Un Circuito de Redes Neurales para la Interpretación de Patrones Ambiguos Mediante la Extracción de Autovectores

Leopoldo Mauro, EXPERT Computación C.A., Caracas
Andreas Meier, Universidad Simón Bolívar, Caracas

RESÚMEN

La característica principal de las redes neurales es su capacidad de reconocer y aprovechar regularidades en los datos, usándolas para clasificar patrones desconocidos. La mayoría de las redes neurales son capaces de dar sólo una clasificación de cada patrón, no un conjunto de clasificaciones alternas, y tampoco indican la confiabilidad de sus resultados. Además, no pueden reconocer nuevos patrones compuestos por sub-patrones conocidos, a menos que se las entrene antes con todas las combinaciones posibles de sub-patrones.

Fundamentándonos en los métodos de extracción iterativa de autovectores empleados en álgebra lineal, proponemos un circuito de tres redes neurales autónomas, que juntas exhiben la capacidad de producir todas las interpretaciones de un patrón en orden decreciente de importancia, descomponiéndolo en sub-patrones conocidos si es posible.

Describimos además un ejemplo de diseño y ensamblaje de redes neurales a partir de componentes modulares más simples, similar a la metodología empleada en electrónica. Esta clase de circuitos tendrán un papel determinante, sobre todo cuando finalmente aparezcan en el mercado los circuitos integrados neurales.

Un Circuito de Redes Neurales para la Interpretación de Patrones Ambiguos Mediante la Extracción de Autovectores

Leopoldo Mauro, EXPERT Computación C.A., Caracas
Andreas Meier, Universidad Simón Bolívar, Caracas

1. Introducción.

La capacidad principal de la técnica de redes neurales consiste en reconocer, almacenar y aprovechar regularidades en un conjunto de datos. La redes neurales pueden estructurar espacios de datos de manera autónoma, y reconocen con gran facilidad la pertenencia de datos de prueba a alguna de esas estructuras; es decir, reconocen y clasifican patrones.

La característica dominante de las técnicas de clasificación mediante redes neurales es que cada patrón de entrada produce un sólo resultado, y no un conjunto de posibles interpretaciones alternas. Generalmente, una red neural tampoco indica la confiabilidad de sus resultados, por lo que un patron de entrada ambiguo es clasificado según alguna de sus posibles interpretaciones sin indicación del hecho. Más aún, la mayoría de las redes no pueden ser entrenadas con patrones ambiguos sin incurrir en alguna penalidad, a veces severa, en la capacidad de almacenamiento o en la confiabilidad de la clasificación. Además, la mayor parte de las redes neurales no pueden detectar que un patrón desconocido está compuesto por dos o más patrones conocidos, a no ser que hayan sido entrenadas previamente con todas las posibles combinaciones de patrones. Esta última es una grave limitación de muchas de las redes neurales existentes, que no parece ser compartida por nuestro cerebro.

El presente trabajo propone un circuito de tres redes neurales autónomas que colaboran para dar al conjunto la capacidad de producir todas las posibles interpretaciones de un patrón en orden decreciente de importancia, y descomponer patrones desconocidos en sus componentes conocidos. Adicionalmente, el circuito produce un "vector de novedad" que contiene los componentes del patrón de entrada que no pudieron interpretarse.

La métodología que proponemos consiste en diseñar y ensamblar redes neurales heterogéneas a partir de componentes modulares más simples de propiedades conocidas, en forma muy similar a lo que se hace en electrónica. La literatura de redes neurales habla muy poco al respecto, con una importante excepción (Fukushima 1980, Fukushima et al. 1983).

2. Métodos de Extracción de Autovectores.

Toda matriz cuadrada ($\mathcal{N}\times\mathcal{N}$) posee $\mathcal{N}$ autovectores y autovalores (algunos de los cuales pueden ser iguales); si además es una matriz real simétrica, sus autovectores y autovalores serán todos reales y ortogonales entre sí. Esta última propiedad es muy importante, pues las matrices de conexión de las redes neurales autoasociativas que se discutirán en este trabajo son todas reales y simétricas.

La extracción algorítmica de autovectores es un tema muy conocido en el álgebra lineal (un buen texto introductorio es Jennings 1977). En el análisis tradicional se ha hecho bastante énfasis en los métodos de extracción directa por transformación de la matriz (diagonalización de Jacobi, transformación de Householder, transformación QR, etc.) por sus ventajas en lo referente a tiempo de cómputo, requerimientos de memoria y errores de redondeo. Pero desde el punto de vista de la computación neural, los métodos iterativos (*power method* y purificación de Aitken) son superiores pues pueden realizarse directamente en el paradigma neural.

