REDES AUTOORGANIZATIVAS II

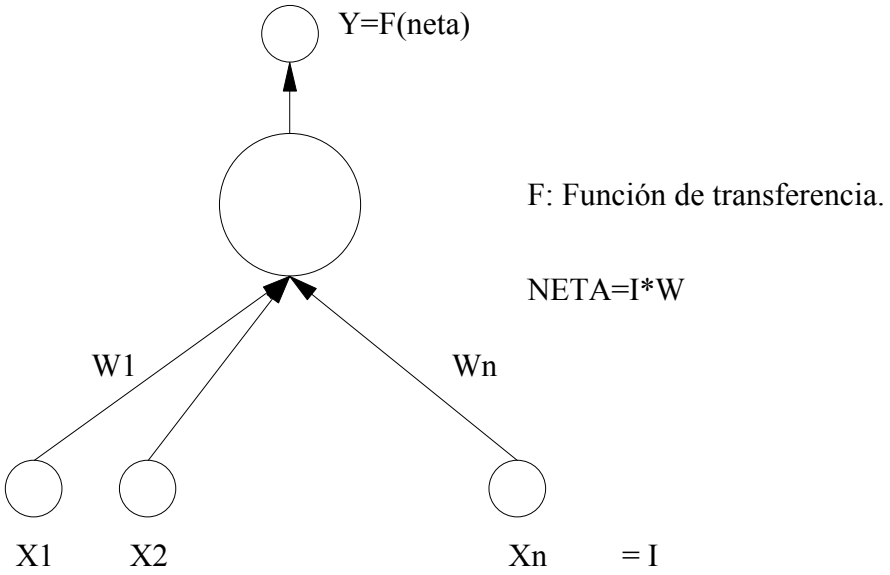
1. Leyes de Grossberg.
    1.1. Red de Contrapropagación.
        - Estructura.
        - Funcionamiento.
        - Limitaciones y Ventajas.

2. Teoría de la Resonancia Adaptiva.
    2.1. Introducción.
    2.2. ART1. Modelo Simplificado.
        - Estructura de la red.
        - Fases de Funcionamiento.
        - Señales externas.
        - Modificación de Pesos.
        - Limitaciones.
    2.3. ART2. Modelo Continuo.

LEYES DE GROSSBERG (Freeman, capt.6; Matlab)

Stephen Grossberg desarrolló 2 modelos de PEs: **Instar** **Outstar**. Los modelos fueron desarrollados para tratar de explicar el fenómeno visual en humanos y animales. Sobre cada uno de los modelos desarrolló también las reglas de aprendizaje que se utilizarían en ambos casos.

Instar



- Este modelo de PE tiene como misión el aprender a reconocer vectores de entrada, de tal manera el cambio que se produce en las conexiones hace tender a éstes hacia el vector de entrada.

  Se supone que el vector de entrada y el de pesos se han normalizados para tener una longitud de 1

La salida de la Instar está gobernada por

la siguiente ecuación diferencial:

$$\dot{y} = -a * y + b * neta \qquad a, b > 0$$

Si resolvemos la ecuación en función del tiempo obtenemos que:

$$y(t) = \frac{b}{a} neta(1 - e^{-at})$$

y el valor de equilibrio es:

$$y_{eq}(t) = \frac{b}{a} neta$$

Si se elimina el vector de entrada en el instante t′, después de haber alcanzado el equilibrio tenemos:

$$y(t) = y_{eq}(1 - e^{-a(t-t')})$$

Si se desea que la Instar responda lo más posible a un vector de entrada determinado, es preciso disponer de un vector de pesos que sean idénticos al vector de entrada. Todo ello implica que:

$$\overset{*}{w} = -cw + dIy \qquad c,d > 0$$

$$Suponiendo \quad y = y_{eq}$$

$$\overset{*}{w} = -cw + dI(wI)$$

Si se suprime ahora el vector de entrada:

$$\overset{*}{w} = -cw \qquad \text{Proceso de Olvido}$$

Para evitar este olvido se modifica la ecuación de tal forma que todo cambio en el vector de pesos, dependa de si hay o no que aprender un nuevo vector de entrada:

$$\overset{*}{w} = (-cw + dIy)U(neta) \quad c,d > 0$$

$$U(neta) = \begin{cases} 1 & neta > 0 \\ 0 & neta = 0 \end{cases}$$

Si interpolamos tenemos:

$$w_{eq} = \frac{d}{c} I \quad y \quad c = d$$

$$\Delta w = a(I - w)$$

con lo cual obtenemos :

$$w(t+1) = w(t) + a(I - w(t))$$

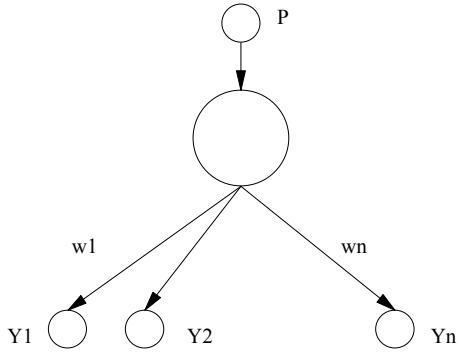
Con el factor de aprendizaje dependiente normalmente del tiempo: a=μF(neta)

Si se hiciese que F fuese un hardlimiter tendríamos la ecuación que rige el aprendizaje competitivo de Kohonen, (cierta similitud).

