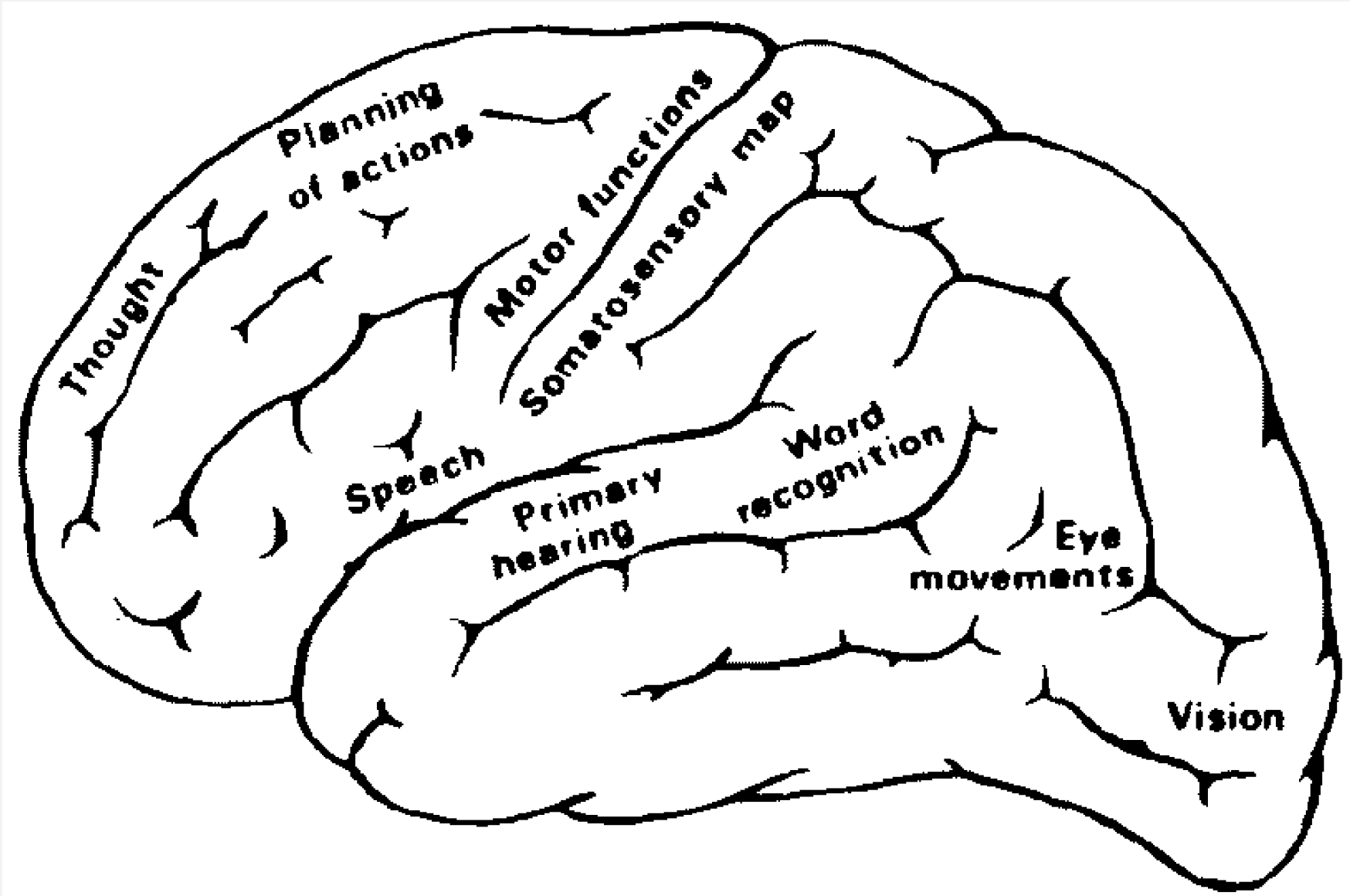
REDES AUTO-ORGANIZADOS – SOM (SELF-ORGANIZING MAP).

Son redes que realizan el proceso de aprendizaje sin la necesidad de contar con valores deseados de salida, es decir, que funcionan en forma no supervisada.

