Coordinación de Ciencias Computacionales

C261-69 Tópicos Avanzados: Redes Neuronales Artificiales

Neurodinámica: *Las Redes de Hopfield*

Dra. Ma. del Pilar Gómez Gil

V:14-02-17

Neurodinámica

- Se refiere al estudio de RNA vistas como sistemas dinámicos no lineales, dando énfasis en el problema de **estabilidad,**
- La presencia de estabilidad siempre implica alguna forma de coordinación entre las partes individuales de un sistema.
- La estabilidad en redes con retroalimentación global (redes recurrentes) es difícil de alcanzar.
- Fundamentalmente, las redes recurrentes pueden usarse como **memorias asociativas**, o como **sistemas de entrada-salida.**
- Neurodinámica se interesa en estudiar estabilidad desde el sentido de **Lyapunov**

(Haykin, 2009)

Neurodinámica (2/2)

- Neurodinámica determinista – el sistema se define por un conjunto de ecuaciones diferenciales no lineales, que definen la evolución exacta del modelo como una función del tiempo (ejemplo redes de Grossber y Hopfield
- Neurodinámica estadística – el sistema está perturbado por la presencia de ruido, y se representa por ecuaciones diferenciales no lineales estocásticas, dando la solución en términos probabilísticos (ejemplo red de Amari)

Sistemas dinámicos

- Un sistema dinámico es aquel cuyo estado cambia con el tiempo
- Un sistema dinámico se puede definir con un **modelo en el espacio de estados** a través de un sistema de ecuaciones diferenciales:

(Campo vectorial)

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{F}(\mathbf{x}(t))$$

Ecuación de estado

$$\boldsymbol{X}(t) = [x_1(t), x_2(t), \ldots\ x_N(t)]$$

Representan a las variables de estado del sistema,
N = orden del sistema

13.2 DYNAMIC SYSTEMS

In order to proceed with the study of neurodynamics, we need a *mathematical model* for describing the dynamics of a nonlinear system. A model most naturally suited for this purpose is the *state-space model*. According to this model, we think in terms of a set of *state variables* whose values (at any particular instant of time) are supposed to contain sufficient information to predict the future evolution of the system. Let $x_1(t)$, $x_2(t)$, ..., $x_N(t)$ denote the state variables of a nonlinear dynamic system, where continuous time t is the *independent variable* and N is the *order* of the system. For convenience of notation, these state variables are collected into an N-by-1 vector $\mathbf{x}(t)$ called the *state vector*, or simply *state*, of the system. The dynamics of a large class of nonlinear dynamic systems may then be cast in the form of a system of first-order differential equations written as

$$\frac{d}{dt} x_j(t) = F_j(x_j(t)), \qquad j = 1, 2, ..., N \tag{13.1}$$

where the function $F_j(\cdot)$ is, in general, a nonlinear function of its argument. We may express this system of equations in a compact form by using vector notation, as shown by

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{F}(\mathbf{x}(t)) \tag{13.2}$$

where the nonlinear function $\mathbf{F}$ is vector valued, each element of which operates on a corresponding element of the state vector:

$$\mathbf{x}(t) = [x_1(t), x_2(t), ..., x_N(t)]^T \tag{13.3}$$

(Haykin, 1999)

