Redes Neuronales

Curso 2008/09

Facultad de Informática

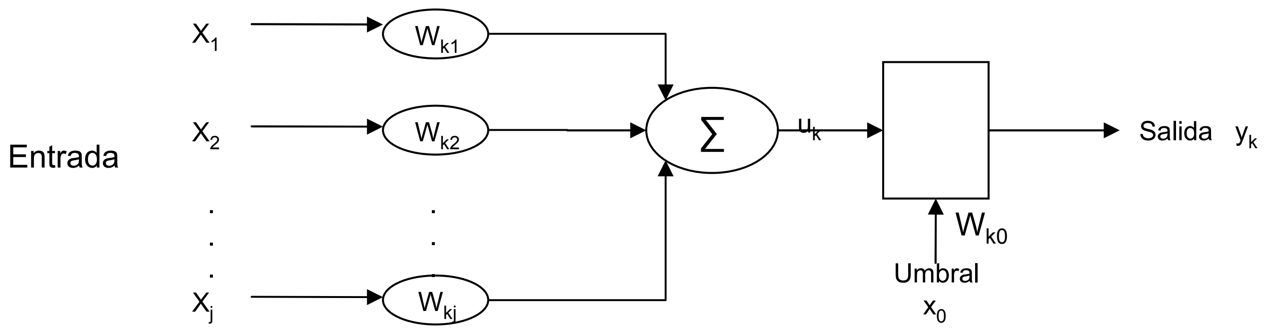
UNIVERSIDADE DA CORUÑA

Sumario

- Introducción
- Red de Hopfield
  - Memoria asociativa
  - Minimización energia
- Máquina de Boltzman

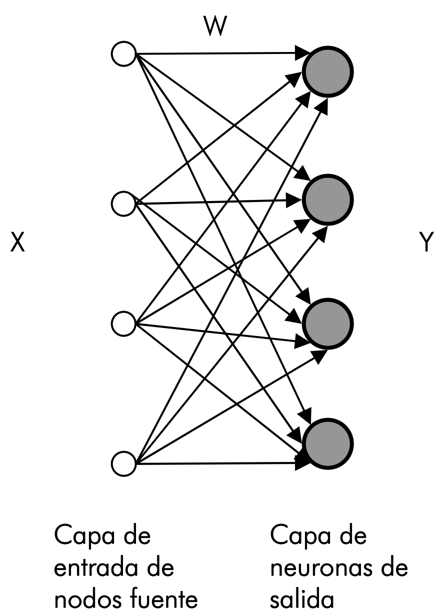
Introducción

- Red de neuronas = modelo computacional, paralelo y distribuido, basado en la conexión de varios elementos de proceso (neuronas) para realizar una función común
- Neurona artificial o elemento de proceso:

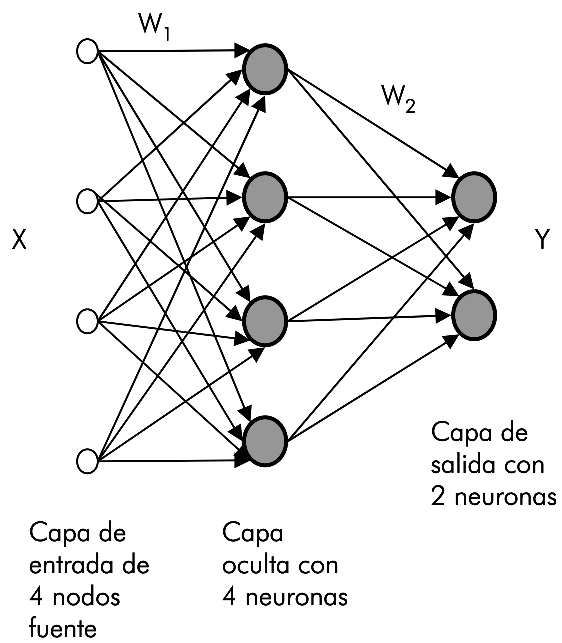


Introducción

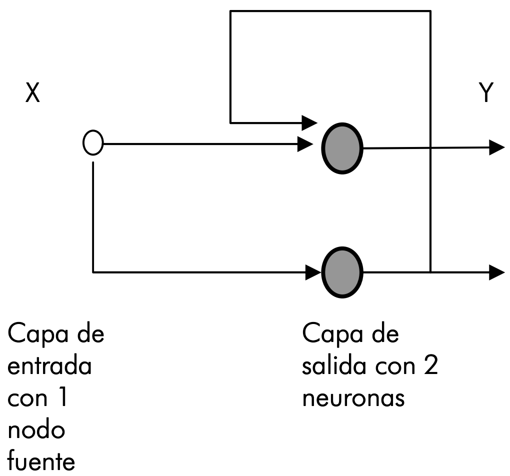
- RNAs monocapa alimentadas hacia delante.



