Computación neuronal. Universidade de Santiago de Compostela, 1995 (351-382)

XII

REALIZACIÓN FÍSICA DE REDES NEURONALES ARTIFICIALES (NEUROCOMPUTADORES)

J. Cabestany; J. M. Moreno; F. Castillo
Universidad Politécnica de Cataluña

1. INTRODUCCIÓN

Las Redes Neuronales Artificiales (RNA) son sistemas adaptativos y masivamente paralelos para el procesado de la información. Su operación está basada en un gran número de elementos de cómputo, organizados de una forma determinada y trabajando de manera cooperativa o competitiva. La organización de una RNA pretende emular la operación del sistema nervioso biológico. A partir de unos elementos de cálculo muy simples (**neuronas**), con un elevado grado de conectividad entre ellos (**sinapsis**) y una cierta organización topológica (**organización en capas**), se pretende la puesta en marcha de estrategias de cálculo masivamente paralelo, que se han mostrado muy eficientes en determinadas aplicaciones, como por ejemplo: el control adaptativo, la detección y clasificación de patrones, el tratamiento de imágenes, el tratamiento de datos con una elevada dimensionalidad y la caracterización de sistemas.

La característica esencial de una RNA es su capacidad de **aprendizaje** a partir de ejemplos ilustrativos. Esto representa una clara alternativa a los procedimientos de tipo simbólico implementados sobre ordenador para el tratamiento de la información, en el sentido de que no hace falta "programar" exactamente el mecanismo de relación entre los espacios de entrada y salida, sino que resulta suficiente la especificación de un algoritmo que permita al sistema aprender de la experiencia para llevar a cabo operaciones de generalización que permitirán la obtención de la salida correcta ante determinados parámetros de entrada que son, a priori, desconocidos para el sistema.

La figura 1 muestra este concepto. Cualquier sistema de procesado pretende obtener la relación correcta entre un espacio de entradas y un espacio de salidas o de respuestas al sistema (figura 1.a). En un sistema convencional de tratamiento, esta relación está fijada por un algoritmo simbólico (un cálculo, una función analítica, un conjunto de operaciones, ...), en cambio en el sistema basado en RNA, la relación entre ambos espacios viene fijada por la propia estructura de la Red Neuronal. En esta estructura existirán determinados grados de libertad que permiten, mediante un procedimiento adecuado (aprendizaje), su correcta adaptación a las características del problema a resolver (figura 1.b).

A lo largo del presente capítulo se realizarán consideraciones sobre la posibilidad de realización física de estas estructuras de RNA, contando con que existen varias alternativas, aunque ninguna de ellas puede ser considerada como definitiva en los momentos actuales. En otro capítulo de este mismo libro [Valderrama y Carrabina, 1995] se puede hallar una presentación de las alternativas que la actual tecnología electrónica presenta para la implementación de los elementos básicos de procesado (los "*neurochips*"), y que en muchas ocasiones son tomados como unidades centrales de una organización especial de sistema de procesado que permite explotar al máximo las características inherentes a un sistema neuronal:

- el cálculo masivamente paralelo.
- la elevada conectividad.

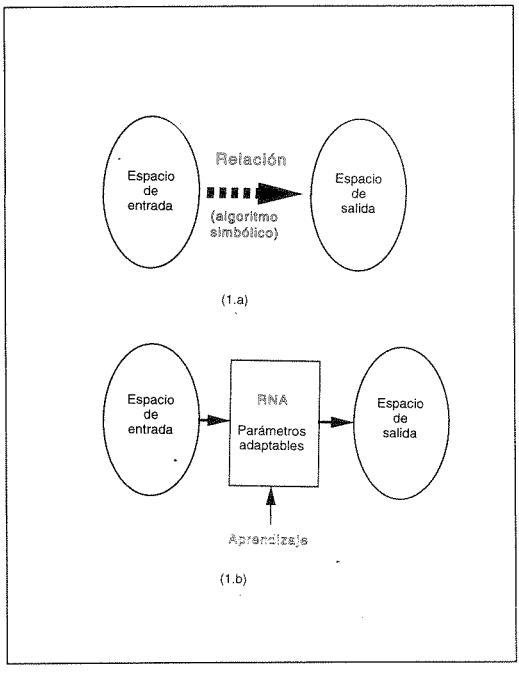

Figura 1. Concepto de procesado. Mediante algoritmo simbólico (a). Mediante una Red Neuronal Artificial (b).

Estos sistemas especialmente pensados y diseñados para la implementación física de RNA acostumbran a ser denominados "**Neurocomputadores**". Generalmente, un Neurocomputador suele ser presentado como una alternativa al procesado en serie, característico de las máquinas del tipo VonNeumann, aunque debe señalarse que sobre estructuras de tipo serie también puede emularse el funcionamiento de las

estructuras de las RNA, a costa de perder algunas de sus prestaciones y ventajas (básicamente, la velocidad de procesado).

