

Aproximación de los problemas de optimización por redes neuronales artificiales

Amparo Ruiz Sepúlveda
Dpto. Lenguajes y Ciencias de la Computación
amparo@lcc.uma.es

y
Rafael Caballero Fernández
Dpto. Economía Aplicada (Matemáticas)
r_caballero@uma.es

Universidad de Málaga

Resumen:

En el presente artículo, pretendemos aportar un conocimiento de cómo puede aproximarse un problema de optimización mediante la dinámica de las redes neuronales artificiales, específicamente mediante las redes de tipo Hopfield. La elección de este tipo de redes radica en el objetivo que nos hemos propuesto en el presente trabajo, la resolución on-line de los problemas de optimización lineal, cuadrática, no lineal y combinatoria. Construimos, desarrollamos y justificamos formalmente una metodología integral que permite recoger los distintos tipos de problemas de optimización y construir una red neuronal de tipo Hopfield que los resuelva de una manera computacionalmente adecuada y formalmente correcta, comparando los resultados obtenidos con los publicados más recientemente.

Palabras clave: *Redes Neuronales, Optimización Combinatoria, Hopfield, Aplicaciones a la Economía*

1.- INTRODUCCIÓN

Las redes neuronales están emergiendo como un área de investigación importante, tanto desde el punto de vista del diseño como de su utilización. Se centra el interés en los aspectos de diseño, con especial énfasis en los algoritmos de aprendizaje, más que sobre la eficiencia de la red; donde es notable el interés de la comunidad académica por las aplicaciones derivadas del uso de redes neuronales. Disciplinas tan diferentes como la biología o las finanzas desarrollan aplicaciones basadas en redes neuronales.

Como era lógico suponer las aplicaciones en ingeniería y las ciencias experimentales han sido numerosas y también en el dominio de las ciencias empresariales y de la gestión. El interés sobre la aplicación de las redes neuronales a estos dominios ha sido creciente en los últimos años. Este interés podemos inferirlo del volumen de artículos publicados en los años recientes y el interés en revistas de ámbitos específicos del dominio de la Investigación Operativa y la Ciencia Empresarial, además de las propias de redes neuronales. Así revistas tan prestigiosas como *Computers and Operations Research*, *The Journal of Intelligent Manufacturing*, *Decision Support Systems*, *International Journal of Forecasting* han publicado números dedicados a este tema, mientras que *European Journal of Operations Research* publicó un número especial en 1996, al igual que otras revistas de contrastado prestigio como *Neural Networks*, *Journal of Computational Intelligence in finance* o *Neural Computing* reconocen en sus páginas habitualmente artículos relacionados con las materias que abordamos en el presente trabajo. En lo que respecta a nuestro país ha habido un notable incremento de participación de investigadores españoles tanto en aspectos teóricos como en aplicaciones, dentro de éste último apartado cabe destacar entre otros a Gallizo, Otero, Trujillo y Serrano Cinca.

El objetivo del presente trabajo consiste en construir una metodología de aplicación de las redes neuronales de tipo Hopfield a los problemas de

optimización, dedicándole especial atención a un problema de particular interés como es el Problema del Viajante de Negocios, introduciendo mejoras, a nuestro juicio, importantes tanto en la calidad de las aproximaciones obtenidas, como en la velocidad de la convergencia de las redes.

En el apartado 2 establecemos las consideraciones dinámicas básicas a aplicar en los siguientes apartados. Tras estas primeras consideraciones, en el apartado 3, planteamos las estrategias de resolución de los problemas de optimización con redes de memoria asociativa. Una vez identificadas las características operativas del modelo de Hopfield, procedemos a construir y definir una metodología de resolución de los problemas de optimización con redes neuronales, donde exponemos los teoremas que avalan la metodología propuesta en el epígrafe 4.

Una vez construidas las herramientas de resolución de problemas de optimización a partir de redes neuronales, nos centramos, en el apartado 5, en los problemas de optimización combinatoria, específicamente, en el Problema del viajante de negocios. Para ello, proponemos un conjunto de algoritmos basados en la propuesta metodológica desarrollada en el apartado 4, incluyendo estrategias para el escape de mínimos locales, vinculados a la operativa del modelo de Hopfield, junto con un análisis de los resultados de la implementación realizada de tales algoritmos.

Para terminar exponemos las conclusiones que consideramos más relevantes.

Por último y con el objeto de permitir un seguimiento de las propuestas y desarrollos realizados, hemos agrupado las demostraciones de resultados teóricos más relevantes en el anexo I.

2.- CONSIDERACIONES DINAMICAS

Antes de abordar nuestro objetivo central, pasamos, aunque brevemente, a analizar las consideraciones dinámicas que requiere nuestro estudio. Aunque los

principios de las redes de Hopfield, también los incorporan la máquina de Boltzmann y las máquinas de la teoría del campo medio, las redes de Hopfield incluyen un característico y vital principio físico, el de almacenar información en una configuración dinámica estable. Aunque otros autores previos a Hopfield formularon aproximaciones a las redes neuronales artificiales, fue la formulación de Hopfield de 1982 la que introduce adecuadamente los términos del principio antes indicado. La idea de Hopfield consiste en identificar cada patrón ejemplar como un mínimo local de una función, llamada de energía, que representa el estado de la red, estado al que por similitud con los principios físicos en los que se basa se le denomina energía. A partir de ahí se utiliza un mecanismo para minimizar la función de "energía" de la red de tal manera que el valle de energía se constituye en una base de atracción del procedimiento de minimización.

Hopfield introduce una red a partir de la caracterización de los denominados "pesos sinápticos"¹ y a partir de ella define una función de energía que la significa, Hopfield (1982) y que toma la siguiente forma como función cuadrática:

$$H(k) = -\frac{1}{2} Y^T(k) W Y(k) = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N W_{ij} Y_i(k) Y_j(k) \quad (1)$$

donde $\{Y_h(k), h=1, \dots, N\}$ representa el conjunto de estados en el instante k de cada neurona h , y $\{W_{ij}, i, j = 1, \dots, N\}$ el conjunto de valores asignados a cada conexión entre la neurona i y la neurona j ; de esta manera $H(k)$ es el valor de la función de energía para la red completa en el instante de tiempo k . Como podemos observar, sería un caso particular de la red con memoria asociativa bidireccional (BAM), y podría demostrarse² que el límite superior e inferior de $H(k)$ para cualquier k viene dado respectivamente por $-(1/2) \sum_{i=1}^N \sum_{j=1}^N |W_{ij}|$ y $(1/2) \sum_{i=1}^N \sum_{j=1}^N |W_{ij}|$.

¹ El concepto de peso sináptico se "importa" del sistema nervioso, de manera que de allí se recoge la relación entre neuronas, contactos sinápticos, y se traduce al valor que se asigna a las relaciones entre las neuronas del modelo de Hopfield

² La demostración de este resultado puede encontrarse en Ruiz (1999) y de extraordinario valor al concluir que la función de energía de Hopfield, tal como la hemos formulado, es una función acotada.

Sabemos que el potencial sináptico es una función de los estados actuales de las N neuronas de la red, pero ¿bajo qué condiciones podemos asegurar que disminuirá hasta estabilizarse en un mínimo local?

Para responder a la pregunta, formulamos ahora una notable contribución que creemos que es de especial relevancia dado el carácter de generalización que supone encontrar, en la versión recíproca propuesta inicialmente por Hopfield, cuál es la forma de los pesos sinápticos a partir de una función de energía de tipo Hopfield.

Teorema 1

Consideremos una red neuronal con N neuronas de McCulloch-Pitts con histéresis, Zurada (1992), donde las aristas tienen asignados unos pesos sinápticos dados por la matriz W . Sea el vector $Y(k)$ el vector de estados de la red. Si la dinámica de computación de la red en cada iteración viene dada por

$$Y_i(k) = \begin{cases} +1 & \text{si } U_i(k) > T_{i2} \\ Y_i(k) - 1 & \text{si } T_{i1} \leq U_i(k) \leq T_{i2} \\ -1 & \text{si } U_i(k) < T_{i1} \end{cases} ; \text{ con } T_{i1} \leq 0 \leq T_{i2} \quad (2)$$

Eligiendo aleatoriamente una neurona, L , cada vez que actualizamos su estado, podemos afirmar que la función de energía $H(Y(k))$ decrece cuando y sólo cuando el potencial sináptico viene dado por la expresión:

$$U_L(k) = -\omega \frac{\Delta H(Y(k))}{\Delta Y_L(k)} \quad (3)$$

donde $\omega > 0$ es la *tasa de actualización*.

De esta manera tenemos caracterizado al potencial sináptico de una red neuronal de estas características. Como consecuencia de esta caracterización, podemos concluir el siguiente resultado:

Corolario 1

Para una función de energía cuadrática

$$H(\mathbf{y}) = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N w_{ij} y_i y_j + \sum_{i=1}^N \theta_i y_i \quad (4)$$

la dinámica de la red definida no genera un incremento de la función de energía cuando y sólo cuando el potencial sináptico viene dado por la expresión:

$$U_i = \sum_{j=1}^N w_{ij} y_j - \theta_i \quad (5)$$

Estamos ahora en condiciones de abordar la resolución de los problemas de optimización a partir de la construcción de una función de energía "ad hoc" para dicho tipo de problemas.

3 - RESOLUCIÓN DE PROBLEMAS DE OPTIMIZACIÓN CON REDES DE MEMORIA ASOCIATIVA

Hopfield, tras proponer en 1982 la utilización de redes neuronales con unidades binarias o bipolares con actualización asíncrona aleatoria como memoria asociativa, observó que podía obtener las mismas propiedades computacionales utilizando una versión determinística en tiempo continuo con unidades que tienen una función de activación continua y monótonamente creciente, como, por ejemplo, una función alisada Hopfield (1984). Esta red es referenciada algunas veces como una red de Hopfield de tipo gradiente, Zurada (1992), o red de Hopfield con actualización continua Hertz, Krogh y Palmer (1991).

Hopfield y Tank (1985) concluían, en definitiva, que:

- 1) La función de coste de un problema de optimización puede ser expresada como una función cuadrática que define una función de energía para una red neuronal artificial de tipo Hopfield.
- 2) Una solución para el problema de optimización puede construirse utilizando hardware analógico y además radicalmente diferente de la implementación de un algoritmo en un ordenador digital. Los pesos y *sesgo* se pueden determinar

comparando la función de coste para un problema concreto con la función de energía.

Por otra parte, Hopfield y Tank (1985) y (1986) presentaron directamente como ejemplos la utilización de su dinámica para obtener soluciones a problemas de:

- 1) Conversión analógica/digital
- 2) Descomposición/decisión de señal (para determinar la descomposición de una señal particular dado el conocimiento de sus componentes individuales)
- 3) Programación lineal Peretto (1992)
- 4) Problema del viajante de comercio (TSP³).

No obstante, las aplicaciones posibles llevadas a cabo por otros investigadores son mucho más numerosas y han emergido de la idea inicial de Hopfield, especialmente en lo relativo a economía y administración de empresas, de un lado, y la investigación operativa de otro.