- RNAs multicapa alimentadas hacia delante.



- RNAs con retroalimentación.



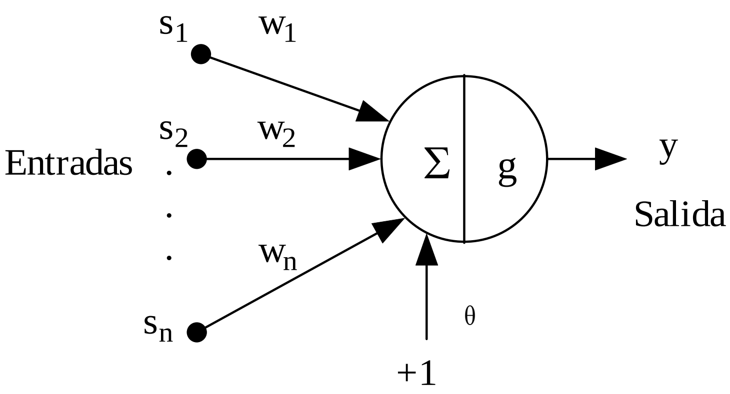
Algunas redes recurrentes

- Red de Hopfield
- Máquina de Boltzman

Red de Hopfield. Introducción

- Propuesta en 1982 por J. J. Hopfield :
  - Almacenar información en un configuración dinámicamente estable
- Basada el modelo de McCulloch-Pitts

Modelo de McCulloch-Pitts

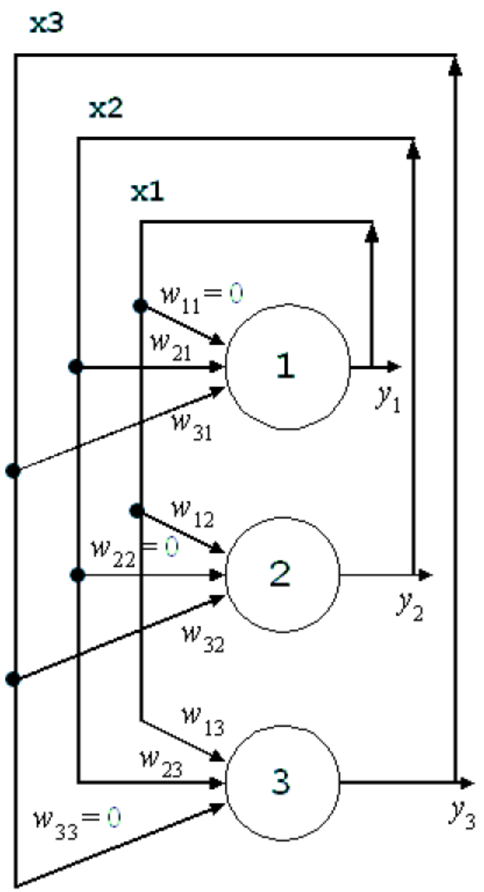


- La función de activación *g* es la función escalón
  - Tienen salida 1 sii $\Sigma w_{ij} s_j \geq \theta_i$
  - 0 en otro caso
- Las neuronas se actualizan síncronamente

Redes de Hopfield. Introducción

- Redes neuronales con "estructura recurrente"
- Inspiradas en "física estadística"
- Características:
  - Unidades de computación no lineales
  - Conexiones sinápticas simétricas
  - Uso abundante de retroalimentación

Diagrama de la red de Hopfield



Red de Hopfield. Introducción (II)

- Los elementos de proceso *i* se actualizan asíncronamente:

  "Escoge un elemento *i* aleatoriamente.

  Si $\Sigma w_{ij} s_j \geq \theta_i$, se activa.

  De otra forma se desactiva."

- La probabilidad de activar dos elementos simultáneamente es 0
- Pesos simétricos $w_{ij} = w_{ji}$

Red de Hopfield. Introducción (III)

- Procedimiento mecánico de actualización del estado de la neurona seleccionada entre aquellas que quieren cambiar, eligiendo la neurona aleatoriamente y una cada vez.
- Se repite hasta que no haya variaciones en el estado

Conceptos básicos

- Sea $y_j = g_j(v_j)$, donde v_j es la activación del elemento j, g_j es la función de activación e y_j la salida.
- El estado de la neurona se puede definir por su activación $v_j(t)$ o salida $y_j(t)$
- Efectos postsinápticos inducidos en la neurona j debidos a las actividades presinápticas de las neuronas $i=1, 2 \ldots N$, siendo $i \neq j$
- Umbral θ

Memoria asociativa o memoria direccionable por el contenido.