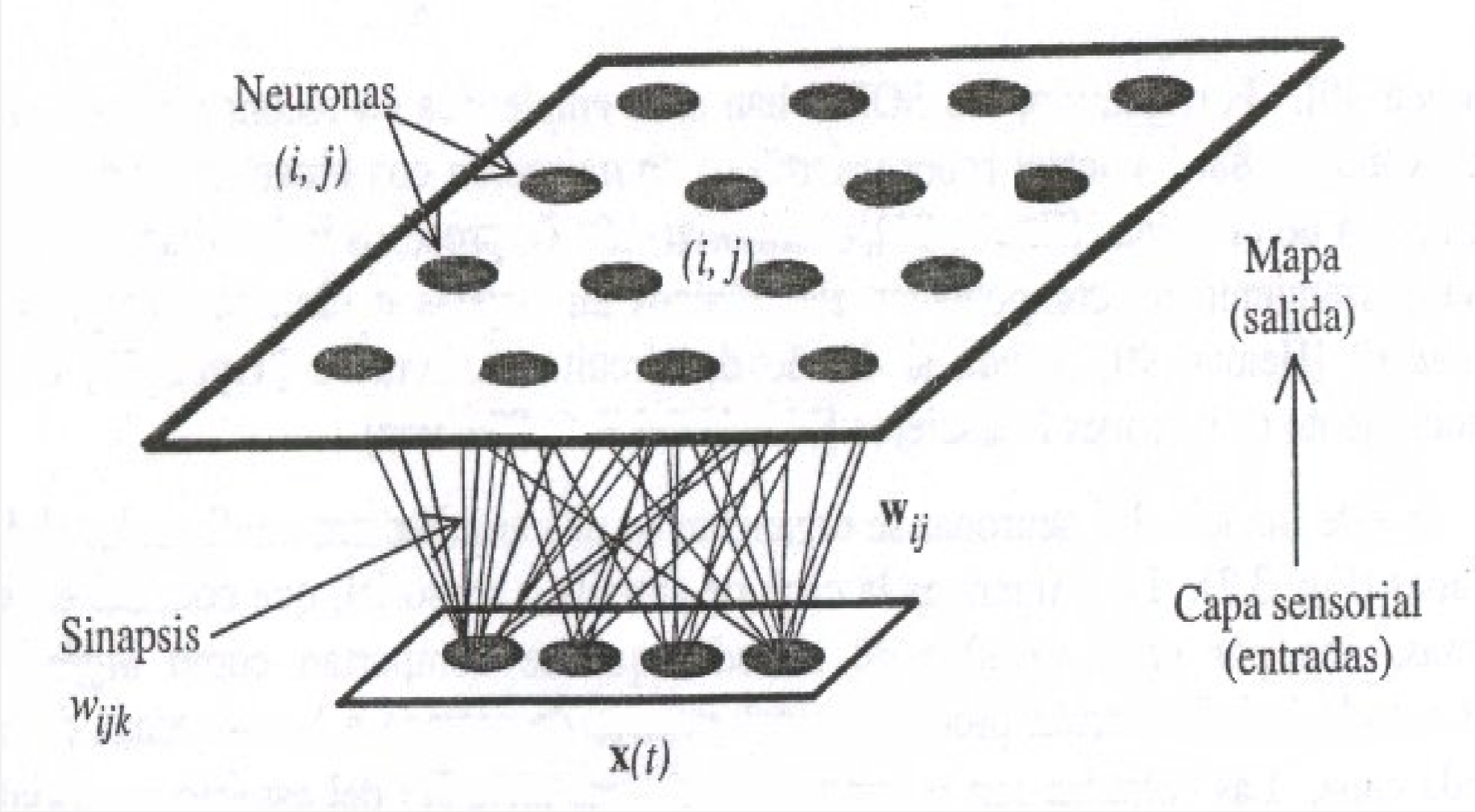

Thought
Planning of actions
Motor functions
Somatosensory map
Speech
Primary hearing
Word recognition
Eye movements
Vision

Por lo tanto este tipo de redes deben ser capaces de encontrar las características principales de las entradas o reconocer patrones sin la propiedad de *"maestro"* que generan los Valores Deseados de Salida.

En este caso la modificación de los pesos de una neurona se condiciona a un concepto de Vecindad, que involucra a neuronas *"vecinas"* dentro de la estructura de la red. Este concepto está basado en estructuras neurobiológicas.

Un proceso de aprendizaje en una estructura auto-organizada, consiste en una constante modificación de los pesos de las redes, en respuesta a señales patrones de entrada presentados, de acuerdo a reglas preestablecidas hasta lograr una configuración final.

Arquitectura de una red SOM



Aprendizaje Competitivo.

En este tipo de redes las salidas de las neuronas compiten entre sí con el objetivo de lograr su activación, con la condición de que, teóricamente, solamente una (o muy pocas) pueden obtener ese objetivo, para cada tipo de patrón de entrada.