Espacio de estado

velocidad instantánea de
la trayectoria

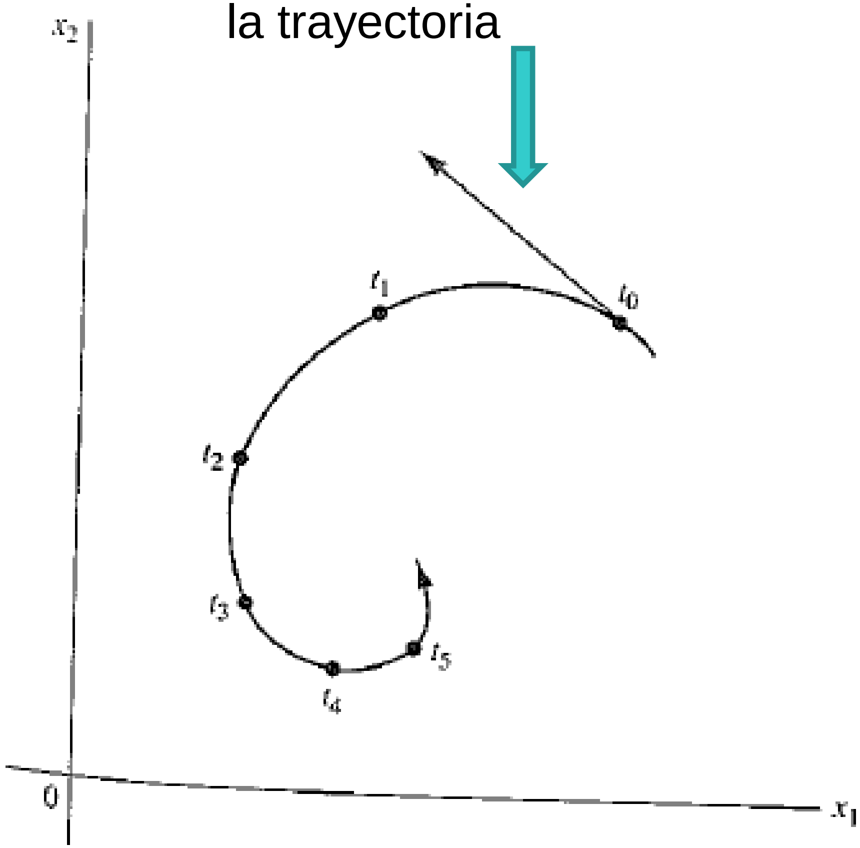


FIGURE 13.1 A two-dimensional trajectory (orbit) of a dynamic system.

El espacio de estado puede ser Euclidiano o no Euclidiano
Ejemplo: esfera, círculo, *Variedad (manifold)*

Cada punto representa un estado en un instante del tiempo

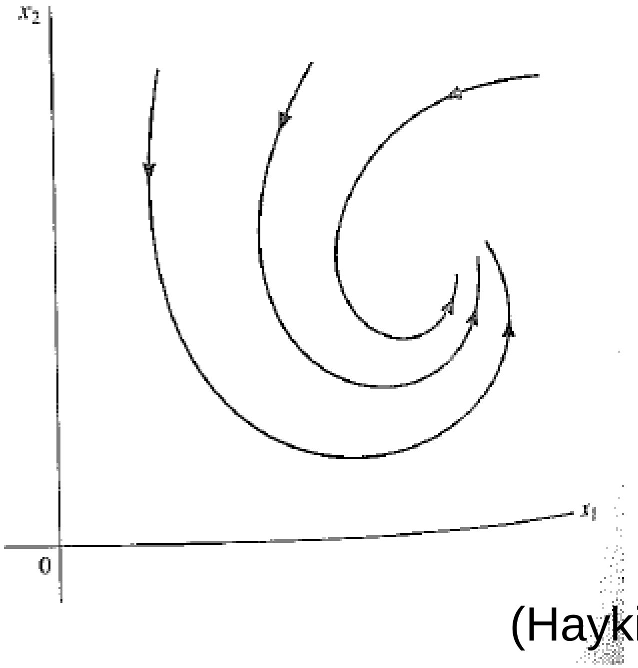
La curva representa una trayectoria y órbita del sistema

(Haykin, 1999)

Descripción gráfica de trayectorias

Una familia de trayectorias….

FIGURE 13.2 A two-dimensional state (phase) portrait of a dynamic system.



Cada línea representa una condición inicial diferente en el sistema dinámico

(Haykin, 1999)

Redes recurrentes inspiradas en Física Estadística

- Unidades de cómputo (neurones) no lineales.
- Conexiones sinápticas (pesos) simétricas.
- Uso abundante de retro-alimentación.

Las Redes de Hopfield

- Hopfield conceptualizó las redes neuronales como sistemas dinámicos con energía y mostró su semejanza con ciertos modelos físicos.
- Hopfield propuso varios modelos de redes recurrentes. En este tipo de redes, la salida de cada neurón se calcula y se retro-alimenta como entrada, calculándose otra vez, hasta que se llega a un punto de estabilidad.
- Supuestamente los cambios en las salidas van siendo cada vez mas pequeños, hasta llegar a cero, esto es, alcanzar la estabilidad.
- Puede ser que una red recurrente nunca llegue a un punto estable.

φ_j

Consideraciones Dinámicas

- Dada una red recurrente de N neurones con acoplamiento simétrico, esto es $w_{ij} = w_{ji}$, donde w_{ij} es la conexión de *i* a *j*, la salida del neurón *j* está dada por la ecuación:

$$X_j = \varphi_j(\upsilon_j)$$

- donde φ_j es la no-linealidad de tipo sigmoide del neurón *j*.
- X_j υ_j son funciones en el tiempo.

