

MREM: Una red recurrente con estados neuronales generalizados

José Muñoz Pérez

Departamento de Lenguajes
y Ciencias de la Computación,
Universidad de Málaga
munozp@lcc.uma.es

**Jose David
Fernández Rodríguez**

Grupo de Estudios en Biomimética,
Departamento de Lenguajes
y Ciencias de la Computación,
Universidad de Málaga
josedavid@geb.uma.es

Resumen

En este trabajo se presenta un paradigma de red neuronal recurrente, de propósito general, con una regla de actualización basada en similitudes entre los estados de los elementos neuronales. Estos elementos neuronales se caracterizan por admitir cualquier conjunto de estados, incluso no numéricos, siempre que se defina una medida de similaridad dentro de dicho conjunto. La función de energía asociada decrece monótonamente durante la dinámica de la red. En concreto, este tipo de red neuronal supone una generalización de las clásicas redes de Hopfield. Como ejemplo de aplicación, se usará este paradigma para resolver un problema de clustering.

1. Introducción

Desde que Hopfield publicó su trabajo seminal sobre el tema en 1982 [7], las redes nombradas de forma homónima han constituido un fértil campo de investigación. En un principio, nacieron como un nuevo tipo de memoria asociativa, aunque después se comprendió su potencial en el campo de la optimización en 1985 [8]. Dentro del paraguas de la optimización cabe un amplísimo abanico de aplicaciones, lo que ha movido a un apreciable número de investigadores hacia este campo, produciéndose una notable cantidad de literatura [12, 13]. También se han investigado exhaustivamente las propiedades de distintos tipos de redes de Hopfield, tanto en su complejidad como en su poder computacional, habiéndose encontrado que son universales en el sentido de equivalencia a máquinas de Turing [14, 11, 6].

1.1. Red de Hopfield básica

En su formulación básica, las redes de Hopfield consisten en un conjunto de neuronas totalmente conectadas que evolucionan en pasos discretos de tiempo. Entre cada par de neuronas i, j se define un determinado peso sináptico w_{ij} ; además, para cada neurona se define un valor umbral θ_i . En cada paso temporal t , cada neurona i posee un estado $s_i(t)$ que puede tomar dos valores posibles, 1 y -1 ¹. El método de actualización también es discreto: en cada paso temporal, se escoge aleatoriamente una

¹Alternativamente, muchos autores usan 0 y 1 como estados.

neurona² y se le aplica la siguiente regla de actualización:

$$s_i(t+1) = \begin{cases} 1 & \text{si } \sum_j w_{ij} \cdot s_j(t) \geq \theta_i \\ -1 & \text{si } \sum_j w_{ij} \cdot s_j(t) < \theta_i \end{cases}$$

Para las demás neuronas, la regla es $s_i(t+1) = s_i(t)$.

A partir de esta dinámica, se puede derivar la siguiente función (denominada *de energía*), que es monótonamente decreciente:

$$E(t) = -\frac{1}{2} \sum_i \sum_j w_{ij} \cdot s_i(t) \cdot s_j(t) + \sum_i \theta_i \cdot s_i(t)$$

Si consideramos que el estado de la red de Hopfield es un punto en un espacio $\{1, -1\}^N$ ³, la dinámica de la red garantiza que se alcanza un estado que es un mínimo local de la función de energía. Bajo esta óptica, al usar una red de Hopfield para resolver un problema de optimización, se debe definir la red de manera que la distribución de pesos y umbrales refleje una combinación de la función a optimizar sujeta a severas penalizaciones que representan las restricciones del problema.

Así, por ejemplo, el problema del viajante de comercio para P ciudades puede codificarse con una red de P^2 neuronas [8], en la que el estado de cada neurona s_{kl} determina si la ciudad k está en la l -ésima posición en la solución. Los pesos y umbrales se definen de manera que en la función de energía las soluciones posean menor energía contra más cortas y a la vez más adecuadas⁴.

1.2. Trabajo relacionado: MAREN

Como ya se ha señalado, se han diseñado numerosas modificaciones al esquema original propuesto por Hopfield, para adaptarlo a dominios de problemas muy diversos. En este respecto, una de las características de las redes de Hopfield originales es que las neuronas tienen dos estados. Aunque dos estados bastan para que se pueda modelar cualquier problema de optimización con una red de Hopfield, existen problemas para los que esta representación puede no ser la más adecuada.