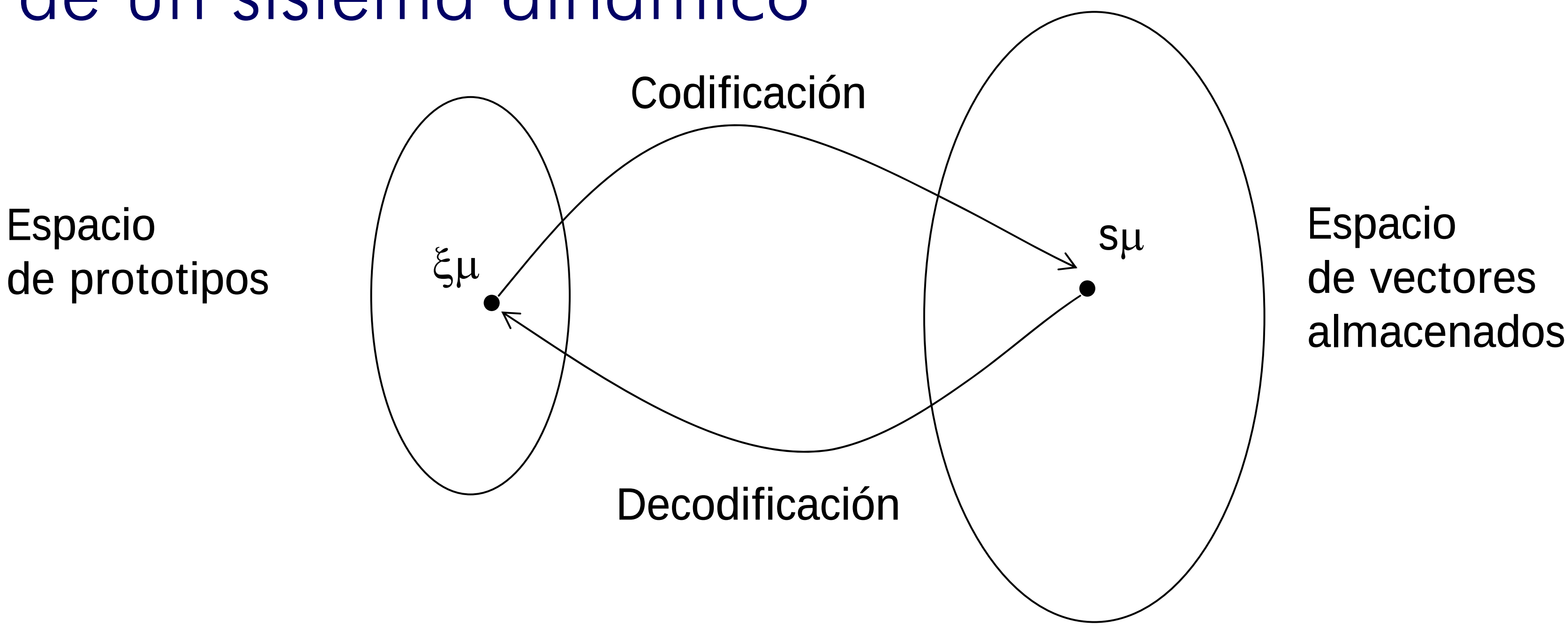
- Función principal:
  - Recuperar un patrón almacenado en la memoria como respuesta a la presentación de una versión con ruido o incompleta de dicho patrón
- Una memoria asociativa se construye a partir de la información que se desea almacenar.

Memoria asociativa (CAM). Ejemplo

- *item* almacenado en memoria es “H.A. Kramers & G.H. Warnier Physi Rev. 60 252 (1941)”.
- *Content-addresable memory (CAM)* sería capaz de recuperar este item con suficiente información parcial.
- “& Wannier (1941)” sería suficiente
- Una memoria ideal podría tratar errores y recuperar esta referencia incluso con la entrada “Wannier, (1941)”.

Memoria asociativa (CAM)

- Mapear una "memoria fundamental o prototipo" ξ_μ en un punto fijo (estable) s_μ de un sistema dinámico



Red de Hopfield. CAM

- Sistema dinámico cuyo espacio de estados contiene un conjunto de puntos fijos (estables) representando las memorias fundamentales o prototipos del sistema.