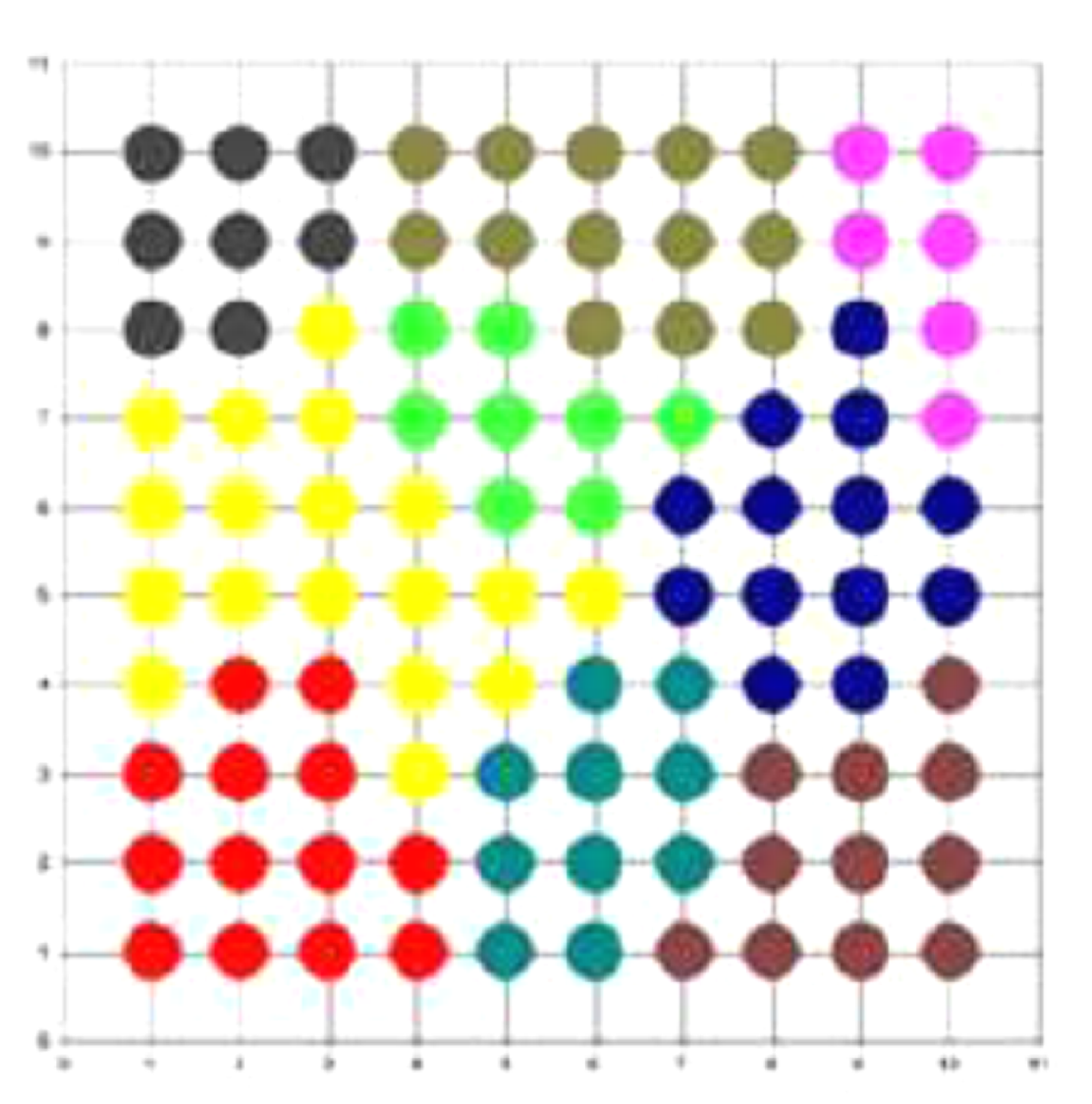
En este tipo de redes, *self-organizing map (SOM)*, las neuronas se distribuyen sobre los nodos de un reticulado de una o dos dimensiones.

Estas neuronas se relacionan selectivamente con las distintas entradas presentadas a la red durante el proceso de aprendizaje competitivo, a fin de lograr un ordenamiento entre las distintas neuronas, de manera de lograr zonas de neuronas activas, asociadas a las diferentes características relevantes de las entradas.

En consecuencia las redes *"self-organizing map"* se caracterizan por la formación de un mapa topológico de los patrones, donde la posición de las neuronas activas (ej. coordenadas en el reticulado), se corresponden con características relevantes de los patrones.

El desarrollo de este tipo de redes se basa en características de funcionamiento del cerebro humano.

Mapa Topológico de Patrones.



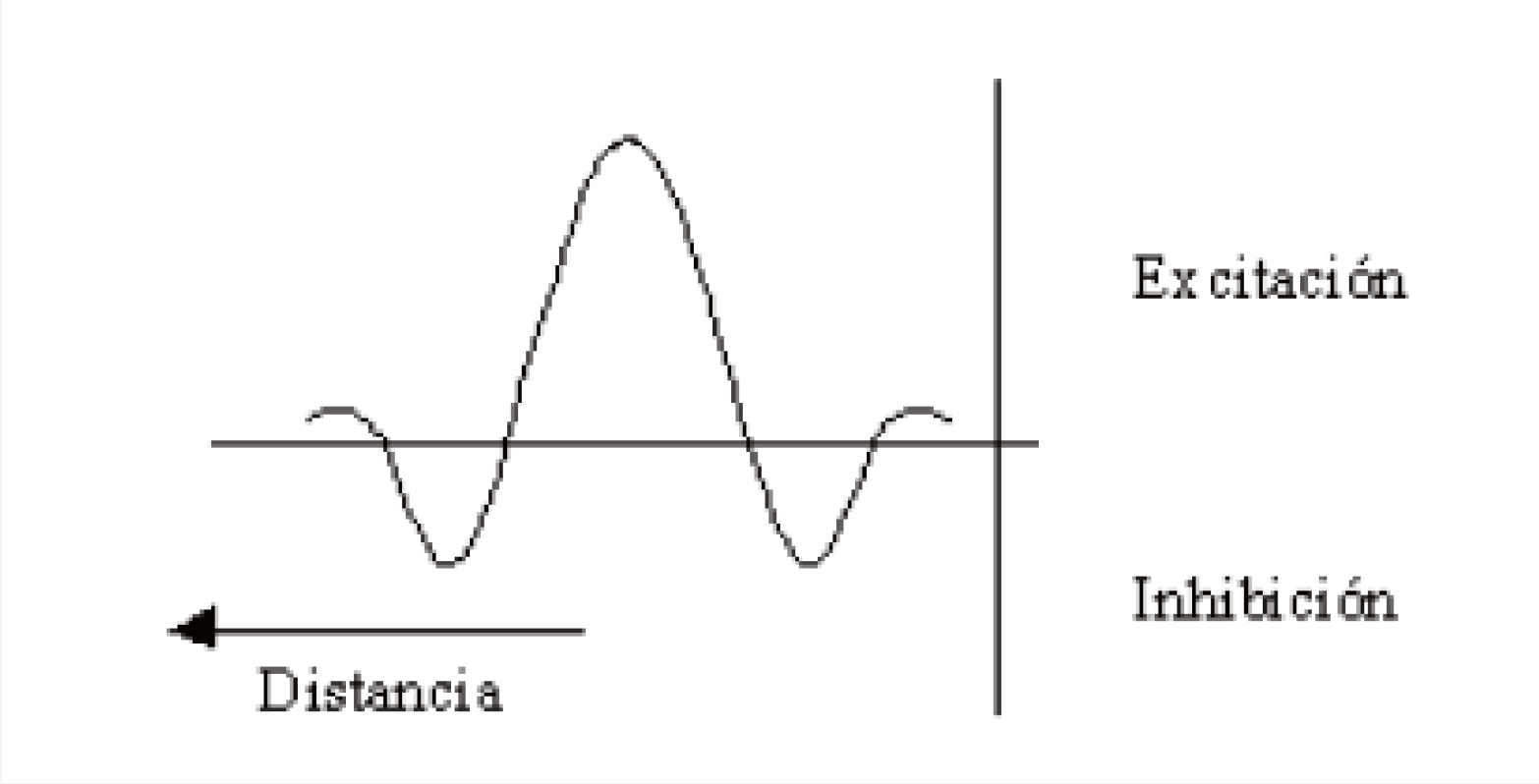
Realimentación lateral en las neuronas biológicas.