Dinámica de las Redes Recurrentes de Hopfield

- Está dada por el conjunto de ecuaciones diferenciales no lineales acopladas del tipo:

$$C_j \frac{\partial v_j}{\partial t} = \sum_{i=1, i \neq j}^{N} W_{ji} \varphi_j(v_j) - \frac{v_j}{R_j} - \theta_j$$

Para j = 1,2, ... N

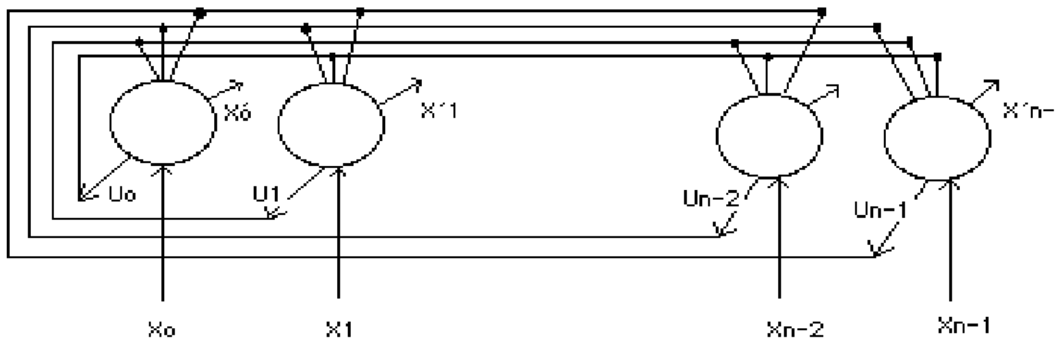
$$\theta_j = umbral$$

- $C_j =$ Controla el cambio del potencial $v_j(t)$ (efecto capacitivo).
- $R_j =$ Pérdidas debido a resistencia en la entrada al elemento j.

Configuración de la Red

- Se utiliza principalmente con entradas binarias.
- Se puede utilizar como una memoria asociativa, o para resolver problemas de optimización.
- Una <u>memoria asociativa</u> o dirigida por contenido es aquella que se puede accesar teniendo una parte de un patrón de entrada, y obteniendo como resultado el patrón completo.
- Hopfield también utilizó sus redes para resolver un problema de optimización: el agente viajero. Además construyó una red con circuitos integrados que convierte señales analógicas en digitales.

Modelo Básico de Hopfield



- n es el número de nodos en la red.
- Las entradas X_o, X_1 ... X_{n-1} se aplican a la red en el tiempo t = 0. Pueden tomar valores de +1 ó -1.
- Las salidas U_o, U_1... U_{n-1} se van calculando y recalculando, hasta que sus valores ya no cambian. Cuando esto sucede, se tiene la salida de la red, y $X'_i = U_i$ para i= 1.. n-1

Algoritmo de Entrenamiento de la red Hopfield

Paso único: Calcule los valores de los pesos que conectan a los nodos, utilizando la siguiente fórmula:

$$t_{ij} = \begin{cases} \sum_{s=0}^{m-1} x_{is} x_{js} & si\ i \neq j \\ 0 & si\ i = j \end{cases}$$

donde t_{ij} es el peso que va del neurón *i* al neurón *j*, y x_{is} es el valor del *i-ésimo* elemento de la *s-ésima* clase; *m* es el número de clases que se desean aprender. En notación matricial, la matriz de pesos se define como:

$$\mathbf{T} = \sum_i \mathbf{X_i^T X_i}\ ,\ \mathrm{t_{ii}} = 0$$

Lo que se conoce como el **producto externo** (*outer product*) de un vector renglón consigo mismo.

Algoritmo de evaluación de la red Hopfield (1/2)

Paso 1. Inicialice la red con un patrón de entrada:

$$U_i(0) = X_i \quad 0 \le i \le n-1$$

donde *n* es el número de nodos en la red

Algoritmo de evaluación de la red Hopfield (2/2)

Paso 3. Itere hasta converger siguiendo la siguiente fórmula:

$$U_j(t+1) = F\left(\sum_{i=0}^{n-1} t_{ij} U_i(t)\right) \quad 0 \leq j \leq n-1$$

donde F es una función escalón definida como:

$$F(x) = \left\{ \begin{array}{c} 1 \text{ si } x > 0 \\ -1 \text{ si } x < 0 \\ U_j(t) \text{ si } x = 0 \quad (\text{sin cambio}) \end{array} \right\}$$

Cuando la red converge, su salida representa al patrón que más se parece al patrón de entrada dado.

Un ejemplo Pequeño del Cálculo de la Matriz de Pesos **T**

$$\mathbf{X_1} = (-1,-1,1)$$
$$\mathbf{X}_2 = (1,\ 1,1)$$

$$\mathbf{X_1^T} \bullet \mathbf{X_1} = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix}$$

$$\mathbf{X_2^T} \bullet \mathbf{X_2} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\mathrm{T} = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 0 \\ 2 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

Haciendo la diagonal $= 0$:

$$\mathrm{T} = \begin{bmatrix} 0 & 2 & 0 \\ 2 & 0 & 0 \\ 0 & 2 & 0 \end{bmatrix}$$