Memoria asociativa (CAM). Funcionamiento

- Representar un patrón particular como un punto de inicio en el espacio de estados
- El sistema evolucionará en el tiempo y convergerá a un prototipo.
- Hay dos fases:
  - Almacenamiento
  - Recuperación

Características operacionales (I)

- Cada neurona tiene dos estados, dependiendo del nivel de activación:
  - "on" $s_i = +1$
  - "off" $s_i = -1$
- Red con N neuronas
  - $S = [s_1, s_2, \ldots, s_N]^T$

Características operacionales (II)

- Un par de neuronas *i, j* están conectados por un peso sináptico w_{ji}
- La activación en la neurona *j* viene dado por:

$$v_j = \sum_{i=1}^{N} w_{ji} s_i - \theta_j$$

- La neurona j cambia su estado de acuerdo a:
  - $s_i = 1$ si $v_i > 0$
  - $s_i = -1$ si $v_i < 0$
  - Si $v_i = 0$, el estado no cambia

Almacenamiento (I)

- *p* vectores N-dimensionales $\{\xi_\mu | \mu = 1,2,\ldots p\}$

  $\xi_{\mu,i}$ indica el elemento *i* del prototipo
- El peso sináptico de la neurona *i* a la neurona *j* se define como:

$$w_{ji} = \frac{1}{N}\sum_{\mu=1}^{p}\xi_{\mu,j}\xi_{\mu,i}; \quad w_{ii} = 0$$

- Matriz de pesos

$$W = \frac{1}{N}\sum_{\mu=1}^{p}\xi_\mu\xi_\mu^T - \frac{p}{N}I$$

Almacenamiento (II)

- No hay auto-feedback
- La matriz de pesos es simétrica, es decir, la influencia de la neurona *i* en la neurona *j* es igual que la influencia de la neurona *j* en la *i*.

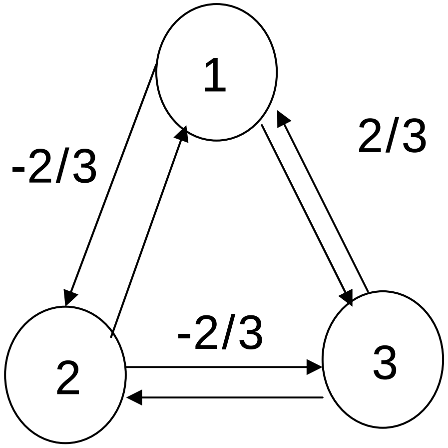
Recuperación

- Un vector N-dimensional **x** se le proporciona a la red
- Sus elementos son +1/-1
- Normalmente, representa una versión incompleta o con ruido de un prototipo.

Recuperación: regla dinámica

- Cada neurona *j* de la red, aleatoriamente, examina su potencial de activación y cambia, si es necesario, su estado
- La actualización del estado de una iteración a la otra es determinista
- La elección de la neurona es aleatoria y asíncrona
- Se realiza hasta que no sea requieran más cambios. Es decir, la red llega a un estado invariable en el tiempo.

Ejemplo



- Estados posibles: $2^3=8$
- Sólo dos de ellos son estables
  - (1,-1,1)
  - (-1,1,-1)

Ejemplo

- Para el estado (-1,1,-1

$$Wy = 1/3\begin{bmatrix} 0 & -2 & 2 \\ -2 & 0 & -2 \\ 2 & -2 & 0 \end{bmatrix}\begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} = 1/3\begin{bmatrix} -4 \\ 4 \\ -4 \end{bmatrix}$$

- Para el estado (1,-1,1)

$$Wy = 1/3\begin{bmatrix} 0 & -2 & 2 \\ -2 & 0 & -2 \\ 2 & -2 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} = 1/3\begin{bmatrix} 4 \\ -4 \\ 4 \end{bmatrix}$$

- Matriz de pesos

$$W = 1/3[-1\ 1\ -1]\begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} + 1/3[1\ -1\ 1]\begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} - 2/3\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = 1/3\begin{bmatrix} 0 & -2 & 2 \\ -2 & 0 & -2 \\ 2 & -2 & 0 \end{bmatrix}$$

Limitaciones

- No todos los prototipos son estables
- Estados falsos. Estados estables de la red diferentes a los almacenados
  - Si ξ es un prototipo de la red también lo es $-\xi$.
  - La combinación de un número impar de prototipos también es un prototipo
- Si se desea que todos los prototipos sean recuperados con una probabilidad próxima a 1, se ha estimado la relación:

$$p <= N/4\ln(N)$$

Función de Energía

- Los pesos se calculan por adelantado y NO forman parte de un sistema dinámico.
- Los vectores de salida cambian en función del tiempo y SÍ forman parte de un sistema dinámico.
- En la teoría de sistemas dinámicos se puede probar un teorema acerca de la existencia de estados estables que emplea el concepto de una función llamada *Función de Lyapuov* o *Función de Energía*:

  "Si se pude hallar una función acotada de las variables de estado de un sistema dinámico, tal que todos los cambios de estado den lugar a una disminución del valor de la función, entonces el sistema posee una solución estable"

Energía de la red de Hopfield

- En el caso de la Red de Hopfield esa función existe

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} s_i s_j w_{ij} + \sum_{i=1}^{N} s_i \theta_i$$

- Si se escoge una unidad *i*, si la función de activación no cambia su estado s_i, la función de energía no cambia.