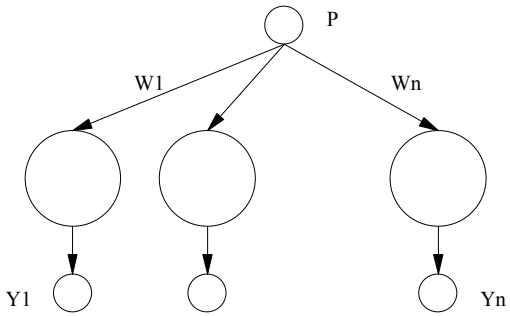
Ejemplo: Freeman pag. 235 (media de vectores) y en el Matlab.

Outstart

- Así como la Instar podía clasificar o reconocer un vector de entrada, este modelo puede reconstruir un vector o recalcularlo a partir de un valor.



EL PE OUTSTAR (Visto como un sólo PE)



El OUTSTAR (Visto como muchos PE de una sóla salida)

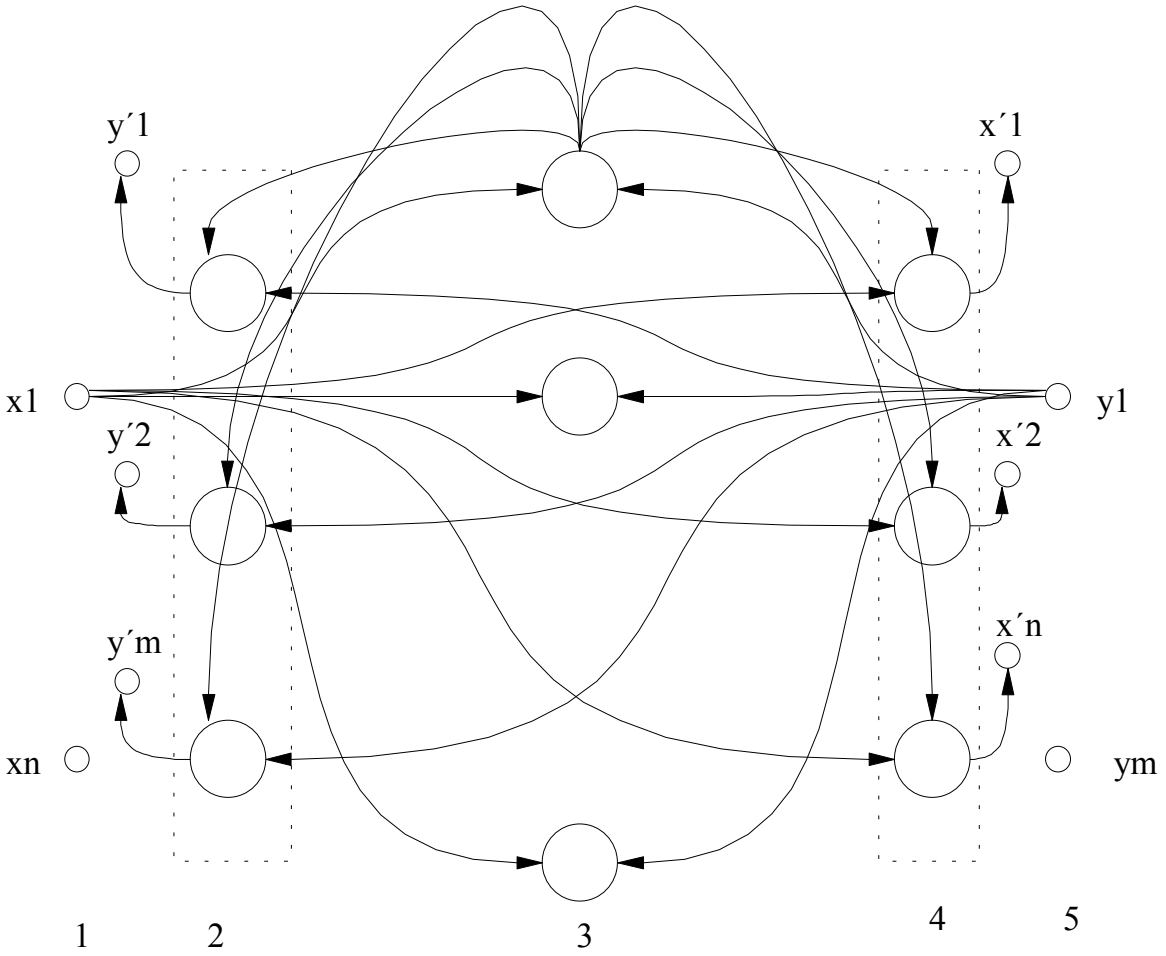
El entrenamiento de este modelo de PE, es muy parecido al del Instar:

$$w_i(t+1) = w_i(t) + \beta(y_i - w_i(t))$$

$$\beta = \mu P \quad normalmente.$$

RED CONTRAPROPAGACIÓN (Freeman)

- No es un nuevo modelo de red, es: una combinación de tipos de redes.



ESTRUCTURA DE UNA RED CONTRAPROPAGACIÓN

COMPONENTES:

- Capa de Entrada (1,5).
- Capa Oculta (3) (PEs Instar. Aprendizaje Competitivo).
- Capa de Salida (2,4) (formada por PEs del tipo Outstart).

Nota: En la bibliografía nos podemos encontrar que la Red de Contrapropagación es una combinación de aprendizaje sin supervisar y supervisado:

Modelo de Kohonen (estructura competitiva) Red Perceptrón (capa oculta y de salida) (supervisado).

FUNCIONAMIENTO:

- Dado un conjunto de pares de vectores (X,Y), entradas y salidas deseadas, este tipo de red puede aprender a asociar un vector X a un vector Y. Se trata en su conjunto de una tarea supervisada.

- Si la relación entre X e Y se puede describir mediante una función continua tal que Y=F(X), entonces la CPN aprenderá dicha correspondencia para todo valor de X en el intervalo especificado por el conjunto de vectores de entrenamiento. Además si existe la inversa de F, entonces, también aprenderá la correspondencia X=F´(Y).

CARACTERÍSTICAS:

Las CNP son interesantes por varios motivos:

- Combinan tipos de redes distintas.
- Usan diferentes algoritmos de aprendizaje en diferentes capas.
- Suponen una ganancia en la velocidad de aprendizaje.
- En algunos problemas es útil combinar aprendizaje supervisado y no supervisado en la misma red.
- El efecto global es el de una red con aprendizaje supervisado, ya que previamente se debe conocer la salida que se desea para cada información de entrada.
- La única limitación va a consistir en la necesidad de que: parecidos vectores de entrada deben producir parecidas informaciones de salida. Si esto no se cumple, será necesario aumentar el número de neuronas de la capa oculta para obtener unos resultados aceptables.

ENTRENAMIENTO DE LA CNP.

- Dado que en la estructura se utilizan dos aprendizajes diferentes, tendremos que considerar cada uno por separado: así se debe de entrenar en primer lugar la estructura competitiva y a continuación la estructura supervisada.

Estructura competitiva.

- La estructura competitiva (Capa de Entrada y Oculta) se entrenan utilizando el algoritmo de aprendizaje de la Instar (Si fuese el caso de que el PE ganador produce una salida 1 y todos los demás 0).

1-Se seleccionan los vectores de entrada y se inicializan los pesos.
2-Se aplica un vector de entrada y se calcula la neurona ganadora en la capa oculta (pesos próximos a la entrada).
3-Se modifican los pesos de la neurona ganadora de la capa oculta con la capa de entrada, siguiendo la regla siguiente:

$$w_{ki}(t+1) = w_{ki}(t) + \mu(t)(x_i(t) - w_{ki}(t))$$

4- Se repiten los pasos 2 y 3 tantas veces como patrones haya.
5- Se repiten el paso 4 hasta que los pesos se hayan estabilizado.

Estructura Supervisada.

- Una vez que se ha entrenado la zona competitiva y tenemos asignada cada una de las neuronas ocultas a una clase diferente se procede a entrenar la capa de salida. Varias Formas:

  - Supongamos que los patrones de entrenamiento están distribuidos en diferentes clases y para cada clase tenemos una salida deseada (**sólo una**). Esto supone que en la capa oculta cada clase activará un sólo PE y nos bastaría entonces hacer que los pesos asociados a dicho PE y los PE de la capa de salida fuesen igual al vector de salida deseado.

  - Si por el contrario, si a diferentes vectores de entrada, pertenecientes a la misma clase (mismo PE en la capa competitiva), se le asociasen salidas deseadas diferentes, entonces se aplicaría un algoritmo de aprendizaje supervisado por corrección de errores.