La realimentación lateral, existente en las neuronas biológicas, se define como una realimentación entre neuronas, que depende de la distancia lateral entre las mismas.

Este tipo de realimentación tiene las siguientes características:

- ✓ una primera zona de excitación centrada en la neurona en consideración.
- ✓ una segunda zona de inhibición, contigua a la anterior.
- ✓ una tercera zona de muy baja excitación (generalmente no considerada).

Forma de Excitación.



Con esta realimentación lateral se logra una concentración de las zonas activas, denominadas burbujas de actividad *(activity bubbles)*, cuya localización está determinada por las características de las entradas.

La salida de una neurona correspondiente a este tipo de comportamiento viene dada por la siguiente expresión:

$$y_j = \varphi\left(I_j + \sum_{k=-K}^{K} c_{jk} y_{j+k}\right), \qquad j = 1,2, \ldots \ldots \ldots N$$

Donde:

φ es la función de transferencia.

$I_j = \sum_{i=1}^{p} w_{ji} x_i$ es la excitación externa.

$\sum_{k=-K}^{K} c_{jk} y_{j+k}$ es la excitación lateral, siendo los c_{jk} los respectivos pesos.

La solución en forma iterativa es:

$$y_j(n+1) = \varphi\left(I_j + \beta \sum_{k=-K}^{K} c_{jk}\, y_{j+k}(n)\right), \qquad j = 1,2,\ldots\ldots\ldots N$$

Con β como factor de realimentación.

Debido al valor limitante de la función de transferencia y del valor de β, para $n \longrightarrow \infty$ los valores de y_j tienden a estabilizarse en un determinado valor y a concentrarse dentro de una región espacialmente limitada *(activity bubble)*.

Esta región está centrada en un punto donde la respuesta inicial $y_j(n)$ es máxima, debido únicamente a la excitación externa I_j

Como consecuencia de este comportamiento se pueden obtener las siguientes conclusiones:

1.- Se puede definir la salida de una neurona como:

$$y_j = \begin{cases} a & \text{para las neuronas dentro de la burbuja} \\ 0 & \text{para el resto} \end{cases}$$

2.- Es posible utilizar el anterior comportamiento para implementar un *"shortcut"* computacional, a fin de emular el efecto de la realimentación lateral, dado que en la práctica es imposible calcularla.

Es decir que es posible eliminar el concepto de la realimentación lateral e introducir el concepto equivalente de Región de Vecindad.

Algoritmo "Self- Organizing Maps"- SOM

El principal objetivo del algoritmo SOM es transformar señales patrones de entrada de una dimensión arbitraria, en una representación discreta de dimensión uno o dos, realizando esta transformación en forma adaptativa y ordenada topológicamente.

Los patrones son presentados a la red de a uno por vez, cada presentación provoca que un grupo localizado de neuronas en la red sean activadas.

Los componentes principales que intervienen en el algoritmo son:

- ✓ un reticulado de neuronas (máxima dimensión: 2).

- ✓ un mecanismo que compare determinadas funciones discriminantes de las entradas y determine las neuronas que mejor verifiquen estas funciones, definiendo de tal forma las neuronas ganadoras.

- ✓ un proceso adaptativo que permite a las neuronas activadas, mejorar la función que las relaciona a las entradas.