Cambios en el estado

- Cambio de una sola componente *k*

$$E=-\frac{1}{2}\sum_{i=1;i\neq k}^{N}\sum_{j=1;\,j\neq k}^{N}s_i s_j w_{ij}+\sum_{i=1;i\neq k}^{N}s_i\theta_i-\frac{1}{2}s_k\sum_{i=1;i\neq k}^{N}s_i w_{ik}-\frac{1}{2}s_k\sum_{j=1;\,j\neq k}^{N}s_j w_{kj}+s_k\theta_k$$

- Variación en *k*

$$\Delta E=-\frac{1}{2}s_k\sum_{i=1;i\neq k}^{N}s_i w_{ik}-\frac{1}{2}s_k\sum_{j=1;\,j\neq k}^{N}s_j w_{kj}+s_k\theta_k=s_k\sum_{i=1;i\neq k}^{N}s_i w_{ik}+s_k\theta$$

- Se suelen considerar umbrales 0

$$\Delta E=\Delta s_k\sum_{i=1;i\neq k}^{N}s_i w_{ik}$$

Cambio de 1 a -1

- Si s_k inicialmente es 1, y $\Sigma_{i \neq k} s_i w_{ik} < 0$ entonces s_k cambia de 1 a -1 y el resto de unidades *i* no cambian
- El cambio en E viene dado por:

$$\Delta E = -\Delta s_k \Sigma_{i \neq k} s_i w_{ik} \leq 0.$$

Cambio de -1 a 1

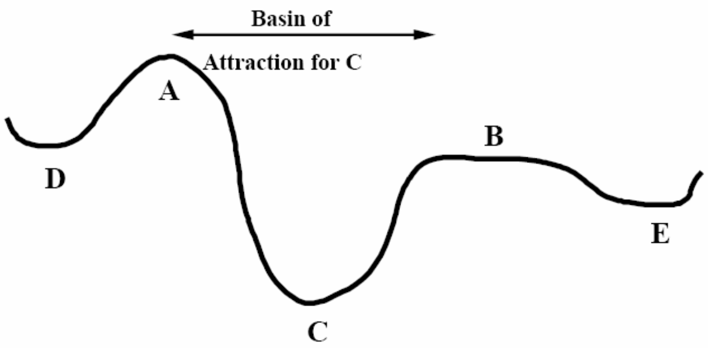
- Si s_k inicialmente es -1, y $\Sigma_{i \neq k} s_i w_{ik} > 0$ entonces s_k cambia de -1 a 1, y el resto de unidades *i* no cambian
- El cambio en E viene dado por:

$$\Delta E = -\Delta s_k \Sigma_{i \neq k} s_i w_{ik} < 0$$

- *Cualquier cambio* $\Delta E \leq 0$

Minimización de la energia

- En cada actualización $\Delta E \leq 0$
- Por tanto, la dinámica de la red tiende a mover E hacia el mínimo.
- Hay mínimos locales.
- No se garantiza el mínimo global.



Problema del viajante (I)

- Hay N ciudades, con una carretera de longitud d_{ij} entre la ciudad i y la ciudad j.
- El viajante quiere encontrar un camino para visitar las ciudades de manera óptima teniendo en cuenta que:
  - Cada ciudad se visita sólo una vez
  - La ruta es lo más corta posible
- Es un problema NP-Completo: los algoritmos conocidos que lo resuelven tienen complejidad exponencial

Problema del viajante (II)

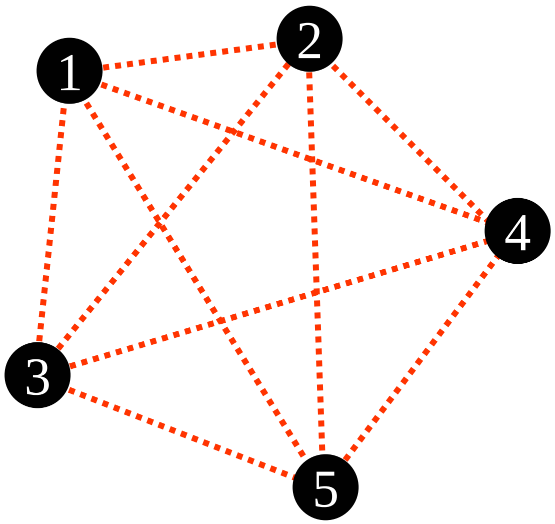
- La neurona N_{ij} indica la decisión de ir de la ciudad *i* a *j*.
- El coste de este movimiento es simplemente d_{ij}.
- Se representa la restricción "visitar cada ciudad una vez" como, para la ciudad *j*, hay sólo una ciudad *i* desde la que se llega directamente.