- Se aplica un vector de entrada y su correspondiente salida deseada.
- Se determina la unidad ganadora en la capa competitiva.
- Se actualizan los pesos de las conexiones que van de los PEs de la capa competitiva a las unidades de salida, utilizando la siguiente expresión (Outstart):

$$w_i(t+1) = w_i(t) + \beta(y_i - w_i(t))$$

$$\beta = \mu P \quad \textit{normalmente.}$$

- Se repiten los pasos del 1 al 3 hasta obtener las salidas deseadas para cada patrón.

Tenemos que hacer ***varias consideraciones***:

- En la capa competitiva para cada patrón sólo se activa un PE con lo que las salidas de los demás PE en dicha capa serán 0.
    - Los únicos pesos que se modificarán en la estructura supervisada serán los que se correspondan a las conexiones de los PEs de la capa de salida con el PE ganador en la capa competitiva.
- Anteriormente se ha dicho que la capa de salida es una Outstar y que en la bibliografía nos podíamos encontrar con que dicha capa fuese del tipo Perceptrón. Esta similitud viene dada por lo siguiente:

La regla de modificación de pesos para el Perceptrón es la siguiente:

$$w(t+1) = w(t) + 2\mu E_k x_k$$

Si consideramos:

- Función de Transferencia de los PEs (capa de salida) es lineal.

- La salida de los PEs (capa oculta) es 0 o 1 dependiendo si es la ganadora o no. La expresión anterior se convierte en la siguiente:

$$w_{hj}(t) + \alpha(y_h - w_{hj}(t)\,x_j)\,x_j \qquad j = neurona \quad ganadora$$

$$w_{hj}(t+1) =$$

$$w_{hj}(t) \qquad\qquad en \quad otro \quad caso$$

Misma Ley que para el Outstart.

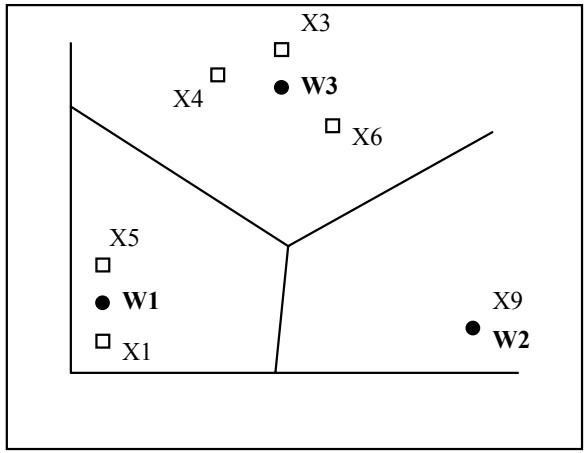
Ejemplo: Consideremos una estructura de red contrapropagación con 2 PE en la capa de entrada, 3 en la capa oculta y 4 en la capa de salida. Tomemos como patrones de entrenamiento los siguientes:

X1=(1,1) ----Y1= (0,0,0,1) X2=(9,1) ----- Y2=(0,0,1,0) X3=(5,9) ----Y3= (0,0,1,1)
X4=(4,8) ----- Y4=(0,1,0,0) X5=(1,3) ----Y5= (0,1,0,1) X6=(6,7) ----- Y6=(0,1,1,0)

La subred de Kohonen de la entrada establecería, después de calcular las distancias entre ellos, que X1 y X5 son de la misma clase, X2 de otra y X3,X4,X6 de otra. Por tanto los prototipos de los pesos de la capa competitiva se pueden calcular como la media aritmética:

W1 = [(1+1)/2,(1+3)/2] = (1,2) W2 = (9,1) W3 = [(5+4+6)/3,(9+8+7)/3] = (5,8)

Si se representan en el plano estos pesos, coincidirán con los centroides o centros de gravedad de los grupos de puntos considerados de la misma clase. Los promedios de las salidas deseadas se almacenarían en los pesos del Perceptrón de salida:



V1 = [(0+0)/2,(0+1)/2,(0+0)/2,(1+1)/2] = (0,0.5,0,1)
V2 = (0,0,1,0)
V3 = [(0+0+0)/3,(0+1+1)/3,(1+0+1)/3,(1+0+0)/3] = (0,0.7,0.7,0.3)

TEORÍA DE LA RESONANCIA ADAPTIVA (ART1) (Neural Computing)

Una de las características de la memoria humana consiste en:

- **Habilidad para aprender nuevos conceptos sin necesitar para ello olvidar otros aprendidos en el pasado.**

Objetivo: Obtener esta misma capacidad en las redes neuronales. Sin embargo, muchas de estas redes tienden a ***olvidar*** informaciones pasadas al tratar de ***enseñarles*** otras nuevas.

Así: Tarea de Clasificación de Patrones.

Problema bien definido ---------- Conjunto de Entrenamiento.
Aprendizaje Correcto ---------- Matriz de Pesos (finales).

Limitaciones:

- Problemas no están bien definidos.
- Añadir nuevos patrones de interés una vez que se ha realizado el entrenamiento con unos iniciales puede suponer el olvido de lo aprendido en la primera fase.