La figura 2 [Glesner y Pöchmüller, 1994] muestra una comparación entre posibles arquitecturas de neurocomputador, de acuerdo con las distintas tecnologías accesibles en la actualidad, y tomando como referencia la velocidad de procesado respecto del número de sinapsis posibles. Están indicadas las prestaciones de determinados sistemas biológicos (mosca, abeja, humano) que a pesar de funcionar a frecuencias mucho menores que los sistemas artificiales, superan ampliamente sus características, debido al alto grado de paralelización de las operaciones, y por tanto, de la conectividad entre los nodos de proceso.

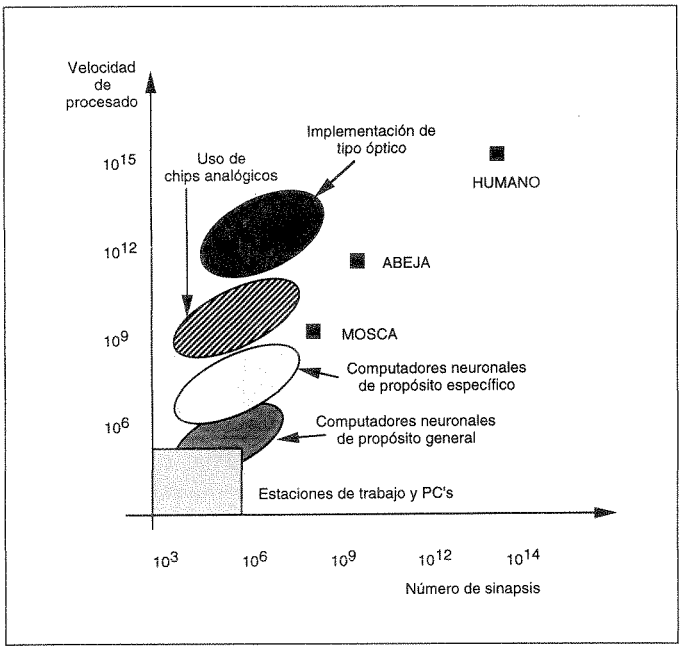

Figura 2. Comparación entre diferentes organizaciones arquitecturales.

Tal y como se verá en el apartado que sigue, existen diferentes posibilidades para la configuración de un neurocomputador. La gran diferencia estriba en los diversos grados de particularización de la configuración (neurocomputador de propósito específico) y la facilidad con que puede adaptarse para la emulación de una RNA con distintas estructuras y algoritmos de aprendizaje (neurocomputador de propósito general).

Resulta ilustrativo no perder de vista que el problema esencial en la utilización de RNA es la determinación de la estructura de la propia Red, adaptada a un cierto

problema. En el proceso hay que hallar las respuestas a las cuestiones básicas reflejadas en la propia figura 3, y que pueden ser resumidas como sigue:

1) ¿Cuántas capas utilizar?

2) ¿Cuántas neuronas hay que disponer por capa?

3) ¿Qué esquema de interconexionado se deberá utilizar?

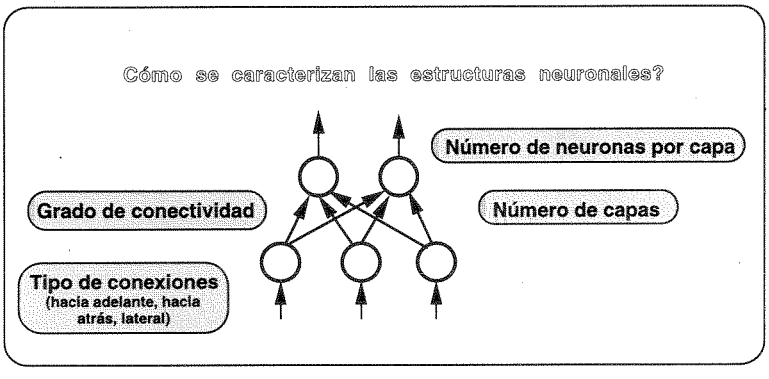

Figura 3. Cuestiones que caracterizan una estructura de RNA.

Dado que no existe una respuesta categórica a las cuestiones planteadas, y tampoco un método establecido para llegar a obtener la solución, en la práctica, se usan métodos de prueba y error a base de realizar ensayos con diversas estructuras y viendo en qué grado se adaptan a la resolución de un problema. Se vislumbra pues, la ventaja de disponer de una arquitectura de cálculo neuronal que permita realizar estos procesos de ensayo de forma ágil y simple, y proporcionando un alto grado de adaptabilidad y programabilidad.

Habida cuenta de las características esenciales de los distintos modelos formales de Neuronas Artificiales, donde primariamente se realizan cálculos relativamente simples de producto (entradas por los pesos relativos a cada sinapsis), acumulación de los distintos productos (modelización de la integración espacio-temporal realizada por el núcleo de las neuronas biológicas), filtrado a través de una determinada función, generalmente no lineal, y el eventual cálculo de distancias para el establecimiento de relaciones de similitud, queda claro que cualquier plataforma que deba actuar de soporte físico para la implementación deberá cumplir unos ciertos requisitos:

a) Capacidad de cálculo suficiente, como mínimo para realizar productos y sumas de forma rápida.

b) Facilidad para implementar la función de activación.

c) Dotación de la precisión suficiente para realizar los cálculos inherentes a los propios algoritmos, sobre todo, los de aprendizaje.

d) Capacidad de memoria suficiente para posibilitar la evolución de los cálculos, y almacenamiento temporal de variables durante la evolución de los algoritmos.