Problema del viajante (IV)

	Ciudades (i,k)				
Paradas (j,l)	1	0	0	0	0
	0	1	0	0	0
	0	0	1	0	0
	0	0	0	1	0
	0	0	0	0	1

Problema del viajante. Crecimiento exponencial

- Número de posibles combinaciones que viene dado por el factorial del número de ciudades (N!/2N)
- N=5 12 caminos



- N=10 181140 de caminos, no abordable

Problema del viajante. Función de coste

$$E = \frac{A}{2}\sum_{i}^{N}\sum_{j}^{N}\sum_{l;l\neq j}^{N} s_{ij}s_{il} + \frac{B}{2}\sum_{i}^{N}\sum_{j}^{N}\sum_{k;k\neq i}^{N} s_{ij}s_{kj} + \frac{C}{2}\left(\sum_{i}^{N}\sum_{j}^{N} s_{ij} - N\right)^2 + \frac{D}{2}\sum_{i}^{N}\sum_{j}^{N}\sum_{k;k\neq i}^{N} d_{ij}\left(s_{ij}s_{kj+1}\right)\left(s_{ij}s_{kj}\right)$$

Problema del viajante. Función de coste

$$E = \boxed{\frac{A}{2}\sum_{i}^{N}\sum_{j}^{N}\sum_{l;l\neq j}^{N} s_{ij}s_{il}} + \frac{B}{2}\sum_{i}^{N}\sum_{j}^{N}\sum_{k;k\neq i}^{N} s_{ij}s_{kj} + \frac{C}{2}\left(\sum_{i}^{N}\sum_{j}^{N} s_{ij} - N\right)^{2} + \frac{D}{2}\sum_{i}^{N}\sum_{j}^{N}\sum_{k;k\neq i}^{N} d_{ij}\left(s_{ij}s_{kj+1}\right)\left(s_{ij}s_{kj}\right)$$

Sólo una ciudad puede aparecer en una parada en la ruta

Problema del viajante. Función de coste

$$E = \frac{A}{2}\sum_{i}^{N}\sum_{j}^{N}\sum_{l;l\neq j}^{N} s_{ij}s_{il} + \boxed{\frac{B}{2}\sum_{i}^{N}\sum_{j}^{N}\sum_{k;k\neq i}^{N} s_{ij}s_{kj}} + \frac{C}{2}\left(\sum_{i}^{N}\sum_{j}^{N} s_{ij} - N\right)^2 + \frac{D}{2}\sum_{i}^{N}\sum_{j}^{N}\sum_{k;k\neq i}^{N} d_{ij}\left(s_{ij}s_{kj+1}\right)\left(s_{ij}s_{kj}\right)$$

Una parada sólo se corresponde a una ciudad

Problema del viajante. Función de coste

$$E = \frac{A}{2}\sum_{i}^{N}\sum_{j}^{N}\sum_{l;l\neq j}^{N} s_{ij}s_{il} + \frac{B}{2}\sum_{i}^{N}\sum_{j}^{N}\sum_{k;k\neq i}^{N} s_{ij}s_{kj} + \boxed{\frac{C}{2}\left(\sum_{i}^{N}\sum_{j}^{N} s_{ij} - N\right)^{2}} + \frac{D}{2}\sum_{i}^{N}\sum_{j}^{N}\sum_{k;k\neq i}^{N} d_{ij}\left(s_{ij}s_{kj+1}\right)\left(s_{ij}s_{kj}\right)$$

En el recorrido sólo hay N ciudades

Problema del viajante. Función de coste

$$E = \frac{A}{2}\sum_{i}^{N}\sum_{j}^{N}\sum_{l;l\neq j}^{N} s_{ij}s_{il} + \frac{B}{2}\sum_{i}^{N}\sum_{j}^{N}\sum_{k;k\neq i}^{N} s_{ij}s_{kj} + \frac{C}{2}\left(\sum_{i}^{N}\sum_{j}^{N} s_{ij} - N\right)^{2} + \boxed{\frac{D}{2}\sum_{i}^{N}\sum_{j}^{N}\sum_{k;k\neq i}^{N} d_{ij}\left(s_{ij}s_{kj+1}\right)\left(s_{ij}s_{kj}\right)}$$