Ejemplo 2

- Almacenar en una Red Hopfield los siguientes patrones:

$$\mathbf{X_1} = (1,\ 1,\ 1, -1)$$
$$\mathbf{X}_2 = (-1,-1,-1, 1)$$

$$\mathbf{X_1^T} \bullet \mathbf{X_1} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ -1 & -1 & -1 & 1 \end{bmatrix}$$

$$\mathbf{X_2^T} \bullet \mathbf{X_2} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & -1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ -1 & -1 & -1 & 1 \end{bmatrix}$$

Ejemplo 2 (cont.)

$$\mathbf{X_1^T} \bullet \mathbf{X_1} + \mathbf{X_2^T} \bullet \mathbf{X_2} = \begin{bmatrix} 2 & 2 & 2 & -2 \\ 2 & 2 & 2 & -2 \\ 2 & 2 & 2 & -2 \\ -2 & -2 & -2 & 2 \end{bmatrix},$$

haciendo la diagonal $= 0$:

$$\mathbf{T} = \begin{bmatrix} 0 & 2 & 2 & -2 \\ 2 & 0 & 2 & -2 \\ 2 & 2 & 0 & -2 \\ -2 & -2 & -2 & 0 \end{bmatrix}$$

Ejemplo 2 (cont.)

Supongamos que deseamos recuperar el patrón mas cercano a:

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 1 & -1 \end{pmatrix}$$

1º. Paso: $\mathbf{U}(\mathbf{0}) = \mathbf{A}$

2º. Paso: $\mathbf{U(0)} \cdot \mathbf{T} = \begin{pmatrix} 1 & 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} 0 & 2 & 2 & -2 \\ 2 & 0 & 2 & -2 \\ 2 & 2 & 0 & -2 \\ -2 & -2 & -2 & 0 \end{pmatrix} = \begin{pmatrix} 6 & 6 & 6 & -6 \end{pmatrix}$

$$\mathbf{U}(\mathbf{1}) = \mathbf{F}\left(\mathbf{U(0)} \cdot \mathbf{T}\right) = \begin{pmatrix} 1 & 1 & 1 & -1 \end{pmatrix}$$

En este punto se cumple que **U(1)** es igual al **U(0)**, por lo que el sistema ya está estable y el proceso termina.

El patrón mas parecido a **A** es (1 1 1 -1)

Ejemplo 2 (cont.)

Ahora hallaremos el patrón mas parecido a $\mathbf{A} = \begin{pmatrix}-1 & -1 & -1 & -1\end{pmatrix}$

$$\mathbf{U}(\mathbf{0}) \cdot \mathbf{T} = \begin{pmatrix}-1 & -1 & -1 & -1\end{pmatrix} \begin{pmatrix} 0 & 2 & 2 & -2 \\ 2 & 0 & 2 & -2 \\ 2 & 2 & 0 & -2 \\ -2 & -2 & -2 & 0 \end{pmatrix} = \begin{pmatrix}-2 & -2 & -2 & 6\end{pmatrix}$$

$$\mathbf{U}(\mathbf{1}) = \mathbf{F}(\mathbf{U}(\mathbf{0}) \cdot \mathbf{T}) = \begin{pmatrix}-1 & -1 & -1 & 1\end{pmatrix} \qquad \mathbf{U}(\mathbf{1}) \neq \mathbf{U}(\mathbf{0})!$$

$$\mathbf{U}(\mathbf{1}) \cdot \mathbf{T} = \begin{pmatrix}-1 & -1 & -1 & 1\end{pmatrix} \begin{pmatrix} 0 & 2 & 2 & -2 \\ 2 & 0 & 2 & -2 \\ 2 & 2 & 0 & -2 \\ -2 & -2 & -2 & 0 \end{pmatrix} = \begin{pmatrix}-6 & -6 & -6 & 6\end{pmatrix}$$

$$\mathbf{U}(\mathbf{2}) = \begin{pmatrix}-1 & -1 & -1 & 1\end{pmatrix} \qquad \mathbf{U}(\mathbf{2}) = \mathbf{U}(\mathbf{1}) \Rightarrow FIN!$$

