

Modelos Neuronales Competitivos: Kohonen & ART

Pedro Isasi Viñuela

Dep. Computación

Universidad Carlos III - Madrid

email: isasi@ordago.uc3m.es

1.- Introducción a las redes autoorganizativas

El estudio e investigación sobre las Redes de Neuronas Artificiales (RNA) ha sufrido un incremento notable, a lo largo de la última década, debido en gran parte al nacimiento de nuevas teorías y modelos artificiales así como a la evolución tecnológica que ha permitido la implantación de dichos modelos. Sin embargo, el estudio de los mecanismos que permiten al hombre comportarse de la forma que lo hace, y que le distingue en gran parte del resto de los animales, eso que ha dado en llamarse desde los anales de la historia “Inteligencia”, no es algo nuevo, sino que ha venido preocupando a los hombres desde el nacimiento del pensamiento filosófico.

Fue necesario esperar tres mil años para que cesase la discusión que mantenían los antiguos Griegos sobre los papeles específicos del corazón y cerebro. Mientras que para algunos el corazón era el centro motriz de sentimientos e inteligencia, por contra no fue hasta el siglo XIX cuando la teoría sobre el cerebro como órgano central del organismo, se extendió definitivamente por Europa.

Sin embargo, incluso en la actualidad, el papel del cerebro como determinante del comportamiento humano, es origen de gran controversia. Clásicamente, se ha pensado que el cerebro controlaba únicamente aquellas partes del organismo más cercanas a los comportamientos animales, como la visión, los órganos motrices; es decir aquello que sirve de enlace entre los seres vivos y su entorno; se incluía dentro de este campo de acción del cerebro funciones específicas de los

seres humanos pero a las cuales se encontró un fundamento biológico claro, como la capacidad de hablar, o de leer.

Ultimamente, una gran parte de los científicos se mueven en una dirección más determinista de aquellos aspectos que clásicamente tenían un valor más espiritual, más etéreo, es decir más alejado de la pura explicación biológica, como los sentimientos: el miedo, la angustia, la alegría, el amor; la capacidad intelectual; los mecanismos del entendimiento; y el comportamiento que llamamos inteligente son objeto de estudio por parte de los neuropsicólogos y neurobiólogos en la creencia de que existen mecanismos biológicos deterministas capaces de explicar estas ideas.

Las RNA han contribuido, en parte, al estudio y al descubrimiento de teorías en esta dirección. Una buena manera de probar una teoría biológica es la de construir con su base un modelo artificial que tenga el comportamiento que tienen los sistemas biológicos, en el área restringida de la teoría, por supuesto. Así pues, por un lado las RNA sirven de ayuda a esa inquietud del ser humano por desentrañar su propio comportamiento. Por otra parte, cualquiera puede darse cuenta de que el cerebro humano es superior a los computadores digitales para muchas tareas. Por ejemplo, el procesado de información visual que un niño puede hacer al poco tiempo de nacer es mucho mejor y más rápido que el que podrían hacer, en la actualidad, los supercomputadores más rápidos. El cerebro, además, posee otras características que serían deseables en los sistemas artificiales:

- Es robusto y tolerante a fallos. En cada día de nuestra vida sufrimos la pérdida de células nerviosas sin que esto afecte significativamente a nuestras capacidades.
- Es flexible. Se puede ajustar de una manera sencilla y automática a un nuevo entorno.
- Es capaz de manejar información inconsistente, difusa o con un alto grado de ruido, de una manera espectacularmente eficaz.

- Es altamente paralelo, lo que redundará en una gran velocidad de actuación imprescindible para la supervivencia en ambientes naturales.

Así pues, la investigación e imitación del comportamiento biológico puede servir de gran avance en el estado actual de los sistemas artificiales, y es así como lo han entendido el gran número de investigadores que, de forma gradual, han ido interesándose por esta nueva ciencia.

1.1.- Características de las Redes de Neuronas Biológicas

Cualquier modelo de redes de neuronas que quiera acercarse al comportamiento de los sistemas biológicos debe contar con una serie de características esenciales, que son las que proporcionan a los sistemas biológicos gran parte de sus capacidades.

Una de las principales características de las Redes de Neuronas Biológicas (RNB), es su dimensión, en el cerebro existen entre 10^{11} y 10^{14} neuronas, y cada una de ellas posee entre 1.000 y 10.000 conexiones con otras tantas células (Simpson 1990), esto deja a los sistemas artificiales varios órdenes de magnitud por debajo.

El funcionamiento general de los sistemas nerviosos es altamente paralelo. Es este grado de paralelismo el que permite que, con una velocidad de señal relativamente baja (1 mseg), el cerebro pueda reaccionar con rapidez ante decisiones, continuamente necesarias, en las que existen un gran número de variables. Dichas decisiones resultan extraordinariamente complejas de plantear, y, en muchos casos, irresolubles en tiempo real, para los mejores computadores secuenciales, a pesar de poseer estas velocidades de 6 órdenes de magnitud superiores a las RNB.

En los sistemas nerviosos, se realizan procesos de convergencia y divergencia de la información. Mediante la existencia de múltiples conexiones de entrada y de salida para cada neurona, se produce una multiplicación del uso de una determinada información (divergencia), así como una combinación de diversas señales de entrada para producir un efecto conjunto en una sola neurona (convergencia).

Las RNB poseen una plasticidad intrínseca, mediante la cual están dotadas de la capacidad de aprender a partir de la experiencia y adaptarse a entornos nuevos con extrema facilidad.

Poseen una cierta determinación genética, que viene determinada por la existencia de un mapa de conexiones prefijado. Esto significa, ante todo, que el diseño principal de la arquitectura de las distintas RNB está predeterminado genéticamente, y ha sido seleccionado de entre los modelos posibles por su valor adaptativo para cada especie. Así pues, la arquitectura general de las conexiones está diseñada de antemano, si bien la potencia relativa de dichas conexiones serán fijadas a lo largo del aprendizaje, lo cual no entra, en absoluto, en contradicción con la idea anterior de plasticidad, como en un principio podría parecer.

La capacidad de aprendizaje de las RNB, se debe a la capacidad del organismo de modificar la permeabilidad sináptica entre neuronas, lo que incide en el poder que una neurona tiene de excitar a otra con la que está conectada, y por ende en la facilidad de propagación de los impulsos electro-químicos por el entramado de la red.

Esa capacidad de modificación de la permeabilidad sináptica puede estar regida por dos mecanismos diferentes, que dan lugar a los dos grandes bloques en que pueden dividirse las RNA. Son los llamados sistemas supervisados y no supervisados.

Básicamente, en los sistemas de aprendizaje supervisado, disponemos de un profesor externo o de algún tipo de información global, que nos permitirá decidir cosas del tipo: cuando dejar de aprender, con que intensidad aprender o cada cuanto tiempo aprender. Podemos, también, manejar cierto tipo de información de error que nos permite ponderar la rectificación que debemos introducir en la Red. En cambio, en los sistemas de aprendizaje no supervisado, no tenemos nada de esto, tratamos de descubrir, mediante información local y de control, las propiedades emergentes de los datos. Aquí, nos dedicaremos íntegramente a este último tipo de RNA.

1.2.- Aprendizaje no Supervisado

1.2.1.- Características

Las RNA con aprendizaje no supervisado, son aquellas que no necesitan de un "profesor" o "supervisor" externo para realizar su aprendizaje. Son capaces de modificar sus parámetros internamente, adaptándose al entorno de la mejor manera posible. Biológicamente, existen, claros ejemplos de aprendizaje no supervisado, así como de supervisado.

Se trata, básicamente, de que la red debe descubrir, por si sola, características, regularidades, correlaciones, o categorías en los datos de entrada, y obtenerlas de forma codificada a la salida. Por tanto, estas unidades y conexiones, podemos decir que muestran cierto grado de autoorganización.

El aprendizaje supervisado solo puede hacer algo útil si, en los datos de entrada, existe cierto tipo de redundancia. Sin redundancia sería imposible encontrar patrones o características en los datos, lo cual se asemeja necesariamente a ruido aleatorio. En este sentido, la redundancia es fuente de conocimiento [BAR89]. De una manera más técnica, el contenido total de información del conjunto de los datos de entrada, es menor que el máximo que podría ser soportado por el mismo canal, esa diferencia es la redundancia.