De esta forma puede afirmarse que, inicialmente, las especificaciones de cualquier implementación de estructuras neuronales, deberá pasar por la consideración de las siguientes características:

- Cierta adaptabilidad y controlabilidad durante el proceso de determinación de la propia estructura y topología (ver figura 3).
- Capacidad de memoria.
- Capacidad de comunicación.
- Potencia de cálculo.

2. CATEGORIZACIÓN DE LAS IMPLEMENTACIONES. LA OPCIÓN NEUROCOMPUTADOR

El objetivo que nos ocupa es la implementación física de las RNA. Resulta claro que una alternativa válida es la utilización de un sistema con una determinada capacidad de procesado y que exhiba unas características de paralelismo masivo, ésto es un **Neurocomputador**. Pueden hallarse en la bibliografía varios intentos formales de definición de un Neurocomputador. Por ejemplo: "*Un sistema autónomo que realiza tareas difícilmente especificables en forma algorítmica, con tolerancia a fallos y capaz de trabajar en tiempo real* [Pino y col., 1992]". O bien "*Sistemas que están constituidos por una matriz de unidades de proceso, con un determinado número de primitivas, dotadas de cierto esquema de interconexión, y con capacidad de operación concurrente* [Treleaven, 1989]". En cualquier caso estamos hablando de sistemas con unas prestaciones determinadas, de acuerdo con el razonamiento general realizado en el apartado anterior, y capaces de hacer una emulación de la funcionalidad neuronal de una forma lo más eficiente posible.

Sin embargo de forma previa a cualquier otra discusión, conviene establecer claramente algunos criterios relevantes que permitan una clara categorización y distinción entre las diferentes opciones de implementación. Estos criterios pueden ser [Glesner y Pöchmüller, 1994]:

- El tipo de RNA realizado.
- La tecnología con que está implementada.
- La cascadabilidad.
- El tipo de mapeado de la RNA sobre los elementos de proceso.
- La flexibilidad.

Si se atiende exclusivamente al tipo de RNA, que generalmente comporta un algoritmo de aprendizaje asociado (Hopfield, Perceptron multicapa, mapas de Kohonen, Cuantificación vectorial, ART, ...) se detecta el problema del grado de adaptabilidad del sistema utilizado a la estructura de RNA. Un sistema flexible permitirá la implementación de un cierto número de tipos de RNA, a costa de ceder otras características tales como la velocidad de procesado, que será la característica inherente a un sistema bien "adaptado" a un tipo concreto de estructura neuronal.

Haciendo consideraciones de tipo tecnológico, respecto a las implementaciones de los elementos constitutivos de una estructura de RNA (neuronas, sinapsis, conexiones) quedó ya claro en el anterior apartado que no existe una directiva definitiva y clara. Además, en [Valderrama y Carrabina, 1995] se discuten las distintas posibilidades desde la óptica de la tecnología electrónica (básicamente, en lo que respecta a la dualidad analógico/digital) valorándose las diferentes ventajas e inconvenientes, en lo que hace referencia a la precisión, la velocidad, capacidad de conectividad, posibilidad de constituir memorias, la inmunidad al ruido, las posibilidades de incorporar el propio algoritmo de aprendizaje, etc.

Dado que con las disponibilidades actuales de la tecnología resulta prácticamente imposible la inclusión de todo un sistema RNA en un espacio reducido (por ejemplo, un único *chip*), la estrategia seguida para solucionar este problema, es la construcción de sistemas modulares que permitan la conexión en cascada de módulos simples. De esta forma es posible hablar de "**arquitecturas conectables en cascada**". El problema se encuentra reproducido en múltiples niveles de abstracción: a nivel de la microelectrónica donde se construyen sistemas VLSI complejos a partir de otros más simples, y a nivel de la electrónica de sistemas, cuando interconectando distintos módulos simples se logra avanzar en la funcionalidad del sistema completo. Se verá que esta estrategia es ampliamente utilizada en el constitución de los Neurocomputadores, ya que, habitualmente, por un lado se constituye el núcleo de cálculo a base de poner en cascada varios elementos de procesado (que podrán albergar en su interior más de una neurona), y por otro lado se combinan varios de estos elementos entre si, o bien se facilita la interconectividad con otros elementos del sistema (puertos de comunicación, memoria, posibilidad de almacenamiento permanente de información, ...), constituyendo así el sistema general de procesado neuronal.

Si se atiende a la forma cómo la RNA se mapea ("se traslada") sobre el posible soporte físico, aparecen una serie de posibilidades:

- Mapeado orientado a neurona.
- Mapeado orientado a la sinapsis.