Lo que se ha intentado mostrar en esta descripción es lo que Grossberg denomina el ***Dilema de la Estabilidad y Plasticidad del Aprendizaje***. Este dilema plantea los siguientes interrogantes:

- Poder aprender nuevos patrones (***Plasticidad del aprendizaje***).
- Poder retener los patrones previamente aprendidos (***Estabilidad del aprendizaje***).

Conseguir una red que pueda dar respuesta a uno de estos interrogantes es sencillo. No lo es, si se pretende diseñarla para que solucione ambos.

En respuesta a este dilema, Grossberg y Carpenter desarrollaron la ***denominada Teoría de la Resonancia Adaptiva (ART)***.

- Se aplica a sistema competitivos.
- Lo que se pretende es ***categorizar*** (clusterizar).
- Aprendizaje Sin Supervisar.

IDEA:

Para solucionar el dilema de la plasticidad y estabilidad, el modelo ART propone añadir a las redes un ***mecanismo de realimentación*** entre las neuronas competitivas de la capa de salida de la red y la capa de entrada. Este mecanismo facilita el aprendizaje de nueva información sin destruir la ya almacenada.

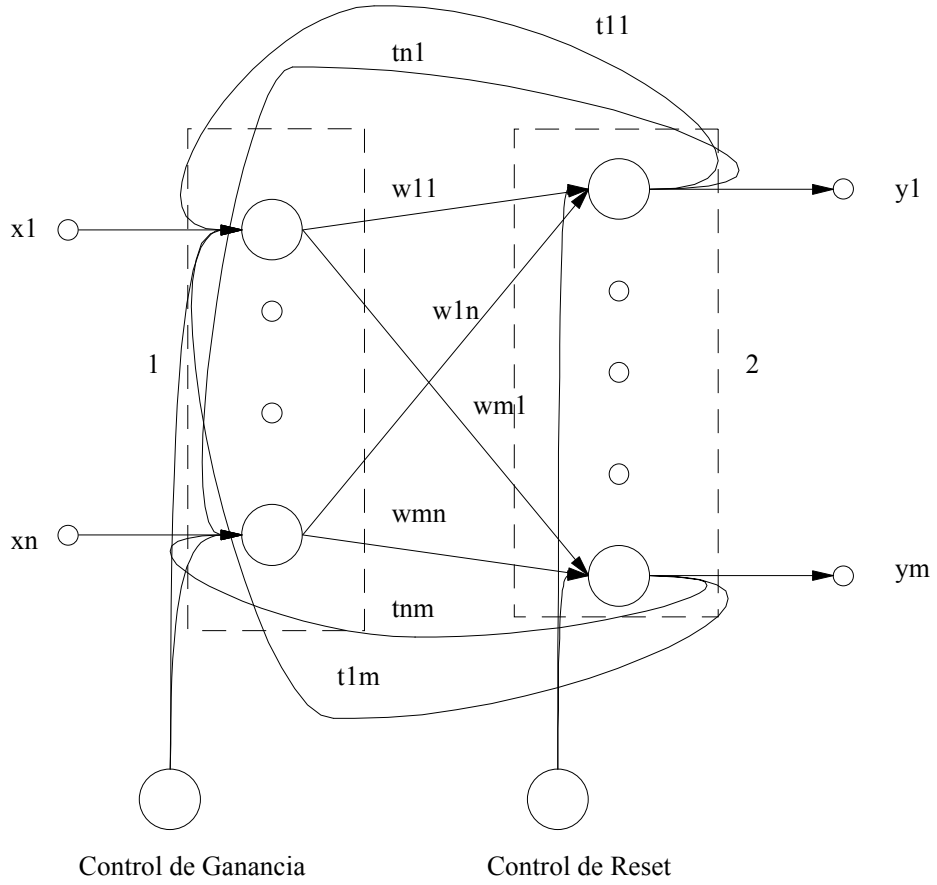
La teoría de la resonancia adaptiva se basa en la idea de hacer ***Resonar*** la información de entrada con los representantes o prototipos de las categorías que reconoce la red.

> Si entra en resonancia con alguno, es suficientemente similar, la red considera que pertenece a dicha categoría y únicamente realiza una pequeña adaptación del prototipo almacenado representante de la categoría para que incorpore algunas características del dato presentado.
>
> Cuando no resuena con ninguno, no se parece a ninguno de los existentes, ***recordados por la red*** hasta ese momento, la red se encarga de crear una nueva categoría con el dato de entrada como prototipo de la misma.

Dos operaciones: ***Entrenamiento y Adaptación***.

ARQUITECTURA Y FUNCIONAMIENTO.

- Una red ART básicamente consta de ***dos capas entre las que se establecen conexiones hacia adelante y hacia atrás*** y dos tipos de señales de control.