El tipo de patrón que una red con aprendizaje no supervisado puede detectar depende de la arquitectura de la misma. Pero, es interesante conocer que tipo de cosas puede "contarnos" una red representadas en sus salidas; existen varias posibilidades:

- Familiaridad. Un único valor continuo de salida nos puede decir como de parecido es un valor de entrada al "valor tipo" o media de una serie de valores presentados con anterioridad. La red, a su vez, puede ir aprendiendo lo que se entiende por "valor tipo".
- Análisis de las componentes principales. Se trata de detectar cuales de las componentes del conjunto de entrada caracterizan en mayor grado al conjunto de datos, de forma que las demás pudieran eliminarse sin una significativa pérdida de información.
- Clustering. A partir de un conjunto de entrada no sólo nos interesa conocer el valor representativo del conjunto, sino si existen

varios de dichos valores. Es decir, si podemos dividir el conjunto de datos en diferentes categorías, y decidir a que clase pertenece cada dato.

- **Prototipado.** Igual que en el caso anterior, pero en vez de obtener como resultado a que clase pertenece el dato de entrada, obtenemos un prototipo o ejemplar de la clase a que pertenece dicho dato de entrada.

- **Codificación.** Podemos obtener, a la salida, una versión codificada del dato de entrada; es decir, un dato de menor dimensión que mantenga el máximo de información que le sea posible. Esto podría utilizarse para compresión de datos, antes de ser transmitidos por un canal de un limitado ancho de banda, por el cual la señal original no podría transmitirse. Siempre y cuando, claro está, que exista la posibilidad de construir una red capaz de restaurar, con posterioridad, el dato original.

- **Extracción y relación de características.** La idea es realizar un mapa topológico de los datos de entrada, a través del diseño de la red, de tal forma que patrones de entrada parecidos, produzcan respuestas similares en células cercanas en la anterior topología. Con esto, se espera que si existe una organización global, en función de la topología definida de los patrones de entrada, sea puesta de relieve a la salida.

Estos ejemplos no tienen por que ser disjuntos, y se pueden combinar de varias maneras. Por ejemplo, el problema de codificación, podría realizarse mediante análisis de componentes principales, o por "clustering", lo cual recibe el nombre de cuantización de vectores "vector quantization". También podemos utilizar el método de análisis de las componentes principales, para reducir la dimensión de los datos, como paso previo a "clustering" o detector de características.

A veces los métodos de aprendizaje no supervisado pueden dar mejores resultados en situaciones, incluso, donde es posible aplicar métodos de aprendizaje supervisado:

- El método de BackPropagation multicapa es tremendamente lento, en parte debido a que los mejores pesos de una capa dependen de todos los otros pesos de todas las otras capas. Esto se podría evitar,

en alguna medida, mediante un método no supervisado, o una solución híbrida, en la cual algunas capas se autoorganizan justo antes de pasar sus datos a una red supervisada.

- Incluso después del entrenamiento, en una red con aprendizaje supervisado, podría ser conveniente realizar algún tipo de aprendizaje no supervisado con objeto de que la red pueda adaptarse de forma gradual a los cambios que puedan producirse en su entorno.

Las arquitecturas de las redes con aprendizaje no supervisado, suelen ser bastante simples, la complejidad de las mismas radica en sus leyes de aprendizaje. Muchas de ellas tienen una sola capa, la mayoría son feed-forward (dirigidas hacia delante), con la notable excepción de la Teoría de la Resonancia Adaptativa. Suele haber, también, bastantes menos unidades que entradas, salvo en el caso de los Mapas de Kohonen en que, normalmente, ocurre lo contrario. Otra característica común a todas ellas es que suelen estar más cerca de las estructuras neurobiológicas que de las puramente computacionales, y se basan en estudios cerebrales, y tienden a imitar sus características y comportamiento.

Tal vez, los modelos más característicos de redes no supervisadas sean los de Kohonen y Grossberg, siendo los más mencionados en la literatura, y los más utilizados en casos prácticos. Es por ello que se han elegido como muestra de lo que es y lo que puede hacerse con los métodos de aprendizaje no supervisados.

1.3.- Regla de Hebb

Como ya se ha mencionado, en las RNA con aprendizaje no supervisado no existe ninguna información externa que nos indique si estamos produciendo resultados equívocos, ni que nos ayude a decidir como y en que grado modificar las conexiones. Entonces, ¿cómo podemos decidir esas modificaciones?, ¿en qué nos basaremos para dirigir el aprendizaje de nuestra red?

Donald Hebb, realizó una primera aproximación a la resolución de estas preguntas. En 1949 [HEBB49], enunció una regla que lleva su nombre y que ha pasado a ser un clásico en las RNA. La regla se refería a cierto comportamiento biológico que Hebb descubrió que se producía en las células cerebrales. Su enunciado es el siguiente: "Cuando un Axón de

una célula A está lo suficientemente cerca como para excitar a una célula B, y toma parte repetidamente en el proceso de disparo de dicha célula, se produce algún tipo de cambio metabólico en una de las células (o en las dos), que hace que la eficacia con la que A disparaba a B se vea incrementada".

Como se ve, esta regla de modificación sináptica no depende de ningún factor externo, simplemente hace que las células vayan influyéndose unas a otras; es decir, se autoconfigure la permeabilidad sináptica de la Red a partir de las reacciones a los estímulos recibidos. Esta regla ha demostrado ser de gran eficacia en las RNA, y es casi universalmente utilizada en los sistemas no supervisados.

Esta regla, tal y como fue enunciada por Hebb, deja abierto un gran abanico de posibilidades, de manera que, cuando una regla de aprendizaje en una RNA tiene forma análoga a lo expresado por Hebb, se dice que posee un aprendizaje tipo Hebb. Es decir, no existe una implementación de la regla de Hebb, sino un cúmulo de reglas de tipo Hebb.

A continuación, se detallan una serie de interpretaciones matemáticas, todas ellas utilizadas en diferentes modelos, de la regla de Hebb:

$$\Delta W_{ij} = a_i a_j$$

$$\Delta W_{ij} = \eta [a_i - \bar{a}_i][a_j - \bar{a}_j]$$

$$\Delta W_{ij} = \Delta a_i \Delta a_j$$

$$\Delta W_{ij} = \eta a_i W_{ij} \Delta a_j$$

$$\frac{dW_{ij}}{dt} = -W_{ij} + a_i a_j$$

$$\frac{dW_{ij}}{dt} = -W_{ij} + \sigma(a_i) \sigma(a_j)$$

$$\frac{dW_{ij}}{dt} = -W_{ij} + \frac{d[\sigma(a_i)]}{dt} \frac{d[\sigma(a_j)]}{dt}$$

Donde:

W_{ij} -> Peso de la conexión de las células i y j .

a_i -> Valor de activación de la célula i .

\bar{a}_j -> Media de los valores de activación.

$\sigma(x)$ -> Función de umbral, tipo sigmoideal.

2.- Mapas Autoorganizativos De Kohonen

Una de las propiedades del cerebro más pasadas por alto en los modelos artificiales existentes es aquella del significativo orden de sus unidades de proceso. Este orden hace que, unidades estructuralmente idénticas, tengan una diferente funcionalidad debida a parámetros internos que evolucionan de forma diferente según dicha ordenación de las células.

Esta propiedad "topológica" del cerebro parece ser de fundamental importancia para la representación de cierto tipo de información, en las imágenes visuales, en las abstracciones, etc. Estos mapas topológicos se encuentran presentes en la corteza cerebral y se encargan de diversas tareas de tipo sensorial y motor.

Cuando pensamos, en el procesado de información del subconsciente, realizamos una comprensión de la información formando representaciones reducidas con los aspectos más relevantes de las mismas, sin que se produzca por ello ninguna pérdida de conocimiento acerca de las interrelaciones entre ellas. El propósito de este procesado inteligente de la información parece ser la creación de imágenes del mundo observable a varios niveles de abstracción.