En [4] se describe una variante del modelo de Hopfield denominada MAREN. La modificación consiste en que las neuronas pueden tomar $M \geq 2$ estados consecutivos y ordenados $1 \dots M$. En este artículo se consideran redes con pesos para las interconexiones neuronales, pero con umbrales fijados a 0. La función de energía se transforma en:

$$E(t) = \sum_i \sum_j w_{ij} \cdot \text{sgn}(s_i(t) - s_j(t))$$

Donde la función $\text{sgn}(x)$ es la función que devuelve el signo de x ⁵. La regla de actualización es consecuentemente modificada de modo que la función de energía sea monótonamente decreciente:

$$s_i(t+1) = \begin{cases} s_j(t) + 1 & \text{si } k_i > 0 \\ s_j(t) & \text{si } k_i = 0 \\ s_j(t) - 1 & \text{si } k_i < 0 \end{cases}$$

$$k_i = \text{sgn}(\Delta E_i^+(t)) - \text{sgn}(\Delta E_i^-(t))$$

Donde $\Delta E_i^{+/-}(t)$ es la diferencia de energía entre el estado de la red cambiado respecto al estado original. Para $\Delta E_i^+(t)$ el estado se cambia incrementándolo, y para $\Delta E_i^-(t)$, decrementándolo:

²En lo sucesivo, cuando se mencione que se escoge aleatoriamente una neurona se entiende que se pueden usar otros métodos, como por ejemplo escoger cíclicamente todas las neuronas, siguiendo un orden arbitrario dentro de cada ciclo.

³Donde N es el número de neuronas de la red de Hopfield.

⁴Es decir, que cada ciudad se visite una y sólo una vez.

⁵ $\text{sgn}(x)$ devuelve tres valores posibles: +1 si x es positivo, -1 si es negativo, o 0 si es nulo.

$$\begin{aligned}\Delta E_i^+(t) &= E(t) [s_i(t) \leftarrow s_i(t) + 1] - E(t) \\ \Delta E_i^-(t) &= E(t) [s_i(t) \leftarrow s_i(t) - 1] - E(t)\end{aligned}$$

Gracias a esta modificación, muchos problemas pueden codificarse de forma más sencilla, con un menor número de neuronas. Retomando el problema del viajante de comercio, en la misma publicación en que se describe MAREN [4] se ofrece una forma de codificarlo: para un problema con N ciudades, se usa una red con N neuronas, cada una de las cuales puede estar en N estados distintos. Así, cada neurona representa una ciudad, y cada estado, la posición en el recorrido solución. Como es usual, los pesos sinápticos se determinan de forma que en la función de energía se penalicen los recorridos largos respecto a los cortos, y además se penalicen las configuraciones que no son soluciones al problema.

2. Descripción de MREM

En [9] se analiza cómo solucionar el problema del viajante mediante una red multivaluada. Dado que MAREN es una red multivaluada desarrollada con anterioridad, la toman como referencia con la que comparar la nueva red. Se propone una codificación del problema más alejada de la interpretación original de Hopfield que MAREN, lo que permite, entre otras cosas, simplificar las cosas y lograr mejorar el rendimiento que consigue la red MAREN para resolver este tipo de problemas. De forma muy interesante, se propone la siguiente generalización de la función de energía de la red de Hopfield original:

$$E(t) = \sum_i \sum_j w_{ij} \cdot g(s_i(t), s_j(t))$$

Donde $g(m, n)$ es una función que mide la similaridad entre los estados m y n . En [1] se aplica una red multivaluada al problema de las N reinas, partiendo de una función de energía similar.

Tras los dos artículos previamente mencionados, este enfoque se desarrolla con más detalle en trabajos posteriores, como [3, 2]⁶, puliéndolo y presentándolo como un nuevo modelo de red neuronal: MREM.