La idea principal aparece expresada de forma simplificada en la figura 4. Ambos casos constituyen un intento de progreso hacia la idea de la implementación del paralelismo y del incremento de la potencia de cómputo (cálculo cooperativo).

El término "Mapeado orientado a neurona" hace referencia al hecho de que cada elemento de procesado físico disponible alberga una "neurona" con sus sinapsis receptoras, o bien un conjunto de neuronas (figura 4.a). Cada elemento deberá ser capaz de realizar las operaciones de cómputo características del elemento que mapea.

Los diseños basados en la "orientación a sinapsis" permiten un alto grado de paralelismo de las implementaciones físicas. Los elementos de procesado disponibles se asignan a cada sinapsis (o grupo de sinapsis) y a cada neurona (figura 4.b).

En general, la característica de **flexibilidad** resulta muy importante para los sistemas electrónicos, y por tanto, también para las opciones de realización física de modelos de RNA. Cuanto más flexible es un sistema determinado, mejor relación calidad-precio posee. Para un usuario de RNA resulta sumamente importante el acceso a una implementación flexible en el sentido de poder adaptarla a gran cantidad de modelos y diferentes algoritmos de aprendizaje, sin más que cambiar un conjunto de parámetros de programación o de configuración. También es verdad

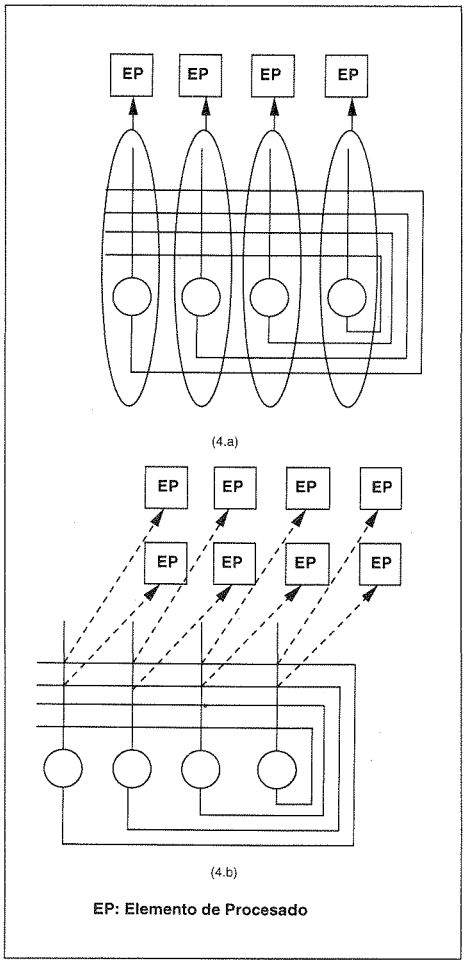

Figura 4. Dos estrategias de mapeado. Mapeado orientado a neurona (a). Mapeado orientado a sinapsis (b).

que la flexibilidad de un sistema está en franca contraposición con otras de sus características, como puede ser la velocidad de ejecución.

La estructura genérica de un Sistema Neurocomputador, que utilice tecnología digital, puede considerarse desglosada entre varios subsistemas [Kung, 1993]: la Matriz de Proceso, una Red de Interconexión, un Banco de Registros/memoria para almacenamiento local de información y una serie de puertos de comunicación con el exterior y otros computadores. La figura 5 muestra dicha estructura. En ella se puede resaltar la funcionalidad de alguna de las partes implicadas:

* El **Computador principal**. Puede utilizarse para el almacenado de grandes cantidades de datos, así como en las tareas de pre y post procesado de la información, aparte puede gestionar el programa de control de los sistemas de interfase e interconexión. En general, suele usarse una estación de trabajo ("*WorkStation*") para este cometido.

* La **Red de interconexión**. Sirve para proporcionar el mapeado de procesadores entre sí, o bien, entre procesadores y módulos de memoria para acomodar y mejor adaptar ciertas necesidades globales de comunicación, con el objetivo de lograr una mejora de las características de velocidad del sistema.

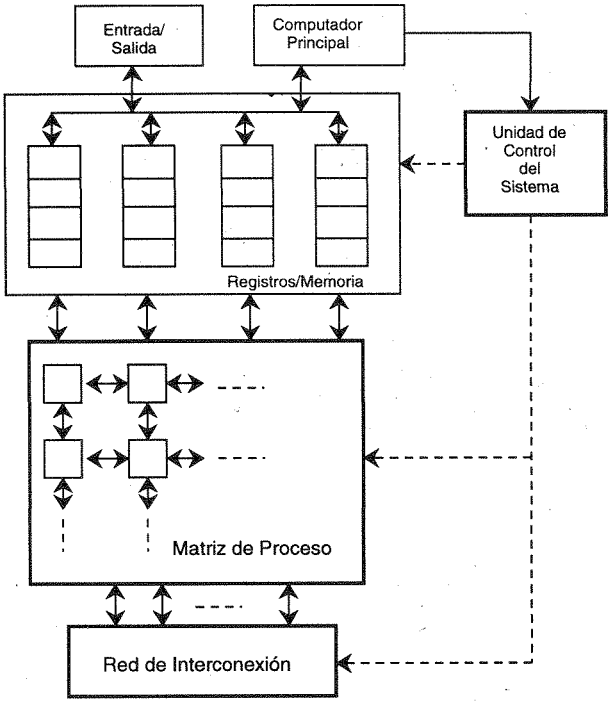

Figura 5. Estructura genérica de un Sistema Neurocomputador.