Representación del sistema dinámico de Hopfield

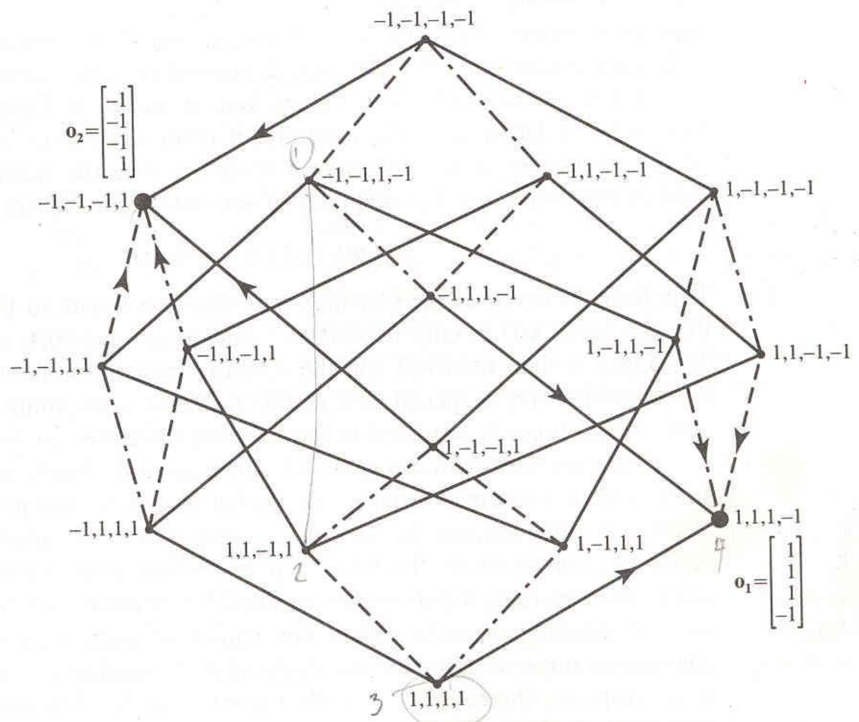


Figure 2.11 Four-dimensional hypercube with two equilibrium states in Example 2.2. (Zurada 92)

Ejemplo de Aprendizaje con Hopfield

- La siguiente figura, publicada en (Lippman 87), muestra el resultado obtenido al construir una memoria asociativa utilizando una red de Hopfield con 120 nodos.
- La red fue entrenada con los patrones mostrados en la parte superior de la figura. Después de entrenada, se le mostró a la red el número "3", distorsionado de manera aleatoria,
- Cambiando cada bit con una probabilidad de 0.25. Este patrón se aplicó en el tiempo t = 0. Las salidas obtenidas en t = 0 y en las primeras 7 iteraciones se muestran en la parte posterior de la figura.

Ejemplo del comportamiento de la red Hopfield (Lippman 87)

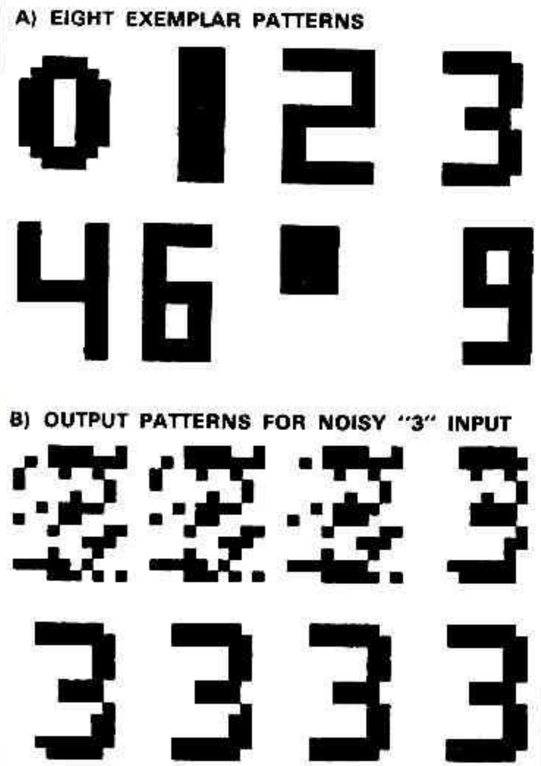


Figure 5. An example of the behavior of a Hopfield net when used as a content-addressable memory. A 120 node net was trained using the eight exemplars shown in (A). The pattern for the digit "3" was corrupted by randomly reversing each bit with a probability of .25, and then applied to the net at time zero. Outputs at time zero and after the first seven iterations are shown in (B).

Ventajas y Desventajas de las Redes de Hopfield

- Prácticamente no existe tiempo de entrenamiento, ya que éste no es un proceso adaptativo, sino simplemente el cálculo de una matriz (T).
- Las redes de Hopfield son bastante tolerantes al ruido, cuando funcionan como memorias asociativas.
- El número de patrones a almacenar (o aprender) es bastante limitado comparado con el número de nodos en la red. Según Hopfield, el número de clases a aprender no puede ser mayor de 0.15 veces el número de nodos en la red.

Las Redes de Hopfield Para Optimización

- En 1985, Hopfield y D.W. Tank publicaron un artículo llamado “Neural Computation of Decisions in Optimization Problems”[1], en el cual proponen una solución al problema del agente viajero (Traveling Salesman Problem, TSP).
- TSP consiste en planear el itinerario de un vendedor que tiene que visitar N ciudades, de manera que la distancia recorrida sea mínima.
- El vendedor empieza en una determinada ciudad y viaja visitándolas todas, solo una vez. Se supone que todas las ciudades están conectadas entre sí, y se cuenta con la información de la distancia entre todas las ciudades.