Distancia total

Problema del viajante. Función de coste

$$E = \frac{A}{2}\sum_{i}^{N}\sum_{j}^{N}\sum_{l;l\neq j}^{N} s_{ij}s_{il} + \frac{B}{2}\sum_{i}^{N}\sum_{j}^{N}\sum_{k;k\neq i}^{N} s_{ij}s_{kj} + \frac{C}{2}\left(\sum_{i}^{N}\sum_{j}^{N} s_{ij} - N\right)^{2} + \frac{D}{2}\sum_{i}^{N}\sum_{j}^{N}\sum_{k;k\neq i}^{N} d_{ij}\left(s_{ij}s_{kj+1}\right)\left(s_{ij}s_{kj}\right)$$

Se compara con la expresión general de la función de energía de Hopfield:

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\sum_{l=1}^{N} w_{ij,kl}s_{ij}s_{kl} + \sum_{i=1}^{N}\sum_{j=1}^{N}\theta_{ij}s_{ij}$$

$$w_{ij,kl} = -A\delta_{ik}(1-\delta_{jl}) - B\delta_{jl}(1-\delta_{ik}) - C - Dd_{ik}(\delta_{jl} - \delta_{jl+1})$$

Neuronas estocásticas

- La regla determinista para activar una neurona j:
  - $s_j=1$ si $v_j>0$
  - $s_j=-1$ si $v_j<0$
  - Si $v_j=0$ el estado no cambia
- Se modifica por:
  - $s_j=1$ con $P(v_j)$
  - $s_j=-1$ con $1-P(v_j)$
  - Si $v_j=0$, $s_j=+/-1$ con probabilidad de 1/2

Neuronas estocásticas (II)

- P(v) debe satisfacer los valores límite:
  - P(V)=0 si v→-∞
  - P(V)=1 si v → ∞
- Además, deber incrementarse monotónicamente con v

$$P(v) = \frac{1}{1 + \exp(-2v / T)}$$

T (pseudotemperatura) es un parámetro para controlar el nivel de ruido y por tanto la incertidumbre de activar la neurona

Enfriamiento simulado (simmulated annealing)

- Método general para evitar caer en mínimos locales mediante “saltos” a estados con más energía
- Analogía con el proceso de un artesano construyendo una espada de una aleación.
  - Calienta el metal, entonces lentamente lo enfría mientras martillea la hoja para darle la forma deseada.
  - Si enfría la hoja rápidamente, el metal formará parches
  - Si el metal se enfría lentamente mientras se le da forma, formará una aleación uniforme

Máquina de Boltzman

- Tiene similitudes y diferencias con la red de Hopfield
- Boltzmann-Hopfield Similitudes:
  - Unidades de procesado con estados binarios(± 1)
  - Conexiones simétricas
  - Para actualizar, se escoge aleatoriamente una neurona y sólo una a la vez
  - No hay auto-alimentación

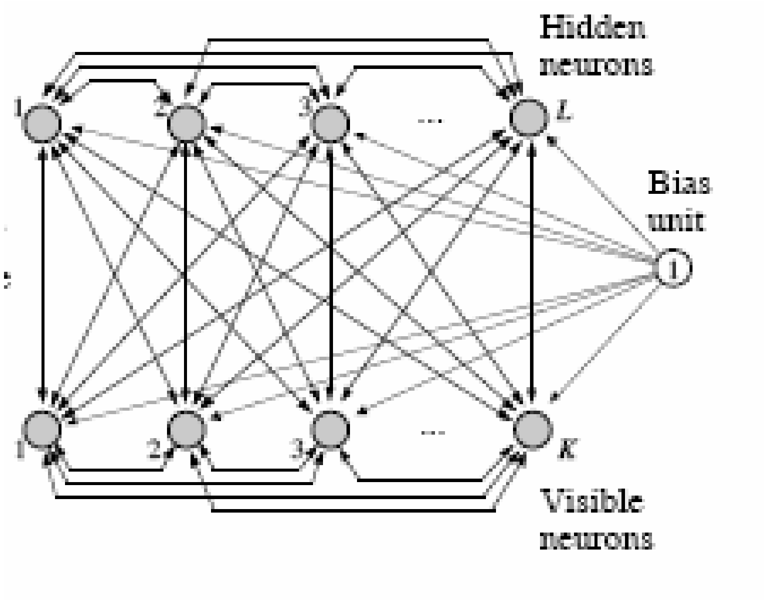
Hopfield-Boltzman diferencias

- Permite el uso de *neuronas ocultas*.
- Usa *neuronas estocásticas* con un mecanismo de activación probabilístico
- Se puede entrenar mediante una forma probabilística de supervisión

Estructura de la máquina de Boltzman

- Las neuronas son de dos grupos: visibles y ocultas
  - Visibles: proporcionan una interfaz entre la red y su entorno
  - Durante la fase de entrenamiento, las neuronas visibles están "fijas (clamped)" en estados específicos del entorno
  - Las neuronas ocultas operan libremente, se usan para explicar las restricciones subyacentes en los vectores de entrada del entorno