* La **Matriz de Proceso**. Incluye un número determinado de elementos de procesado con una cierta cantidad de memoria asociada. Dependiendo del tamaño relativo de los elementos individuales de procesado, así como de su organización dentro de esta Matriz aparecen distintos tipos o categorizaciones de la propia Matriz de Proceso.

Desde un punto de vista amplio, los Neurocomputadores pueden ser catalogados según su aplicabilidad y flexibilidad en:

* Neurocomputadores de propósito general.
* Neurocomputadores de propósito específico.

Todo ello en función de la cantidad de modelos para redes neuronales que sean capaces de soportar correctamente. Desde el punto de vista de la constitución interna, al considerar un Neurocomputador, deberá tenerse en cuenta tres aspectos fundamentales:

1) Su grado de programabilidad (altamente relacionado con la anterior clasificación).

2) El número de procesadores físicos que integran su Matriz de Proceso.

3) La complejidad individual de cada procesador.

La figura 6 muestra una catalogación de los distintos sistemas, desde las memorias RAM hasta los computadores secuenciales convencionales, teniendo en cuenta el número de nodos y la complejidad intrínseca de cada uno de ellos [basado en Seitz, 1984].

Resulta evidente que conforme se va ganando en generalidad de uso, la complejidad de los nodos aumenta. Para el caso de los Neurocomputadores específicos, por ejemplo, se están considerando máquinas capaces de implementar uno o muy pocos modelos, en base a un gran número de nodos de procesado de relativa sencillez de cálculo.

Si atendemos al número de procesadores físicos que integran la Matriz de Proceso, y de acuerdo con la idea expresada en la figura 6, se distinguen dos tipos de Neuroprocesadores: aquellos que están construidos alrededor de un **único procesador**, y los que responden a una estructura de **procesador múltiple**. Evidentemente, los primeros supondrán una complejidad individual del procesador superior a la de los segundos.

Los sistemas con una Matriz de Proceso configurada con un sólo procesador (generalmente se trata de un microprocesador de elevadas prestaciones) se caracterizan por su elevado grado de flexibilidad y adaptabilidad a distintas RNA. En cambio,

los segundos exhiben unas mejores prestaciones de eficiencia de cálculo, que dependen sin embargo de la forma como se distribuya la carga computacional a lo largo de los distintos procesadores implicados. Consideraremos dos grandes clases de estas Matrices de Proceso: las **Arquitecturas Sistólicas** y las **Arquitecturas Generales Paralelas**.

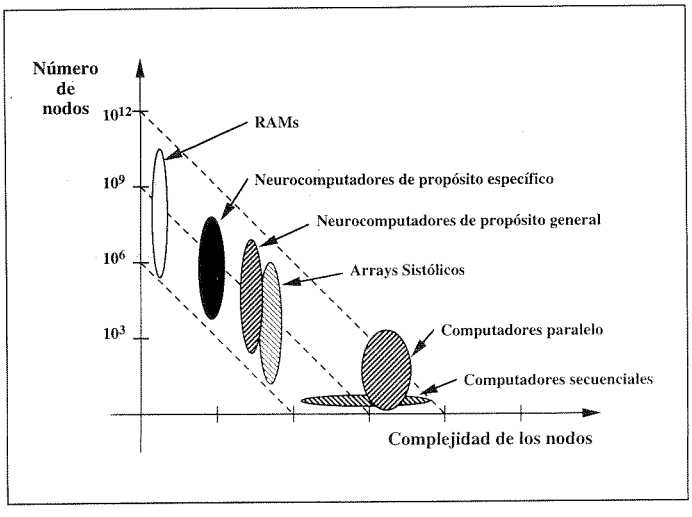

Figura 6. Categorización de distintos sistemas en función del número de nodos de procesado y de su complejidad intrínseca.

Un Sistema Sistólico tal como fue definido en [Kung y Leiserson, 1978] es *una red de procesadores que rítmicamente calculan y pasan datos a través del sistema.* Son muy interesantes porque presentan un buen balance entre tareas de cálculo y de entrada/salida de datos. Ello los hace candidatos a ser utilizados en la implementación de realizaciones de tipo neuronal, donde se desea una alta eficacia de cómputo. Además este tipo de esquemas presentan posibles realizaciones simples y regulares, debido a la conectividad local entre procesadores. En [Ienne, 1993] se indica que existen tres arquitecturas distintas para la implementación de RNA:

* **Arquitectura Sistólica en Anillo** (**SRA**, del inglés "*Systolic Ring Array*").