Por último, es importante resaltar que es posible demostrar sólo que, dadas las restricciones propuestas, la red converge a un mínimo local de la función de energía. Sin embargo, en general, dicho mínimo local no es el mínimo global. Por eso la aproximación de Hopfield es mejor acomodarla a problemas donde hay varios mínimos locales que dan soluciones satisfactorias y es más importante aproximar rápidamente a una “buena” solución que hacer el proceso mucho más largo y tener la mejor solución posible. Uno puede argumentar que éstos son la clase de problemas que los sistemas biológicos tienen que resolver Peretto (1992); pero no resulta siempre fácil decidir si un problema de optimización particular con un conjunto específico de parámetros será adecuado para que sea resuelto utilizando la dinámica de la red de Hopfield. No obstante, para los problemas de optimización combinatoria sí podemos afirmar que las características intrínsecamente paralelizables de la dinámica de Hopfield, se ajustan

³ Del inglés **T**ravelling **S**alesman **P**roblem.

adecuadamente a los métodos de resolución eficientes y eficaces que se utilizan para resolver este tipo de problemas.

Esta dinámica, como veremos a continuación, se centra para cada neurona en considerar el valor de activación de las restantes y combinar todos ellos, con la ponderación que supone el peso sináptico para cada conexión (contribución de cada neurona a la activación de la otra), hasta alcanzar el valor de entrada en la neurona que corresponda para ver si se activa o no, cambia o no de estado, para continuar con su contribución a las restantes neuronas.

3.1- Características operativas del modelo de Hopfield

Para analizar estas características operativas del modelo de Hopfield nos vamos a basar, como antes, en el modelo de neurona de McCulloch-Pitts como unidad básica de procesamiento.

Sabemos que en este modelo cada neurona tiene dos estados definidos por los niveles de activación que pueden actuar sobre ella. El estado “activo” de la neurona i lo denotaremos por $s_i = +1$, mientras que el estado “desactivado” de la misma neurona lo indicaremos por $s_i = -1$. Para una red de N neuronas, el estado de la red vendrá definido por un vector s , cuyas componentes serán los estados de cada neurona.

También debemos asumir que el estado discreto de estas neuronas corresponde con el límite en el tiempo de las respectivas variables x_i , en el caso de valores continuos que obtendríamos de un modelo continuo, bajo las siguientes condiciones:

- a) La red se aproxima a un estado estable o de equilibrio en el tiempo.
- b) Las ponderaciones sinápticas w_{ij} representan la contribución de las salidas s_i a la activación de la neurona j .
- c) La función de activación de la neurona j será la suma de todas las contribuciones de las neuronas de la red a este fin.

$$v_j = \sum_{i=1}^N w_{ij} s_i - \theta_j \quad (6)$$

Donde θ_j es el umbral de activación fijado externamente, y por tanto, el valor v_j representa la entrada neta de la neurona j que sirve para su activación o no en el instante que corresponda.

Conforme a estas consideraciones el estado de la neurona j cambia siguiendo la siguiente regla:

$$s_i(k) = \begin{cases} +1 & \text{si } v_i(k) > T_{i2} \\ s_i(k-1) & \text{si } T_{i1} \leq v_i(k) \leq T_{i2} \\ -1 & \text{si } v_i(k) < T_{i1} \end{cases} ; \text{ con } T_{i1} \leq 0 \leq T_{i2} \quad (7)$$

Es decir, nos encontramos que la evolución de la red de Hopfield representa una realidad que viene representada por el estado de cada neurona y las relaciones que entre ellas existen, con un peso asignado que viene recogido por los "pesos sinápticos".

De esta manera, como ya hemos visto, la evolución dinámica del modelo viene representado por la función de energía, de manera que tiende a los estados "naturalmente" estables, valles de la función y estos "valles" se encuentran asociados con los mínimos locales de la citada función.

Por todo ello, para alcanzar los estados estables debemos hacer evolucionar en el tiempo la aproximación a los mismos y la gran ventaja es que esa evolución, que se encuentra garantizada, converge hacia alguno de los valores estables.

4 - METODOLOGÍA DE RESOLUCIÓN DE PROBLEMAS DE OPTIMIZACION CON REDES NEURONALES.

Ya hemos indicado como Hopfield y Tank (1985) utilizaron redes recursivas de tipo Hopfield para resolver problemas de optimización. La idea fundamental consiste en:

- a) Una red recurrente construida de acuerdo con los principios enumerados en los párrafos anteriores siempre converge hacia un estado de equilibrio

estable, invariante dinámico.

- b) Los invariantes dinámicos que se alcanzan se corresponden con mínimos locales de la función de energía definida.
- c) La función de energía depende de los pesos sinápticos asignados.
- d) Para resolver los problemas de optimización debemos asignar unos pesos sinápticos que representen el problema a resolver.

En estas condiciones para resolver un problema de optimización deberemos seguir los siguientes pasos:

1. Encontrar una correspondencia entre el conjunto de estados posibles y el conjunto de soluciones posibles del problema.
2. Construir una función de energía que incorpore la función objetivo del problema de optimización y la satisfacción de las restricciones.
3. Deducir los pesos sinápticos de la función de energía.
4. Ejecutar la red siguiendo la dinámica de Hopfield.

Podemos observar como la clave de la metodología consiste en encontrar la función de energía que represente el problema pues, tras su formulación analítica podemos deducir la topología de la red que representa. Por otra parte, la función de energía debe incorporar, como decíamos, la información suficiente para poder evolucionar a un óptimo local de la función objetivo satisfaciendo las restricciones correspondientes. Para ello es preciso que tenga una estructura similar a la siguiente:

$$E(x) = E_{\text{coste}}(x) + \sum \lambda_i E_i(x) \quad (8)$$

Donde $E(x)$ es la función de energía, $E_{\text{coste}}(x)$ la función de coste que incluye, habitualmente, la función objetivo y $E_i(x)$ la penalización si no satisface la restricción i del problema, siendo λ_i la ponderación de la restricción i .

Una vez realizadas estas consideraciones, bastará concretar cual es la relación entre la función de energía y los problemas de optimización para conseguir que la evolución de la red nos lleve a un estado estable que coincida con el óptimo buscado.

4.2 - Relación entre la función de energía de Hopfield y los problemas de optimización

Un problema de optimización puede expresarse en forma estándar como:

$$\text{Min } f(x): \text{ con } x \in X, g(x) \leq 0, h(x) = 0$$

Donde $X \subset \mathbb{R}^p$.

$$\begin{aligned} f: X \subset \mathbb{R}^p &\longrightarrow \mathbb{R} \\ g: X \subset \mathbb{R}^p &\longrightarrow \mathbb{R}^n \\ h: X \subset \mathbb{R}^p &\longrightarrow \mathbb{R}^m \end{aligned}$$

Para un problema general expresado de esta forma, definimos la función de Lagrange o Lagrangiana como:

$$L(x; \lambda, v) = f(x) + \lambda \cdot g(x) + v \cdot h(x), \text{ con } \lambda \geq 0, \lambda \in \mathbb{R}^n, v \in \mathbb{R}^m$$

Puede demostrarse que los óptimos de esta función verifican las denominadas condiciones de primer orden, que pasan a ser condiciones necesarias con la cualificación de restricciones, en todo caso las también denominadas condiciones de Kuhn-Tucker nos conducen a afirmar que la búsqueda de nuestros óptimos tenemos

que realizarla, exclusivamente, entre los puntos que verifican:

Condiciones de primer orden

$$\nabla f(x^*) + \lambda^* \cdot \nabla g(x^*) + \nu^* \cdot \nabla h(x^*) = 0 \quad (9)$$

donde $\lambda^* \geq 0$, $\lambda^* \in \mathbb{R}^n$ y $\nu^* \in \mathbb{R}^m$

A partir de este problema, así formulado, proponemos a continuación una construcción de la función de energía que nos permita modelizar y representar los problemas de optimización mediante redes neuronales.

Sea

$$E(x; \lambda, \nu; \kappa, \kappa') = L(x; \lambda, \nu) + \kappa m^2(x) + \kappa' h^2(x) \text{ siendo } \kappa, \kappa' \in \mathbb{R}; \kappa, \kappa' > 0 \quad (10)$$

Donde $m(x)$ es una transformación de $g(x)$ que garantice el valor nulo para puntos que verifican la restricción correspondiente y coincida con $g_i(x)$ para el conjunto de puntos que no verifican la restricción.

Es decir:

$$m_i(x) = \begin{cases} 0 & \text{si } g_i(x) \leq 0 \\ g_i(x) & \text{si } g_i(x) > 0 \end{cases} \quad (11)$$

y por tanto

$$\nabla m_i = \begin{cases} 0 & \text{si } g_i(x) \leq 0 \\ \nabla g_i(x) & \text{si } g_i(x) > 0 \end{cases}$$

$$L(x; \lambda, \nu) = f(x) + \lambda \cdot g(x) + \nu \cdot h(x), \text{ con } \lambda \geq 0; \lambda \in \mathbb{R}^n; \nu \in \mathbb{R}^m$$

El gradiente de E sería entonces:

$$\nabla E = \nabla L + 2\kappa m \cdot \nabla m + 2\kappa' h \cdot \nabla h$$

$$\begin{aligned}
&= \nabla f + \lambda \cdot \nabla g + \nu \cdot \nabla h + 2\kappa m \cdot \nabla m + 2\kappa' h \cdot \nabla h \\
&= \nabla f + \sum_{i \in I_1} \lambda_i \nabla g_i + \sum_{i \in I_2} \{\lambda_i + 2\kappa g_i\} \nabla g_i + \sum_{j=1}^m \{\nu_j + 2\kappa' h_j\} \nabla h_j \quad (12) \\
&\text{donde } I_1 = \{i / g_i(x) \leq 0\} \text{ e } I_2 = \{i / g_i(x) > 0\}
\end{aligned}$$

En estas condiciones podemos observar cómo la dirección del gradiente de la función energía E, es la composición de varias direcciones, de manera que:

- a) ∇f marca la dirección de máximo crecimiento de f
- b) ∇g_i apunta en la dirección de máximo crecimiento de g_i , pero como está precedido por el factor $(\lambda_i + 2\kappa g_i)$, sólo cuando el mismo sea positivo apuntará en la dirección indicada. Es decir:
 - I. Si $g_i(x) > 0$, dado que $\lambda_i, \kappa \geq 0$ entonces $(\lambda_i + 2\kappa g_i) \geq 0$
 - II. Si $g_i(x) \leq 0$, dado que $\lambda_i \geq 0$ entonces el factor es no negativo.
- c) ∇h_j apunta en la dirección de máximo crecimiento de h_j , pero como está precedido por el factor $(\nu_j + 2\kappa' h_j)$, sólo cuando el mismo sea positivo apuntará en la dirección indicada. Es decir:
 - I. $(\nu_j + 2\kappa' h_j) > 0$ cuando $h_j(x) > -\nu_j / (2\kappa')$
 - II. $(\nu_j + 2\kappa' h_j) < 0$ cuando $h_j(x) < -\nu_j / (2\kappa')$

Tras estas consideraciones que nos permiten identificar los comportamientos de la función de energía, podemos proceder a establecer formalmente la relación entre los mínimos locales del problema original y los óptimos de la función de energía E.