Arquitectura de la máquina de Boltzman

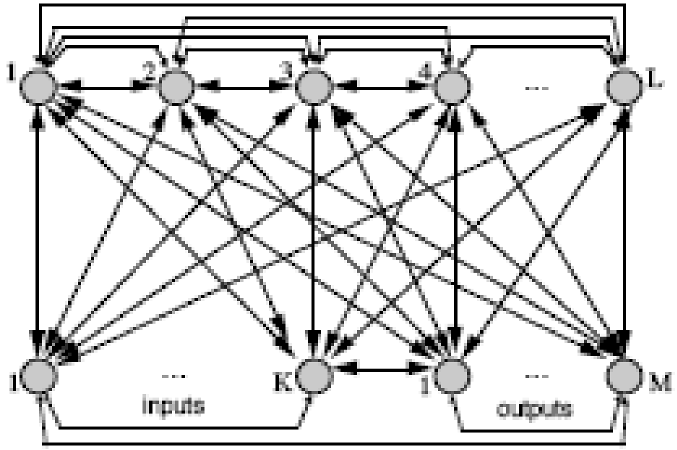


Máquina de Boltzman (I)

- Modelado de una distribución de probabilidad que se especifica mediante patrones “fijados (clamping)” en los neuronas visibles con la probabilidad adecuada
- La red puede realizar *pattern completion*

Máquina de Boltzman (II)

- Las neuronas "visibles" se pueden dividir en entradas y salidas



Máquina de Boltzman(II)

- Realiza asociación mediante supervisión
- Las neuronas de entrada reciben información del entorno y las de salida proporcionan un resultado a un usuario final
- Inferencia hacia delante: infiere los patrones de salida de las entradas

Aprendizaje

- Objetivo: Categorizar patrones de entrada de acuerdo a la distribución de Boltzman
- Se asume:
  - Cada vector de entrada persiste el tiempo suficiente para permitir que la red alcance el *equilibrio térmico*
  - No hay estructura en orden secuencial en la que los vectores del entorno estén fijados (*clamped*) a unidades visibles de la red.

Minimización de la energía (I)

- Elegir una neurona oculta al azar, neurona j, y cambiarle el estado (voltear) de s_j a $-s_j$ a una temperatura T con probabilidad

$$\Pr ob(s_j \rightarrow -s_j) = \frac{1}{1 + Exp(-\Delta E_j / T)}$$

donde *Ej es el cambio de energía* resultante del cambio de estado.

Minimización de la energía (II)

- Se define la función de energía como:

$$E = -\frac{1}{2}\sum_{i}\sum_{j;i\neq j} w_{ji} s_i s_j$$

- Este sumatorio aborda tanto neuronas ocultas como visibles
- La condición i ≠j indica no autoalimentación
- *wji* es el peso de la unidad *i* a la unidad *j*
- Hay un umbral externo θj aplicado a la unidad j

Minimización de la energía (III)

- Si el procedimiento de "voltear" se aplica repetidamente a las neuronas, la red alcanza un equilibrio térmico
- En este equilibrio, las unidades cambiarán de estado, pero la probabilidad de encontrar la red en cualquier estado permanece constante y obedece a la distribución de Boltzman.

Aprendizaje de Boltzman

- Primero considerar una temperatura (T) elevada, descenderla gradualmente hasta que alcanza el equilibrio térmico en una serie de temperaturas, como describe el procedimiento de enfriamiento simulado

Modelo perfecto

- Un determinado conjunto de pesos sinápticos se dice que constituye un modelo perfecto de la estructura del entorno si conduce exactamente a la misma distribución de probabilidad que los estados de las neuronas visibles
- Para alcanzarlo, el número de neuronas ocultas tiene que ser exponencialmente mayor que las visibles

Bibliografía

- S. Haykin, “Neural Networks. A comprehensive foundation”, McMillan College Publishing Company, 1994
- R. Rojas, “Neural Networks”, Springer-Verlag, Berlin, 1996
- J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", *Proceedings of the National Academy of Sciences of the USA*, vol. 79 no. 8 pp. 2554-2558, April 1982.
- Kirkpatrick, S.; C. D. Gelatt, M. P. Vecchi. “Optimization by simulated annealing”,Science. New Series 220 (4598): 671-680, 1983
- E. Aarts & J. Korst “Simulated Annealing and Boltzmann Machines: a stochastic approach to combinatorial optimization and neural computing”. Chichester: John Wiley & Sons, 1989.

Applet:

http://www.cbu.edu/~pong/ai/hopfield/hopfieldapplet.html