* **Arquitectura Sistólica en Anillo con *bus* global** (**SRAGB**, del inglés "*Systolic Ring Array with Global Bus*").

* **Arquitectura Sistólica con Matriz Bidimensional** (**SSAA**, del inglés "*Systolic Square Array Architecture*").

En la figura 7 se presenta el principio de organización correspondiente a la arquitectura **SRA** ("*Systolic Ring Array*"). Está compuesta por una fila unidimensional de procesadores cerrada en anillo. La arquitectura permite el mapeado de RNA

usando las ideas anteriormente establecidas de "mapeado orientado a neurona" (figura 7.a) y "mapeado orientado a sinapsis" (figura 7.b). En el primer caso se lleva a cabo un mapeado uno a uno entre las neuronas de la RNA a emular y los procesadores físicos disponibles en el anillo. Cada unidad almacena los pesos de conexión (sinapsis) asociados a cada neurona, lo cual permite obtener en cada paso de emulación la función de activación después de realizar el producto de los respectivos pesos por las componentes del vector de entrada.

Cuando la arquitectura SRA emplea el concepto de "mapeado de sinapsis" se cambia el flujo de la información para obtener al final de un ciclo de emulación la salida de una neurona.

Si a la organización anterior se le añade un **Bus Global** de comunicación entre todos los procesadores físicos, aparece el esquema de la figura 8 que corresponde a la filosofía **SRAGB** (*"Systolic Ring Array with Global Bus"*). El principal propósito de este bus es el aumento de flexibilidad del sistema al permitir comunicación directa y no local entre procesadores. Esta característica puede redundar en un aumento de la velocidad en aquellas implementaciones en que se desea un flujo de datos inspirado en el paralelismo sináptico.

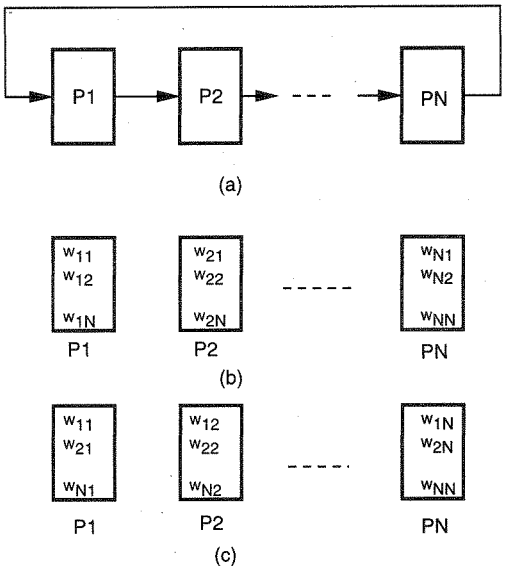

Figura 7. Organización de la arquitectura SRA.

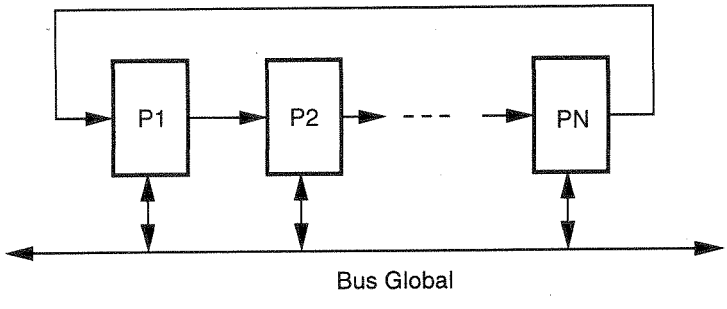

Figura 8. Organización de la arquitectura SRAGB.

Cuando se utiliza una arquitectura bidimensional tal como la mostrada en la figura 9, se está intentando explotar al máximo el paralelismo inherente a los modelos de RNA. En esta estructura, las componentes del peso sináptico asociado a cada neurona están distribuidos entre las unidades de la misma columna de la matriz. Esta arquitectura puede ser designada por las siglas **SSAA** (*"Systolic Square Array Architecture"*). Considerando correctamente la introducción de los vectores de entrada por la parte superior, y de las componentes del vector error por las entradas laterales, pueden ser implementadas perfectamente las fases de ejecución y aprendizaje de un gran número de algoritmos neuronales. La arquitectura está constituida por elementos de procesado de muy poca capacidad de cómputo, y es altamente eficiente para mapear los esquemas de cálculo paralelo propios de las RNA. La gran limitación de esta arquitectura es la pequeña flexibilidad que presenta. Ello representa que su eficacia disminuye drásticamente cuando el tamaño del problema no está correctamente adaptado a las dimensiones de la matriz.