2.1. Power Method.

El fundamento del *power method* es el siguiente. Sea $\mathbf{u}$ un vector cualquiera, èxpresado en la base de los autovectores de una matriz simétrica $\mathbf{W}$ como

$$\mathbf{u} = \rho_1 \mathbf{q}_1 + \rho_2 \mathbf{q}_2 + \ldots + \rho_n \mathbf{q}_n \tag{1}$$

Obtengamos

$$\begin{aligned} \mathbf{u}^{(1)} = \mathbf{W}\,\mathbf{u} &= \mathbf{W} \sum_i \rho_i \mathbf{q}_i = \sum_i \rho_i \mathbf{W}\,\mathbf{q}_i \\ &= \rho_1\lambda_1\mathbf{q}_1 + \rho_2\lambda_2\mathbf{q}_2 + \ldots + \rho_n\lambda_n\mathbf{q}_n \end{aligned} \tag{2}$$

Repitiendo esta operación k veces se obtiene

$$\mathbf{u}^{(k)} = \mathbf{W}\,\mathbf{u}^{(k-1)} = \rho_1\lambda_1^k\mathbf{q}_1 + \rho_2\lambda_2^k\mathbf{q}_2 + \ldots + \rho_n\lambda_n^k\mathbf{q}_n \tag{3}$$

Si hemos numerado los autovectores de tal manera que $|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_n|$, podemos hacer la aproximación

$$\mathbf{u}^{(k)} \approx \rho_1\lambda_1^k\mathbf{q}_1 \tag{4}$$

para k suficientemente grande. Vemos que $\mathbf{u}^{(k)}$ tiende a ser proporcional a $\mathbf{q}_1$, el autovector dominante de $\mathbf{W}$, a medida que k crece.

Puesto que los autovectores no cambian ante un escalamiento, resulta conveniente evitar el crecimiento indefinido de $\mathbf{u}^{(k)}$ siguiendo a cada premultiplicación por $\mathbf{W}$ con una normalización. El *power method* puede entonces resumirse en las siguientes ecuaciones recurrentes

$$\mathbf{v}^{(k)} = \mathbf{W}\,\mathbf{u}^{(k)} \tag{5a}$$

$$\mathbf{u}^{(k+1)} = \phi\left(\mathbf{v}^{(k)}\right) \tag{5b}$$

donde ϕ es una función vectorial de normalización, usualmente una de las siguientes

$$\phi(\mathbf{x}) = \mathbf{x} \,/\, \|\mathbf{x}\| \tag{6}$$

$$\phi(\mathbf{x}) = \mathbf{x} \,/\, \max x_i \tag{7}$$

El *power method* siempre puede aplicarse a matrices simétricas. Su convergencia es buena si los autovectores dominantes poseen autovalores diferentes, pero en el caso de autovalores similares la convergencia puede ser muy lenta.

2.2. Purificación de Aitken.

Si el *power method* permite obtener el autovector dominante de una matriz, el método conocido como "purificación de Aitken" (Aitken 1937) permite obtener los demás autovectores, o autovectores subdominantes. Este método está fundamentado en que, si la matriz simétrica $\mathbf{W}$ posee a $\mathbf{q}_1$ como autovector de autovalor λ_1, la nueva matriz

$$\mathbf{W}' = \mathbf{W} - \left(\lambda_1 / \|\mathbf{q}_1\|^2\right) \mathbf{q}_1 \mathbf{q}_1^T \tag{8}$$

posee los mismos autovectores que $\mathbf{W}$, con la excepción que el autovalor asociado con $\mathbf{q}_1$ en $\mathbf{W}'$ es 0. Esto es fácil de demostrar premultiplicando $\mathbf{q}_1$ por $\mathbf{W}'$:

$$\begin{aligned} \mathbf{W}'\,\mathbf{q}_1 &= \mathbf{W}\,\mathbf{q}_1 - \left(\lambda_1 / \|\mathbf{q}_1\|^2\right) \mathbf{q}_1 \mathbf{q}_1^T \mathbf{q}_1 \\ &= \lambda_1 \mathbf{q}_1 - \left(\lambda_1 / \|\mathbf{q}_1\|^2\right) \mathbf{q}_1 \|\mathbf{q}_1\|^2 \\ &= (\lambda_1 - \lambda_1)\,\mathbf{q}_1 = 0\,\mathbf{q}_1 \end{aligned} \tag{9}$$

Este proceso puede extenderse a la eliminación de varios autovectores repitiendo las operaciones indicadas en (8) para cada uno. El método de Aitken para la obtención de autovectores subdominantes consiste, entonces, en la obtención del autovector dominante por medio del *power method*, y su eliminación por medio de (8). Al aplicar nuevamente el *power method* ya no se obtendrá más $\mathbf{q}_1$, sino el siguiente autovector en orden decreciente de autovalor, el cual puede nuevamente eliminarse repitiendo el proceso de purificación.

Esta metodología de purificaciones sucesivas es sensible a los errores en la determinación del autovector a ser eliminado, ya que la iteración del *power method* amplificará dichos errores hasta que interfieran con la convergencia. Por ello el método de Aitken rara vez se emplea para obtener más que los primeros pocos autovectores de una matriz.

3. Redes Neurales Autoasociativas.

En términos generales, una red neural autoasociativa es aquella que, por diseño o entrenamiento, es capaz de reproducir en su salida una copia (posiblemente "limpia") de su entrada. En este artículo nos centraremos tan sólo en las redes autoasociativas de una sóla capa con arquitectura recurrente. Estas redes se caracterizan por poseer una matriz de conexión simétrica, cuyos autovectores dominan la evolución en el tiempo del sistema iterativo resultante. En efecto, el propósito de estas redes es el de reproducir en su salida al vector de entrada, con tal que éste se encuentre entre aquellos que la red reconoce (sus autovectores).