Así pues, uno de los rasgos esenciales de los sistemas biológicos es el de disponer de un sistema capaz de extraer de manera automática las características más relevantes de un conjunto de datos, en principio arbitrario, y de ámbito heterogéneo. Se trata de un sistema de los denominados de "*clustering*" que sea capaz de adaptarse, de forma automática, a los estímulos recibidos y producir una clasificación en

función de las características mencionadas, sin ningún mecanismo externo que determine dicha clasificación.

Esto es, en parte, posible gracias a la ya mencionada propiedad "topológica" de las RNB, según Kohonen (1982) ciertas Redes Neuronales pueden adaptar sus respuestas de tal forma que la posición de la célula que produce la respuesta pasa a ser específica de una determinada característica de la señal de entrada. Esta especificidad se da en el mismo orden topológico para la red que el que existe entre las características de las señales de entrada.

Esto quiere decir que, la estructura topológica de la red, absorbe a su vez aquella que se produce entre las características de los datos; y, por tanto, el sistema no es sólo capaz de realizar una clasificación de los estímulos, sino que, además, nos pondrá de relieve y conservará las relaciones existentes entre las diferentes clases obtenidas.

2.1.- Interacción lateral

Existen varios modelos que tratan de explicar como se pueden producir las anteriores propiedades a partir de sistemas neuronales. Estudios realizados sobre el neocortex cerebral [HUB65], demuestran que este está formado esencialmente por capas de células bidimensionales interconectadas, en cada capa, por conexiones laterales. Estas conexiones producen una interacción lateral de unas características muy relevantes.

Cada neurona está conectada con otras de su entorno de manera que produce una excitación en las más próximas y una inhibición en las más alejadas (Figura 1). Tanto la excitación como la inhibición laterales son gradualmente más débiles a medida que nos alejamos de la neurona en cuestión.

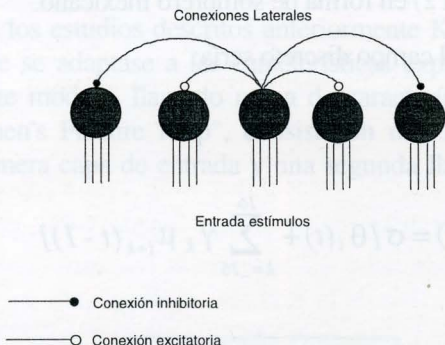


Figura 1.- Neuronas con conexión lateral.

Este mecanismo hace que, cuando un estímulo produce una reacción en una célula, las células de su inmediato entorno se vean influenciadas por dicha reacción, de una manera positiva las más cercanas y negativa las más alejadas; así como que, a medida que nos vamos alejando, esa influencia vaya progresivamente debilitándose. De esta manera, el orden de las neuronas influye directamente en la forma en que los estímulos van a ser propagados a través de la red, y en la respuesta de la misma.

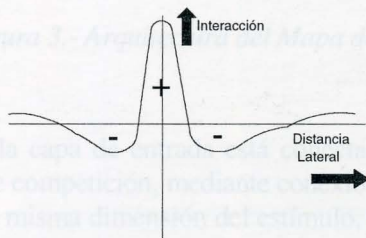


Figura 2.- Función de "Sombrero Mexicano" para la interacción lateral.

La función que establece esta interacción lateral viene a menudo definida por una función (Figura 2) en forma de sombrero mexicano.

Su formulación en el campo discreto sería:

$$\mu_i(t) = \sigma[\theta_i(t) + \sum_{k=-16}^{16} \gamma_k \mu_{i+k}(t-1)] \quad \mathbf{1}$$

Donde:

$\mu_i(t)$ es la salida de la unidad i .

$\sigma[x]$ es una función no lineal de tipo sigmoide.

γ_k es un coeficiente extraído de muestrear en el tiempo la señal de la Figura 2, de forma que, para $-1 < k < 1$, tenga un valor positivo, y negativo en el resto.

De esta forma, y en un ámbito reducido, en la Ecuación 1, entre -16 y 16, definimos una interacción lateral que proporciona a la red las características mencionadas con anterioridad.

2.2.- Mapas de Kohonen

A partir de los estudios descritos anteriormente Kohonen (1982) diseñó un modelo que se adaptase a las características expuestas de los modelos biológicos. Este modelo, llamado mapa de características de Kohonen, en Inglés "Kohonen's Feature Map", consiste en una Red Neuronal de dos capas, una primera capa de entrada y una segunda llamada de competición (Fig. 3).

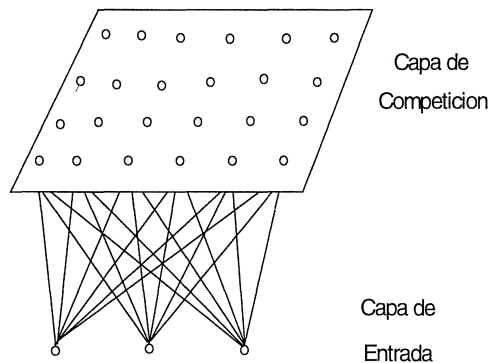


Figura 3.- Arquitectura del Mapa de Kohonen.

Cada célula de la capa de entrada está conectada con cada una de las células de la capa de competición, mediante conexiones ponderadas. La capa de entrada tendrá la misma dimensión del estímulo, y será excitada por este. Denotaremos al estímulo por un vector $\varepsilon = \{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n\}$, y a la salida de la capa de competición por otro vector $\tau = \{\tau_1, \tau_2, \dots, \tau_n\}$. Las conexiones entre la capa de entrada y de competición serán una matriz que tendrá una ordenación característica.

Se define para la salida la siguiente ecuación:

$$\tau_i = \sum_{j=1}^n \mu_{ij} \varepsilon_j \quad 2$$

Cada conjunto ordenado $(\mu_{i1}, \mu_{i2}, \dots, \mu_{in})$ puede verse, pues, como una especie de "imagen" que debe ser comparada con otro conjunto ordenado $(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n)$. El objetivo de la Red será el de adaptar sus parámetros de manera que, cada unidad, esté especialmente sensibilizada a un dominio de la señal de entrada en orden regular.

La comparación puede realizarse siguiendo varias medidas de distancia, en la ecuación 2 se utilizó el producto escalar de dos vectores; sin embargo, también se puede utilizar la distancia euclídea entre vectores:

Si llamamos:

$$x = [\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n]^T \quad y \quad m_i = [\mu_{i1}, \mu_{i2}, \dots, \mu_{in}]^T$$

Entonces se trata de calcular:

$$\|x, m_c\| = \min_i \|x, m_i\|.$$

Siendo $\|a, b\|$ la medida de distancia utilizada.

El proceso consiste, pues, en presentar un estímulo y propagarlo a través de la red, según una determinada función de comparación o de distancia, y elegir como ganadora a la célula que produzca una menor señal en la capa de competición.

Una vez realizado el proceso descrito, lo que queremos es que nuestra red responda de manera similar a estímulos parecidos, para ello aplicamos una regla de aprendizaje tipo Hebb, es decir reforzamos más aquellas unidades que hayan respondido en mayor grado al estímulo, de forma proporcional al valor del estímulo. Esto podría formalizarse de la siguiente forma:

$$\frac{d\mu_{ij}}{dt} = \alpha(t) \{ \tau_i(t) \varepsilon_j(t) - \Gamma[\tau_i(t)] \mu_{ij}(t) \} \quad 3$$

Donde tenemos que $\alpha(t)$ es una función de "olvido" decreciente con respecto al tiempo, y $\Gamma(x)$ es una función de umbral de la salida. Esta función será máxima para aquellas células que den una respuesta mínima, y mínima para aquellas que la den máxima. Es decir, dada la métrica anterior aplicaremos la función $\Gamma(x)$ a la salida de las células de la capa de competición, con la intención de obtener mayor respuesta a estímulos que se aproximen más a los almacenados en las conexiones, y viceversa.

La ecuación 3 puede discretizarse, para lo cual se define en primer lugar el concepto de vecindario. Un vecindario $N_c(S_i)$ de la célula c se define como el conjunto de aquellas células que estén a menos de una distancia S_i de c . La distancia S_i es discreta, lo que quiere decir que, para cada célula, definimos una serie de conjuntos $N_c(S_1)$, $N_c(S_2)$, etc., de la forma expuesta en la Figura 4.