[1] Biological Cibernetics Vol. 52 pp. 141 –154 , 1985

Las Redes de Hopfield Para Optimización (cont.)

- Este es un problema combinatorio del tipo *NP-Complete* (NP = *Non-deterministic Polynomial*).
- Hay N!/2N posibles soluciones. Por ejemplo, si N=60 existen aproximadamente $69.34X10^{78}$ posibles soluciones.
- La solución encontrada por Hopfield y Tank no es la óptima, pero las respuestas se encuentran mas o menos rápidamente, tal vez después de N^3 cálculos.
- Esta solución depende de parámetros difíciles de asignar, muchas veces obtiene resultados malos, pero es ingeniosa.

DEFINICION DEL PROBLEMA

- Supóngase que hay N ciudades: A,B,C,D... a visitar y que la distancia de la ciudad *x* a la *y* está dada por el valor d_{xy}.

- Una ruta posible se puede representar como un conjunto de N renglones, uno por cada ciudad, formada de 0´s y 1´s.
- En cada renglón hay un solo 1, el cual indica la posición de esa ciudad en la ruta. Una ruta válida tiene solo un uno por renglón y un uno por columna.

Ejemplo de representación

- Suponiendo que el agente debe visitar 4 ciudades, llamadas A, B, C y D, se puede definir a un posible "tour" con la siguiente matriz:

	1	2	3	4
A	0	1	0	0
B	0	0	0	1
C	1	0	0	0
D	0	0	1	0

- Las columnas indican el orden de visita, y los renglones la ciudad a visitar en dicho orden
- Esta matriz representaría al "tour" C-A-D-B.
- Cada posición en la matriz será referenciada como Vxy

Matriz de representación

- Asimismo, la matriz se puede representarse en un vector con N^2 elementos, donde N es el número de ciudades.
- A su vez, este vector puede representarse en una red de Hopfield de N^2 neurones.
- El objetivo del entrenamiento es hacer converger la red hacia una ruta válida, en el cual exista la mínima energía posible.

Función de Energía

- La representación del problema establece la restricción de que hay un solo “uno” por renglón y un solo “uno” por columna. Además, sabemos que el objetivo es minimizar la distancia entre ciudades.
- Hopfield y Tank definieron una formula de energía que contiene dichas restricciones:

$$E = \frac{A}{2}\sum_{x}\sum_{i}\sum_{j\neq i} V_{xi}V_{xj} + \frac{B}{2}\sum_{i}\sum_{x}\sum_{y\neq x} V_{xi}V_{yi} + \frac{C}{2}\left(\sum_{x}\sum_{i} V_{xi} - n\right)^2 +$$

$$\frac{D}{2}\sum_{x}\sum_{y\neq x}\sum_{i\neq x} d_{xy}V_{xy}(V_{y(i+1)} + V_{y(i-1)})$$

- Donde V_{xy} representa la salida del neuron que muestra que la ciudad *x* está en la posición *y*.

Función de Energía (cont.)

- En la función se puede notar lo siguiente:
  - El primer término es cero si y solo si hay un solo “uno” en cada renglón. De otra manera, el término toma un valor mayor que cero.
  - El segundo término es cero si y solo si hay solo “uno” en cada columna.
  - El tercer término será cero si y solo si hay exactamente N “unos” en la matriz tour.
  - El cuarto término representa la longitud de un tour válido.
  - Considérese que los subíndices de V están definidos en términos de módulo n, esto es:
  - $V_{j+N}=V_j$

Cálculo de salida la red

- La salida de cada nodo se calcula como:

$$Vx,i = g(\mu x,i) = 1/2(1+\tanh(\mu x,i/\mu o))$$

- $\mu_{x,i}$ representa la entrada de cada neuron. Su valor deberia cambiar de manera que se reduzca la energía del sistema, este cambio se pude escribir como:

$$\mu_{x,i} = \mu_{x,i} + \left(\frac{\delta\mu_{x,i}}{\delta t}\right)\frac{1}{\Delta t}$$

donde:

$$\frac{\delta_{\mu x,i}}{\delta_t} = \frac{\mu_{x,i}}{\tau} - A\sum_{j\neq i} V_{x,j} - B\sum_{y\neq x} V_{y,i} - C\left(\sum_x \sum_j V_{x,j} - n\right)^2 - D\left(\sum_y d_{x,y}(V_{y,i}+1+V_{y,i}-1)\right)$$

Cálculo de salida la red (cont.)