Teorema 2

Sea $E(x; \lambda, \nu; \kappa, \kappa')$ la función energía y sea $(x^*; \lambda^*, \nu^*; \kappa, \kappa')$ un punto que satisface las condiciones de primer orden definidas previamente. Si $E(x^*; \lambda^*, \nu^*; \kappa, \kappa')$ es un mínimo local de $E(x; \lambda, \nu; \kappa, \kappa')$ con respecto a x para algunos valores

finitos κ y $\kappa' > 0$ entonces $f(x^*)$ es un mínimo local de $f(x)$ del problema de optimización en forma estándar.

Una consideración elemental del anterior teorema nos permite afirmar que si el punto $(x^*; \lambda^*, \nu^*; \kappa, \kappa')$ es un mínimo local estricto, entonces el punto x^* es un mínimo local estricto del problema original.

El teorema recíproco, no es cierto; existen varios contraejemplos que lo justifican⁴. Por tanto tendremos que reconstruir las hipótesis para conseguir la tesis del recíproco.

Teorema 3

Sea $(x^*; \lambda^*, \nu^*)$ un punto que satisface las condiciones de primer orden. Si x^* es un mínimo local de $f(x)$ de un problema de optimización asociado a la función de energía E , entonces:

a) $(x^*; \lambda^*, \nu^*; \kappa, \kappa')$ es un mínimo de la función de energía E con respecto a la variable x para algunos κ, κ' finitos positivos.

o bien

b) en el límite cuando $\kappa, \kappa' \rightarrow \infty$ $E(x^*; \lambda^*, \nu^*; \kappa, \kappa')$ aproxima un mínimo local de la función de energía.

Al igual que en el teorema anterior, podemos afirmar que en el supuesto que el mínimo sea estricto el apartado a) del teorema se satisface estrictamente.

Podemos observar, también, cómo el contraejemplo del recíproco del teorema [2], recogido en la notaⁱⁱⁱ, queda integrado en el teorema [3], extendiendo de manera importante el teorema [2]. Por tanto, con los dos teoremas anteriores [2],[3] y dadas las características de la dinámica de la red de Hopfield, hemos caracterizado la función de energía que representa a los problemas de optimización expresados en forma estándar, estableciendo la obtención de soluciones a los

⁴ Rochafellar en carta dirigida a D.A. Pierre en 1973 propone el siguiente ejemplo, donde $f(x) = x_2 - x_1^4 - x_1 x_2$, $h(x) = x = 0$. Para el citado ejemplo la función de energía posee un punto de silla en $(0,0)$ y el máximo lo obtiene en $\pm((8\kappa')^{-1/2}, -(32\kappa'^3)^{-1/2})$

problemas generales de optimización con restricciones, mediante redes de tipo Hopfield.

En este punto es conveniente resaltar cuáles han sido los aspectos significativos de la demostración de ambos teoremas [2] y [3]. En este sentido es preciso destacar como la única propiedad que se ha utilizado en los mismos relativa a las funciones de penalización ha sido su no negatividad, por tanto podrían sustituirse las mismas por composiciones de las funciones de restricción que mantuviesen dicha propiedad, por ejemplo: $\max\{0, g_i(x)\}, |g_i(x)|$.

En el caso de utilizar como funciones de penalización $\max\{0, g_i(x)\}$ ó $|g_i(x)|$, las condiciones de segundo orden se utilizan con una dificultad añadida de no poder obtenerlas en algunos puntos, al no ser derivables de segundo orden tales funciones en tales puntos. Este problema podría superarse sin más que añadir un sesgo independiente de la variable x que alejase los puntos de discontinuidad de la primera derivada de los puntos de interés en nuestro análisis, lo que al ser un término independiente de la variable x no afectaría al resultado de la obtención de los valores óptimos y, por otra parte, al ser conocido el sesgo introducido puede recuperarse el valor de la función objetivo sin dificultad alguna.

Tras estas consideraciones estamos en condiciones de proponer una red de Hopfield para los problemas de optimización que, teniendo en cuenta los resultados de este apartado, caracterice y resuelva dichos problemas. Un estudio detallado de los problemas de optimización no lineal, cuadrática o lineal puede encontrarse en Ruiz(1999)

5. - PROBLEMAS DE OPTIMIZACION COMBINATORIA.

EL PROBLEMA DEL VIAJANTE DE COMERCIO

Abordamos ahora la resolución de un tipo de problemas, los de optimización combinatoria, donde la potencialidad de las redes neuronales se manifiesta en grado superlativo. Como caso particular de este tipo de problemas

hemos elegido el problema del viajante, tanto por su ejemplificación y dureza ⁵ de su solución, como por la simplicidad de formulación que lleva aparejada la complejidad de la obtención de su solución; además de la abundancia de trabajos en este sentido donde se manifiesta la importancia científica y aplicada del mismo. Por todo ello le dedicaremos una especial atención.

El problema del viajante es un problema de optimización combinatoria que ha sido objeto de estudio por parte de muchos investigadores. La mayoría de los intentos por resolver este problema con una red de Hopfield han utilizado exclusivamente el método de las funciones de penalización. Sin embargo, este método presenta serios problemas de mínimos locales y para solucionarlos se han empleado técnicas de recalentamiento simulado o algoritmos de escape de mínimos locales. Por otra parte, cuando se trata de problemas de grandes dimensiones, el recalentamiento simulado es impracticable debido a su lentitud y los otros algoritmos se vuelven ineficientes por lo que deben combinarse con otras técnicas, como por ejemplo, métodos de partición de red.

Podríamos decir, incluso, que el problema del viajante es un problema que ha ejercido una fascinación particular en los investigadores de optimización combinatoria, probablemente porque es fácil de formular pero bastante difícil de resolver, una visión muy clara y que describe muy bien el mecanismo de obtención de soluciones, puede encontrarse en: <http://onso.cps.unizar.es/grupos/n-rulas>

Formulación del problema del viajante de comercio (TSP):

Dadas N ciudades numeradas desde 1 hasta N y donde d_{XY} representa la distancia entre la ciudad X y la ciudad Y , el TSP consiste en encontrar el

⁵ Término computacional de la complejidad de la obtención de la solución. Los problemas NP-duros son una clase de problemas de al menos la complejidad de los NP-completos. Siendo los problemas NP-completos aquellos problemas que o son resolubles en tiempo polinomial o no lo son. Profundizar en este apartado generaría una abundante disertación que, a nuestro juicio, no es objeto de este trabajo.

circuito cerrado más corto que visite exactamente una vez cada ciudad.

Unicamente se va a considerar el TSP simétrico donde $d_{XY}=d_{YX}$.

El problema de encontrar el recorrido óptimo es NP-completo, es decir, no se conoce un algoritmo que lo resuelva en tiempo polinomial. Aunque para este tipo de problemas siempre existe algún algoritmo elemental que determina la solución óptima (enumeración completa), dicho algoritmo no es eficiente ya que el tiempo de cálculo crece exponencialmente con el tamaño del problema.

Dada la dificultad práctica para resolver de forma exacta este tipo de problemas, comenzaron a aparecer una serie de algoritmos basados en métodos heurísticos que proporcionan soluciones factibles (es decir, soluciones que satisfacen las restricciones del problema) las cuales, aunque no optimizan la función objetivo, al menos se acercan al valor óptimo en un tiempo de cálculo razonable.

En particular, para resolver el TSP a través de una red neuronal, es necesario encontrar una representación adecuada en términos de una función de energía de manera que el mínimo global de dicha función se corresponda con la solución del problema.

El modelo de Hopfield y sus versiones modificadas convierten un problema de optimización con restricciones en otro problema de optimización equivalente sin restricciones utilizando el método de las funciones de penalización. Así, si se desea convergencia, los términos de penalización que ponderan la violación de las restricciones deben ser suficientemente grandes y, por lo tanto, la función objetivo queda en un segundo plano, obteniéndose de este modo, un mayor número de soluciones válidas pero de muy baja calidad. Por otra parte, si se desea calidad en las soluciones, los términos de penalización no deben ser muy grandes para no anular a la función objetivo. Con esto, las soluciones obtenidas se encuentran más próximas al óptimo global a costa de tener un número menor de caminos válidos. Este es el problema que presentan los métodos basados en las funciones de penalización: *si se desea convergencia rápida, la calidad de las soluciones es*

baja, y si lo que se pretende es obtener soluciones con una buena calidad, la convergencia es lenta.

5.1 - Propuesta metodológica de resolución del problema del viajante con redes neuronales artificiales

Recientemente, Li (1996) propone una nueva aproximación para resolver el TSP con una red de Hopfield. Este método, que combina los multiplicadores de Lagrange y las funciones de penalización para resolver un problema de optimización con restricciones, supera los problemas que ambos métodos presentan cuando se utilizan por separado.

Como muestra Li (1996), se obtienen mejores resultados en cuanto a convergencia y calidad que en los hasta ahora desarrollados basados en el modelo original de Hopfield. El método propuesto resuelve el conflicto entre la calidad de la solución y la convergencia que ha sido el mayor problema con las redes neuronales de tipo Hopfield. Sin embargo, aunque la convergencia no se puede mejorar más, puesto que con el método se obtiene el 100% de caminos válidos, las soluciones encontradas se siguen correspondiendo con mínimos locales. Es más, para problemas de grandes dimensiones la calidad va decreciendo (la solución se aleja cada vez más del óptimo buscado).

Para evitar el problema de los mínimos locales se han venido utilizando diferentes técnicas heurísticas de búsqueda. Algunas de ellas utilizan algoritmos basados en el recalentamiento simulado. La base de estos algoritmos consiste en que mediante la introducción de ruido, la red ocasionalmente aceptará un estado con un nivel de energía mayor con el fin de escapar de los mínimos locales Cerny (1985). Sin embargo, la excesiva lentitud de esta técnica lo convierte en un método impracticable para problemas de grandes dimensiones.

Otros intentos por encontrar el mínimo global de la función de energía utilizan algoritmos de escape de mínimos locales Cheung y Lee (1992), de tal manera que si la red queda atrapada en un estado de mínimo local, se usa el algoritmo para encontrar un nuevo estado de menor o igual energía.

Recientemente, Peng, Gupta y Armitage (1996) utilizan un algoritmo sobre una red de Hopfield para resolver el TSP que supera en calidad y rapidez al recalentamiento simulado. Sin embargo, la eficacia del algoritmo decrece a medida que aumentan las dimensiones del problema por lo que estos autores proponen un algoritmo que combina el algoritmo con una técnica de partición aleatoria de red para mejorar los mínimos locales en problemas de gran tamaño. Sin embargo, este método tampoco resuelve el problema y, además, introduce una mayor complejidad a la hora de resolver el TSP.