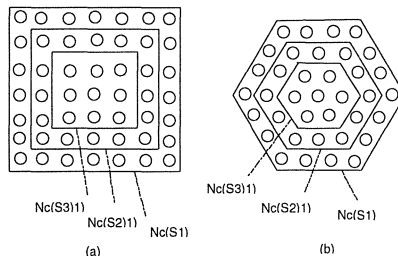


Figura 4.- Dos ejemplos de diferentes topologías de vecindarios .

El vecindario es lo que define la ordenación topológica de las células del sistema, y sería equivalente a las conexiones laterales del modelo de interacción lateral descrito con anterioridad. Las reglas de modificación sináptica definidas a partir de los vecindarios introducen, en el modelo, unas características topológicas parecidas a las del neocortex cerebral.

La función $\alpha(t)$ de olvido de la ecuación 3 podría definirse también en función del vecindario, bien de forma continua, o de forma discreta, de tal manera que su valor se fuera debilitando con la distancia a la célula i , para la que se calculan los nuevos pesos.

Por otra parte la función $\Gamma(\tau_i)$ se puede hacer igual a τ_i cuando i pertenezca a N_c e igual a cero en caso contrario, así pues la ecuación 3 quedaría:

$$\frac{d \mu_{ij}}{dt} = \alpha(t)[\varepsilon_j(t) - \mu_{ij}(t)] \text{ para } i \in N_c \quad 4$$

$$\frac{d \mu_{ij}}{dt} = 0 \text{ para } i \notin N_c$$

De esta manera, definimos un espacio alrededor de la célula ganadora en el cual las eficacias sinápticas de las células se verán incrementadas. Este incremento, expuesto en la ecuación 4, será igual para todas ellas, y será nulo para aquellas células exteriores al vecindario.

El valor α , se irá decrementando a lo largo del tiempo para evitar que los estímulos que se presenten en primer lugar tengan mayor relevancia que los que lo hagan en último, y debe cumplir las inecuaciones:

$$\sum_{s=0}^{\infty} \alpha(s) = \infty, \quad \sum_{s=0}^{\infty} \alpha(s)^2 < \infty$$

Por último, el vecindario puede tener una evolución similar, se elegirá un vecindario amplio al principio, para luego ir disminuyéndolo hasta llegar a la unidad.

Si elegimos la distancia euclídea como medida de similaridad entre vectores, la ecuación (4) quedará:

$$\|x(t_k) - m_c(t_k)\| = \min_i \|x(t_k) - m_i(t_k)\|$$

Y la actualización de los pesos:

$$m_i(t_{k+1}) = m_i(t_k) + \alpha(t_k)[x(t_k) - m_i(t_k)] \quad \text{para } i \in N_c$$

$$m_i(t_{k+1}) = m_i(t_k) \quad \text{en caso contrario.}$$

El algoritmo general de los Mapas topológicos de Kohonen, en forma de tabla, quedaría como se aprecia en la tabla 1.

- | | |
|----------------|--|
| Paso 1: | Inicializar pesos
Asignar a los pesos valores pequeños aleatorios. |
| Paso 2: | Presentar una nueva entrada.
El conjunto de aprendizaje se presenta cíclicamente hasta llegar a la convergencia de la red. Actualizar Alfa. |
| Paso 3: | Propagar el patrón de entrada hasta la capa de competición y obtener los valores de salida de las células de dicha capa. |
| Paso 4: | Seleccionar la célula <i>c</i> cuya salida sea mayor. |
| Paso 5: | Actualizar las conexiones entre la capa de entrada y la célula <i>c</i> , así como las de su vecindad, según su grado de vecindad. |
| Paso 6: | Volver al paso 2. |

Tabla 1.- Algoritmo de los Mapas Topológicos de Kohonen

Una demostración del funcionamiento de la red, nos la ofrecen los Mapas Topológicos Bidimensionales de Kohonen. Estos Mapas simplemente distribuyen los elementos de la red a lo largo de un conjunto de ejemplo de patrones de entrada. Si, además, hacemos que tanto la dimensión de los patrones, como la de las neuronas sea bidimensional,

podremos realizar una representación gráfica de los mismos y tener una buena referencia visual de lo que esta pasando en la red.

La aplicabilidad más generalizada de los Mapas de Kohonen, es la de clasificador, o extractor de características. Es el caso de cuando tenemos un conjunto de datos, que queremos agrupar según una serie de características de los mismos, pero que desconocemos. Teniendo la peculiaridad, además, de que la Red nos dará la clasificación y nos relacionará las clases entre sí.

En el ejemplo de la figura 5 se ha representado un conjunto de 2000 puntos distribuidos uniformemente a lo largo de un espacio bidimensional. Cada uno de estos puntos se corresponde con un patrón del conjunto de entrada a la red. Es decir, la capa de entrada de la red consta de dos neuronas, que determinan las coordenadas x e y respectivamente de cada uno de los puntos. En la capa competitiva, se han colocado un total de 100 neuronas distribuidas en un cuadrado de 10x10. Cada una de las neuronas de la capa competitiva está conectada a todas las neuronas de la capa de entrada, en este caso sólo dos, y esas conexiones determinan la posición de la neurona en el espacio bidimensional de los patrones. Es decir, se correspondería a las coordenadas x e y de la neurona en dicho espacio.

Por tanto, tendremos una representación gráfica de las neuronas de la capa competitiva como puntos dentro del espacio definidos por los valores de los pesos de las conexiones para dichas neuronas.

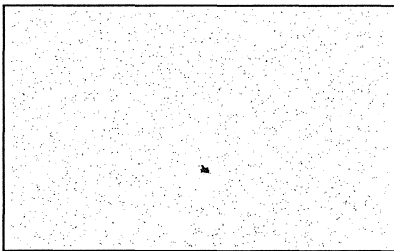


Figura 5.- Distribución uniforme, 1 iteración.

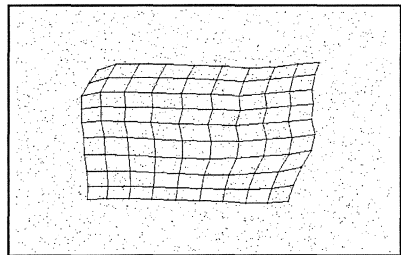


Figura 6.- Distribución unif. 18 iteraciones.

El significado de la distribución de las neuronas en un cuadrado es el siguiente: a cada neurona la conectamos con sus cuatro vecinas; es decir, la de arriba, abajo, izquierda y derecha. No se trata de una conexión física, sino de una relación entre neuronas a efectos de definición del vecindario. Definimos al grado de vecindad, como el número de células que habrá que recorrer para llegar desde una célula a otra, incluida la de destino. En la figura 6 se representan las neuronas conectadas por líneas según la definición anterior.

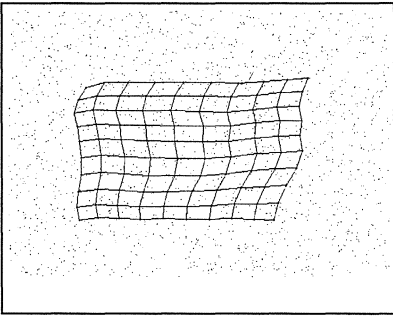


Figura 7.- Distribución uniforme para 24 iteraciones.

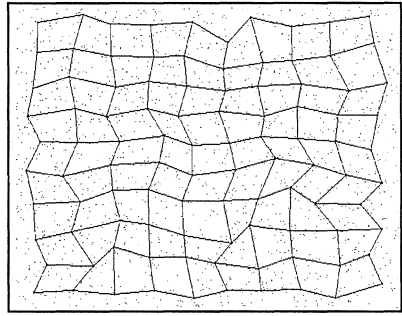


Figura 8.- Distribución uniforme, 88 iteraciones.