2.1. Función de energía

En [3, 2], al cómputo energético desarrollado en [9, 1] se le añade una contribución que cumple las funciones del valor umbral en la formulación básica de Hopfield:

$$E(t) = -\frac{1}{2} \sum_i \sum_j w_{ij} \cdot g(s_i(t), s_j(t)) + \sum_i \theta_i(s_i(t))$$

Bajo este enfoque, se relajan los requisitos que debe cumplir el conjunto de estados neuronales $\mathcal{M} = \{m_1, \dots, m_N\}$. En vez de ser un conjunto numérico (totalmente ordenado) como en MAREN, simplemente se requiere que sea un conjunto finito con elementos numéricos o discretos, y que se puedan definir sendas funciones que tomen como argumentos elementos del conjunto de estados \mathcal{M} :

- una función $g(m, n)$ que mide numéricamente la similaridad entre pares de elementos. Aparte de lo anterior no se añaden más restricciones al tipo de funciones que se pueden usar, aunque las funciones de similaridad suelen cumplir las siguientes tres propiedades:
 - $\forall m, n \in \mathcal{M} : f(m, n) = f(n, m)$ (simetría)
 - Dado un valor numérico c que denote la máxima similaridad, y que la función devuelva valores mayores contra más similares sean sus argumentos:

⁶Parece que existe un artículo [10] en el que también se presenta MREM, aunque no se ha podido acceder al texto del mismo.

- $\forall m, n \in \mathcal{M} \mid m \neq n : f(m, n) \leq c$
- $\forall m \in \mathcal{M} : f(m, m) = c$

- una función $\theta_i(m)$ que devuelve un valor umbral para cada posible estado.

2.2. Regla de actualización

Como en los modelos anteriores, en cada paso temporal se escoge aleatoriamente una neurona i , a la que se le aplica la regla de actualización, dejando las demás neuronas sin cambios. La regla cambia la neurona al estado m que produzca una mayor minimización de la función de energía⁷. La ecuación que implementa la regla es:

$$s_i(t+1) = \operatorname{argmax}_{m \in \mathcal{M}} \left(\frac{1}{2} \cdot \sum_j (w_{ij} \cdot g(m, s_j(t)) + w_{ji} \cdot g(s_j(t), m)) - \frac{1}{2} \cdot w_{ii} \cdot g(m, m) - \theta_i(m) \right)$$

Con esta ecuación, se tienen en cuenta las contribuciones de la neurona i al cómputo de similaridad tanto por la izquierda $g(m, s_j(t))$ como por la derecha $g(s_j(t), m)$. Se resta el término $\frac{1}{2}w_{ii}g(m, m)$ para contarla solamente una vez, y además se tiene en cuenta la contribución del correspondiente umbral $\theta_i(m)$.

Por supuesto, si se quiere obtener una dinámica lo más parecida posible a la red de Hopfield original, y/o si la red es simétrica tanto en los pesos sinápticos como en la función de similaridad, se puede simplificar la expresión anterior obviando los términos por la derecha:

$$s_i(t+1) = \operatorname{argmax}_{m \in \mathcal{M}} \left(\frac{1}{2} \cdot \sum_j (w_{ij} \cdot g(m, s_j(t))) - \theta_i(m) \right)$$

2.3. Notación matricial

Se puede conseguir una notación más compacta usando vectores y matrices. El estado global en el paso t de una red con N neuronas se puede considerar un vector $\mathbf{S}^t = (s_1^t, \dots, s_N^t)$ cuyas componentes son los estados de cada neurona. Dado que el conjunto de estados \mathcal{M} es finito, toda la información necesaria para especificar una instancia concreta de una red puede especificarse como una 4-tupla $\mathcal{H} = (\mathcal{M}, \mathbf{W}, \mathbf{G}, \Theta)$ [2], donde:

- Como ya se ha definido previamente, \mathcal{M} es el conjunto de estados
- \mathbf{W} es una matriz cuadrada; cada elemento W_{ij} es el peso sináptico entre la neurona i y la neurona j
- \mathbf{G} es una matriz cuadrada; cada elemento G_{kl} es $g(k, l)$, es decir, la medida numérica de similaridad entre el estado k y el estado l
- Θ es una matriz con N filas y $|\mathcal{M}|$ columnas; cada elemento Θ_{pq} es el valor umbral para la neurona p cuando ésta se encuentra en el estado q