- Los valores de los parámetros A,B,C,D, μ, Δt y μ_0 deben tomar un valor inicial.
- Desgraciadamente la convergencia de la red va a depender de estos valores, los cuales son realmente difíciles de establecer.
- Hopfield y Tank han sugerido que el valor inicial sea:
- $$\mu_0 = {}^{1}\!/_{N}$$

continúa…

CALCULO DE LA RED (Cont.)

añadiendo un poco de ruido para romper la simetria esto es:

$$\mu o = \mu o + noise(0.1\ \mu\ inicial)$$

Donde:

$$noise(x) = \#\ \text{al azar entre 0 y x}$$

- En algunas pruebas que realizaron Hopfield y Tank utilizaron los siguientes valores:

A=1,000	ή=1.5*N
B=1,000	μo=0.02
C=0.75*D*N	t=1
D=1,000	Dt=20,000

- Hopfield y Tank mostraron sus resultados con un experimento con 10 ciudades. De 20 corridas que hicieron, 16 resultaron con rutas validas y cerca del 50% de las 50 soluciones fueron de las mas cortas halladas por el metodo exhaustivo.[Hopfield 87]

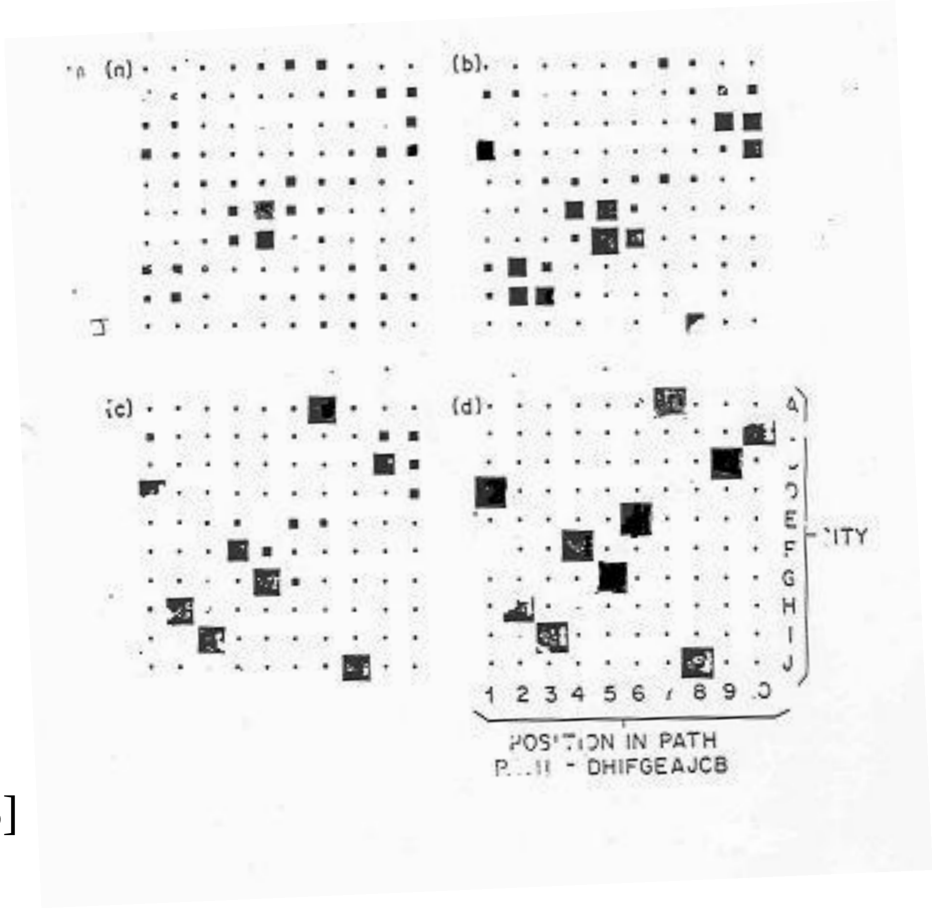
CALCULO DE LA RED (Cont.)

- Las salidas cambian de manera descrita al principio de intervalos de tiempo.
- La función de energía de este sistema esta definida por:

$$E = \frac{1}{2}\left[x - \sum_j 2^j OUT_j\right]^2 + \sum_j (2^{2^{j-1}})\left[OUT_j(1 - OUT_j)\right]$$