Lo que ocurre en la simulación, como puede apreciarse en las figuras 4, 5 y 6, es que, a medida que presentamos los patrones de entrada, la red los va aprendiendo, lo cual quiere decir que varía el valor de sus conexiones, siguiendo el algoritmo de la tabla 1. Este cambio de valor de las conexiones viene expresado, gráficamente, por una traslación de la neurona en cuestión. De esta manera, conseguiremos que, donde haya más densidad de puntos, exista también mayor densidad de neuronas. La conexión topológica de la capa de competición hace que la representación de las células sea en forma de rejilla, en donde las líneas significan la vecindad entre las células. Esto quiere decir que podremos comprobar que nuestra red está comportándose de forma correcta, sin más que

observar si la rejilla esta extendida, es decir si no se producen cruces entre las líneas que la componen. Esto es así debido a la influencia que cada célula tiene con sus vecinas en un grado proporcional al de su vecindad. Por lo tanto, esta influencia provoca que el mapa se desenvuelva en forma de rejilla. No hay que olvidar que, al principio, la red esta distribuida aleatoriamente por todo el espacio y, por tanto, se encuentra totalmente retorcida.

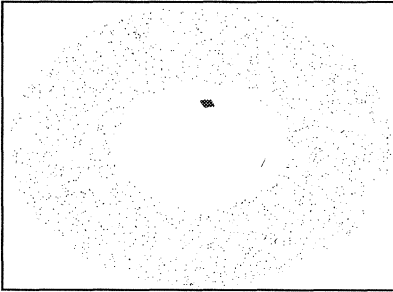


Figura 9.- Distribución toroidal, 1 iteración.

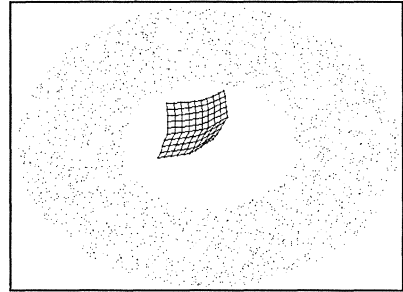


Figura 10.- Distribución toroidal, 11 iteraciones.

Se presentan, aquí, tres ejemplos con otras tantas distribuciones de puntos, todas ellas de 2000 puntos y con 100 neuronas en la capa competitiva. El primer ejemplo (Figura 4, 5 y 6), se trata de una distribución uniforme en tres de sus fases, después de la primera iteración en que todas las células están en un núcleo central, en la Figura 5, después de 11 iteraciones, se puede apreciar como la rejilla está totalmente desenredada, y va distribuyéndose de manera uniforme, al igual que los puntos. En la Figura 6, en cambio, se aprecia, después de 88 iteraciones, como existe una distribución prácticamente uniforme de la rejilla para que la rejilla fuese más perfecta se necesitarían un número de puntos y de células mayor.

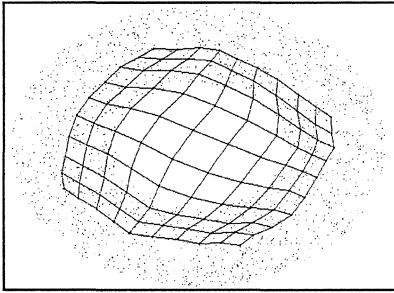


Figura 11.- Distribución toroidal, 20 iteraciones.

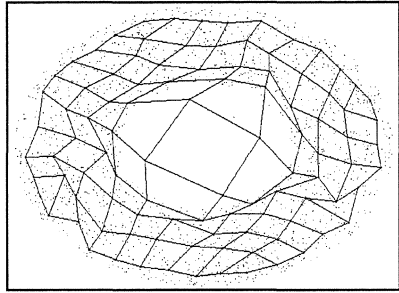


Figura 12.- Distribución toroidal, 88 iteraciones.

En el segundo ejemplo la distribución que se ha elegido es la de un toroide con 2000 puntos. La misma red del caso anterior produjo la salida que se recoge en las figuras 7 a la 9. En la Figura 7 es lo mismo que en la 4 del ejemplo anterior, y en la Figura 8 para 12 iteraciones vemos como, también, se ha desenrollado la rejilla pero ya no de una manera tan uniforme como antes. En cambio en la Figura 9 se puede ver, claramente, como la rejilla adquiere la forma del toroide sin que se produzcan por ello cruces en las líneas de la rejilla.

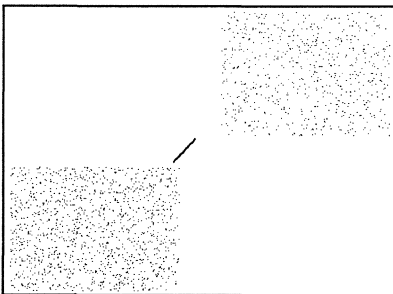


Figura 13.- Dos distribuciones uniformes, 1 iteración.

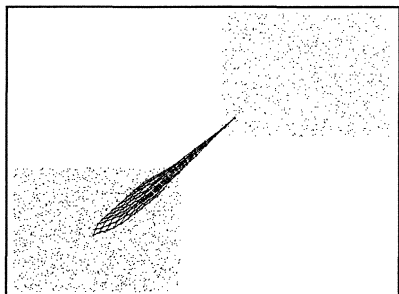


Figura 14.- Dos distribuciones uniformes, 12 iteraciones.

En el último ejemplo, Figura 10, 11 y 12, tenemos dos distribuciones uniformes, en la parte superior izquierda e inferior derecha respectivamente. Cuando llevamos ya varias iteraciones, Figura 11 con 18 iteraciones, se aprecia como la rejilla se estira formando una diagonal de izquierda a derecha y de arriba a abajo, en busca de las dos distribuciones de puntos; para, al final, Figura 12 con 88 iteraciones, comportarse como si de dos redes se tratará, una para cada distribución, sin perderse las relaciones topológicas entre los elementos de la red. Se ve, pues, claramente con estos tres ejemplos como una misma red de Kohonen puede clasificar, de forma automática y no supervisada, distintas distribuciones de puntos sin perder sus propiedades topológicas.

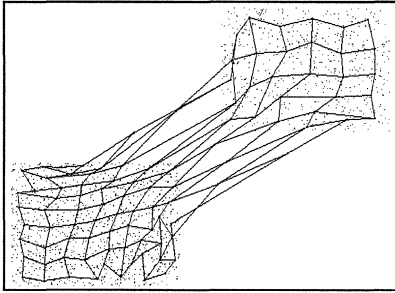


Figura 15.- Dos distribuciones uniformes, 88 iteraciones.

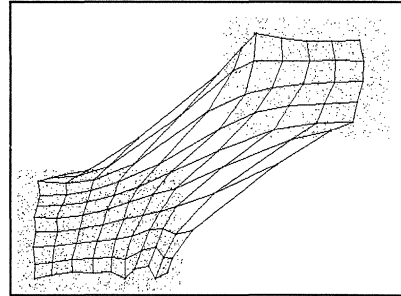


Figura 16.- Dos distribuciones uniformes, 25 iteraciones.

Se puede apreciar, también, como las neuronas que aparecían como vecinas al principio de la simulación, ahora, ocupan posiciones contiguas dentro de la distribución; es decir, neuronas conectadas topológicamente adquieren, mediante aprendizaje, características similares. De esta forma, podemos saber, sin más que seguir las conexiones topológicas de las células, que unidades comparten similares características, y cuales son, por el contrario, opuestas.

Para el caso del ejemplo esto puede parecer absolutamente trivial, pero cuando la dimensión de los datos aumenta considerablemente, el poder relacionar grupos de datos y poder agrupar puntos o disminuir la dimensión de los datos, es de gran importancia.

Otro ejemplo es el descrito por Durbin y Willshaw [DUR87], en el que utilizan un método numérico, que más tarde Angéniol [ANG88] modificaría aplicando una red de Kohonen para resolver el problema del viajante, en inglés TSP (traveling salesman problem). El TSP es un problema clásico en el mundo de la computación debido a su complejidad, y a que muchos de los problemas de complejidad similar o incluso menor, pero de más aplicabilidad, pasan por la solución al TSP.