Cuando el vector de prueba no corresponde exactamente a algún vector reconocido, la red ha de responder con el vector conocido más cercano (de acuerdo a cierta métrica). En otras palabras, el vector de prueba es descompuesto de acuerdo a la base definida por los autovectores de la matriz de conexión, y el componente de mayor norma (o más importante) es el resultado de la red.

3.1. Matrices Autoasociativas.

Las redes neurales autoasociativas recurrentes se basan en la siguiente observación (Amari 1977, Anderson et al. 1977). Tomemos una matriz autoasociativa $\mathbf{W}$ construida a partir de un vector columna $\mathbf{a}$ según la siguiente operación (reminiscente del método de Aitken):

$$\mathbf{W} = \mathbf{a}\,\mathbf{a}^T \tag{10}$$

Multiplicando dicha matriz por el "vector de prueba" $\mathbf{p}=\mathbf{a}$ obtenemos

$$\mathbf{a}' = \mathbf{W}\,\mathbf{p} = \mathbf{a}\,\mathbf{a}^T\,\mathbf{a} = \mathbf{a}\,\|\mathbf{a}\|^2 = \lambda\,\mathbf{a} \tag{11}$$

En otras palabras, la matriz $\mathbf{W}$ posee al vector $\mathbf{a}$ como uno de sus autovectores, con autovalor $\lambda=\|\mathbf{a}\|^2$ (positivo para todo $\mathbf{a}$ no nulo).

Es posible construir la matriz $\mathbf{W}$ combinando múltiples vectores $\mathbf{a}_i$ (usualmente denominados "vectores de entrenamiento"):

$$\mathbf{W} = \sum_i \mathbf{a}_i\,\mathbf{a}_i^T \tag{12}$$

Al multiplicar $\mathbf{W}$ por un vector de prueba $\mathbf{p}=\mathbf{a}_j$ tendremos

$$\begin{aligned}\mathbf{a}_j' = \mathbf{W}\,\mathbf{p} &= \left(\sum_i \mathbf{a}_i\,\mathbf{a}_i^T\right)\mathbf{a}_j \\ &= \mathbf{a}_j\,\mathbf{a}_j^T\,\mathbf{a}_j + \sum_{i\neq j} \mathbf{a}_i\,\mathbf{a}_i^T\,\mathbf{a}_j \\ &= \lambda_j\,\mathbf{a}_j + \sum_{i\neq j} \mathbf{a}_i\,(\mathbf{a}_i^T\,\mathbf{a}_j)\end{aligned} \tag{13}$$

Cuando los diversos $\mathbf{a}_i$ son ortogonales entre sí, el segundo término de (13) desaparece y

los vectores $\mathbf{a}_i$ constituyen los autovectores de la matriz **W**. Cuando los $\mathbf{a}_i$ no son ortogonales, (13) puede expresarse como

$$\mathbf{a}'_j = \lambda_j \mathbf{a}_j + \sum_{i \neq j} \varepsilon_{ji} \mathbf{a}_i \tag{14}$$

donde ε_{ji} es la proyección del vector $\mathbf{a}_j$ sobre $\mathbf{a}_i$ ($\mathbf{a}_i^T \mathbf{a}_j$). Aplicando el ***power method*** a $\mathbf{a}'_j$ obtenemos por inducción, para *k* suficientemente grande

$$\mathbf{q}_j = \lambda_j \mathbf{a}_j^{(k)} + \sum_{i \neq j} \varepsilon_{ji} \mathbf{a}_i^{(k)} \tag{15}$$

donde $\mathbf{q}_j$ es el autovector cuya "semilla" es $\mathbf{a}_j$. En otras palabras, cada autovector de **W** es una combinación lineal de los diversos $\mathbf{a}_i$. Puede demostrarse que los autovectores minimizan la "medida de error"

$$E = \sum_{I,\ J} |\mathbf{a}_i^T \mathbf{q}_j| \ / \ (\mathbf{q}_j^T \mathbf{q}_j) \tag{16}$$

En efecto la matriz **W** ha adquirido un nuevo conjunto de autovectores que reemplaza a los vectores de entrenamiento originales. Esta sensibilidad de las matrices de asociación ante la no-ortogonalidad de sus vectores de entrenamiento limita grandemente su uso práctico. Una solución a este problema se plantea en la próxima sección.

3.2. Brain-State-in-a-Box (BSB).

La red neural Brain-State-in-a-Box (Anderson 1977) es un sistema iterativo definido por la siguiente ecuación de recurrencia:

$$\mathbf{a}^{(k+1)} = \sigma\left(\mathbf{W}\, \mathbf{a}^{(k)}\right) \tag{17}$$

donde σ es una función vectorial no lineal saturable de la forma:

$$\sigma(\mathbf{x}) = \begin{cases} \mathbf{x} & si \ \max |x_i| < 1 \\ \mathbf{x} \ / \max |x_i| & si \ \max |x_i| \geq 1 \end{cases} \tag{18}$$

El propósito de su peculiar forma se hará evidente más adelante.