Desarrollo del algoritmo

Sea x un vector patrón de entrada:

$$x = [x_1, x_2, \ldots\ldots\ldots x_p]$$

El vector de pesos de la neurona j viene dado por:

$$w_j = [w_{j1}, w_{j2}, \ldots\ldots\ldots w_{jp}] \qquad j = 1,2 \ldots\ldots N$$

Teóricamente, para determinar el mejor *"matching"* de las neuronas con el vector de entrada, se computa el producto:

$$x \cdot w_j^T \ , \qquad j = 1,2 \ldots \ldots . . N$$

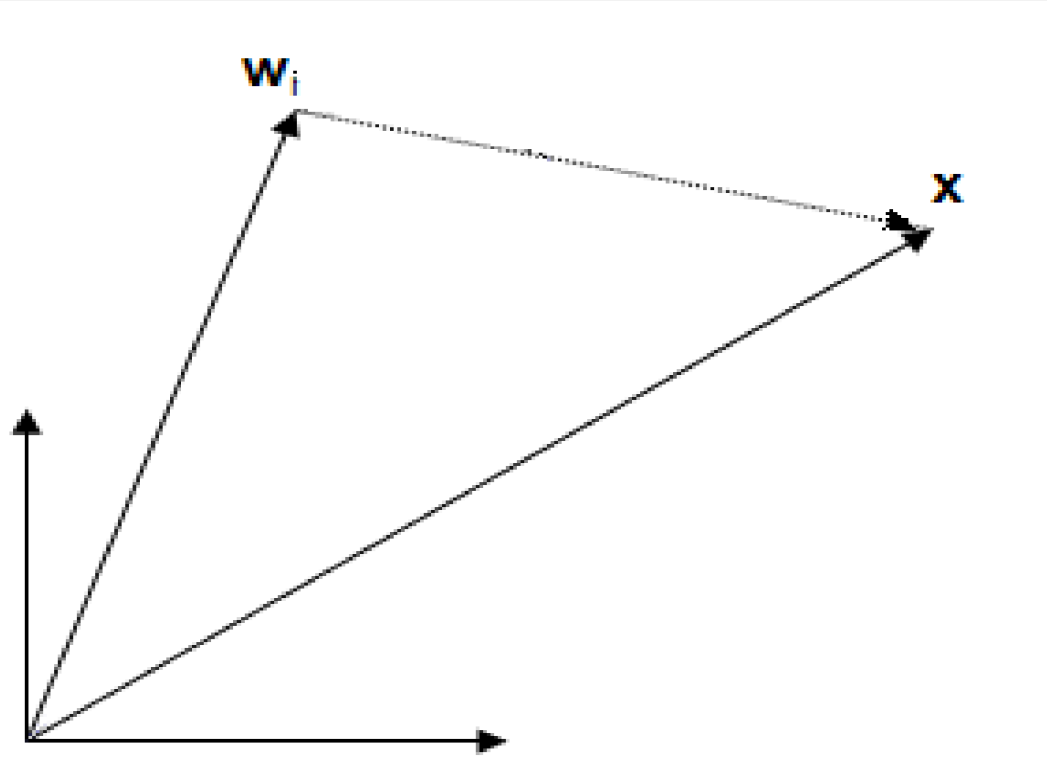
Y luego se selecciona el mayor.

De esta manera se determina la neurona con mayor excitación externa y por lo tanto la localización de la neurona ganadora.

Sin embargo en la práctica es conveniente utilizar un criterio de *"matching"* que permita normalizar los vectores de pesos.

El criterio utilizado en este caso es la distancia Euclídea entre los vectores, buscando determinar la distancia mínima:

$$D_i = |x - w_i|$$

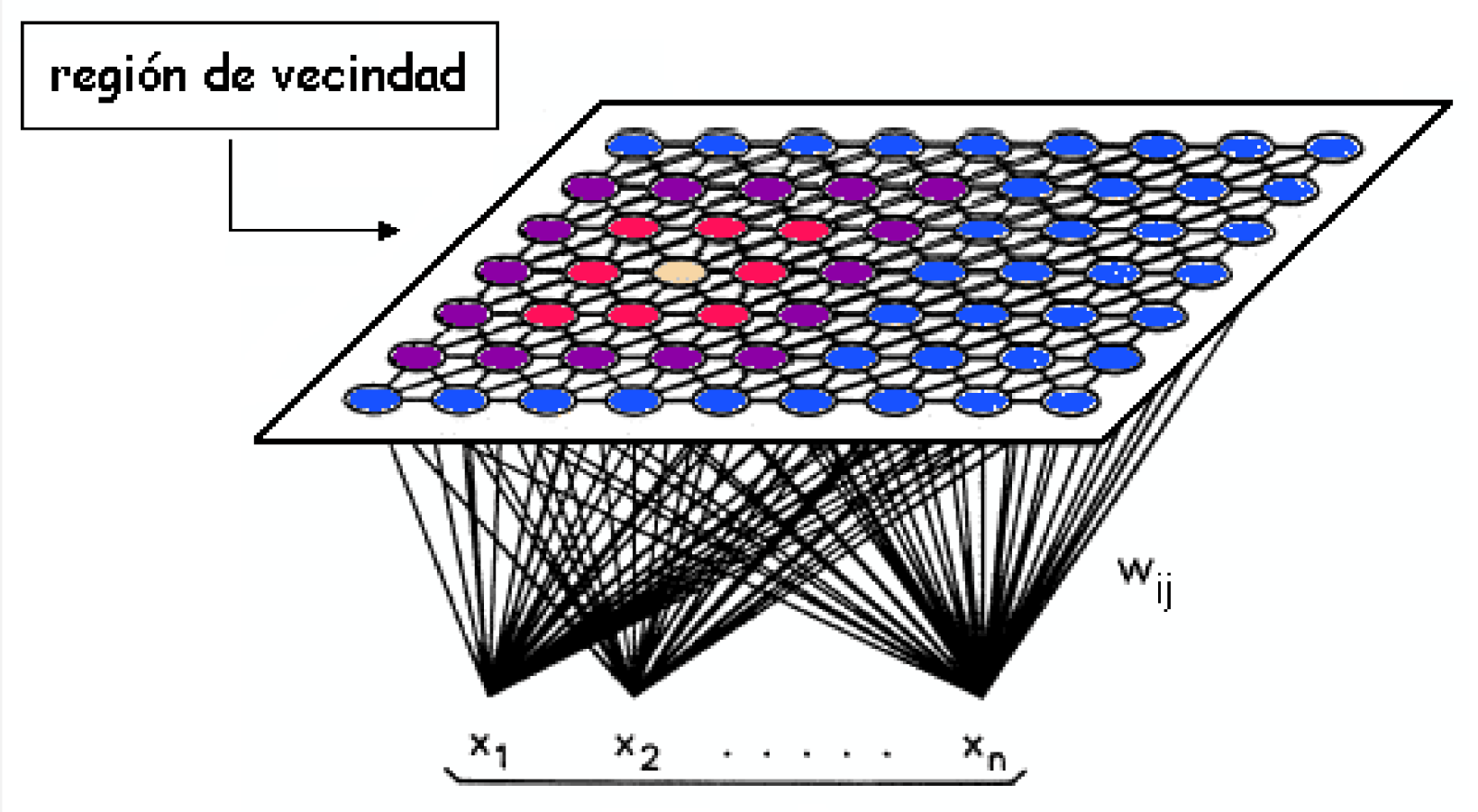


Determinada la neurona con la mínima distancia, ésta representa al patrón de entrada presentado, de esta manera datos de entrada que pueden ser de una dimensión mayor, son representados por un vector de dos dimensiones (máximo).

Dependiendo de la aplicación, la respuesta de la red puede ser:

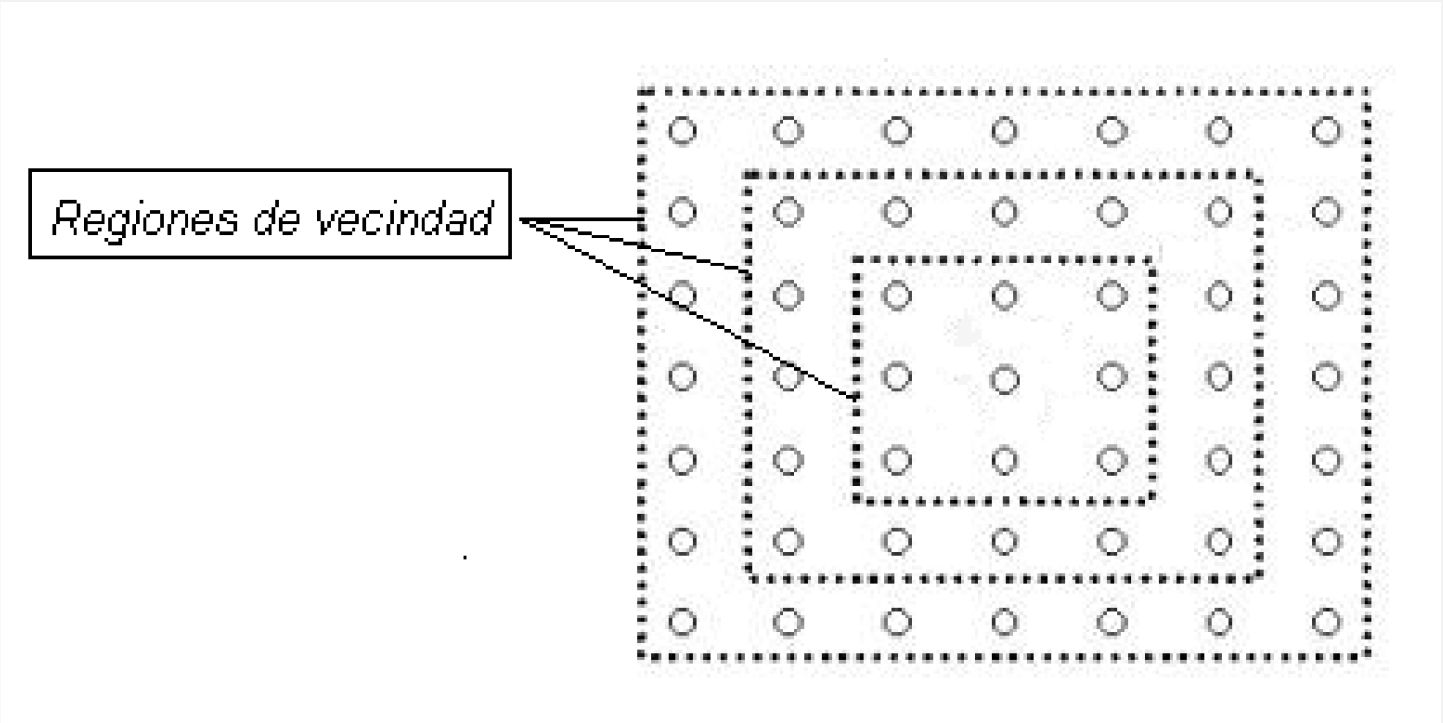
- ✓ la posición de la neurona ganadora en el reticulado.
- ✓ el vector de pesos de la neurona ganadora.