El problema del viajante consiste en describir el recorrido que debería hacer un viajante de comercio, que deba visitar “n” ciudades, y regresar a la ciudad de la que partió, de tal forma que recorra la menor distancia posible. La solución es una lista ordenada de ciudades, que empiecen y acaben en la misma, y que recorran el resto una sola vez. Este problema está categorizado dentro de los “np-completos” y su complejidad es factorial a la dimensión del problema; es decir, si para resolver un TSP con 10 ciudades tardásemos 1 milésima de segundo, para uno de 20 sería del orden de 2 años si lo tratamos de resolver por búsqueda ciega.

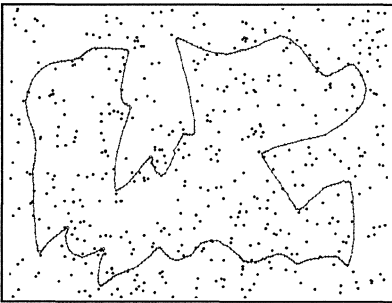


Figura 17.- Problema del TSP para 500 ciudades.

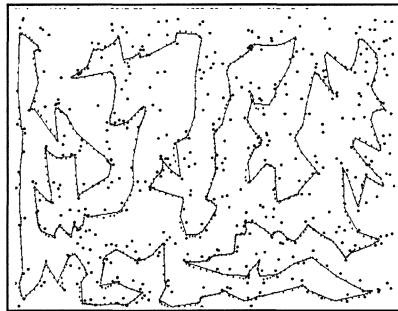


Figura 18.- TSP de 500 ciudades después de 9 iteraciones.

El método de resolución del TSP con Kohonen podría ser el siguiente: representamos las ciudades por puntos en un espacio bidimensional de la misma manera explicada en el apartado anterior. Creamos una red con sólo unos pocos nodos, digamos tres, y los unimos en una cuerda cerrada; es decir, cada célula tiene dos vecinas, una siguiente y una anterior, y esto para todas las células. La representación de las células de la capa competitiva sería la misma que se eligió para el apartado anterior. Aplicamos el método de Kohonen, con una única variante, y es que cada vez que una célula sea la ganadora de más de una ciudad, dividimos a la célula en dos y vemos cuál de las dos gana la competición, repitiéndose el algoritmo para esta nueva arquitectura de la red. Puede suceder que, una célula que antes se correspondía con una ciudad, ahora la haya perdido en favor de una ya existente o de otra que haya aparecido posteriormente. Estas células, si en un número de iteraciones no han ganado en ninguna competición, se eliminan de la red.

De esta forma, evitamos que existan más células que ciudades, y el algoritmo esté arrastrando lastres innecesarios que no se corresponden con ninguna ciudad. A la vez que se garantiza que, al final, tendremos exactamente el mismo número de células que de ciudades; es decir, que cada célula se corresponda con cada ciudad, y al final las coordenadas de la ciudad y de la célula sean las mismas. Esto es así debido a que el valor de la tasa de aprendizaje, o de olvido, va disminuyendo, así como la vecindad. De tal manera que, al final, cuando tengamos el mismo número de ciudades que de células el vecindario sea uno, y no exista arrastre, simplemente cada célula caerá gravitacionalmente hacia su ciudad correspondiente.

Lo que ocurre en la red es que, a medida que vamos presentando las ciudades, aquella célula que se encuentre más cercana a la ciudad, será la ganadora de la competición, y se acercará a la ciudad, arrastrando en su movimiento a las células vecinas. De tal modo que el camino se irá acercando cada vez más a las ciudades, hasta coincidir con todas ellas. Además, el camino que vendrá dado por el mapa topológico de las células de la capa competitiva, se irá adaptando de manera elástica a una solución en que la vecindad evita el que se produzcan cruces.

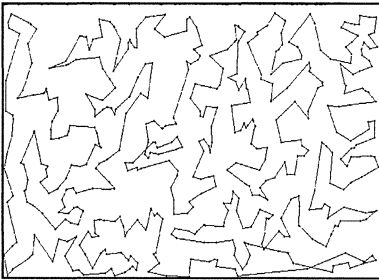


Figura 19.- TSP 500 ciudades, después de 18 iteraciones

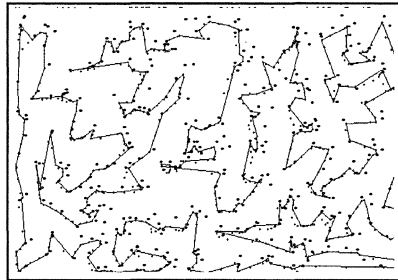


Figura 20.- Solución final del TSP, mediante Kohonen.

Como se aprecia, en las figuras 19 y 20, dos células, a pesar de estar muy próximas físicamente, si no son vecinas topológicamente, en este caso vecinas en el recorrido, la influencia que una tiene en la otra es escasa y evita así que esa ciudad pueda atraer a células que aún estando más próximas no lo sean en el recorrido, lo cual daría resultados incorrectos.

Este es un buen ejemplo de las características topológicas de una red, en las que esa topología guía de manera principal el comportamiento de la red y produce los resultados que, de otra forma, no se obtendrían. En este caso es una topología unidimensional que representa el hecho físico de ser la ciudad visitada en siguiente lugar.

En las figuras 13, 14 y 15, se pueden ver los resultados obtenidos para una dimensión de 500 ciudades, en tres fases de evolución de la simulación, al principio, con 1 sola iteración, Figura 13, en la que vemos que existe un número elevado de células pero es menor que el de ciudades; después de 9 iteraciones donde la goma está más estirada, y ya se atisba la solución que nos propondrá el método, y al final, después de 65 iteraciones, donde ya cada célula ha coincidido con una ciudad y nos da la solución final. Como puede apreciarse la solución es una muy buena solución, pero no tiene por que ser la óptima, el método no nos lo

garantiza, se trata de un buen método de búsqueda de solución aproximada al TSP, en un tiempo de varios minutos (para el caso del ejemplo).

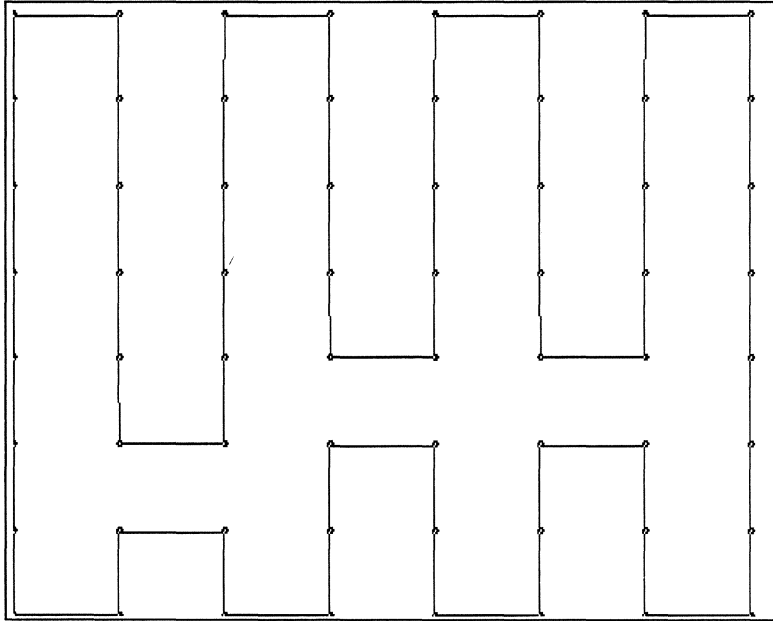


Figura 21.- Solución al problema del TSP con una distribución premeditada de 64 ciudades.

Por último, en la figura 21, vemos como para un caso de 64 ciudades premeditadamente distribuidas, el método nos da la solución óptima.

3.- Teoría de la Resonancia Adaptativa - ART

La Teoría de la Resonancia Adaptativa, nace de la necesidad de superar lo que Grossberg definió como el dilema de la Estabilidad-Plasticidad (Grossberg-1980, Grossberg-1982a, Grossberg-1982b), y que lo enuncia de la siguiente manera:

"Por un lado, el hecho de haberme puesto a vivir en Los Angeles no debe impedirme el poder encontrar, sin esfuerzo, la casa de mis padres en Boston, cuando en el futuro vaya a visitarles. Por otro, el hecho de haber crecido en Boston no debe impedirme el poder aprender lugares y direcciones de Los Angeles al irme a vivir allí"

Es decir, se trata de diseñar un sistema que reaccione de forma "plástica" a nuevos estímulos, pero de manera "estable" a aquellos que sean irrelevantes. Tenemos, también, la necesidad de poder conmutar entre estos dos estados plástico y estable, cuando sea necesario, para evitar la degradación de lo ya aprendido.