Con esta notación, la función de energía se escribe como:

$$E^t = -\frac{1}{2} \sum_i \sum_j W_{ij} \cdot g_{S_i^t, S_j^t} + \sum_i \Theta_{i, S_i^t}$$

Y la regla de actualización como:

⁷Por supuesto, si cualquier otro estado eleva la función de energía, se conserva el estado actual.

$$S_i^{t+1} = \operatorname{argmax}_{m \in \mathcal{M}} \left(\frac{1}{2} \cdot \sum_j (W_{ij} \cdot G_{m,S_j^t} + W_{ji} \cdot G_{S_j^t,m}) - \frac{1}{2} \cdot W_{ii} \cdot G_{m,m} - \Theta_{i,m} \right)$$

Nuevamente, si queremos obtener una dinámica más similar a la de la red de Hopfield original, y/o si tanto \mathbf{W} como \mathbf{G} son simétricas, la expresión anterior se puede simplificar a:

$$S_i^{t+1} = \operatorname{argmax}_{m \in \mathcal{M}} \left(\frac{1}{2} \cdot \sum_j (W_{ij} \cdot G_{m,S_j^t}) - \Theta_{i,m} \right)$$

3. Características y aportaciones de MREM

Este tipo de redes pueden ser consideradas como una generalización de muchos modelos discretos anteriores. En concreto, en [3] se citan en este respecto los modelos de redes neuronales MAREN, Phasor, Q-Ising, y el de Potts. Las funciones genéricas $g(m, n)$ y $\theta_i(m)$ se instancian a las propias de cada modelo concreto. Por ejemplo, para el modelo básico de Hopfield con estados 1 y -1, las funciones se instancian a:

$$\begin{aligned} g(m, n) &= m \cdot n \\ \theta_i(m) &= \theta_i \cdot m \end{aligned}$$

Donde el factor θ_i en el término derecho de la segunda ecuación es el valor numérico real usado como umbral para cada neurona i en la formulación original de la red de Hopfield.

Como ya apuntaron los creadores de MAREN [4], el uso de redes multivaluadas permite una aproximación más adecuada a los problemas reales que las redes con neuronas binarias. MREM proporciona mayor libertad a la hora de modelar problemas de optimización, al poder diseñar libremente el conjunto de estados y las funciones asociadas, incluso sin necesidad de que los estados sean numéricos. Esto permite en muchos casos expresar de forma implícita todas o casi todas las restricciones del problema⁸. De esta forma, se evitan algunos de los inconvenientes clásicos de las redes de Hopfield [2]:

- La necesidad de incluir estas restricciones como términos de penalización en la función de energía.
- Consecuentemente al punto anterior, se mejora tanto la estabilidad de la dinámica como la velocidad de convergencia.
- También relacionado con el primer punto, se reduce la necesidad de incluir gran cantidad de parámetros, y de tener que ajustar finamente sus valores para que la red funcione satisfactoriamente.

4. Aplicaciones

En [3] se presenta MREM como un nuevo modelo de red para implementar memorias asociativas, analizándose desde un punto de vista teórico la capacidad de almacenamiento del modelo.

En lo que concierne a la aplicación a problemas de optimización, en [2] se aplica este modelo al problema del viajante de comercio. La codificación es la misma que la usada para MAREN [4], que ya se explicó en la Subsección 1.2, y el diseño de la red se basa en el trabajo preliminar [9].

⁸En otras palabras, se evita o se limita la circunstancia de que haya configuraciones de la red que no se correspondan a soluciones al problema.

4.1. Clustering

En este apartado se describirá cómo adaptar MREM para resolver un problema de clustering⁹, teniendo en cuenta que esta clase de problemas pueden considerarse como problemas de optimización. A grandes rasgos, se puede describir un problema de clustering como la agrupación de N muestras, cada una dotada de M características, en k clases o *clusters*, basando la agrupación en la similaridad entre las características de las muestras. Así, las muestras se pueden considerar como N puntos en un espacio M -dimensional, y el clustering como *el problema de agrupar aquellas que están cerca entre sí*. La cantidad de clases k puede ser fija o variable (es decir, determinada en función de la configuración de las muestras en el N -espacio).