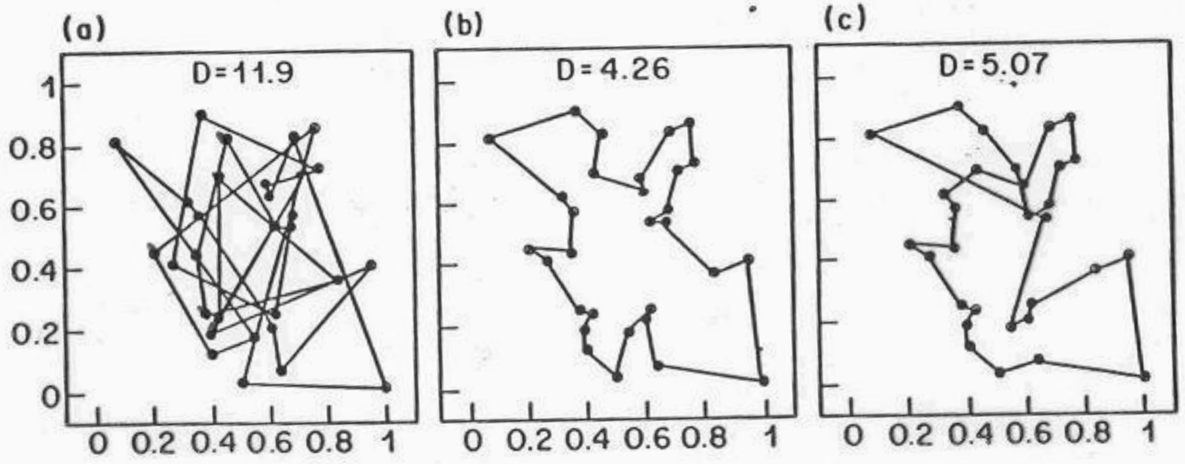
- Cuando E es minimizada, las salidas deseadas se han alcanzado.

Ejemplo. Valores en diferentes etapas de aprendizaje



[Hopfield & Tank 85]

AlgunosTours hallados



[Hopfield & Tank 85]

Otras Aplicaciones de la Red de Hopfield

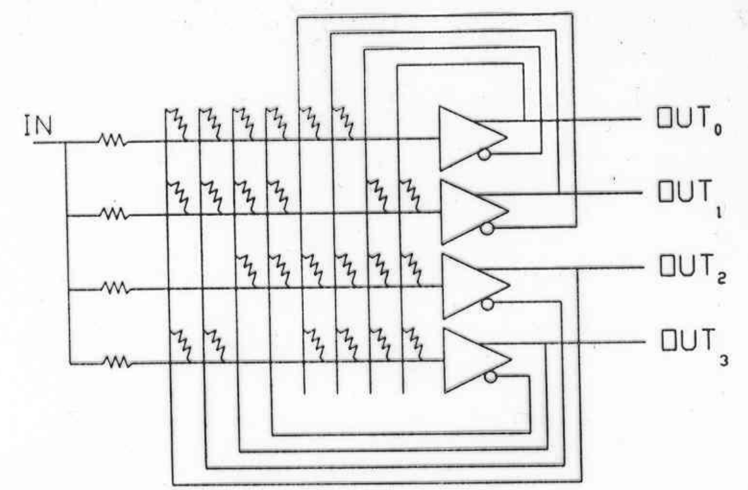


Figure 6-4. Four-Bit Analog-to-Digital Converter Using Hopfield Net

[Zurada 92]

Otras aplicaciones de la Red de Hopfield

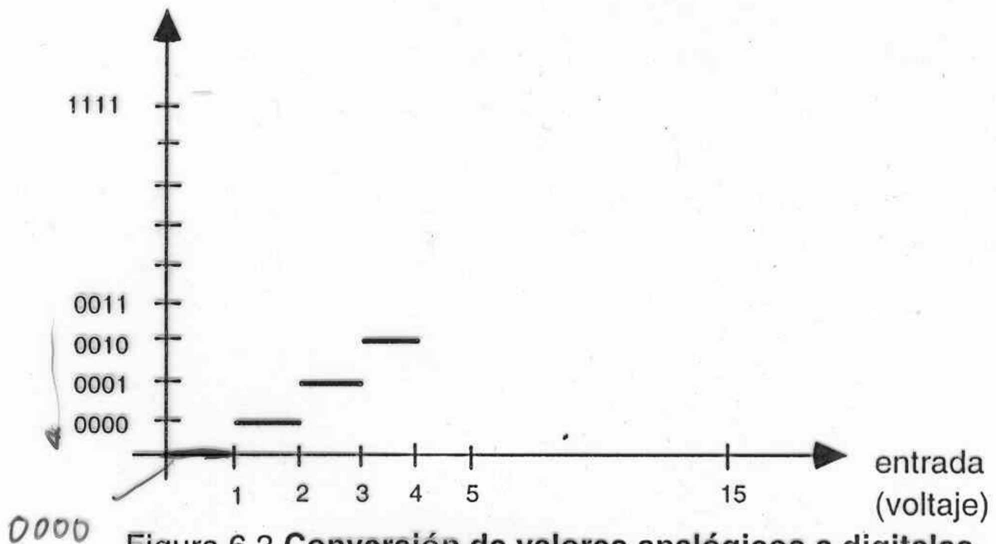


Figura 6.3 **Conversión de valores analógicos a digitales**

[Zurada 92]