En este apartado se propone un algoritmo de escape de mínimos locales pero en lugar de aplicarlo al modelo de Hopfield original toma como base la metodología que proponemos en este capítulo puesto que es con el que mejores resultados se han obtenido hasta el momento.

La aproximación inicial se basó en la formulación del TSP para la programación entera:

$$\begin{aligned}
 \text{Min} \quad & \sum_{X=1}^N \sum_{\substack{Y=1 \\ X \neq Y}}^N \sum_{i=1}^N d_{XY} v_{Xi} (v_{Yi+1} + v_{Yi-1}) \\
 \text{Sujeto a} \quad & \sum_{X=1}^N v_{Xi} = 1 \quad \forall X \in \{1, \dots, N\} \\
 & \text{y} \quad \sum_{i=1}^N v_{Xi} = 1 \quad \forall i \in \{1, \dots, N\} \\
 & \text{y} \quad v_{Xi} \in \{0, 1\} \quad \forall X, i \in \{1, \dots, N\}
 \end{aligned} \tag{17}$$

donde d_{XY} representa la distancia entre las ciudades X e Y . $v_{Xi}=1$ si la ciudad X es visitada en la posición i dentro del camino. Los subíndices X e Y se refieren a las ciudades mientras que los subíndices i y j se refieren a las posiciones en el camino. La primera restricción indica que cada posición contiene exactamente una ciudad y la segunda, representa que cada ciudad ocupa exactamente una posición.

Basándose en esta formulación, Hopfield y Tank construyeron una red de $N \times N$ neuronas completamente conectadas, donde cada columna se corresponde con

una ciudad y cada fila con la posición en la que es visitada la ciudad. El estado de las neuronas se almacena en una matriz $V_{N \times N}$.

Para resolver el TSP, el modelo de Hopfield convierte un problema de optimización con restricciones en otro sin restricciones utilizando el método de las funciones de penalización. Para ello, se propuso la siguiente función de energía que intenta, simultáneamente, imponer las restricciones y obtener el menor valor de la función objetivo:

$$E(v) = \frac{A}{2} \sum_{X=1}^N \sum_{i=1}^N \sum_{\substack{j=1 \\ j \neq i}}^N v_{Xi} v_{Xj} + \frac{B}{2} \sum_{i=1}^N \sum_{X=1}^N \sum_{\substack{Y=1 \\ Y \neq X}}^N v_{Xi} v_{Yi} + \\ + \frac{C}{2} \left\| \sum_{X=1}^N \sum_{i=1}^N v_{Xi} - N \right\|^2 + \frac{D}{2} \sum_{X=1}^N \sum_{Y=1}^N \sum_{i=1}^N d_{XY} v_{Xi} \left\| v_{Yi+1} + v_{Yi-1} \right\| \quad (18)$$

donde A, B y C son constantes que ponderan las restricciones que debe cumplir el problema, mientras que la constante D permite ponderar la función objetivo.

Al principio cada v_{Xi} se elige aleatoriamente con valores en el intervalo $[0,1]$. A medida que la red progresa, el estado de las neuronas va cambiando de manera que la energía total disminuye hasta que la red se estabiliza en una de las esquinas del hipercubo unidad ($v_{Xi} \in \{0,1\}$).

La relación entre el estado de las neuronas, v , y la salida de las neuronas, u , viene dada por una ecuación similar a la logística definida en (4):

$$v_{Xi} = \frac{1}{1 + e^{-u_{Xi}/\tau}} \quad (19)$$

Aunque con esta aproximación es posible encontrar caminos válidos, la convergencia a soluciones factibles falla incluso cuando $N=10$. Es más, las soluciones válidas encontradas tienen una baja calidad y a medida que el tamaño del problema crece, la calidad se va haciendo cada vez peor (se va alejando cada vez más del óptimo global).

Es por esto que muchos investigadores han intentado varios métodos para superar este esquema básico. Estos métodos incluyen modificaciones de la función de energía o técnicas para evitar los mínimos locales.

Es el caso de Brandt y otros (1988) que proponen una función de energía modificada donde los estados válidos se ven favorecidos en detrimento de la minimización de la función objetivo, y aunque la convergencia aumenta, la calidad de las soluciones continúa siendo muy baja. Protzel, Palumbo y Arras (1993) intentan mejorar la calidad de las soluciones para la formulación de Brandt utilizando diferentes parámetros. S. Abe (1993) utiliza la función de energía del modelo original de Hopfield excepto para los términos que representan las restricciones, que incluyen menos parámetros:

$$E(v) = \frac{A}{2} \sum_X \left\| \sum_i v_{Xi} - 1 \right\|^2 + \frac{B}{2} \sum_i \left\| \sum_X v_{Xi} - 1 \right\|^2 + \frac{D}{2} \sum_{X=1}^N \sum_{Y=1}^N \sum_{i=1}^N d_{XY} v_{Xi} (v_{Yi+1} + v_{Yi-1}) \quad (20)$$

Sin embargo, con estos métodos es difícil obtener convergencia y alta calidad simultáneamente ya que, por una parte, los términos de penalización deben ser lo suficientemente grandes para obtener convergencia a soluciones factibles, pero, por otra, a medida que dichos términos se van haciendo más fuertes, la función objetivo se vuelve más débil lo que significa una baja calidad en los resultados.

Haciendo uso de la metodología propuesta podemos superar los problemas anteriormente enunciados, ya que las restricciones se satisfacen sin necesidad de que los términos de penalización sean excesivamente grandes, y, por tanto aliviamos el típico problema de oscilación que presentan los tradicionales multiplicadores de Lagrange.

Al igual que los métodos anteriores, el problema de optimización combinatoria se convierte en un problema con restricciones:

$$\text{Min } E(v) = \frac{1}{2} \sum_{X=1}^N \sum_{i=1}^N \sum_{\substack{Y=1 \\ Y \neq X}}^N d_{XY} v_{Xi} \left[v_{Yi+1} + v_{Yi-1} \right] \quad (21)$$

sujeto a $C_k(v) = 0$, $v_{Xi} \geq 0 \quad \forall X, i$

$E(v)$ es la función objetivo, mientras que $C_k(v)$ son funciones reales que representan restricciones de igualdad y toman el valor 0 cuando la restricción se satisface:

$$\begin{aligned} C_1^i(v) &= \sum_{X=1}^N v_{Xi} - 1 = 0 \quad \forall i \\ C_2^X(v) &= \sum_{i=1}^N v_{Xi} - 1 = 0 \quad \forall X \\ C_3^{Xj}(v) &= \sum_{\substack{j=1 \\ j \neq i}}^N v_{Xi} v_{Xj} = 0 \quad \forall X, i \\ C_4^{Xj}(v) &= \sum_{\substack{Y=1 \\ Y \neq X}}^N v_{Xi} v_{Yi} = 0 \quad \forall X, i \\ C_5^{Xj}(v) &= v_{Xi} \left[1 - v_{Xi} \right] = 0 \quad \forall X, i \end{aligned} \quad (22)$$

Para llegar a la solución final v^* , se tiene que cumplir una condición adicional:

$$v_{Xi}^* \in \{0,1\} \quad \forall X, i$$

De esta manera la solución final representará una de las esquinas del hipercubo $[0,1]^{N \times N}$.

Para construir la siguiente función de energía, aplicamos la metodología propuesta y la dinámica de Hopfield:

$$E(v, \lambda, \kappa) = E(v) + \sum_{k=1}^5 \lambda_k C_k(v) + \frac{1}{2} \sum_{k=1}^5 \kappa_k \left[C_k(v) \right]^2 \quad (23)$$

Con esta formulación no sólo se obtiene el 100% de convergencia en todas las soluciones, sino que además la calidad de las soluciones supera ampliamente a

las obtenidas con el modelo de Hopfield y sus versiones con una función de energía modificada basada simplemente en la adecuada consideración de las penalizaciones. Sin embargo, para problemas de grandes dimensiones, las soluciones encontradas por el método propuesto siguen representando mínimos locales.

Con el fin de mejorar los mínimos locales, muchos investigadores han intentado combinar la red de Hopfield con diferentes métodos. Uno de ellos es el recalentamiento simulado, consistente éste en una técnica de búsqueda local basada en los conceptos de termodinámica, que se ha venido utilizando desde principios de los años 80 para resolver problemas de optimización Kirkpatrick, Gelatt y Vecchi (1983), Cerny (1985). Sin embargo, a diferencia de la mayoría de métodos de búsqueda local, en los que se emplea una estrategia estrictamente descendente (es decir, el cambio de estado únicamente se realiza si se encuentra una solución mejor), esta técnica permite, con una cierta probabilidad (cada vez menor, a medida que nos vamos acercando a la solución óptima), el paso a soluciones peores con el fin de reducir la posibilidad de caer en un mínimo local y quedar atrapados en ellos.

Básicamente, consiste en un proceso iterativo que gradualmente va disminuyendo un parámetro de control denominado *Temperatura*. Mediante la introducción de ruido en la red, ésta aceptará ocasionalmente un estado con un nivel de energía mayor con el fin de escapar de los mínimos locales. El proceso comienza con una alta temperatura permitiendo una alta probabilidad de aceptar un incremento de energía. A medida que la red evoluciona, la temperatura se va reduciendo gradualmente, disminuyendo también la posibilidad de cambios a soluciones peores cuando ya nos hallamos más cerca del óptimo buscado (el óptimo global).

Aunque se ha demostrado que este algoritmo es capaz de encontrar siempre la solución óptima, independientemente de la solución inicial de la que se parta, la garantía de encontrar este óptimo global sólo se alcanzará tras un número infinito

de pasos en el peor de los casos. Por lo tanto, la convergencia sólo puede ser aproximada en la práctica. Incluso para problemas sencillos, el tiempo necesario para conseguir la convergencia es exponencial ya que si se desea llegar al óptimo global, la reducción de temperatura durante el proceso de convergencia debe realizarse muy lentamente. Concretamente, para el TSP de 100 ciudades sería impracticable ya que necesitaría varios días de cálculo para llegar a una solución.

Así pues, para problemas de gran tamaño, es necesario aplicar otros métodos que permitan la convergencia a una solución más rápidamente. En este sentido, varios investigadores han utilizado algoritmos de escape de mínimos locales Cheung y Lee (1992), de tal manera que si la red queda atrapada en un estado de mínimo local, se usa el algoritmo para encontrar un nuevo estado de menor o igual energía.

Otra posible solución se aporta desde el algoritmo Peng, Gupta y Armitage (1996), inspirado por el recalentamiento simulado y que, básicamente, consiste en una combinación de una técnica de perturbación de la red sobre el modelo de Hopfield, de su aplicación podríamos obtener una buena estrategia de búsqueda.

Después de analizar todas las posibilidades seleccionamos una combinación adecuada de la aportación que, a nuestro juicio, realiza cada una de ellas en cada fase de la solución del problema y de ahí, al aplicarla a la formulación y propuesta de resolución antes planteada, pasamos a considerar la propuesta metodológica para la solución de los problemas de optimización combinatoria, en su particular acepción del problema TSP, procediendo a evaluar, posteriormente, la aportación específica de cada una.