La adaptación del modelo MREM a problemas de clustering es muy sencilla. Recordando que la red se define mediante una 4-tupla $\mathcal{H} = (\mathcal{M}, \mathbf{W}, \mathbf{G}, \Theta)$:

- El conjunto de estados contiene un estado m_l (pueden ser numéricos, aunque no es necesario) por cada cluster: $\mathcal{M} = \{m_1, \dots, m_l, \dots, m_k\}$.
- Hay una neurona por cada muestra, de modo que el estado de la neurona i señala el cluster al que pertenece la muestra asociada a i .
- La matriz de pesos sinápticos \mathbf{W} contiene como valores w_{ij} las distancias d_{ij} entre las muestras asociadas a cada neurona i y cada neurona j en el espacio N -dimensional. Como distancia, puede usarse prácticamente cualquier métrica, de acuerdo la naturaleza de las características de las muestras (valores numéricos continuos/enteros, valores discretos, etc.). Por ejemplo, para valores numéricos continuos, la más intuitiva es la euclídea, pero hay muchas otras, por ejemplo: manhattan, de tablero de ajedrez, mahalanobis... En general, se deberá usar la distancia que mejor caracterice la separación de las muestras en clusters bien diferenciados, ponderando la decisión respecto al coste computacional de calcular dicha distancia.
- La función de similaridad puede ser tan sutil y adaptada al problema concreto como se desee; aunque para problemas de clustering se puede usar una muy simple:

$$g(m, n) = \begin{cases} 1 & \text{si } m = n \\ 0 & \text{si } m \neq n \end{cases}$$

Nótese que para esta función de similaridad, la matriz asociada es la identidad: $\mathbf{G} = \mathbf{I}_k$.

- Los umbrales se pueden obviar, usando una matriz nula: $\Theta = \mathbf{0}_{N \times k}$.

En este contexto, la función de energía puede interpretarse como la suma de las distancias de cada muestra al resto de muestras del cluster al cual está asignada. Evidentemente, un cluster formado por elementos apartados por distancias relativamente grandes incrementará notablemente el valor de la función de energía. Como la dinámica de la red tiende a minimizar este valor, el resultado es que se mejora el clustering hasta que se alcanza una solución localmente óptima.

4.1.1. Agrupación de patrones de expresión genética

En este apartado se describirá la instanciación de una red MREM a un problema de clustering concreto: la clasificación de unas muestras de expresión genética en 5 grupos. Hay 38 muestras, cada una de las cuales registra el nivel de expresión de 18 genes. Por tanto, instanciamos el problema:

- Conjunto de estados: $\mathcal{M} = \{clase_1, clase_2, clase_3, clase_4, clase_5\}$
- Número de neuronas: $N = 38$
- Matriz de pesos sinápticos \mathbf{W} : distancias euclídeas en el espacio 18-dimensional.
- Función de similaridad y umbrales: los ya mencionados en el apartado anterior.

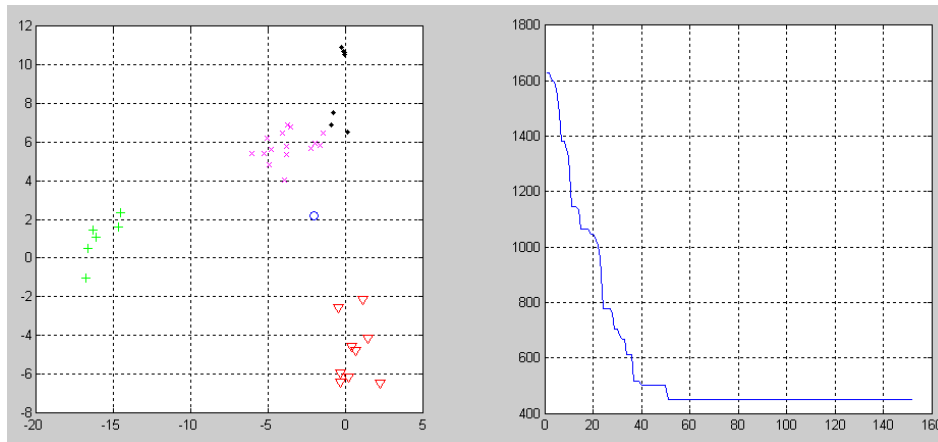
⁹Se puede encontrar una introducción al clustering en [5].