La matriz BSB es una matriz autoasociativa, construida de acuerdo al método de la sección anterior, sumada a una matriz identidad. La matriz resultante posee los mismos autovectores que la matriz asociativa original, pero sus autovalores son siempre mayores o iguales a 1. En efecto, si **q** es un autovector de **W**, tenemos

$$(\mathbf{W} + \mathbf{I})\, \mathbf{q} = \mathbf{W}\, \mathbf{q} + \mathbf{I}\, \mathbf{q} = \|\mathbf{q}\|^2 \mathbf{q} + \mathbf{q} = (\|\mathbf{q}\|^2 + 1)\, \mathbf{q} \tag{19}$$

Los vectores de entrenamiento y los de prueba deben construirse a partir de elementos

tomados del conjunto discreto {-1, 0, +1}, con significado convencional de *falso, desconocido* y *cierto*, respectivamente. No es necesario normalizar los vectores.

La ecuación (17) puede representarse por medio del siguiente diagrama de bloques, similar a los empleados en teoría de control:

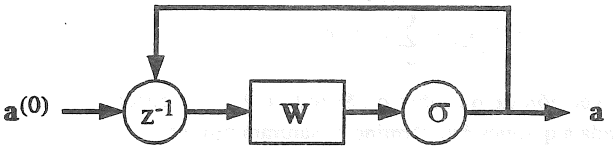

Figura 1

donde el bloque z^{-1} produce un retardo de una unidad de tiempo, con valor inicial $\mathbf{a}^{(0)}$. El vector **a** es el "vector de estado" de este sistema de tiempo discreto.

Los puntos estables no nulos del sistema sólo se hallan en las esquinas del hipercubo definido por la función σ, dondes todos los componentes del vector **a** son +1 ó -1. No pueden haber puntos estables en el interior del hipercubo (excepto en el origen) pues la presencia de autovalores mayores que 1 causa que el circuito posea realimentación positiva, y hace que los componentes del vector de estado crezcan hasta saturarse en alguna esquina. En efecto, las trayectorias del estado del sistema están restringidas al interior de la "caja" definida por la función σ– de allí el nombre de la red. El algoritmo BSB itera hasta que el vector **a** deja de cambiar, por lo que su terminación está garantizada.

En el interior del hipercubo el sistema BSB es lineal, y responde a los métodos del álgebra lineal. Tan pronto uno de los elementos del vector de estado se satura ("tropezándose" con alguna pared de la caja), la función σ se comporta (a excepción del signo) como la función ϕ definida en (7). En efecto, la iteración definida por el algoritmo BSB es muy similar al *power method*, pero con una diferencia importante: BSB produce el autovector más cercano a $\mathbf{a}^{(0)}$, y no necesariamente el autovector dominante de **W**. Supongamos que el vector de prueba es $\mathbf{a}^{(0)}=\sum_i \rho_i \mathbf{q}_i$ (expresado según su proyección sobre la base autovectorial). Aplicando k iteraciones de BSB, y asumiendo que el vector de estado permanece dentro del hipercubo, tenemos

$$\mathbf{a}^{(1)} = \mathbf{W}\,\mathbf{a}^{(0)} = \left(\sum_i \mathbf{q}_i \mathbf{q}_i^T\right)\left(\sum_i \rho_i \mathbf{q}_i\right) = \sum_i \rho_i \lambda_i \mathbf{q}_i \qquad (20)$$

$$\mathbf{a}^{(k)} = \mathbf{W}\,\mathbf{a}^{(k-1)} = \sum_i \rho_i \lambda_i^k \mathbf{q}_i \qquad (21)$$

Podríamos pensar que (21) se aproximará al autovector dominante, como ocurre en el *power method*. Sin embargo, la presencia de σ encierra las trayectorias de **a** en una caja; o dicho de otra forma, el primer autovector cuyo factor $\rho_i \lambda_i^k$ se haga mayor que 1 hará que $\mathbf{a}^{(k)}$ quede dividido por ese mismo factor (debido a (18)). Si el autovector "ganador" es $\mathbf{q}_j$, los factores de los demás autovectores se convertirán en $(\rho_i/\rho_j)(\lambda_i/\lambda_j)^k < 1$. Esta propiedad de generar el autovector más cercano al vector de prueba da a la red BSB cierta inmunidad al ruido. Si descomponemos al vector de prueba en un autovector $\mathbf{q}_j$ y un pequeño término de

error $\mathbf{q}_\varepsilon$ ortogonal a $\mathbf{q}_j$, la iteración BSB nos da

$$\begin{aligned}\mathbf{a}^{(1)} = \mathbf{W}\,\mathbf{a}^{(0)} &= \Big(\sum_i \mathbf{q}_i\,\mathbf{q}_i^T\Big)\,(\mathbf{q}_j+\mathbf{q}_\varepsilon) \\ &= \mathbf{q}_j\,\mathbf{q}_j^T\,\mathbf{q}_j + \sum_{i\neq j}\mathbf{q}_i\,\mathbf{q}_i^T\,\mathbf{q}_\varepsilon \\ &= \lambda_j\,\mathbf{q}_j + \sum_{i\neq j}\varepsilon_i\,\lambda_i\,\mathbf{q}_i \end{aligned} \qquad (22)$$

donde ε_i es la proyección de $\mathbf{q}_\varepsilon$ sobre $\mathbf{q}_i$. Si todo $\varepsilon_i\lambda_i < \lambda_j$, entonces sucesivas iteraciones harán que (21) tienda a $\mathbf{q}_j$, pues éste término se saturará primero.