Teniendo en cuenta la metodología propuesta y los resultados obtenidos hemos de volver a nuestro objetivo central, resolver el TSP mediante la metodología propuesta en el presente trabajo, lo que implica minimizar la función:

$$E(v, \lambda, \kappa) = E(v) + \sum_{k=1}^5 \lambda_k C_k(v) + \frac{1}{2} \sum_{k=1}^5 \kappa_k \|C_k(v)\|^2 \quad (24)$$

Así, para encontrar una solución local al problema se debe resolver el siguiente sistema de ecuaciones diferenciales:

$$\begin{aligned} \frac{dv_{Xi}}{dt} &= -\mu \nabla_v E(v, \lambda, \beta) = -\mu \left[\frac{\partial E(v)}{\partial v_{Xi}} + \sum_{k=1}^5 \lambda_k \frac{\partial C_k(v)}{\partial v_{Xi}} + \sum_{k=1}^5 \beta_k C_k(v) \frac{\partial C_k(v)}{\partial v_{Xi}} \right] \\ \frac{d\lambda_k}{dt} &= +\rho \nabla_\lambda E(v, \lambda, \beta) = \rho C_k(v) \end{aligned} \tag{25}$$

La dinámica de la red se implementa según las siguientes ecuaciones:

$$\begin{aligned} u_{Xi}^{(t+1)} &= u_{Xi}^{(t)} - \mu \left[D \sum_{\substack{Y=1 \\ Y \neq X}}^N d_{XY} v_{Yi+1}^{(t)} + v_{Yi-1}^{(t)} \right] + \frac{\partial E_L(v)}{\partial v_{Xi}^{(t)}} + \frac{1}{2} \frac{\partial E_P(v)}{\partial v_{Xi}^{(t)}} \\ \frac{\partial E_L(v)}{\partial v_{Xi}^{(t)}} &= \lambda_1^{i(t)} + \lambda_2^{X(t)} + \lambda_3^{X_i(t)} \sum_{\substack{j=1 \\ j \neq i}}^N v_{Xj}^{(t)} + \lambda_4^{X_i(t)} \sum_{\substack{Y=1 \\ Y \neq X}}^N v_{Yi}^{(t)} + \lambda_5^{X_i(t)} [0.5 - v_{Xi}^{(t)}] \\ \frac{1}{2} \frac{\partial E_P(v)}{\partial v_{Xi}^{(t)}} &= \beta_1 \left[\sum_{X=1}^N v_{Xi}^{(t)} - 1 \right] + \beta_2 \left[\sum_{i=1}^N v_{Xi}^{(t)} - 1 \right] + \beta_3 v_{Xi}^{(t)} \left[\sum_{\substack{j=1 \\ j \neq i}}^N v_{Xj}^{(t)} \right]^2 \\ &+ \beta_4 v_{Xi}^{(t)} \left[\sum_{\substack{Y=1 \\ Y \neq X}}^N v_{Yi}^{(t)} \right]^2 + \beta_5 v_{Xi}^{(t)} [1 - v_{Xi}^{(t)}] [0.5 - v_{Xi}^{(t)}] \\ v_{Xi}^{(t+1)} &= \frac{1}{1 + e^{-u_{Xi}^{(t+1)}/\tau}} \\ \lambda_k^{t+1} &= \lambda_k^t + \rho \beta_k C_k^{t+1} \end{aligned} \tag{26}$$

Este proceso se repite hasta que la red encuentra una solución válida. Para que la solución se considere válida deben cumplirse las siguientes tres condiciones:

$$\begin{aligned} \|v^{t+1} - v^t\|_\infty &< 0.01 / N \\ \left| \sum_{X=1}^N v_{Xi}^{(t)} - 1 \right| &< 0.01 \quad \text{y} \quad \left| \sum_{i=1}^N v_{Xi}^{(t)} - 1 \right| < 0.01 \end{aligned}$$

$$1 - v_{Xi}^{(t)} < 0.01 \quad \text{o} \quad v_{Xi}^{(t)} < 0.01$$

Las ecuaciones anteriores representan que el estado es estable, que es una solución posible y que es un estado sin ambigüedad, respectivamente.

A estas ecuaciones de evolución [40], le unimos la obligación de partir de un estado válido y mediante perturbación modificarlo y hacerlo evolucionar hasta alcanzar, según los criterios definidos, un estado estable válido.

A la unión de las ecuaciones [40] y elegir como punto inicial un recorrido inicial válido lo vamos a notar como algoritmo de **Recorrido Inicial Válido (RIV)**, donde es preciso resaltar la incorporación de un parámetro **D** en la ecuación que rige la evolución de la variable **u**, que nos servirá para impulsar el proceso de convergencia de la evolución dinámica de la red, como ya veremos más adelante. Hasta aquí la propuesta de resolución del problema TSP aplicando nuestra metodología. Veamos ahora una mejora más para realizar un escape de mínimos no deseados.

5.2 - Estrategia de escape de mínimos locales

A diferencia del algoritmo de Peng, Gupta y Armitage (1996), el algoritmo de escape de mínimos locales que proponemos, toma como base nuestro método e introduce ruido en el estado de las neuronas en lugar de distorsionar los parámetros de la red (los pesos de conexión y los términos de sesgo). Con esto se evita el tener que almacenar la matriz de pesos de conexión, de manera que, únicamente se necesitan dos matrices NxN para mantener el estado y la salida de las neuronas (V_{NxN} y U_{NxN}) reduciendo así los requerimientos de memoria. Esto va a permitir, además, que el algoritmo se pueda utilizar, incluso, para problemas de grandes dimensiones.

El algoritmo utiliza las ecuaciones anteriores, que representan la dinámica de la red, para efectuar cada iteración, a este algoritmo le denominaremos como RIVSA al tomar como referencia el algoritmo RIV y sumarle la técnica de recalentamiento simulado (Simulated Annealing)

El funcionamiento del algoritmo es el siguiente: Sea H una red de Hopfield. La red se inicializa con un camino válido (el camino por defecto) que constituye un estado estable para la red. Se introduce ruido en la red y se hace progresar hasta que ésta se estabiliza en un nuevo estado estable H' (un nuevo camino válido). Si el nuevo estado H' tiene un nivel de energía menor que H se toma H' como camino inicial y se repite el proceso. En caso contrario, se vuelve a repetir el proceso pero volviendo a utilizar H como estado inicial. La estrategia consiste en repetir este proceso un número M fijo de veces, de manera que, para cada iteración, se toma como camino inicial, el camino más corto encontrado hasta el momento. Después de repetir el proceso durante M iteraciones, la solución final de una determinada ejecución será el camino con menor distancia encontrada durante todas las iteraciones.

En la primera iteración (para $M=1$) se parte del camino por defecto como camino inicial válido. El camino por defecto consiste en unir simplemente las ciudades etiquetadas ordenadas según el valor de la etiqueta. Así por ejemplo, para una red con 10 ciudades, el camino por defecto es 1-2-3-4-5-6-7-8-9-10. En general, para un TSP de N ciudades, el estado de las $N \times N$ neuronas se inicializa de la siguiente manera:

$$v_{Xi}^{(0)} = \begin{cases} 1 & \text{si } X = i \\ 0 & \text{si } X \neq i \end{cases}$$

Puesto que esta inicialización representa un estado estable, se debe introducir ruido δ en las neuronas para desestabilizar la red y hacer que evolucione a un nuevo estado válido con un nivel de energía menor.

$$v_{Xi}^{(0)} = v_{Xi}^{(0)} + \delta$$

El funcionamiento del algoritmo RIVSA sería de la siguiente manera:

En la matriz $S_{N \times N}$ se almacena siempre el mejor camino válido encontrado hasta el momento. $f(X)$ representa la función que calcula la distancia total para un camino dado.

1. Inicializar el camino solución como el camino por defecto

$$s_{Xi} = \begin{cases} 1 & \text{si } X = i \\ 0 & \text{si } X \neq i \end{cases}$$

$$D_s = f(S)$$

2. Repetir M veces

$$2.1 \quad v_{Xi} = s_{Xi} + \delta$$

$$2.2 \quad u_{Xi} = -\tau \ln \left(\frac{1 - v_{Xi}}{v_{Xi}} \right)$$

2.3 Hacer progresar la red siguiendo las ecuaciones del método que proponemos hasta encontrar un nuevo mínimo local, es decir, hasta que la red se estabilice.

$$2.4 \quad D_a = f(U)$$

$$2.5 \quad \text{Si } D_a < D_s$$

$$D_s = D_a$$

$$s_{Xi} = v_{Xi}$$

3. Alcanzada la solución final, el camino óptimo se identifica en la matriz S cuya distancia total es D_s

Notación:

δ es un número aleatorio uniformemente distribuido en el intervalo $[0, \alpha]$ con $\alpha \leq 1$, que se genera al comienzo de cada iteración. Este número aleatorio se utiliza para introducir ruido en la red con el fin de distorsionarla cuando se encuentra en un estado estable que representa un mínimo local.

D_a contiene la distancia del camino encontrado en cada iteración mientras que D_s contiene la distancia del mejor camino encontrado hasta el momento.

Obtenemos entonces una estrategia basada en nuestra metodología que permite resolver de una manera fiable y computacionalmente razonable los problemas de optimización combinatoria, en especial el denominado problema del viajante. Los resultados de la implementación realizada demuestran, Ruiz Sepúlveda (1999), la bondad empírica de la metodología aquí propuesta en cada uno de sus apartados. Mostramos, a continuación, un resumen de los resultados obtenidos.

5.3 - Análisis de los resultados computacionales

Para comprobar la efectividad de los algoritmos antes propuestos, RIV y RIVSA, se van a comparar con el modelo propuesto por Brandt y otros (1988) al que denominaremos HB y con el algoritmo propuesto por S. Z. Li (1996).

Los algoritmos se van a aplicar al problema del viajante con 16, 22 y 48 ciudades. En el artículo de S. Z. Li también se analiza el problema del viajante de 10 ciudades pero en este caso la calidad de la solución no se puede mejorar puesto que ya con el algoritmo, Li (1996) se obtiene la distancia mínima (la solución óptima) para el 100% de las ejecuciones.

Las coordenadas de las ciudades se toman de la librería TSPLIB [Bixby y Reinelt \(1999\)](#). Los ficheros “ulysses16.tsp”, “ulysses22.tsp” y “att48.tsp” contienen el conjunto de datos para el problema que nos ocupa con 16, 22 y 48 ciudades respectivamente. En la librería TSPLIB también se dan los caminos óptimos para cada problema junto con la distancia total. Sin embargo, el método de cálculo utilizado se basa en las distancias geográficas, y las coordenadas (x,y) vienen expresadas en grados y minutos.