4.1.2. Ejemplo

Los resultados de un proceso de clustering sobre el problema descrito en la Subsubsección 4.1.1 se pueden ver en la Figura 1. La implementación está realizada en MATLAB, y el tiempo de ejecución es de aproximadamente 0.7 segundos en un AMD Turion 1.61GHz con 1Gb de RAM. Como modificación a lo descrito en secciones anteriores, en lugar de escoger en cada paso una neurona aleatoriamente, se escogen según una secuencia arbitraria cíclica. Como consecuencia, el número de pasos se define como un múltiplo (4, en este ejemplo) del número total de neuronas, para darle 4 oportunidades de actualizarse a cada una. Por tanto, se itera el algoritmo de actualización durante $4 \cdot 38 = 152$ iteraciones (pasos temporales).

La red neuronal parte de una configuración inicial aleatoria. La función de energía disminuye de valor rápidamente: tras 51 iteraciones (pasos temporales), desciende hasta el mínimo local de 449.4991. Como se puede apreciar, el proceso de clustering es coherente respecto a la proyección calculada mediante el análisis de componentes principales, ya que los clusters calculados por la red no se solapan en dicha proyección.

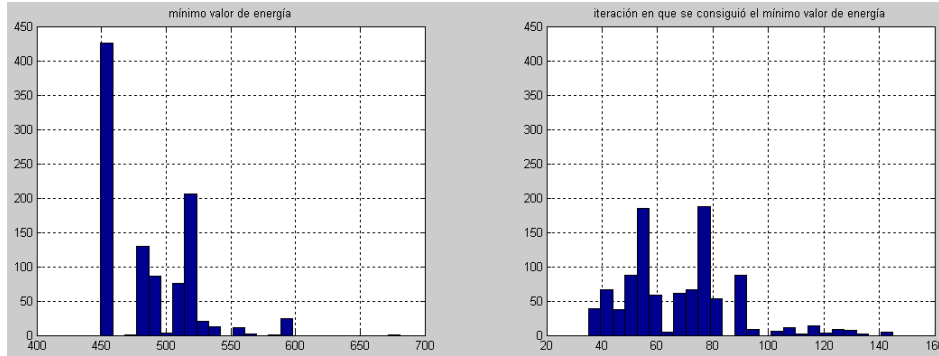
Figura 1: Resultados de un proceso de clustering. *Izquierda*: proyección en 2 dimensiones del resultado obtenido en el espacio 18-dimensional, mediante un análisis en componentes principales. Cada marca representa una muestra, y cada tipo de marca indica el cluster al que ha sido asignada cada muestra. Las magnitudes de las abcisas y las ordenadas se derivan del proceso de proyección. *Derecha*: evolución de la función de energía a lo largo del proceso. En las abcisas, los pasos de tiempo; en las ordenadas, el valor de la función de energía.



4.1.3. Estadísticas

Tras visualizar gráficamente el proceso de clustering descrito en la Subsubsección 4.1.2, ejecutamos 1000 procesos de clustering (cada uno comenzando en una posición aleatoria distinta) para estudiar estadísticamente el rendimiento de MREM para este tipo de clustering. Estos procesos siguen la misma dinámica y se ejecutan durante el mismo número de iteraciones que el ejemplo anterior. Nos centraremos en dos medidas de rendimiento, cuyos histogramas se pueden ver en la Figura 2:

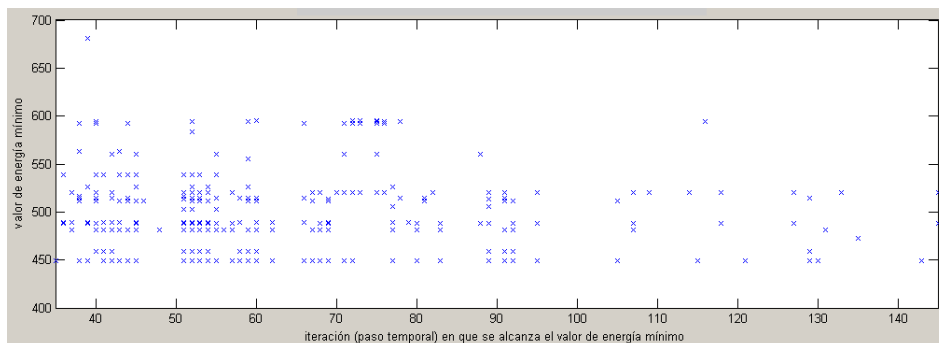
Figura 2: Histogramas de rendimiento sobre 1000 procesos de clustering. *Izquierda*: Histograma del mínimo valor de energía alcanzado en cada proceso. *Derecha*: Histograma que recoge el paso temporal en la que se alcanzó el mínimo valor de energía de cada proceso.



- El valor de energía mínimo alcanzado por la red. Conforme menor sea, mejor será el clustering obtenido. En el histograma se puede apreciar que 426 procesos (42.6%) consiguieron resultados de aproximadamente 455; los resultados del resto de procesos se distribuyen de forma desigual entre valores mayores. Sólo 40 procesos (4%) se estancaron en valores superiores a 500.
- La iteración (paso temporal) en la que se obtuvo el valor de energía mínimo en cada proceso de clustering. Tras la agrupación de datos para el histograma, la distribución parece aproximadamente bimodal (modas en torno a 50 y 70). Sólo 59 (5.9%) procesos tardan más de 100 pasos temporales en estabilizarse.

Los histogramas no permiten apreciar si hay correlación entre ambas medidas. En la Figura 3 se puede ver un gráfico en el que se ha dibujado una cruz por cada proceso. Las coordenadas X de cada cruz se corresponden con la iteración en la que el proceso se estabiliza, y las Y con el mínimo valor de energía alcanzado. Se puede ver cualitativamente que la correlación entre ambas variables es débil, en el sentido de que no se puede ajustar una función entre ambas variables con márgenes de error estrechos. Sí se puede apreciar que para los relativamente pocos procesos en los que el valor de energía se estabiliza por encima de 550, suele hacerlo en menos de 80 iteraciones.

Figura 3: Comparación de ambas medidas de rendimiento: como se puede apreciar, no existe una correlación fuerte entre el valor mínimo de la función de energía y la iteración (paso temporal) en la que se alcanzó dicho valor en cada proceso.



4.1.4. Trabajo futuro

El método de clustering descrito adolece de un sesgo de tamaño: tiende a formar clusters igualados en tamaño, ya que si hay dos clusters relativamente cercanos con tamaños muy diferentes, la dinámica

de la red será tal que el más pequeño tenderá a *robar* muestras periféricas al más grande. Se puede ver un ejemplo en la parte derecha de la Figura 1, en la que el cluster representado mediante puntos negros tiene asignadas algunas muestras que parecen más cercanas a las muestras del cluster representado mediante cruces rosas. Esto se debe a que en la función de energía se considera la suma total de distancias entre las muestras del cluster, con lo que los clusters más pequeños que sus vecinos se encuentran en una situación energéticamente favorable para *robarles* muestras a sus vecinos mayores. Una solución alternativa podría ser el uso del promedio de las distancias euclídeas como medida de distancia.

Otro aspecto que debe ser investigado a fondo es una comparativa exhaustiva a nivel teórico y estadístico entre este nuevo método de clustering y otros previamente existentes y reconocidos como estándares, como el método de las K-medias.

5. Conclusiones

Las redes MREM constituyen una generalización que incluye como casos particulares tanto las redes de Hopfield originales como muchas de sus extensiones, como las redes MAREN. Por tanto, proporciona un marco teórico dentro del cual unificar la investigación sobre redes de Hopfield.

Al igual que las redes de Hopfield originales, se pueden usar como memorias asociativas (con algunas ventajas añadidas, como la existencia de un método que permite que la red no aprenda los patrones complementarios, con un coste de aumento del número de neuronas [3]). También, como otras redes de Hopfield, se pueden usar para resolver problemas de optimización. En este sentido, la generalidad de las redes MREM proporciona libertad a la hora de diseñar la forma de adaptar la red a cada problema concreto, permitiendo evitar algunos de los problemas de las redes de Hopfield convencionales en dicho aspecto. Por ejemplo, la necesidad de incluir las restricciones al problema de optimización como factores de penalización en la función de energía, con todos los inconvenientes que ello acarrea.