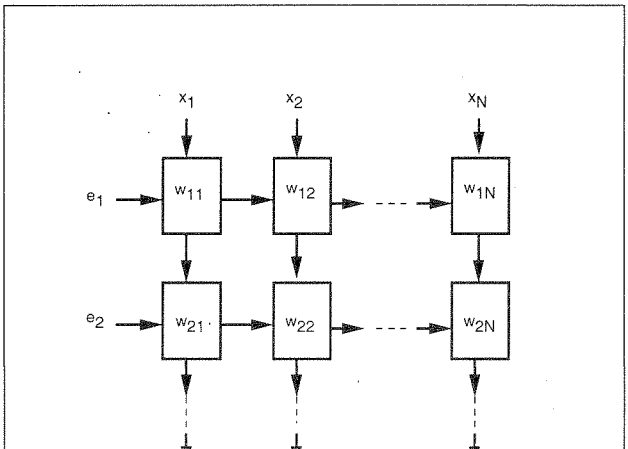

Figura 9. Organización de la arquitectura SSAA.

Una nueva posibilidad, ya dentro de la familia de las Arquitecturas Generales en Paralelo es la que se basa en un **Bus Distribuido BBA** (del inglés, *"Broadcast Bus Architecture"*). La organización aparece en la figura 10, y está compuesta por una serie de procesadores, todos ellos accesibles a través de un bus común de entrada y dotados de un bus común de salida. Aparte, existe un bus adicional (no representado en la figura 10) que sería el encargado de suministrar a los procesadores la información sobre la instrucción que deben realizar.

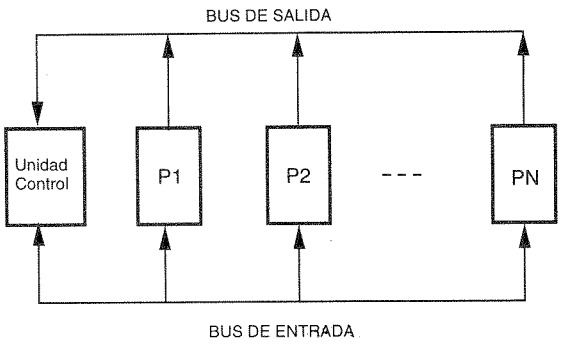

Figura 10. Organización de la arquitectura BBA.

La gran mayoría de los Neurocomputadores actuales responden a organizaciones del tipo esquematizado en la anterior figura 5, con Matrices de Proceso que responden a alguna de las arquitecturas presentadas en este apartado.

3. PARÁMETROS QUE CARACTERIZAN A UN NEUROCOMPUTADOR

El interés por disponer de una implementación física de los modelos de RNA parece fuera de toda duda. Esta impresión, razón del presente capítulo, viene reforzada por el amplio abanico de aplicaciones prácticas de las Redes Neuronales que precisan de tiempos de ejecución realmente pequeños. Todo ello ha motivado el interés y el trabajo de investigación y desarrollo de gran número de grupos universitarios, así como de grandes compañías del sector de la fabricación de circuitos semiconductores (VLSI), tal es el caso de Philips, Siemens, Intel, Hitachi y AT&T. Estas compañías normalmente desarrollan los "Elementos de Procesado" que forman, de acuerdo con una determinada arquitectura, el núcleo de proceso ("Matriz de Proceso" en la anterior figura 5) de un cierto sistema de procesado neuronal o **Neurocomputador**.

Llegados a este punto hará falta concretar algunas características que faciliten al usuario de este tipo de productos, las tareas de especificación de propiedades deseadas para la emulación de uno o varios modelos de RNA, así como las necesidades de velocidad que una cierta aplicación impone al sistema neuronal físico.

Conviene establecer primeramente la diferencia que existe entre un "**Acelerador Neuronal**" o "**Coprocesador Neuronal**", y un **Neurocomputador** independiente. Puede afirmarse que un Acelerador Neuronal tiene las siguientes características [Ienne, 1993]:

- Un Acelerador Neuronal no es un sistema autónomo. Generalmente está conectado con algún ordenador convencional (una estación de trabajo o PC), a través del propio bus de aquel sistema, o bien a través de un puerto de comunicación rápido. El ordenador principal ejecuta el interfase de usuario y todo el "*software*" no neuronal . El acelerador, que estará diseñado para ejecutar eficientemente los algoritmos neuronales, recibe el conjunto de datos de entrada a procesar desde el ordenador principal, con el indicativo de qué modelo de RNA utilizar (en el caso de que el acelerador permita varios modelos). Al final de la ejecución, el ordenador obtiene los resultados del acelerador.

- Un acelerador neuronal debe tener, normalmente, la capacidad de procesar varios modelos de RNA. Sin embargo, éste no es un requisito imprescindible, sino deseable. Algunos aceleradores están orientados a un modelo/algoritmo determinado (Perceptron Multicapa con implementación del aprendizaje con el algoritmo de "*Back Propagation*", ...).

- El tamaño y forma de la RNA emulada por el acelerador debe ser reconfigurable para suministrar al usuario toda la flexibilidad que puede requerir su uso para más de una aplicación, o bien su optimización sobre todo en la fase de depuración.

A continuación aparece una breve lista de características deseables para un acelerador neuronal o neurocomputador con capacidad para emular más de un modelo de RNA (multi-modelo):

* La **Escalabilidad**. La concepción de una determinada arquitectura deberá contemplar la evolución de la tecnología, y deberá estar diseñada para poder acceder sin demasiado esfuerzo a mayores cotas de integración, lo cual permitirá la implementación de un mayor número de elementos de procesado en un mismo espacio físico. Esta premisa fija, desde luego, un cierto "Estilo de diseño" que permita esta migración tecnológica.