Esto requiere que el sistema tenga incorporada alguna estimación, que le permita decidir cuando un estímulo es irrelevante y cuando no, ya que queremos realizar todo ésto de una manera no supervisada.

Grossberg hace una distinción, que luego ha sido utilizada por varios autores, entre dos tipos de memoria. La memoria a corto plazo y la memoria a largo plazo:

- Memoria a corto plazo (STM -Short Term Memory).- Se refiere a la capacidad biológica de recordar algo que acaba de ocurrir en un instante de tiempo corto, independientemente del número de veces que haya ocurrido. Se identifica en términos de RNA, con los valores de activación de las neuronas.
- Memoria a largo plazo (LTM -Long Term Memory).- Se refiere a la capacidad de recordar cosas que ocurrieron en un instante de tiempo lejano, con tal de que hayan ocurrido el suficiente número de veces. Computacionalmente, podría ser el equivalente a los pesos de las conexiones.

Para solucionar el problema de la estabilidad-plasticidad, Grossberg introduce en su modelo dos subsistemas, y un término de control:

- Subsistema Atencional.- Es, básicamente, una red de aprendizaje competitivo, que se encarga de reaccionar ante estímulos nuevos y de aprenderlos.

- Subsistema Orientador.- Se encarga de distinguir entre que estímulos son relevantes para el sistema y cuales no.
- Gain Control.- Se trata de un sistema que lleva el control de los módulos que van a actuar y gobierna las señales entre los distintos módulos.

En la Figura 22 se puede ver la arquitectura general de un sistema ART: La capa F1 es, como en el aprendizaje competitivo, la capa de entrada, su dimensión será la misma que la de los estímulos y se encargará de propagar la señal de entrada a la capa superior F2. Esta última es, básicamente, una capa competitiva donde existen los mismos tipos de conexiones que en el aprendizaje competitivo; es decir, cada célula inhibe a todas las demás con una fuerza fija e igual para todas, y recibe una señal excitatoria, de si misma, también con la misma fuerza.

Uno de los principales problemas que teníamos, con el aprendizaje competitivo, era el de la inestabilidad. Grossberg (1975) ya indicaba que este problema podría ser solucionado introduciendo en el sistema mecanismos de aprendizaje recurrentes y de correlación entre los estímulos aprendidos y los que se presenten al sistema.

Por tanto, en el ART, añadimos a las conexiones existentes entre la capa F1 y la F2, llamadas conexiones “botton-up”, otras nuevas entre la capa F2 y la F1, llamadas conexiones “top-down”, las cuales permiten un proceso de aprendizaje de valores esperados o predicciones con las que poder comparar los estímulos de entrada y que pueden producir reacciones orientadoras en situaciones de desigualdad.

Usando estos mecanismos, un sistema ART puede generar códigos de reconocimientos de forma adaptativa y sin la presencia de ningún profesor externo. Puede, también, de forma adaptativa, conmutar entre un estado estable y uno plástico de forma que sea capaz de aprender eventos significativamente nuevos sin perder la estabilidad ante aquellos que sean irrelevantes. Es capaz, pues, de distinguir entre eventos familiares y no familiares y esperados y no esperados.

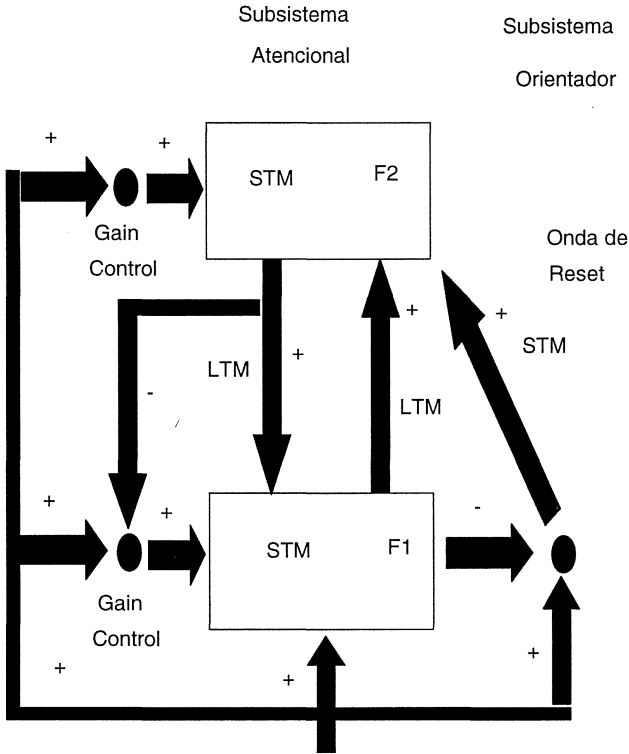


Figura 22.- Arquitectura de un sistema ART.

Los eventos familiares se procesan en el subsistema atencional de forma parecida a como se realiza en el modelo de aprendizaje competitivo. Este subsistema también es capaz de aprender las predicciones “top-down” que ayudarán a estabilizar los códigos “bottom-up” aprendidos de los eventos familiares.

Sin embargo, el subsistema atencional, por sí sólo, es incapaz de mantener al mismo tiempo representaciones estables de categorías

familiares y crear nuevas categorías para estímulos no familiares que puedan ir apareciendo. Para realizar esta labor, se introduce el subsistema orientador que inhibe al subsistema atencional cuando se recibe un estímulo no familiar.

Las interacciones que se producen entre estos dos subsistemas ayudan a decidir cuando un nuevo estímulo es familiar y está bien representado por uno de los códigos de reconocimiento, o si no es familiar y necesita de un nuevo código de reconocimiento.

Los sistemas ART tienen cuatro características esenciales que los distinguen de otros modelos de detección de características:

1.- Unidades computacionales autoescalables.- El contexto de un estímulo puede afectar a su modo de reconocimiento, de tal modo que señales que aparezcan en un estímulo pueden ser tratadas como ruido, pero pueden ser parte del propio estímulo en otro contexto. Vemos como partes de la señal de entrada, que son tratadas como ruido, cuando afectan a un sistema en un cierto grado de autoorganización, pueden ser tratadas como señal cuando perturban al mismo sistema en un diferente grado de autoorganización. El modelo, de forma automática, auto-escala sus unidades computacionales para incluir al contexto y ser sensible a definiciones de señal y de ruido dependientes del aprendizaje.

2.- Acceso a memoria autoajutable.- No existe ningún método de búsqueda de las señales en el sistema; por el contrario, el modelo es capaz de realizar una búsqueda paralela que, de forma adaptativa, actualice el orden de búsqueda para mantener la eficiencia a medida que los códigos de reconocimiento se hacen más complejos debido al aprendizaje.

3.- Acceso directo a los códigos de reconocimiento.- Una de las más impresionantes características del comportamiento humano es lo rápidamente que pueden reconocer objetos familiares. En ART, a medida que el aprendizaje se convierte en auto-consistente y predictivamente exitoso, el proceso de búsqueda es más directo. No importa como sea de grande y de complejo el código aprendido, los estímulos familiares acabarán por disparar de forma directa el código que los recoge, sin búsqueda, así como aquellos estímulos no familiares cuyo parecido sea suficientemente alto.

4.- El entorno como profesor.- En un sistema ART, si a un error de reconocimiento le acompañamos de una penalización en el entorno, el sistema se hará mas vigilante, que se podría traducir por un cambio en el estado atencional del sistema, lo cual incrementará la sensibilidad entre las diferencias de los estímulos y los valores predictivos.

Veamos ahora las reacciones que se producirían en la red a la llegada de un estímulo cualesquiera I. El estímulo I se transforma en un patrón S de activaciones a través de los nodos, o de forma más abstracta "directores de características", de F1. Esta señal es propagada a través de las conexiones botton-up mediante la suma ponderada de la señal S por los valores de los pesos de las conexiones botton-up, hasta transformarse en una nueva señal T. A la transformación de S en T se la llama "filtro adaptativo". Esta señal de entrada T, se transforma rápidamente en otra señal Y en la capa F2 a través de las conexiones excitatorias e inhibitorias existentes en dicha capa.