Cada una de las capas existentes en la red realiza tareas diferentes:

Capa 1 (Competitiva).
Operaciones:
Entrada y Comparación.

Capa 2. Operaciones:
Salida y Reconocimiento.

OPERACIÓN

- Se puede describir la acción de la red en términos de la actividad de cada capa para una serie de fases:

  - **Fase de Inicialización.**
  - **Fase de Reconocimiento.**
  - **Fase de Comparación.**
  - **Fase de Búsqueda.**

Fase de Inicialización

- La fase de inicialización consiste en inicializar los diferentes parámetros de la red y determinar como serán las señales control.

- Los pesos se inicializan siguiendo el siguiente criterio:

  wji = 1/(1+N). N = nº entradas (conexiones hacia adelante).
  tij = 1 (conexiones hacia atrás).

SEÑALES DE CONTROL

Las señales de control se utilizan para controlar el flujo de datos a través de la red y conmutar entre las diferentes fases.

Control de Ganancia: controla el flujo para la capa de entrada y conmuta entre dos estados:

Entrada de Datos. Comparación.

- Esta señal toma valores binarios y dichos valores suelen depender de varias señales:

  - Salidas de las Neuronas de la capa de Salida ponderadas por pesos (-N).
  - Vector de entrada ponderado por pesos (1).

Señal de Ganancia (SG)

SG es uno siempre que se le presenta a la red un patrón de entrada válido. Si en la capa competitiva existe alguna neurona activa SG es forzado a 0.

Control de Reset: Es el más simple de los dos.

$$SG = \begin{cases} 1 & \sum_{i=1}^{N} xi - N\sum_{j=1}^{M} y_j \geq 0.5 \\ 0 & \sum_{i=1}^{N} xi - N\sum_{j=1}^{M} y_j < 0.5 \end{cases}$$

Controla el flujo de los datos para la capa de salida. Sirve para conmutar también entre dos estados: ***Reconocimiento y Búsqueda.***

- Toma también valores binarios y dichos valores activan o desactivan los nodos en la capa de reconocimiento (capa de salida). Lo mismo que en el caso anterior estos valores suelen depender de diferentes señales.

    - Salidas de las neuronas,capa de entrada ponderadas por (-1).
    - Vector de entrada ponderado por ρ ***(parámetro de vigilancia***).

De tal manera la señal de Reset será de la siguiente forma:

La señal de reset tiene como ***misión desactivar la neurona ganadora en la capa de salida cuando un vector de entrada no es suficientemente similar al prototipo de una clase***. Esta medida de similaridad depende del factor o parámetro de reset.

$$SR = \begin{cases} 1 & \sum_{i=1}^{N} \rho x_i - \sum_{i=1}^{N} neta_i^1 > 0 \quad \text{Reset} \\ 0 & \sum_{i=1}^{N} \rho x_i - \sum_{i=1}^{N} neta_i^1 \leq 0 \end{cases}$$

FASE DE RECONOCIMIENTO.

- En la fase de reconocimiento un vector de entrada es presentado a la red, activando una neurona en la capa de salida.

- Las neuronas en la capa de entrada reciben 3 entradas:

Vector de entrada. ***Salidas de la 2 capa.*** ***Control de ganancia.***

- En esta fase la señal de ganancia es 1 ya que ninguna neurona de la capa de salida está activada. Las neuronas de la capa de entrada se activan utilizando la regla 2/3. Es decir si 2 o más entradas están activadas la neurona se activa produciendo un 1, en otro caso produce un 0.

Para las neuronas de la capa de salida la operación que se realiza es la siguiente:

$Yj = 1$ si $\Sigma Wji*Xi$ es máximo.
$Yj = 0$ en otro caso.

Entradas Binarias (distancia mínima) (Operación AND).

FASE DE COMPARACIÓN.

- Dos vectores son presentados en la capa de entrada: vector de entrada y vector producido por la capa de salida (una neurona activada).
    - Señal de control de ganancia es 0.
    - Se conmuta entre 2 operaciones: ***Reconocimiento*** (entrada) y ***Comparación***.
- Al estar la señal de ganancia puesta a 0, la salida de la capa de entrada es un AND entre el vector de entrada y el vector producido por la capa de salida ponderado por los pesos feedback.

Test de Vigilancia.

- El resultado obtenido en la capa de entrada después de la fase de comparación es enviado al control de Reset. Simplemente significa una medida de similaridad entre el vector de entrada y el vector prototipo obtenido por la capa de salida.