Como novedad, en este trabajo se ha instanciado una red MREM para resolver problemas de clustering, y se ha presentado un ejemplo concreto, aunque como ya se ha señalado, el enfoque todavía puede ser mejorado, y queda pendiente una comparativa con otros métodos de clustering, como el de las K-medias.

Referencias

- [1] CASERMEIRO, E. M., MUÑOZ, J., AND BENÍTEZ, R. A recurrent multivalued neural network for the n-queens problem. In *IWANN* (2001), pp. 522–529.
- [2] CASERMEIRO, E. M., MUÑOZ-PÉREZ, J., AND MERINO, E. D. An n-parallel multivalued network: Applications to the travelling salesman problem. In *IWANN* (2003), pp. 406–413.
- [3] CASERMEIRO, E. M., MUÑOZ-PÉREZ, J., AND GARCÍA-BERNAL, M. An associative multivalued recurrent network. In *IBERAMIA 2002: Proceedings of the 8th Ibero-American Conference on AI* (London, UK, 2002), Springer-Verlag, pp. 509–518.
- [4] ERDEM, M., AND OZTURK, Y. A new family of multivaluated networks. *Neural Networks 6* (1996), 979–989.
- [5] EVERITT, B., LANDAU, S., AND LEESE, M. *Cluster Analysis*. 2001.
- [6] FLOREEN, P., AND OPONEN, P. Complexity issues in discrete hopfield networks. Tech. rep., University of Helsinki, Department of Computer Science, 1994.
- [7] HOPFIELD, J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the USA* 79, 8 (1982), 2554–2558.

- [8] HOPFIELD, J., AND TANK, D. Neural computation of decisions in optimization problems. *Biological Cybernetics* 52 (1985), 141–152.
- [9] MÉRIDA-CASERMEIRO, E., GALÁN-MARÍN, G., AND MUÑOZ-PÉREZ, J. An efficient multivalued hopfield network for the traveling salesman problem. *Neural Processing Letters* 14 (2001), 203–216.
- [10] MÉRIDA-CASERMEIRO, E., AND MUÑOZ-PÉREZ, J. Mrem: An associative autonomous recurrent network. *Journal of Intelligent and Fuzzy Systems* 12, 3,4 (2002), 163–173.
- [11] ORPONEN, P. The computational power of discrete hopfield nets with hidden units. *Neural Computation* 8, 2 (1996), 403–415.
- [12] SMITH, K. Neural networks for combinatorial optimization: A review of more than a decade of research. *INFORMS Journal on Computing* 11 (1999), 15–34.
- [13] SULEHRIA, H. K., AND ZHANG, Y. Hopfield neural networks: a survey. In *AIKED'07: Proceedings of the 6th Conference on 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases* (Stevens Point, Wisconsin, USA, 2007), World Scientific and Engineering Academy and Society (WSEAS), pp. 125–130.
- [14] ŠÍMA, J., AND ORPONEN, P. Continuous-time symmetric hopfield nets are computationally universal. *Neural Computation* 15, 3 (2003), 693–733.



José Muñoz Pérez nació en Cazorla (España) en 1956. Se licenció en Matemáticas en la Universidad de Granada (España) en 1974, y obtuvo su doctorado en Matemáticas por la Universidad de Sevilla (España) en 1978. En el periodo 1976-89 fue Profesor Asociado en la Universidad de Sevilla, y desde esa fecha ha sido profesor (y más tarde catedrático) en la Universidad de Málaga (España).



Jose David Fernández Rodríguez nació en Málaga (España) en 1982. En la Universidad de Málaga obtuvo sendos títulos como Ingeniero Técnico en Informática de Sistemas y como Ingeniero Superior en Informática en 2004 y 2007, respectivamente. Desde 2007 cursa sus estudios de doctorado en la Universidad de Málaga, siendo alumno del Máster Oficial con Mención de Calidad en Ingeniería del Software e Inteligencia Artificial.