En las pruebas realizadas se han tomado las coordenadas como puntos del plano real y el cálculo de distancia se realiza mediante la distancia euclídea. Aunque este método constituye una buena aproximación al cálculo de las distancias

geográficas, el camino óptimo puede variar levemente entre ambos métodos. Las coordenadas se normalizan para que (x_i, y_i) queden comprendidas en el intervalo $[0,1]$. Una vez realizada la normalización, la distancia total (utilizando el método de distancia euclídea) para los caminos óptimos dados en la librería TSPLIB tienen una longitud de 2.36316, 2.41279 y 4.32452 para los problemas problema del viajante de 16, 22 y 48 ciudades respectivamente.

Cada caso se va a ejecutar 100 veces. En cada ejecución, las variables v se inicializan con el camino por defecto para los algoritmos Li (1996) y RIV. Posteriormente, a cada v se le añade un ruido aleatorio que será diferente para cada una de las 100 ejecuciones.

5.3.1 -Valores iniciales

1. Algoritmo de Brandt (HB)

$$A=B=2, C=4, D=1, \tau=10, \mu=0.1$$

v se inicializa aleatoriamente alrededor del hipercubo unidad

$$v_{Xi}^{(0)} = 0.5 + \delta$$

$$\text{con } \delta \in [-10^{-6}, 10^{-6}]$$

2. Algoritmo Li (1996)

$$\lambda_k=0 \forall k, \beta_1=\beta_2=100, \beta_3=\beta_4=1, \beta_5=0, D=10^6, \tau=10^7, \mu=100$$

ρ se incrementa desde uno hasta 100 de acuerdo con $\rho \leftarrow 1.01\rho$

v se inicializa aleatoriamente alrededor del hipercubo unidad

$$v_{Xi}^{(0)} = 0.5 + \delta$$

$$\text{con } \delta \in [-10^{-6}, 10^{-6}]$$

3. Algoritmo RIV

$$\lambda_k=0 \forall k, \beta_1=\beta_2=100, \beta_3=\beta_4=1, \beta_5=0, D=10^6, \tau=10^7, \mu=100,$$

$$\rho=1000, \Delta D=0.99$$

v se inicializa con el camino por defecto

$$v_{Xi}^{(0)} = \begin{cases} 1 & \text{si } X = i \\ 0 & \text{si } X \neq i \end{cases}$$

$$v_{Xi}^{(0)} = v_{Xi}^{(0)} + \delta$$

δ es un número aleatorio.

Para el problema del viajante de 16 $\delta \in [0.0, 0.40000]$

Para el problema del viajante de 22 y 48 $\delta \in [0.0, 0.52000]$

El parámetro D se decreta de acuerdo con $D \leftarrow D * \Delta D$ cada cierto número de iteraciones:

Para el problema del viajante de 16 y 22 ciudades D se decreta cada 10 iteraciones.

Para el problema del viajante de 48 ciudades D se decreta cada 1000 iteraciones.

4. Algoritmo RIVSA

El proceso de recalentado se repite cinco veces para cada ejecución ($M=5$)
Las inicializaciones son idénticas a las del algoritmo RIV. Sólo cambia el ruido α que se añade al camino inicial cuando se obtiene un estado con un nivel de energía más bajo. Así, a partir de la segunda iteración (cuando $M>1$), V se inicializa con el camino más corto encontrado hasta el momento y se añade un ruido aleatorio δ

Para el problema del viajante de 16 ciudades $\delta \in [0.0, 0.82000]$

Para el problema del viajante de 22 y 48 ciudades $\delta \in [0.0, 0.52000]$

El parámetro D se inicializa a 10^6 en cada ciclo, es decir, para cada valor de M. A medida que la red evoluciona hacia un nuevo estado estable, D se decreta de acuerdo con $D \leftarrow D * \Delta D$ cada cierto número de iteraciones:

Para el problema del viajante de 16 y 22 ciudades D se decreta cada 10 iteraciones.

Para el problema del viajante de 48 ciudades D se decrementa cada 1000 iteraciones.

5.3.2 -Resultados

En cada una de las figuras siguientes se muestran las distancias encontradas para los problemas del viajante con 16, 22 y 48 ciudades en los cuatro algoritmos analizados. Los datos del algoritmo HB y Li (1996) se toman del artículo de S. Z. Li (1996).

Por motivos de claridad, la escala utilizada para representar los datos del algoritmo HB (figura 1) es diferente de los demás algoritmos. La calidad de las soluciones para el algoritmo HB es más pobre y las distancias encontradas son más grandes, lo que implicaría que de utilizar la misma escala para todos los algoritmos, los resultados obtenidos para Li (1996), RIV y RIVSA se distinguirían con mayor dificultad.

Representamos en los ejes de ordenadas el número de ejecuciones que alcanza la distancia del recorrido obtenido y que se representa en los ejes de abcisas. En definitiva es un gráfico de frecuencias donde se representa la frecuencia absoluta con que se alcanzan las distancias de los recorridos óptimos en el marco de las ejecuciones realizadas.

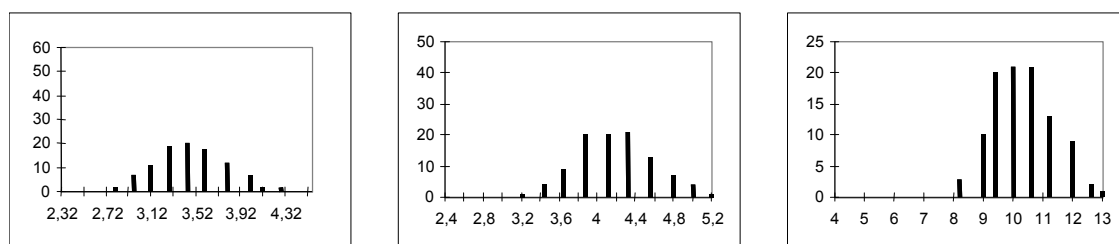


Figura 1. Distancias encontradas con el algoritmo HB para problema del viajante de 16, 22 y 48 ciudades

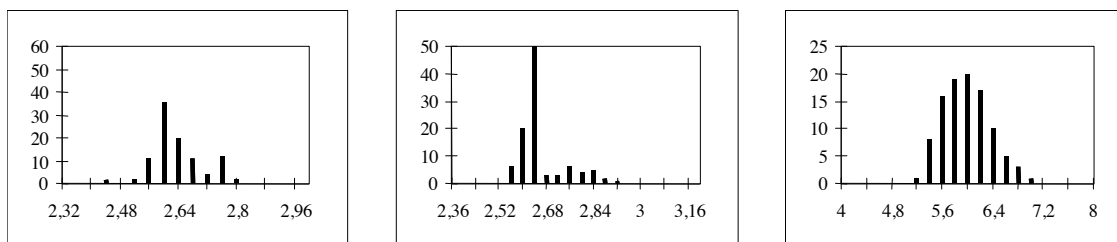


Figura 2. Distancias encontradas con el algoritmo Li (1996) para problema del viajante de 16, 22 y 48 ciudades

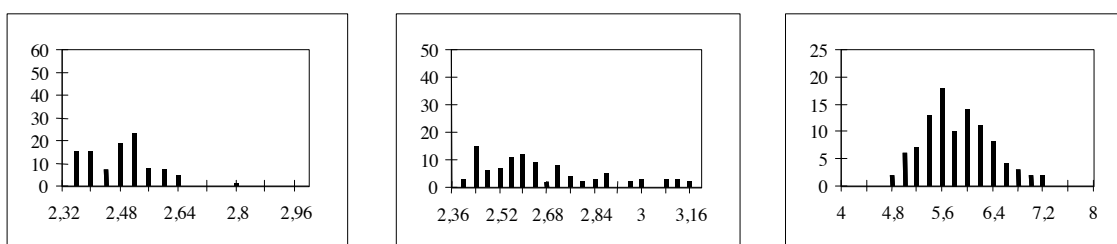


Figura 3. Distancias encontradas con el algoritmo RIV para problema del viajante de 16, 22 y 48 ciudades

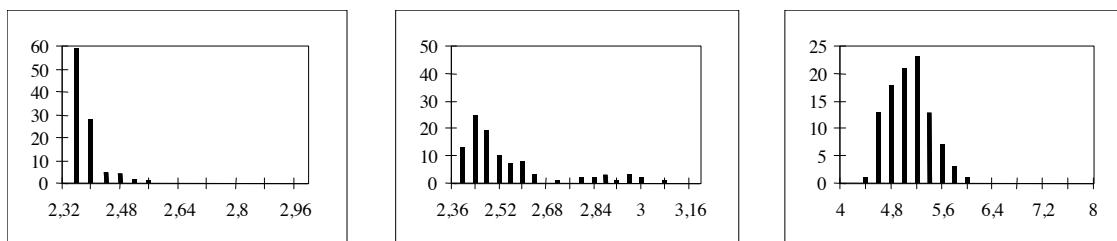


Figura 4. Distancias encontradas con el algoritmo RIVSA para problema del viajante de 16, 22 y 48 ciudades

Las tablas I, II y III resumen los datos obtenidos con cada algoritmo para los problemas con 16, 22 y 48 ciudades respectivamente. En cada tabla se muestra la media de distancias encontradas, el error producido, la distancia mínima encontrada y el número medio de iteraciones para llegar a la solución.

Algoritmo	Media de Distancias	Error	Mínima Distancia Encontrada	Nº Medio de Iteraciones
HB	3.4696	47.02%	2.8000	65
Li (1996)	2.6356	11.53%	2.4400	13215
RIV	2.4824	5.05%	2.3643	795
RIVSA	2.3848	0.92%	2.3593	4500

Tabla I. Datos encontrados para el problema del viajante de 16 ciudades

Algoritmo	Media de Distancias	Error	Mínima Distancia Encontrada	Nº Medio de Iteraciones
HB	4.1852	73.42%	3.2000	66
Li (1996)	2.6620	10.31%	2.5600	21536
RIV	2.6596	10.21%	2.4285	816
RIVSA	2.5468	5.53%	2.4060	5025

Tabla II. Datos encontrados para el problema del viajante de 22 ciudades

Algoritmo	Media de Distancias	Error	Mínima Distancia Encontrada	Nº Medio de Iteraciones
HB	10.2700	137.37%	8.2000	298
Li (1996)	5.9800	38.28%	5.2000	4633
RIV	5.8360	34.95%	4.8769	2544
RIVSA	5.0800	17.47%	4.5993	13735

Tabla III. Datos encontrados para el problema del viajante de 48 ciudades

En primer lugar, y como demuestran los resultados obtenidos, con un método de funciones penalización (el algoritmo original de Hopfield) las distancias encontradas en todos los casos analizados son bastante peores que con el método de Li (1996). Sin embargo, la velocidad de convergencia del algoritmo HB supera al resto de algoritmos utilizados.