El funcionamiento de la red BSB se puede resumir informalmente de la siguiente manera: dado un vector inicial $\mathbf{a}^{(0)}$, el autovector "más cercano" (con mayor proyección sobre $\mathbf{a}^{(0)}$) guía la evolución de **a** hasta un punto de la "pared" del hipercubo cercano a su esquina. Si $\mathbf{a}^{(0)}$ es idéntico a algún autovector, el estado evolucionará directamente hacia la esquina apropiada. Si hay varios autovectores suficientemente cercanos, cada uno de ellos tratará de guiar independientemente al estado hacia su propia esquina; como resultado, **a** se moverá en una dirección intermedia. El autovector con mayor autovalor triunfará ligeramente, y su ventaja crecerá con cada iteración hasta hacer que **a** se dirija a la esquina asociada con dicho autovector. Si los vectores de entrenamiento con los que se creó la red no son del todo ortogonales, la dinámica BSB forzará de todos modos al estado a evolucionar hacia la esquina apropiada, aunque los autovectores ya no "apuntan" exactamente a las esquinas. Si la condición de ortogonalidad no es violada demasiado, estos "vectores guía" apuntan lo suficientemente cerca de las esquinas como para que la dinámica BSB sea capaz de corregir el error cometido.

4. Extracción de Autovectores con Redes Neurales.

Una simple inspección del diagrama de bloques del *power method* y de la red BSB, así como lo dicho en las secciones anteriores, nos han de convencer que ambos algoritmos son miembros de una misma familia, con propiedades matemáticas similares.

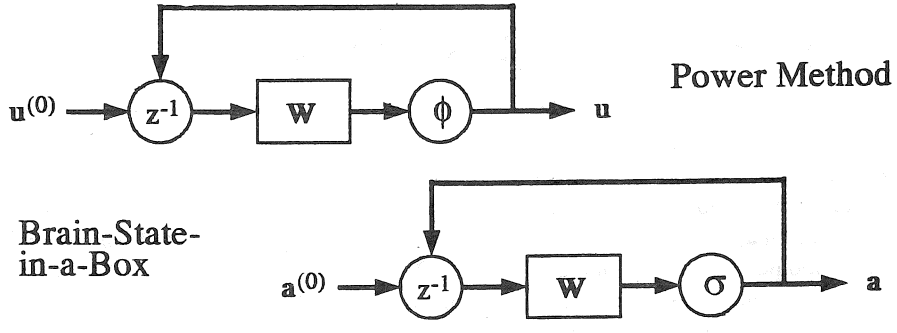

Figura 2

Ambos métodos son iterativos; su convergencia es controlada por ϕ en el caso del *power method* y por σ en el caso de BSB; y la evolución dinámica de su estado es determinada por las propiedades autovectoriales de la matriz **W**. Su diferencia más notable estriba en que BSB es un sistema no-linear, muy sensible a su vector inicial $\mathbf{a}^{(0)}$, mientras que el *power method* es linear y esencialmente insensible a $\mathbf{u}^{(0)}$. Es interesante notar como las debilidades de un método son las virtudes del otro.

Como ya hemos visto, es posible extender el *power method* para obtener los autovectores subdominantes. Nos podemos entonces preguntar, ¿será posible extender BSB para obtener los componentes autovectoriales secundarios de su vector de prueba?

4.1. Realización del método de Aitken.

La respuesta es: sí, mediante la implementación de la metodología de purificación de Aitken. Para comprenderlo, veamos el diagrama de bloques del método de Aitken tal como se aplica al *power method:*

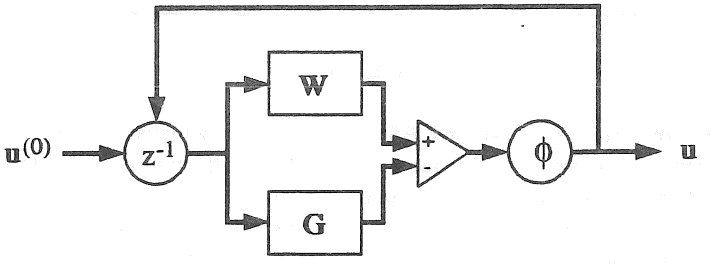

Figura 3

donde la matriz **G** contiene los términos de purificación de (8), que para obtener el *k*-ésimo autovector son

$$\mathbf{G} = \sum_{i<k} (\lambda_i / \|\mathbf{q}_i\|^2)\, \mathbf{q}_i \mathbf{q}_i^T \qquad (23)$$

Podríamos pensar que lo único necesario para adaptar la Figura 3 al uso de una red BSB es cambiar la función ϕ por σ. Sin embargo no es conveniente implementar directamente el diagrama con redes neurales, pues en efecto nos veríamos forzados a combinar las matrices **W** y **G** en una sóla (como se hace en el método de Aitken original), ya que ambas están íntimamente acopladas a través de la linealidad del bloque sumador. Si combinamos las matrices, el circuito se reduce a una sóla red neural similar a BSB, pero con una regla de aprendizaje difícil de implementar debido a los diferentes requerimientos de los componentes **W** y **G** de la matriz integrada. Los detalles de entrenamiento e interrogación de esta red híbrida son tan engorrosos que la hacen poco práctica.