La comparación se hace valorando la siguiente relación de semejanza:

$$relación \quad de \quad semejanza = \frac{\|X * Y\|}{\|X\|} = \frac{\left\|X * T_j\right\|}{\|X\|}$$

$$\|X * Y\| = \sum_{i=1}^{N} x_i * y_{ij} = \sum_{i=1}^{N} x_i * t_{ij} \qquad j = neurona \quad ganadora$$

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

- Al trabajar con valores binarios (0/1), ***el producto aritmético equivale al lógico (AND***), y por tanto, lo que se representa es: el número de componentes binarios con valor 1 que coinciden entre el vector de entrada y el de salida.

    - En el caso de que fuesen completamente iguales, este valor coincidiría con el denominador, y la relación de semejanza sería la unidad (representando el 100%). Si no coincidiesen en ningún bit, entonces sería 0 (representado el 0%).

- Una vez calculada la relación de semejanza entre ambas informaciones, se compara dicha relación con el ***parámetro de vigilancia ρ,*** cuyo valor será fijado por el usuario en un rango de 0 a 1 y que influirá en el número de clases que establecerá la red, ya que cuando mayor sea su valor, se está pidiendo al sistema que discrimine con mayor precisión, de tal forma que si el valor asignado es 1 se estará indicando que a una clase sólo pueden pertenecer patrones que sean exactamente iguales.

  - Si se cumple que ***Relación de Semejanza < ρ,*** entonces la neurona vencedora j, en la capa de salida no representa la apropiada categoría a la que pertenece la información de entrada y es desestimada, ***se resetea o elimina del conjunto de posibles neuronas vencedoras***, y se vuelve a presentar el patrón a la estructura.

  - En el caso contrario se ha encontrado la categoría apropiada al vector de entrada y se deben actualizar los pesos para confirmar este hecho:

Actualización de pesos:

- Los pesos feedforward (W) tienen el mismo valor que los feedback (T), pero normalizado:

$$W=T/(\gamma+\Sigma T)$$

γ es un valor pequeño, normalmente se utiliza 0.5, aunque en la inicialización de los pesos se recomiendo como valor la unidad (Lippman).

- Si la neurona de salida j no es la vencedora, el valor de su salida es cero, con lo que no debe producirse variación en el peso correspondiente. Por el contrario, si la neurona es la ganadora si se debe producir variación en los pesos asociados a ella.

La regla de cambio de pesos es la siguiente: $t_{ij}(t+1) = t_{ij}(t) * x_i$

Los pesos Wji se obtienen normalizando los anteriores:

$$w_{ji}(t+1) = \frac{t_{ij}(t) * x_i}{\gamma + \sum_{i=1}^{N} t_{ij}(t) * x_i}$$

FASE DE BÚSQUEDA:

- Si en un primer momento se ha determinado que la primera neurona ganadora no representa la categoría del vector de entrada, se debe empezar la búsqueda por diferentes categorías establecidas por la red.

- Al no representar la categoría del vector de entrada la señal de control de reset, desactiva dicha neurona ganadora.

  Esta acción tiene un doble efecto: por una parte fuerza su salida a 0 y la excluye para las sucesivas búsquedas y por otro lado la señal de control de ganancia es puesta a 1 (ninguna neurona de la capa de salida es ganadora).

- El vector de entrada es entonces de nuevo aplicado a la estructura repitiendose entonces los pasos previamente expuestos. En estas circunstancias podemos llegar a 2 situaciones completamente diferentes:

  - Si todavía quedan neuronas de las salida que no hayan sido desestimadas, se repite el proceso de búsqueda de la categoría a la que pertenece la entrada.

  - Si no queda ninguna neurona candidata porque ya se ha repetido el proceso, entonces se produce una ***situación de saturación*** de la red, ya que no se puede crear una nueva categoría para el patrón, al no existir ninguna neurona de salida que pueda asociarse a la nueva clase. Podría solucionarse el problema ampliando el número de neuronas de salida de la red de forma dinámica, ya que los nuevos pesos no afectarían a representantes ya almacenados en el resto de conexiones que componen la memoria a largo plazo de la red.

ALGORITMO GENERAL:

1- Se inicializan pesos y parámetro de vigilancia.

2- Se aplica una nueva entrada.

3- Se determina la neurona ganadora en la capa de salida.

4- Se compara la categoría que representa la neurona ganadora con la entrada.

Si pertenece a la categoría, se va al paso 5.
Si no pertenece a la categoría se desactiva la ganadora y se vuelve al paso 2.

5- Se adaptan los pesos.

6- Se desactivan todas las neuronas de la capa de salida.

7- Se vuelve al paso 2.

LIMITACIONES DE LA RED ART.