El camino óptimo para el problema del viajante de 16 ciudades según la librería TSPLIB es 1-14-13-12-7-6-15-5-11-9-10-16-3-2-4-8, cuya distancia total, una vez normalizadas las coordenadas y utilizando el método de distancia euclídea es 2.36316. Como muestra la tabla I, la media de distancias encontradas con los

algoritmos RIV y RIVSA disminuyen el error cometido en 6.48% y un 10.61% respectivamente, con respecto a la distancia media encontrada para el algoritmo Li (1996). Por otra parte, el camino óptimo encontrado con el algoritmo RIV es únicamente un 0.05% (2.3643 el algoritmo RIV frente a 2.36316 definido por la librería TSPLIB) más largo que el óptimo. El 30% de los caminos encontrados con el algoritmo RIV tienen una distancia menor que la mínima distancia encontrada con el algoritmo Li (1996). Con el algoritmo RIVSA no sólo se encuentran un 88% de caminos por debajo de la distancia mínima encontrada para Li (1996), sino que además el algoritmo encuentra un camino con una distancia menor a la contemplada como óptima en el artículo de S. Z. Li Li (1996). Se trata del camino 1-3-2-4-8-15-5-11-9-10-7-6-14-13-12-16, cuya distancia es 2.359299. En cuanto al número de iteraciones requeridas para obtener convergencia, se observa que los dos algoritmos propuestos en el artículo superan ampliamente en velocidad al algoritmo Li (1996). Concretamente, los algoritmos RIV y RIVSA son 16 y casi 3 veces más rápidos que el algoritmo Li (1996).

Para el problema del viajante de 22 ciudades, los algoritmos RIV y Li (1996) no difieren mucho en cuanto a distancia media encontrada. Sin embargo, la velocidad del algoritmo RIV es muy superior a la del algoritmo Li (1996) (RIV es 26 veces más veloz que Li (1996)). Por el contrario, con el algoritmo RIVSA sí se obtiene una mejora considerable en la calidad de las soluciones, y a pesar de que el tiempo de procesamiento aumenta, aún sigue siendo menor que el requerido para el algoritmo Li (1996) (RIVSA es 4 veces más rápido que Li (1996)). Las mínimas distancias encontradas con los algoritmos RIV y RIVSA también son menores que para el algoritmo Li (1996), obteniéndose un 31% y un 67% respectivamente de caminos con distancias inferiores a la distancia mínima encontrada con Li (1996). Al igual que ocurre en el problema del viajante de 16 ciudades, con el algoritmo RIVSA se encuentran mejores caminos que el contemplado en Li (1996). Concretamente, según la librería TSPLIB el camino óptimo es 1-14-13-12-7-6-15-5-11-9-10-19-20-21-16-3-2-17-22-4-18-8, cuya distancia euclídea, después de

normalizar las coordenadas, es 2.413279. Con el algoritmo RIV se encuentran cuatro caminos con una distancia inferior:

1-22-18-4-17-2-3-16-20-21-19-10-9-11-5-15-6-7-12-13-14-8	2.408987
1-12-13-14-7-6-15-5-11-9-10-19-20-21-16-3-2-17-4-18-22-8	2.408661
1-3-2-17-4-18-22-8-14-13-12-7-6-15-5- 11-9-10-19-20-21-18	2.407807
1-8-22-18-4-17-2-3-16-20-21-19-10-9-11-5-15-6-7-12-13-14	2.406074

Por último, la calidad de las soluciones también se mejora para el problema del viajante de 48 ciudades. Con RIV y RIVSA se encuentra un 8% y un 53% de caminos más cortos que la menor distancia encontrada con el algoritmo Li (1996). El algoritmo RIV sigue siendo más rápido que el Li (1996). Sin embargo, no ocurre lo mismo con el algoritmo RIVSA que para el caso de 48 ciudades es el más lento de todos los expuestos.

Los resultados para 51 y 101 ciudades confirman las conclusiones indicadas en el anterior párrafo, mejorando en los mismos términos de calidad y convergencia las soluciones obtenidas por los iniciales algoritmos propuestos para la resolución del problema del viajante de comercio.

Efectivamente en las figuras 5 y 6 que siguen mostramos para los distintos algoritmos aquí estudiados las distancias obtenidas y las mejoras de velocidad resultantes. Igualmente en las tablas IV y V recogemos la información pertinente para el problema de viajante con 51 y 101 ciudades.

Al igual que en los gráficos anteriores, en el eje de ordenadas representamos frecuencias y en el eje de abcisas las distancias de los recorridos óptimos encontrados.

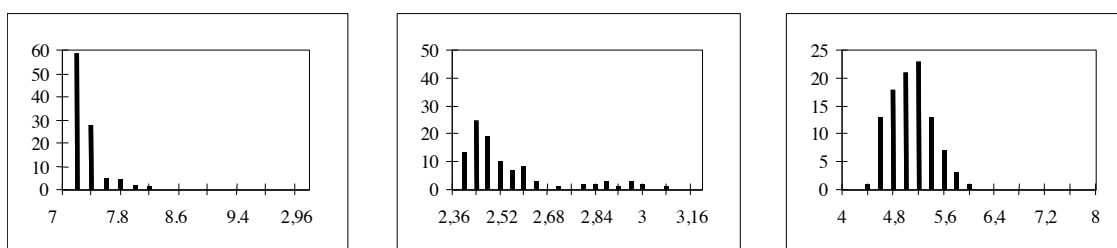


Figura 5. Distancias encontradas con el algoritmo Li (1996), RIV y RIVSA para el problema del viajante de 51 ciudades

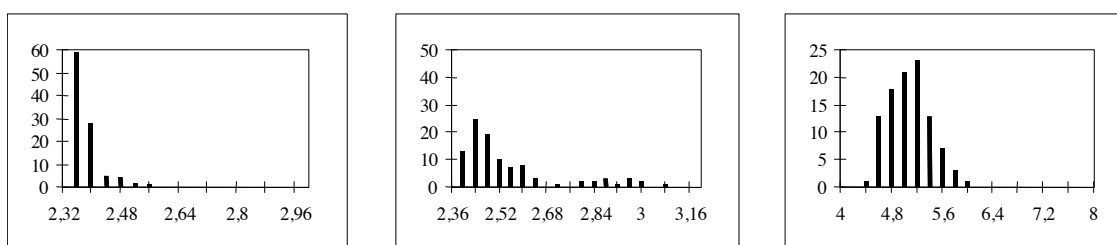


Figura 6. Distancias encontradas con el algoritmo Li (1996), RIV y RIVSA para el problema del viajante de 101 ciudades

Algoritmo	Media de Distancias	Error	Mínima Distancia Encontrada	Nº Medio de Iteraciones
Li (1996)	9.6001	40.66%	9.1974	2887
RIV	8.5848	25.78%	8.0144	1413
RIVSA	7.6648	12.30%	7.2072	7038

Tabla IV. Datos encontrados para el problema del viajante de 51 ciudades

Algoritmo	Media de Distancias	Error	Mínima Distancia Encontrada	Nº Medio de Iteraciones
Li (1996)	13.9354	60.55%	12.6767	6937
RIV	12.8787	48.37%	11.8380	3564
RIVSA	11.0103	26.85%	10.5292	23867

Tabla V. Datos encontrados para el problema del viajante de 101 ciudades

6 - CONCLUSIONES

De los resultados que hemos presentado en este trabajo podemos concluir que:

- a) El algoritmo construido en base a nuestra metodología, al que denominamos RIV, genera unas soluciones que superan a las encontradas por los algoritmos más recientes, tanto en la calidad, proximidad a la solución, como, y sobre todo, en el tiempo de procesamiento.
- b) Añadiendo una estrategia de escape de mínimos locales basada en el recalentamiento simulado al algoritmo RIV, disminuimos su velocidad de convergencia, pero mejoramos considerablemente la precisión de las soluciones obtenidas.
- c) Comparando el algoritmo con estrategia de escape de mínimos locales con los resultados obtenidos por un algoritmo de recalentamiento simulado, obtenemos, de nuevo, unos resultados que confirman una aportación que realizamos en el presente trabajo: Disminuye el tiempo de convergencia. Aproxima mejor la solución óptima dada por la biblioteca TSPLIB para un conjunto de ciudades, pequeño, mediano y grande; siendo la mejora más ostensible cuando mayor es el tamaño del problema considerado.
- d) La combinación de la metodología propuesta con una estrategia de escape de mínimos locales no deseables, permiten confirmar una mejora en la obtención de las soluciones de problemas de optimización combinatoria y una disminución del tiempo de ejecución requerido.

Del estudio detallado de los resultados podemos observar como algunas de las características de la evolución y comportamiento de los mismos nos hacen presumir que puede ser interesante aplicar otro tipo de dinámica de red y obtener resultados interesantes.

7 - REFERENCIAS BIBLIOGRÁFICAS

Abe, S. (1993): "Global convergence and suppression of spurious states of the Hopfield neural networks". *IEEE Trans. Circuit and Systems*, 40, 246-257.

Bixby, B. & Reinelt, G. (1999): Travelling Salesman Problem Library (on line). Available, <ftp://softlib.cs.rice.edu/pub/tsplib/tsplib.tar>

Brandt, R. D., Wang, Y., Laub, A. J. & Mitra, S. K. (1988): "Alternative networks for solving the travelling salesman problem and the list-matching problem." *Proceedings of the IEEE International Conference on Neural Networks*, 2, 333-340.

Cerny, V. (1985): "Thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm". *Journal of Optimization Theory and Applications*, 45, 41-45.

Cheung, K. W. & Lee, T. (1992): "Neural networks for global optimization". *Proc. RNNS/IEEE Symp. Neuroinformatics and Neurocomputing '92*, 53-63.

Gallizo, J.L. y Moreno, J.M. (1992): "Towards Integral Decisional Systems in Management Control", *15th Annual Congress of the European Accounting Association, abril 1992, Madrid*.

Hertz, J. A., Krogh, A. S. & Palmer, R. G. (1991): "Introduction to the Theory of Neural Computation", *Lecture Notes Volume I in the Santa Fe Institute Studies in the Sciences of Complexity*, Addison-Wesley Publishing Co.

Hopfield, J.J. (1982): "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", *Proceedings of the National Academy of Sciences of the U.S.A.*, 79, 2554-2558.

Hopfield, J.J. (1984): "Neurons with graded response have collective computational properties like those of two-state neurons". *Proceedings of the National Academy of Sciences of the U.S.A.* 81, 3088-3092.

Hopfield, J. J. & Tank, D. W. (1985): "Neural computation of decisions in optimization problems", *Biological Cybernetics*, 52, 141-152.

Hopfield, J.J. & Tank, D.W. (1986): "Computing With Neural Circuits: A Model", *Science*, 233, 625-633.

Kirkpatrick, JR., Gelatt, C.D. & Vecchi, M.P. (1983): "Optimization by simulated annealing". *Science*, 220: 671-680.

Li, S. Z. (1996): "Improving convergence and solution quality of Hopfield type neural networks with augmented Lagrange multipliers". *IEEE Trans. on Neural Networks*, 7, 1507-1516.

Otero, J.M. & Trujillo, F. (1992): "Predicción multivariante y multiperiodo mediante una red neuronal: estudio de un caso". *VI Reunión Asepelt España. Granada*.