Puesto que las propiedades de **W** y **G** son diferentes, vale la pena aislarla dichas matrices

en redes neurales independientes. Hay dos restricciones importantes que debemos tomar en cuenta al diseñar el circuito final. La primera es que la matriz **W** deberá formar parte de una red BSB para que pueda cumplir con su parte en el *power method* subyacente. La segunda consiste en que **G** deberá recibir directamente como entrada el vector de estado **u** para que pueda "aprenderlo" por medio de una variación apropiada de (23).

Luego de algunas manipulaciones (en las que se aprovechan las propiedades de z^{-1}, la linealidad parcial de σ, y el hecho que **G** es inicialmente 0) logramos convertir la Figura 3 de acuerdo a las restricciones antes indicadas:

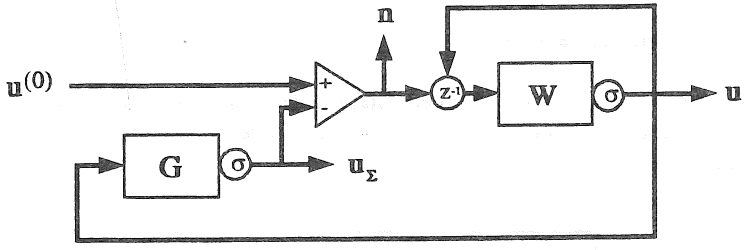

Figura 4

donde se puede apreciar como la sub-red BSB (que incluye a la matriz **W**) fué colocada en el lazo de realimentación negativa del bloque sumador, que en este montaje se comporta como lo haría un amplificador operacional en un circuito electrónico.

Aunque las transformaciones realizadas para pasar de la Figura 3 a la 4 no preservan exactamente la dinámica temporal del sistema original, sí preservan su flujo de información y a la vez introducen nuevas e interesantes propiedades.

4.2. Propiedades requeridas para G.

Un punto crítico del circuito de la Figura 4 es la matriz **G**. Debido a su posición, esta matriz recibe como entrada el autovector principal (extraido por la matriz **W**) contenido en la salida del bloque sumador. Para poder cumplir con su función dentro del método de Aitken, **G** deberá "aprender" este autovector y, por medio de su dinámica no-lineal asociada, reproducirlo constantemente como su salida, para que el bloque sumador pueda cancelarlo de la entrada de la sub-red BSB y así permitirle a ésta extraer el próximo autovector.

Puesto que BSB produjo el autovector canonico, desconocemos con que factor de escala éste se halla en la entrada $\mathbf{u}^{(0)}$ del circuito, por lo que también ha de ser responsabilidad de **G** hallar dicho factor. El método que se eligió para ello es acumular gradualmente el autovector con cada presentación del mismo a la entrada de **G**, de forma tal que su factor de escala a la salida crezca con cada nueva presentación, hasta cancelarlo totalmente en $\mathbf{u}^{(0)}$. Este comportamiento es similar a la carga gradual de un condensador, por lo que podemos decir que la red que incorpora a **G** "integra" los autovectores que recibe como entrada.

A medida que BSB produce secuencialmente los autovectores contenidos en su entrada,

G los debe ir acumulando independientemente, reproduciéndolos todos juntos y con el factor de escala apropiado. Es de hacerse notar que las entradas de G siempre serán ortogonales entre sí por haber sido obtenidas directamente de la salida de la sub-red BSB. Esto es indispensable para que la matriz G pueda mantener la independencia en el almacenamiento y recuperación de los diversos vectores que la componen.

Es necesario que la memoria de G decaiga con el tiempo. De no ser así, el circuito no podrá responder a los autovectores contenidos en los nuevos patrones presentados en $\mathbf{u}^{(0)}$. Sin este decaimiento el circuito quedaría "ciego" a los autovectores del primer patrón presentado, haciéndole imposible responder a ellos en subsiguientes patrones. La constante de tiempo de este decaimiento, similar a la descarga de un condensador, determina la dinámica temporal del resto del sistema y debe ser ajustada para obtener los tiempos de respuesta deseados.

El decaimiento de G, junto con el hecho de hallarse esta matriz en un lazo de realimentación negativa, garantiza además que el proceso de "carga" que genera los factores de escala de los autovectores no prosiga indefinidamente hasta saturar a G: si el factor de escala es demasiado grande, la realimentación negativa tratará de reducirlo hasta el punto exacto en el que el autovector quede anulado en la entrada "positiva" del módulo sumador.

4.3. Implementación de G.

En términos de redes neurales, la matriz G corresponde a una red autoasociativa recurrente, capaz de ser entrenada "en línea." Puesto que sus entradas siempre serán ortogonales entre sí, el entrenamiento puede realizarse sin dificultad usando la regla de Hebb clásica sin peligro de interferencia entre los vectores memorizados.

No cualquier red autoasociativa recurrente puede servirnos en este circuito. Las características dinámicas mencionadas en la sección anterior limitan nuestra elección a dos candidatos: la red *Shunting Grossberg* (Grossberg 1973) y su versión simplificada, la red *Additive Grossberg* (Grossberg 1968). Por simplicidad seleccionamos la red *Additive Grossberg* (que en lo sucesivo denominaremos AG) para implementar las funciones requeridas en G.