- Aunque la red ART es una de los más potentes modelos con capacidad autoorganizativa, sin supervisión, existen algunas limitaciones importantes a tener en cuenta que dificultan la tarea de clasificación realizada por la red. Estas limitaciones se refieren:
  - Su ***dependencia del tipo y orden de las informaciones aprendidas***.
  - ***La influencia del parámetro de vigilancia*** en el número de categorías creadas por la red, ya que pequeños cambios de este parámetro pueden originar un gran número de categorías.
- Se trata de una red con una gran sensibilidad ante el ruido o distorsión, también ante el desplazamiento de los datos de entrada, que puede dar lugar a una gradual degradación de los prototipos correspondientes, debido al carácter destructivo (operación AND de datos de entrada y prototipo) de la regla de aprendizaje que realiza la adaptación de los pesos. Sin embargo, si estos datos de entrada son sometidos a un adecuado preprocesamiento, la red ART puede ser útil para realizar una posterior tarea de clasificación o ***categorización***.

- También este modelo presenta el problema de la ineficiencia en cuanto a las necesidades requeridas para el almacenamiento de los pesos de las conexiones entre neuronas. Hay que tener en cuenta que se precisan 2*N para representar cada categoría de N bits.

EJEMPLO: Supongamos una estructura de red ART con 3 neuronas en la capa de entrada y 2 neuronas en la capa de salida.

Sean X1=(1,1,0,0) X2=(0,0,1,1) X3=(1,1,1,0)

Se inicializan pesos: $T = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$ $W = \left(\begin{array}{c|c} 0.2 & 0.2 \\ 0.2 & 0.2 \\ 0.2 & 0.2 \\ 0.2 & 0.2 \end{array}\right)$

Se presenta el primer vector de entrada X1. Como es la primera información de entrada, no hay ninguna neurona vencedora, se almacena directamente el vector X1 en los pesos de la primera neurona de la capa de salida.

Así los nuevos pesos serán: $T = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}$ $W = \left(\begin{array}{c|c} 0.4 & 0.2 \\ 0.4 & 0.2 \\ 0 & 0.2 \\ 0 & 0.2 \end{array}\right)$

Se presentan el segundo patrón, se produce entonces la competición entre las neuronas de la capa de salida con los siguientes resultados: y1=0 y2=0.4
Por tanto resulta ganadora la neurona 2, por lo tanto se vuelven a actualizar los pesos:

$$T = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \qquad W = \left(\begin{array}{c|c} 0.4 & 0 \\ 0.4 & 0 \\ 0 & 0.4 \\ 0 & 0.4 \end{array}\right)$$

Se introduce ahora el 3 patrón de entrada, produciéndose la competición con los siguientes resultados: y1=0.8 y2=0.4
La vencedora es la neurona 1 de la capa de salida, de la que era el prototipo el patrón (1100). A continuación se produce el retorno a través de las conexiones feedback, comparándose el vector X3 con el prototipo de esta clase (1100), obteniéndose la siguiente relación de semejanza: 0.6. Si el criterio se satisface, se considerarían de la misma clase, se ajustan pesos (siguen siendo los mismos). Si no se satisface el criterio se llegaría a una situación de saturación.

MEJORA DEL MODELO. LA RED ART2.

En 1987, Carpenter y Grossberg propusieron ***una versión continua del modelo de resonancia adaptiva, que denominaron ART2***, que era capaz de trabajar con valores de entrada reales, en lugar de binarios. Esta nueva red tiene la misma arquitectura y funcionamiento que la original, pero en este caso los pesos de las conexiones feedback y feedforward son iguales (T=W). El funcionamiento es muy similar.

1. Se inicializan pesos y parámetro de vigilancia. Se presenta un patrón de entrada.

2. Cada neurona de la capa de entrada recibe el valor del componente correspondiente real del vector X y lo envía a todas las neuronas de la capa de salida a través de las conexiones correspondientes W.

3. Se determina la neurona ganadora en la capa de salida para dicha entrada. En este paso se plantea una *diferencia* con respecto al modelo discreto ya que en este caso se supone que la neurona vencedora es aquella j que verifica una ***mínima diferencia (distancia euclídea)*** entre el patrón de entrada y los pesos de las conexiones entre esa neurona j y las de la capa de entrada.

4.La neurona ganadora j, envía su salida (1) a través de las conexiones (Tji).

5.Se compara la información de entrada X con la información Yj recibida. En este modelo la relación de semejanza utilizada es:

Relación de Semejanza=Σ|xi-tji|

6.Se compara la Relación se Semejanza con el parámetro de vigilancia establecido por el usuario, que en este modelo tendrá un rango válido (0..máximo valor de las componentes de X).

7.Si no se satisface dicha comparación se desactiva la neurona j y vuelve al paso 3. Si se satisface se ajustan pesos. La regla utilizada es:

$$w_{ji}(t+1) = t_{ij}(t+1) = \frac{x_i + w_{ji}(t) * Num_j(t)}{Num_j(t) + 1}$$

Numj(t) es el número de vectores de entrada que han sido considerados hasta el instante t de la clase j.

8)- Desactivar todas las neuronas de la capa de salida y volver al paso 2.