Para determinar la región de neuronas vecinas a la ganadora, que conjuntamente representan al patrón de entrada, se utiliza la *Región de Vecindad* $\Lambda_{i(x)}(n)$, que determina la vecindad de la neurona ganadora.



Se adopta un valor grande de la función $\Lambda_{i(x)}(n)$ al comenzar el proceso iterativo, para luego hacerlo disminuir con el paso de las iteraciones. Siempre centrada en las diferentes neuronas ganadoras.

Evolución de la región de vecindad:



Durante el entrenamiento la neurona con la mínima distancia, en conjunto con las neuronas vecinas, ajustan sus pesos con el objetivo de acercarse aún más a los valores de las entradas.

Para implementar este proceso de ajuste se utiliza el siguiente proceso de aprendizaje de *Hebb* modificado:

$$\frac{dw_j}{dt} = \eta\, y_j\, x - g(y_j)\, w_j$$

Donde:

η es parámetro de aprendizaje del algoritmo.

$-g(y_j)\,w_j$ es el *"forgetting term"* utilizado para evitar la saturación de los pesos.

Y:

w_j es el vector de pesos de la neurona j

$g(y_j)$ es una función escalar de la salida y_j

Con la única condición:

$g(y_j) = 0$ para $y_j = 0$ y para todo j .

En forma similar a la formación de la burbuja de actividad, la respuesta de la neurona j tendrá valores de saturación alta o baja, en función de que se encuentre dentro o fuera de la Región de Vecindad, por lo tanto:

$$y_j = \begin{cases} 1 & \text{para las neuronas dentro de la Región de Vecindad} \\ 0 & \text{para el resto} \end{cases}$$

De la misma forma se puede definir:

$$g(y_j) = \begin{cases} 0 & \text{para neuronas inactivas} \\ \alpha & \text{para neuronas activas} \end{cases}$$

Donde α es una constante positiva.

En consecuencia se puede simplificar la ecuación de modificación de los pesos:

$$\frac{dw_j}{dt} = \begin{cases} \eta\, x - \alpha\, w_j & \text{para neuronas dentro de } \Lambda_{i(x)}(n) \\ 0 & \text{para neuronas fuera de } \Lambda_{i(x)}(n) \end{cases}$$

Haciendo $\alpha = \eta$ resulta:

$$\frac{dw_j}{dt} = \begin{cases} \eta\left(x - w_j\right) & \text{para neuronas dentro de } \Lambda_{i(x)}(n) \\ 0 & \text{para neuronas fuera de } \Lambda_{i(x)}(n) \end{cases}$$

Finalmente el proceso de adaptación de los pesos es:

$$w_j(n+1) = \begin{cases} w_j(n) + \eta(n)\left(x - w_j(n)\right) & j \; \varepsilon \; \Lambda_{i(x)}(n) \\ \\ w_j(n) & \text{resto} \end{cases}$$

Para el parámetro de aprendizaje $\eta(n)$ se recomienda comenzar (primeras 1000 iteraciones) con un valor mayor (siempre menor a la unidad), para luego reducirlo . La forma de variación de $\eta(n)$ no es crítica.

La primera fase del proceso de adaptación se denomina fase de ordenamiento y es donde se produce el verdadero ordenamiento de los vectores de pesos $w_j(n)$.

La segunda fase del proceso denominada fase de convergencia, más larga que la anterior, produce un ajuste fino de los pesos, donde el valor de $\eta(n)$ puede tomar valores muy bajos (0.01).

En cuanto a la Región de Vecindad $\Lambda_{i(x)}(n)$, puede tomar diferentes formas: cuadrada, romboidal o gaussiana.

Su valor comienza generando una zona que puede contener todas las neuronas de la red, para luego disminuir su radio en función del tiempo. Esta disminución se puede implementar en forma lineal durante las primeras 1000 iteraciones.

Durante la fase de convergencia $\Lambda_{i(x)}(n)$ debe contener solamente las neuronas vecinas a la neurona ganadora, para concluir el entrenamiento incluyendo solamente a esta última.

Resumen del algoritmo.

Lo esencial del algoritmo es permitir implementar en forma relativamente sencilla, consideraciones más complejas como la regla modificada de *Hebb* y la realimentación lateral.

Consta de las siguientes etapas:


Inicialización.
Presentación de los patrones de entrada.
Determinación de la neurona "ganadora".
Entrenamiento.

Etapas.

1.- Inicialización:
Selección de los valores iniciales de los vectores de los pesos, con la única condición que $w_j(0)$ sean todos diferentes. Generalmente se implementa en forma aleatoria y con valores de pesos pequeños.

