hamming entre los patrones de

Cuadro 1: Resultados de clasificación medios en 10 ejecuciones del algoritmo para los parámetros indicados donde  $n_{aprox}$  indica el número de clases encontradas.

K	$n = 3$		$n = 4$	
	$n_{aprox}$	Error (%)	$n_{aprox}$	Error (%)
75	3.3	0.80	5.0	1.72
150	3.1	0.06	5.0	3.00
300	3.4	0.30	5.0	0.60
450	3.9	0.68	6.4	1.80
600	4.1	1.12	6.0	0.70
750	3.1	0.06	5.2	0.32

entrada y su centroide seguirán una binomial  $B(40, 0.0225)$ . Los patrones están igualmente repartidos entre los  $n$  centroides.

Los resultados obtenidos se muestran en el cuadro 1. Se puede observar cómo, no sólo el error de clasificación es bajo (del orden del 0.06 % al 3 % de media, alcanzándose en la mayoría de los casos el resultado óptimo, es decir, 0 % de error), sino que además esta técnica consigue dar con el número exacto o aproximado de grupos en los que está realmente dividiendo el conjunto de patrones en casi la totalidad de las simulaciones realizadas. De hecho, esta técnica es capaz en muchos casos de recuperar los centroides iniciales, usados para generar los datos, a partir de éstos.

## 9. Conclusiones

En este trabajo hemos presentado un nuevo punto de vista, según el cual la limitación, en la capacidad de memorizar patrones, de la red recurrente no debe ser considerada como determinante, sino que puede ser usada para el propósito de clasificación, no supervisada, de patrones discretos.

Se ha desarrollado el modelo neuronal MREM, generalización del modelo de Hopfield discreto basado en neuronas multivaluadas, como memoria auto-asociativa y se han presentado ciertos resultados que muestran técnicas para que la red no incorpore patrones espúreos en la fase de aprendizaje, reduciendo el número de mínimos locales de la función de

energía que no están asociados a ningún patrón de entrada.

Mediante una ligera modificación de la regla de aprendizaje de Hebb, hemos dotado, a esta primera fase del proceso, de un mecanismo para reforzar el aprendizaje de las relaciones que se encuentran entre distintos patrones, incorporando así conocimiento concerniente a varios patrones simultáneamente.

Además, las simulaciones confirman la idea expresada en este trabajo, llegando a resultados de clasificación óptima en la mayoría de los casos.

Este trabajo presenta una nueva línea de investigación abierta, partiendo del hecho de que es posible dar nuevas modificaciones de la regla de aprendizaje que potencien otros aspectos de las relaciones entre patrones.

## Referencias

- [1] Mahmut H. Erdem y Y. Ozturk, *A New family of Multivalued Networks*, Neural Networks **9**, 6, 979-989, 1996.
- [2] Gloria Galán Marín y José Muñoz Pérez, *Design and Analysis of Maximum Hopfield Networks*, I.E.E.E. Transactions on Neural Networks, **12**, 329-339, 2001.
- [3] Gloria Galán Marín y José Muñoz Pérez, *A New Input-output Function for Binary Hopfield Neural Networks*, Lecture Notes in Computer Science, **1606**, 311-320, 1999.
- [4] Gloria Galán Marín, Enrique Mérida Casermeiro y José Muñoz Pérez, *Modelling Competitive Hopfield Networks for the Maximum Clique Problem*, Computers & Operations Research, **30-4**, 603-624, 2002.
- [5] L. Gislén, C. Peterson and B. Söderberg, *Complex Scheduling with Potts Neural Networks*, Neural Computation. **9**, 805-831, 1992.
- [6] Donald O. Hebb, *The Organization of Behavior*, New York: Wiley, 1949.
- [7] J.J. Hopfield, *Neural networks and physical systems with emergent collective computa-*

- tional abilities*, Proc. of National Academy of Sciences USA, **79**, 2254-2558, 1982.
- [8] J. J. Hopfield, *Neurons with graded response have collective computational properties like those of two-state neurons*, Proceedings of the National Academy of Sciences USA, **81**, 3088-3092, 1984.
- [9] S. Jankowski, A. Lozowski y J. M. Zurada, *Complex-valued Multistate Neural Associative Memory*, IEEE Transactions on Neural Networks, Vol 7, 6, 1491-6, 1996.
- [10] G.A. Kohring, *On the Q-state Neuron Problem in Attractor Neural Networks*, Neural Networks, Vol **6**, 573-581, 1993.
- [11] D. López-Rodríguez y E. Mérida-Casermeiro, *Matrix Bandwidth Minimization: A Neural Approach*, Proceedings of International Conference of Computational Methods in Science and Engineering, ICCMSE, **1**, 324-327, 2004.
- [12] E. Mérida Casermeiro, *Red Neuronal recurrente multivaluada para el reconocimiento de patrones y la optimización combinatoria*, Tesis doctoral. Univ. Málaga, España, 2000.
- [13] Enrique Mérida Casermeiro, José Muñoz-Pérez, *MREM: An associative autonomous recurrent network*, Journal of Intelligent and Fuzzy Systems 12(3-4), 163-173, 2002.
- [14] E. Mérida Casermeiro, J. Muñoz Pérez, R. Benítez Rochel, *A recurrent multivalued neural network for the N-queens problem*, Lecture Notes in Computer Science **2084**, 522-529, 2001.
- [15] Enrique Mérida Casermeiro, José Muñoz-Pérez, M. A. García-Bernal, *An Associative Multivalued Recurrent Network*, IBERAMIA 2002, 509-518, 2002.
- [16] Enrique Mérida Casermeiro, José Muñoz-Pérez, *MREM: An associative autonomous recurrent network*, Journal of Intelligent and Fuzzy Systems 12(3-4), 163-173, 2002.
- [17] E. Mérida-Casermeiro y D. López-Rodríguez, *Multivalued Neural Network for Graph MaxCut Problem*, Proceedings of International Conference of Computational Methods in Science and Engineering, ICCMSE, **1**, 375-378, 2004.
- [18] A.J. Noest, *Discrete-state phasor neural nets*, Physical Review **A38** 2196-99, 1988.
- [19] Yusuf Ozturk y Hüseyin Abut, *System of associative relationships (SOAR)*, Proceedings of ASILOMAR, 1997.
- [20] H. Rieger, *Storing an extensive number of grey-toned patterns in a neural network using multistate neurons*, Journal of Physics, **A23**, L1273-79, 1990.
- [21] F.Y. Wu, *The Potts Model*, Reviews of Modern Physics, **54(1)**, 1982.
- [22] Jacek M. Zurada; Ian Cloete; Etienne Van der Poel, *Generalized Hopfield networks for associative memories with multi-valued stable states*, Neurocomputing, Vol 13, 135-49, 1996.