Las propiedades de la red AG derivan directamente de las ecuaciones diferenciales vectoriales que la definen. En su forma más general éstas son:

$$\dot{\mathbf{a}} = -\beta\mathbf{a} + \alpha\,\sigma(\mathbf{G}\,\mathbf{a}) + \gamma\mathbf{i} \tag{24}$$

$$\dot{\mathbf{G}} = -\mu\mathbf{G} + \eta\,\mathbf{a}\,\mathbf{a}^{T} \tag{25}$$

donde σ es una función sigmoide de la forma

$$\sigma(x) = (1 + e^{-\omega x})^{-1} \tag{26}$$

La ecuación (24) determina el comportamiento dinámico de la red ante los estímulos. En particular, el factor β determina la rata de decaimiento exponencial del vector de actividad neural **a**; el factor α controla el grado de reverberación causado por la matriz autoasociativa G, lo cual define cuan sostenida es la respuesta intrínseca de la red a los múltiples vectores

almacenados en G; y el factor γ determina la constante de tiempo de "carga" de la actividad ante la estimulación causada por el vector de entrada i.

La ecuación (25), por su parte, define la regla de aprendizaje de la red AG, que es un caso particular de la ecuación (23) llevada al límite contínuo. Los factores μ y η determinan el grado de "plasticidad" de la memoria de la siguiente manera: μ regula la constante de tiempo de decaimiento de la memoria, y η controla la rata de aprendizaje Hebbiano; entre ambos definen el comportamiento integrativo deseado en G. La necesidad de balancear estos dos factores provoca lo que Grossberg denomina el "dilema plasticidad vs. estabilidad"; es decir, la facilidad con la cual G aprende nuevos vectores vs. la durabilidad de los vectores aprendidos. No es posible tener plasticidad y estabilidad a la vez: es necesario establecer un compromiso. En nuestro caso el compromiso se resuelve en favor de la plasticidad, convirtiendo en ventaja lo que para la red AG aislada es un defecto.

Este no es lugar para realizar un estudio detallado de las propiedades dinámicas del sistema de ecuaciones (24)-(25). Por una parte, Amari, Cohen y Grossberg han realizado análisis profundos del comportamiento de una familia de redes que incluye a la red AG como caso particular (Amari 1972, Cohen y Grossberg 1983).

Por otra parte, no es necesario estar familiarizado con todas las peculiaridades de la red AG, ya que en realidad sólo nos interesan aquellas características que ya hemos descrito en secciones anteriores. El hecho de que la red AG esté situada en un lazo de realimentación negativa minimiza el impacto que puedan tener sobre el circuito sus desviaciones con respecto al comportamiento "ideal" deseado. Este fenómeno de "idealización de características" provocado por la realimentación negativa es bién conocido y aprovechado en el diseño de circuitos electrónicos.

5. Comportamiento Dinámico del Sistema.

El comportan\miento dinámico de este circuito es muy interesante, pero a la vez muy difícil de describir formalmente. Aquí tan sólo intentaremos dar una idea de lo que ocurre a medida que los patrones son analizados. Un futuro artículo examinará más detalladamente estos aspectos mediante técnicas gráficas descriptivas (similares a un "osciloscopio simbólico") que aún están en desarrollo.

5.1. La acumulación de autovectores.

Como ya fué dicho, la red AG acumula (o integra) las diversas respuestas de la red BSB. Por una parte lo hace a través de la modificación de la matriz G (lo que Grossberg denomina *long-term memory*), y por otra parte a través de los términos que contienen α y β en la ecuación (24). Este *short-term memory* (así denominado por Grossberg) "sujeta" el autovector el tiempo suficiente como para memorizarlo en el *long-term memory*, y a la vez sirve para combinar linealmente los autovectores a los cuales la red BSB ya ha respondido.

Según hemos mencionado, tanto el *short-term memory* como el *long-term memory* deben decaer para que la red AG pueda responder a nuevos autovectores. El efecto secundario

de este decaimiento es el de permitir a la red AG hallar el factor apropiado para cancelar el autovector: cada presentación de éste "carga" la red, mientras que el factor de decaimiento la "descarga" contínuamente. Gracias a la realimentación negativa, la combinación de ambos procesos estabiliza a la red en el factor correcto para cada autovector memorizado.

5.2. La generación serial de autovectores.

La red BSB responde inicialmente al vector total de entrada del sistema produciendo el autovector más cercano (o el de autovalor más grande, si hay varios igualmente cercanos). Dicho autovector es acumulado en la red AG y cancelado de la entrada de BSB por medio del módulo sumador; entonces BSB responde al siguiente autovector contenido en su entrada, y así sucesivamente. En efecto, la red BSB produce en secuencia (en $\mathbf{u}$, Figura 4) los autovectores que componen al estímulo presentado en $\mathbf{u}^{(0)}$ en orden decreciente de su importancia y autovalores.

El circuito, entonces, responde a un estímulo estático causando la rápida generación de la secuencia de autovectores que lo componen (por la red BSB) y la acumulación de dicha secuencia (por la red AG). Todo ello gracias a la dinámica de esta última, controlada principalmente por los factores α y η. Pero, debido a esa misma dinámica, la recién producida secuencia se repite lentamente una y otra vez a medida que la red AG "olvida" algún autovector bajo el control de los factores β y μ. Esas secuencias "refrescan" los factores de escala de los autovectores que han decaido, manteniéndolos estables dentro de cierto margen de tolerancia determinado primariamente por γ. A este contínuo proceso de "refrescado" de la memoria lo denominaremos *régimen estático de functionamiento.*

Al cambiar el patrón de entrada del circuito, la red AG no responde inmediatamente, sino que su salida cambia lentamente (bajo el control de β y α). Si los cambios del patrón son más rápidos que esa constante de tiempo, éstos no serán reflejados en la salida de la red AG, aunque su autovector principal sí será extraido por la red BSB. A esta forma de respuesta ante estímulos transitorios la denominaremos *régimen dinámico de funcionamiento.* Si el patrón se mantiene por tiempo suficiente, la red AG alcanzará a cambiar su salida al autovector principal y el sistema abandonará el régimen transitorio y entrará en el régimen estático.