Tan pronto como se produce la transformación de S en Y la capa F2 envía esta señal hacia abajo a través de las conexiones top-down. De la misma forma como ocurría en la transformación de S en T, ahora obtendremos una nueva señal V que será el resultado de realizar la suma ponderada de la señal Y por los valores de los pesos de las conexiones top-down. La transformación de Y en V será, por tanto, un nuevo filtro adaptativo. A la señal V se la llamará valor predictivo.

Ahora F1 se verá perturbada por dos fuentes: el estímulo de entrada I y el valor predictivo V. Esto hará que los valores de activación de la célula F1 varíen de los que tenía al principio de recibir el estímulo I, esta variación nos medirá, de alguna manera, el grado de relación entre el estímulo I y el valor predictivo V. Esto se utilizará para resolver el dilema de la estabilidad-plasticidad.

La capa F1 está relacionada con el subsistema orientador a través de una conexión inhibitoria que computa todos los valores de activación de la células de F1. Al recibir un estímulo como este, se introduce a su vez en el "gain control" que está conectado, a su vez, con la capa F1, en F1 se produce una comparación del estímulo consigo mismo, lo cual nos da un valor alto de la señal inhibitoria hacia el subsistema orientador, el cual quedará inhibido.

Más adelante, cuando F1 reciba la señal predictiva V el “gain control” quedará inhibido por las señales top-down que recibe. Con lo cual, en F1, se producirá la comparación entre el estímulo I y el valor predictivo V, lo cual reducirá la inhibición de la capa F1 hacia el subsistema orientador.

Si dicha señal inhibitoria baja por debajo de un cierto umbral, llamado factor de vigilancia, querrá decir que el valor predictivo V era erróneo, y que no debe ser asumido por aquel código de reconocimiento. En este caso, el subsistema orientador envía una señal a la capa F2 del subsistema atencional comunicándole el error, lo cual produce la temporal desaparición de la capa F2 del código de reconocimiento elegido.

Una vez producido esto se repite todo el proceso, una y otra vez, hasta que uno de los valores predictivos sea correcto, se dice entonces que sistema entra en resonancia. En el proceso de resonancia puede ocurrir que acabemos por eliminar todos los códigos de reconocimiento de la capa F2, y por tanto no podamos continuar con la búsqueda de nuevos códigos. El sistema, en este caso, crea un nuevo código para el estímulo, de tal forma que la correlación entre dicho estímulo y el valor predictivo que origine sea absoluto. Diremos que se ha producido una nueva situación relevante que ha dado lugar a la creación de un nuevo código de reconocimiento para estímulos con características similares al recibido.

Esto soluciona el problema existente en otros modelos no supervisados, (Kohonen-1982), y Competitive Learning (Rumelhart y McClelland-1981), de la necesidad de prefijar de antemano el número de códigos de reconocimiento, ya que en ART la red se inicia sin elementos en la capa F2, y van siendo creados a medida que van siendo necesarios; es decir, es el conjunto de los datos de entrada el que nos determinará, a partir de sus propias características, el número de categorías necesarias.

El aprendizaje se produce una vez que un valor predictivo ha sido catalogado como válido, tanto si ha sido después de haberse producido una nuevo código de reconocimiento, como si no. Este aprendizaje afecta tanto a las conexiones “bottom-up”, cuya reglas son parecidas a las del aprendizaje competitivo, y que se detallan a continuación, como a las “top-down”, que se encargarán de modificar los valores predictivos asociados a un conjunto de estímulos, a medida que estos vayan siendo catalogados en el mismo código de reconocimiento.

Las ecuaciones del sistema son las siguientes:

$$\begin{aligned}
 0 \leq i \leq N - 1 \quad N &= \text{numero de nodos de F1} \\
 0 \leq j \leq M - 1 \quad M &= \text{numero de nodos de F2} \\
 0 \leq p \leq 1
 \end{aligned}$$

La inicialización del sistema se realiza siguiendo las siguientes ecuaciones:

$$t_{ij}(0) = 1, \quad b_{ij}(0) = \frac{1}{1 + N}$$

En estas ecuaciones b_{ij} es el peso de la conexión “Bottom-Up”, y t_{ij} es el peso de la conexión “Top-Down”, entre el nodo de entrada i y el nodo de salida j en el tiempo t . Estos pesos definen el ejemplar especificado por el nodo de salida j . El factor de vigilancia es p .

Para calcular los valores en la capa F2 se hará:

$$\mu_j = \sum_{i=0}^{N-1} b_{ij}(t) x_i, \quad 0 \leq j \leq M - 1$$

Siendo μ_j la salida de la célula j de la capa F2, y x_i el elemento i -ésimo de la entrada, que para el modelo ART 1 toma valores binarios (1 ó 0).

Para seleccionar el mejor ejemplar:

$$\mu_j^* = \max_j(\mu_j)$$

Utilizando, para ello, las fórmulas de la interacción lateral, del apartado anterior.

$$\|X\| = \sum_{i=0}^{N-1} x_i = O^{N-1} x_i$$

$$\|TX\| = \sum_{i=0}^{N-1} t_{ij}^* x_i$$

En el Subsistema orientador tendremos: Si

$$\frac{\|TX\|}{\|X\|} > p$$

Entonces, se dispara el Subsistema Orientador.

Por último, nos queda adaptar las conexiones “Top-Down” y “Bottom-Up”:

$$t_{ij}^*(t+1) = t_{ij}(t)x_i$$

$$b_{ij}^*(t+1) = \frac{t_{ij}^*(t)x_i}{0,5 + \sum_{i=0}^{N-1} t_{ij}^*(t)x_i}$$

$$i=0, 1, \dots, N-1$$

Bibliografía

- Angéniol, B., De la Croix Vaubois, G. & Le Texier Y. (1988). Self-Organizing Feature Maps and the Travelling Salesman Problem. *Neural Networks* 1, 289-293.
- Barlow, H.B. (1989). Unsupervised Learning of Constrained Nonlinear Transformations. *Computer*, Junio 1989, 67-76.
- Carpenter A. & Grossberg S. (1987). ART 2: Stable self-organization of Pattern Recognition Code for Analog Input Patterns. *Applied Optics*, 26, 4919-4930.
- Carpenter A., Grossberg S. & Reynolds J. (1991a). ARTMAP: Supervised real-time learning and clasification of nonstationary data by a self-organizing Neural Network. *Neural Networks*, Vol 4, 565-588.
- Carpenter A., Grossberg S. & Rosen D. (1991b). Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System. *Neural Networks*, Vol 4, 759-771.
- Durbin, R. & D. Willshaw. 1987. An Analogue Approach to the Traveling Salesman Problem Using an Elastic Net Method". *Nature* 326, 688-691.
- Grossberg, S. (1980). How Does the Brain Build a Cognitive Code?. *Psychological Review* 87, 1-51.

-
- Grossberg, S. (1982a). *Studies of mind and brain: Neural principles of learning, perception, development, cognition, and motor control*. Boston: Reidel Press.
 - Grossberg, S. (1982b). Processing of expected and unexpected events during conditioning and attention: A psychophysiological theory. *Psychological Review*, 89, 529-572.
 - Grossberg, S. (1987). Competitive Learning: From Interactive Activation to Adaptive Resonance. *Cognitive Science* 11, 23-63.
 - Hebb, D. (1949). *The Organization of Behavior*. New York. John Wiley & Sons.
 - Kohonen, T. (1982). Self-Organized Formation of Topologically Correct Feature Maps. *Biological Cybernetics* 43, 59-69.
 - McClelland, J.L. y Rumelhart, D.E. (1981). An interactive activation model of context effects in letter perception, Part I: An account of basic findings. *Psychological Review*, 88, 375-407.
 - Simpson, P. (1990). *Artificial Neural Systems, Foundations, Paradigms, Applications, and Implementations*. Pergamon Press. New York.