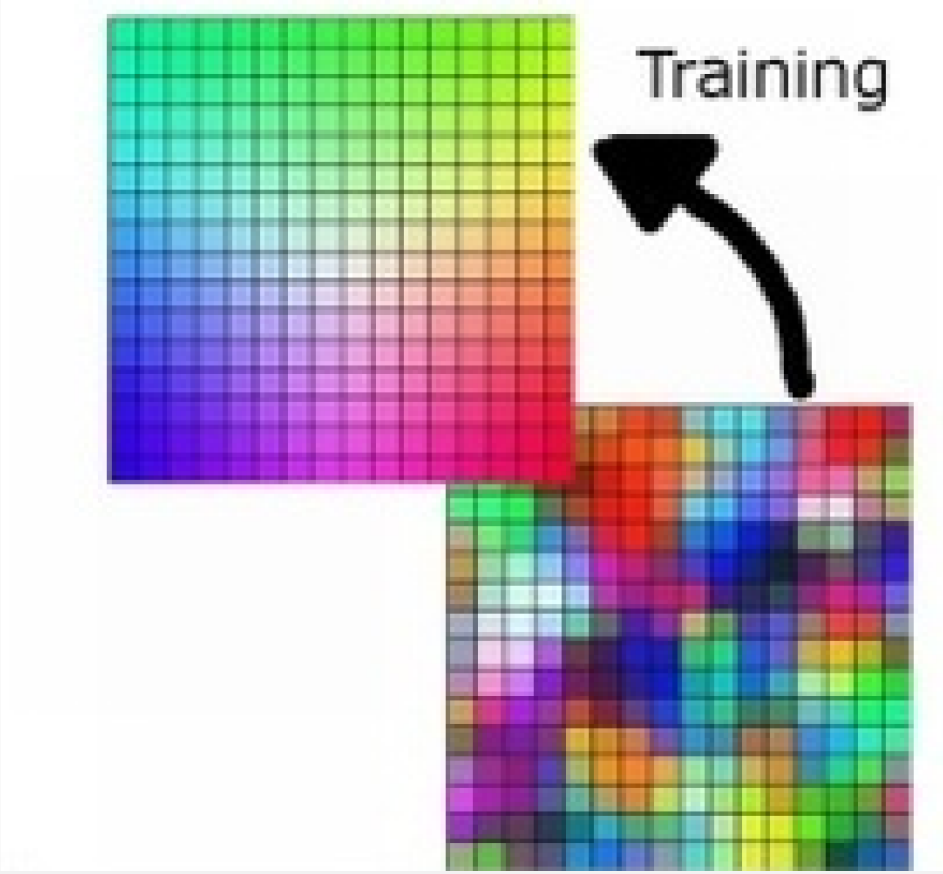
2.- Presentación de los patrones de entrada:
Se realiza de a un patrón por vez.

3.- Determinación de la neurona "ganadora":
Cálculo de la distancia Euclídea entre los vectores de entrada y los vectores de pesos. Elección de la distancia mínima.

4.- Entrenamiento:
Proceso iterativo de ajuste de los pesos de las distintas neuronas de acuerdo a las correspondientes fórmulas, teniendo en cuenta si la neurona se encuentra dentro de la Región de Vecindad o no.

5.- Continuación:
Repetir las etapas 2, 3 y 4 hasta que no se verifiquen cambios sustanciales en la configuración final.

Mapa Topológico de Patrones Antes y Después de Entrenar.



INTERPRETACION DE LOS RESULTADOS

Una vez entrenada la red SOM se procede al etiquetado de la misma, para lo cual se le presentan a la red todos los casos y se identifican las neuronas que representan cada caso.

Los limites entre los diferentes grupos que la red entrenada genera, se determinan comparando las distancias entre los vectores de pesos de las neuronas etiquetadas. Si la distancia es reducida pertenecen al mismo grupo, en cambio si la distancia es grande pertenecen a grupos diferentes.

Modificación del algoritmo.

Un problema que puede surgir es cuando una neurona se transforma en ganadora con una frecuencia mayor que el resto., se introduce en el algoritmo original, un mecanismo de modificación de la distancia llamado *"conscience"*, que registra la frecuencia con que cada neurona resulta "ganadora" .

Sea la distancia entre un vector patrón y el vector peso de la neurona :

$$D_i = |x - w_i|$$

$$= \sqrt{(x_1 - w_{i1})^2 + (x_2 - w_{i2})^2 + \cdots \ldots \ldots + (x_p - w_{ip})^2}$$

La distancia corregida al incorporar el mecanismo de *"conscience"* es:

$$D_i' = D_i - B_i = D_i - \gamma\left(\frac{1}{N} - F_i\right)$$

Donde:

γ es variable en el tiempo, toma un valor inicial grande (2-10) para luego disminuir. El mejor valor depende de cada aplicación.

F_i es la frecuencia ganadora de la neurona i. Al principio:

$$F_i = \frac{1}{N} \rightarrow B_i = 0$$

Para calcular los valores de F_i se utilizan las siguientes ecuaciones:

Para la neurona ganadora:

$$F_{i\,nueva} = F_{i\,anterior} + \beta(1.0 - F_{i\,anterior})$$

Para el resto:

$$F_{i\,nueva} = F_{i\,anterior} + \beta(0.0 - F_{i\,anterior})$$

De esta forma para la neurona ganadora el B_i resulta negativo, en consecuencia la distancia corregida D_i' es mayor disminuyendo por lo tanto la posibilidad de la neurona i a volver a ser *"ganadora"*.

Para el resto de las neuronas la distancia corregida disminuye.