Otero, J.M. & Trujillo, F. (1991): "Red neuronal para la predicción de la tasa de actividad femenina". *V Reunión Asepelt España. Gran Canaria.*

Peng, M., Gupta, N. K. & Armitage, A. F. (1996): "An investigation into the improvement of local minima of the Hopfield Network". *Neural Networks*, 9, 1241-1253.

Peretto, P. (1992): "An Introduction to the Modeling of Neural Networks", Cambridge University Press. Cambridge.

Protzel, P. W., Palumbo, D. L. & Arras, M. K. (1993): "Performance and fault-tolerance of neural networks for optimization", *IEEE Trans. On Neural Networks*, 4, 600-614.

Ruiz Sepúlveda, Amparo (1999): "Una solución integral de los problemas de optimización en intranet, basada en redes neuronales artificiales". *Tesis doctoral.* Facultad de Ciencias Económicas y Empresariales. Universidad de Málaga. Málaga.

Serrano Cinca, C.(1996): "Self Organizing Neural Networks for Financial Diagnosis", *Decision Support Systems*, vol. 17, pp. 227-238.

Serrano Cinca, C. (1997): "Feedforward Neural Networks in the Classification of Financial Information", *European Journal of Finance*, Vol 3, Nº 3, pag 183-202.

Serrano, C. y Martín del Brío, B. (1993): "Predicción de la crisis bancaria mediante el empleo de redes neuronales artificiales", *Revista Española de Financiación y Contabilidad*, vol XXII, nº 74, pag 153-176.

ANEXO I: Demostración de los resultados teóricos presentados en el presente trabajo

Teorema 1

Consideremos una red neuronal con N neuronas de McCulloch-Pitts con histéresis, Zurada (1992), donde las aristas tienen asignados unos pesos sinápticos dados por la matriz W . Sea el vector $Y(k)$ el vector de estados de la red. Si la dinámica de computación de la red en cada iteración viene dada por

$$Y_i(k) = \begin{cases} +1 & \text{si } U_i(k) > T_{i2} \\ -1 & \text{si } T_{i1} \leq U_i(k) \leq T_{i2} \\ -1 & \text{si } U_i(k) < T_{i1} \end{cases} ; \text{ con } T_{i1} \leq 0 \leq T_{i2} \quad (2)$$

Eligiendo aleatoriamente una neurona, L , cada vez que actualizamos su estado, podemos afirmar que la función de energía $H(Y(k))$ decrece cuando y sólo cuando el potencial sináptico viene dado por la expresión:

$$U_L(k) = -\omega \frac{\Delta H(Y(k))}{\Delta Y_L(k)} \quad (3)$$

donde $\omega > 0$ es la *tasa de actualización*.

Demostración

En efecto, si la neurona L se ha modificado, entonces $\Delta Y_L(k) = \pm 2$, y por tanto:

Si $\Delta Y_L(k) = 2$, entonces por (2) $Y_L(k) = 1$ y $U_L(k) > T_{L2} > 0$. De (3) se deduce que $\Delta H(Y(k)) < 0$

Si $\Delta Y_L(k) = -2$, entonces por (2) $Y_L(k) = -1$ y $U_L(k) < T_{L1} < 0$. De (3) se deduce que $\Delta H(Y(k)) < 0$ ♦

Corolario 1

Para una función de energía cuadrática

$$H(\mathbf{Y}(k)) = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N w_{ij} Y_i(k) Y_j(k) + \sum_{i=1}^N \theta_i Y_i(k) \quad (4)$$

la dinámica de la red definida no genera un incremento de la función de energía cuando y sólo cuando el potencial sináptico viene dado por la expresión:

$$U_i(k) = \sum_{j=1}^N w_{ij} Y_j(k) - \theta_i \quad (5)$$

Demostración

Suponemos que:

- En el instante k la unidad L se selecciona para su actualización, donde $1 \leq L \leq N$, por tanto, debemos extraer la contribución a la función energía debida a la unidad L .
- La variación en la función de energía viene dada por $\Delta H(k) = H(k+1) - H(k)$.
- La actualización es asíncrona y, por tanto, sólo la unidad L puede cambiar en el instante de tiempo k y consecuentemente $Y_i(k+1) = Y_i(k)$ para $i \neq L$.
- Todas las unidades tienen salidas bipolares $[Y_i]^2 = 1$ para todo i .
- Si la unidad L cambia su salida entonces $Y_L(k+1) = -Y_L(k)$.
- La ortogonalidad entre los patrones de una red de Hopfield

Tenemos que $\Delta H(k) = Y_L(k) \text{net}_L(k) - W_{LL}$; donde $\text{net}_L(k)$ representa la función de entrada neta a la neurona L en el instante (k) , es decir el valor de la señal combinada de las distintas neuronas y ponderadas por el peso sináptico de la conexión interneuronal correspondiente.

Por otra parte $\Delta Y_L(k) = Y_L(k) - Y_L(k-1)$ y por tanto aplicando el teorema 1, obtenemos que:

$$U_i(k) = net_L(k-1) - \theta_i$$

$$\text{donde } \theta_i = W_{LL} \cdot \omega$$

$$\text{y } \omega = \frac{\Delta Y_L(k)}{Y_L(k-1)}$$

siendo $\omega > 0$ por los apartados a) y b) de la demostración del teorema 1. ♦

Teorema 2

Sea $E(x; \lambda, \nu; \kappa, \kappa')$ la función energía y sea $(x^*; \lambda^*, \nu^*; \kappa, \kappa')$ un punto que satisface las condiciones de primer orden definidas previamente. Si $E(x^*; \lambda^*, \nu^*; \kappa, \kappa')$ es un mínimo local de $E(x; \lambda, \nu; \kappa, \kappa')$ con respecto a x para algunos valores finitos κ y $\kappa' > 0$ entonces $f(x^*)$ es un mínimo local de $f(x)$ del problema de optimización en forma estándar.

Demostración

Sea un punto $x^\Delta = x^* + \varepsilon$ un punto próximo a x^* y admisible. Entonces:

$$E(x^*; \lambda^*, \nu^*; \kappa, \kappa') \leq E(x^\Delta; \lambda^*, \nu^*; \kappa, \kappa')$$

$$\text{con } g(x^\Delta) \leq 0; h(x^\Delta) = 0$$

en estas condiciones la función de energía se reduce a:

$$\begin{aligned} E(x^\Delta; \lambda^*, \nu^*; \kappa, \kappa') &= f(x^\Delta) + \sum_{j \in I_{re}} \lambda_j^* g_j(x^\Delta) + \kappa \sum_{j \in I_{re}} g_j^2(x^\Delta) \\ &= f(x^\Delta) + \sum_{j \in I_{re}} g_j(x^\Delta) [\lambda_j^* + \kappa g_j(x^\Delta)] \end{aligned} \quad (13)$$

siendo $I_{re} = \{j | \lambda_j^* > 0\}$.

Cada sumando de la ecuación (11) verifica que $g_j(x^\Delta) \leq 0; \lambda_j^* > 0$ para $j \in I_{re}$; $0 < \kappa < \infty$, pero dado que $g_j(x^*) = 0$ al ser $\lambda_j^* > 0$ y continua, existe un $\varepsilon > 0$ tal que cuando $\|x^* - x^\Delta\| < \varepsilon$, verifica que $\lambda_j^* > -\kappa_j g_j(x^\Delta)$, y por tanto:

$$E(x^\Delta; \lambda^*, \nu^*; \kappa, \kappa') \leq f(x^\Delta)$$

Pero por otra parte, x^* es un mínimo local de E y de ahí,

$$E(x^*; \lambda^*, \nu^*; \kappa, \kappa') \leq E(x^\Delta; \lambda^*, \nu^*; \kappa, \kappa')$$

y por tanto

$$E(x^*; \lambda^*, \nu^*; \kappa, \kappa') \leq f(x^\Delta)$$

Pero como

$$\begin{aligned} E(x^*; \lambda^*, \nu^*; \kappa, \kappa') &= L(x^*; \lambda^*, \nu^*) + \kappa m^2(x^*) + \kappa' h^2(x^*) \\ &= f(x^*) + \kappa m^2(x^*) \geq f(x^*) \end{aligned} \quad (14)$$

obtenemos que $f(x^*) \leq f(x^\Delta)$ para todo x^Δ tal que $\|x^\Delta - x^*\| < \varepsilon$ con x^Δ verificando las restricciones.

De ahí que x^* es un mínimo local del problema original, con lo que queda demostrado el teorema. ♦

Teorema 3

Sea $(x^*; \lambda^*, \nu^*)$ un punto que satisface las condiciones de primer orden. Si x^* es un mínimo local de $f(x)$ de un problema de optimización asociado a la función de energía E , entonces:

a) $(x^*; \lambda^*, \nu^*; \kappa, \kappa')$ es un mínimo de la función de energía E con respecto a la variable x para algunos κ, κ' finitos positivos.

o bien

b) en el límite cuando $\kappa, \kappa' \rightarrow \infty$ $E(x^*; \lambda^*, \nu^*; \kappa, \kappa')$ aproxima un mínimo local de la función de energía.

Demostración:

Sea x^Δ un punto que no satisface todas las restricciones, entonces por la construcción de la función de energía podemos encontrar unos κ y κ' suficientemente grandes tales que $E(x^\Delta; \lambda^*, \nu^*; \kappa, \kappa') \geq E(x^*; \lambda^*, \nu^*; \kappa, \kappa')$ donde algunos de los κ y κ' pueden crecer ilimitadamente.

Por la misma razón si x^Δ satisface todas las restricciones pero alguna de las desigualdades la satisface estrictamente, tendríamos que para algunos κ suficientemente grandes se verifica que $E(x^\Delta; \lambda^*, \nu^*; \kappa, \kappa') \geq E(x^*; \lambda^*, \nu^*; \kappa, \kappa')$.

Si consideramos ahora x^Δ un punto admisible, es decir, que verifica las restricciones de la siguiente forma:

$$h(x^\Delta)=0; g_j(x^\Delta)=0 \quad j \in I_{re}; g_j(x^\Delta) < 0 \quad j \notin I_{re}; \quad (15)$$

Podemos encontrar un ε suficientemente pequeño tal que para $\|x^\Delta - x^*\| < \varepsilon$, $f(x^\Delta) \geq f(x^*)$.

Con estas consideraciones tenemos que:

Para x^Δ que no satisfacen todas las restricciones o satisfacen alguna de las desigualdades estrictamente,

$$\lim_{\kappa, \kappa' \rightarrow \infty} E(x^\Delta; \lambda^*, \nu^*; \kappa, \kappa') > E(x^*; \lambda^*, \nu^*; \kappa, \kappa') \quad (16)$$

Para x^Δ que satisface las restricciones con igualdad, entonces

$$E(x^\Delta; \lambda^*, \nu^*; \kappa, \kappa') \geq f(x^\Delta) \geq f(x^*) = E(x^*; \lambda^*, \nu^*; \kappa, \kappa')$$

Es decir, se verifica la tesis del teorema. ♦