5.4. El vector de novedad.

Este vector (denominado $\mathbf{n}$ en la Figura 4) es la diferencia entre el vector $\mathbf{u}^{(0)}$ de entrada del circuito y el vector $\mathbf{u}_{\sum}$ de salida de la red AG, y contiene todo aquello que aún no ha sido reconocido por la red BSB. En particular, debido al retardo de la respuesta (o "inercia") de la red AG, los cambios repentinos del input se reflejan inmediatamente en el vector de novedad; tan sólo cuando la red AG ha incorporado la respuesta de la red BSB, ésta desaparece del vector de novedad. Debido al proceso de carga y descarga de la red AG, esta desaparición no es instantánea, sino que sigue una curva exponencial.

El vector de novedad puede ser, por lo tanto, una fuente importante de datos para un posible "sistema de control de atención" en cuanto le puede proveer de todo aquello que no ha podido ser identificado, así como de aquellos estimulos que cambian rápidamente.

6. Conclusiones.

El análisis de patrones superpuestos o ambiguos es un tema difícil para cualquier técnica de computación. Sin embargo, esta situación es muy común en la vida real, y los cerebros tanto del hombre como de los animales la manejan generalmente muy bién. Dotar a las redes neurales de una herramienta que las pueda ayudar en esa tarea nos parece bastante importante.

La capacidad del presente circuito de discriminar los componentes que conforman una muestra desconocida y presentarlos uno a uno, en orden decreciente de importancia, es un paso hacia la integración de la técnica de las redes neurales con los métodos tradicionales de razonamiento simbólico. En efecto, analizar algo desconocido en términos de lo ya conocido, y hacerlo contemplando todas las semejanzas a lo largo de todos los ejes en un análisis multi-dimensional, es lo propio de la asimilación intelectual. Hacer menos es meramente responder ciegamente a estereotipos.

Los elementos matemáticos utilizados nos sirvieron primero de inspiración y luego de herramienta para el análisis del comportamiento del montaje. Pensamos que, en general, los métodos iterativos del álgebra lineal podrían beneficiarse mucho de las técnicas de "ensayo y error" y corrección de errores propias de las redes neurales, así como de la técnica de anulación de errores en lazos cerrados con realimentación negativa que empleamos en este circuito.

El comportamiento dinámico del circuito es difícil de describir, y amerita un estudio más profundo y exaustivo: aún no se han desarrollado buenos "osciloscopios simbólicos" capaces de mostrar en forma visualmente significativa trayectorias complejas en un espacio multi-dimensional, por lo que lamentablemente las técnicas simplificadas de análisis, tan usadas en electrónica para reducir a niveles manejables la complejidad de un circuito, no son directamente aplicables a este caso. Nuestros estudios preliminares, sin embargo, nos permiten poner grandes expectativas en las capacidades del circuito presentado, así como en circuitos similares.

Referencias.

Aitken, A. C. (1937). **The evaluation of the latent roots and vectors of a matrix**, *Proceedings of the Royal Society of Edinburgh, Section A*, **57**, 269-304.

Amari, S-I. (1972). **Characteristics of random nets of analog neuron-like elements**, *IEEE Transactions on Systems, Man, and Cybernetics*, **2**, 643-657.

Amari, S-I. (1977). **Neural theory of association and concept formation**, *Biological Cybernetics*, **26**, 175-185.

Anderson, J., Silverstein, J., Ritz, S., Jones, R. (1977). **Distinctive features, categorical perception, and probability learning: Some applications of a neural model**, *Psychological Review*, **84**, 413-415.

Cohen, M., Grossberg, S. (1983). **Absolute stability of global pattern formation and parallel memory storage by competitive neural networks**, *IEEE Transactions on Systems, Man, and Cybernetics*, 13, 815-826.

Fukushima, K. (1980). **Neocognitron: A self-organizing neural network for a mechanism of**

pattern recognition unaffected by a shift in position, *Biological Cybernetics*, 36, 193-202.

Fukushima, K., Miyake, S., Takayuki, T. (1983). **Neocognitron: A neural network model for a mechanism of visual pattern recognition**, *IEEE Transactions on Systems, Man, and Cybernetics*, 13, 826-834.

Grossberg, S. (1968). **Some nonlinear networks capable of learning a spatial pattern of arbitrary complexity**, *Proceedings of the National Academy of Sciences*, 59, 368-372.

Grossberg, S. (1973). **Contour enhancement, short-term memory, and constancies in reverberating networks**, *Studies in Applied Mathematics*, 52, 217-257.

Jennings, A. (1977). **Matrix Computation for Engineers and Scientists**, *John Wiley & Sons, Chichester*.