* La **Programabilidad**. Una buena estructura debería permitir la reconfiguración a nivel de "*software*" del sistema físico ("*hardware*") disponible. El verdadero reto es lograr que esto sea cierto para un usuario sin un conocimiento muy profundo del "*hardware*" disponible.

* La **Modularidad**. Este término es en cierta forma sinónimo del concepto de "Cascadabilidad" que fue mencionado en el anterior apartado 2, y debería permitir la expansión del actual sistema acelerador o neurocomputador hacia nuevas necesidades originadas por nuevos problemas con distintos flujos de datos o nuevo número de elementos de procesado necesarios. La idea es la

posibilidad de añadir nuevos "módulos" para ampliar la capacidad del sistema, sin tener que acudir al total rediseño de la misma.

* El **soporte de Redes Virtuales**. Cualquier neurocomputador o acelerador neuronal estará compuesto por un determinado número de módulos que permitirán una realización física óptima de RNA de un cierto tamaño. Los problemas reales necesitarán, muy a menudo, de sistemas que permitan la emulación de modelos de RNA de tamaño muy grande, que incluso estarán por encima de las posibilidades de "mapeado directo" de los sistemas neurocomputadores existentes. Es posible atacar esta cuestión cuando se permite el empleo de alguna técnica de partición del problema. Ello redunda, sin embargo, en la pérdida de alguna de las características propias de toda realización neuronal (paralelismo intrínseco, velocidad de procesado, ...).

Es evidente que los dispositivos microprocesadores son candidatos a actuar como núcleo de la Matriz de Proceso de los aceleradores neuronales, aunque no de forma exclusiva, ya que existen también *"procesadores específicamente neuronales"* desarrollados para permitir la implementación física de este tipo de sistemas. Cuando un microprocesador es utilizado para estos menesteres debe ofrecer unas buenas prestaciones de rapidez de cálculo (por ellos mismos, o con la concurrencia de coprocesadores aritméticos específicos), y de gestión de memoria.

Parte de la reciente historia de las realizaciones físicas para la emulación neuronal, está configurada por este tipo de productos organizados alrededor de algún microprocesador comercial de uso general, que gracias a una sistema operativo o "software" de gestión específico puede realizar correctamente un gran número de las prestaciones hasta aquí apuntadas como deseables. Compañías como Texas Instruments, o TRW han sido pioneras en este ámbito. En la Tabla I se muestra un breve resumen de lo dicho:

TABLA I

Compañía	Nombre del Producto	Capacidad Virtual (número procesadores)	Capacidad Virtual de Interconexión	Microprocesador utilizado
Hecht-Nielsen Neurocomp.	ANZA ANZA- plus	30.000 1.000.000	480.000 1.500.000	MC68020 MC68020
Texas Inst.	ODYSSEY	8.000	250.000	TMS32020
TRW	MarkIII MarkIV	8.000 236.000	400.000 5.500.000	MC68020 –

Cuando se dispone de varias posibilidades en lo que a sistemas respecta, se plantea la cuestión de las comparación de las prestaciones a efectos de establecer características de funcionamiento. Indudablemente, la característica más interesante de las realizaciones de tipo neuronal (ya sean los aceleradores, ya sean los neurocomputadores) es la velocidad de procesado. Puede ser instructivo el comparar las medidas utilizadas en este ámbito, con las correspondientes utilizadas en el campo de los ordenadores convencionales. Deberá tenerse en cuenta que existen una serie de diferencias notables entre ambos que pueden dificultar, al menos en parte, el establecimiento de criterios claros:

1) Mientras que es sumamente sencillo tener comercialmente disponibles ordenadores convencionales, para la realización de pruebas y comparaciones, resulta complicado acceder a plataformas neuronales debido al limitado número de productos comerciales, y a que es todavía un campo en constante desarrollo, por lo que muchas veces se están comparando y utilizando prototipos de las arquitecturas.

2) Mientras que en un ordenador convencional el principio de funcionamiento es más o menos común y resulta más simple establecer una prueba de comparación, en los neurocomputadores nos encontramos ante la posible dependencia entre la propia arquitectura física y el modelo de RNA a emular.

El método que tradicionalmente se viene utilizando para cuantificar el funcionamiento de un sistema neurocomputador es la *medida del número de operaciones de multiplicación y acumulación que es capaz de realizar en una unidad de tiempo.* Normalmente se cuantifican los millones de operaciones, dando lugar a los **MCPS** (del inglés "***Millions of Connections Per Second***"). Otra medida habitual es sobre *la capacidad de actualización de los pesos por unidad de tiempo,* dando lugar a los **MCUPS** (del inglés "***Millions of Connections Updates Per Second***"). Ésto significa que aparte del cálculo de los productos con acumulación, se consideran las operaciones necesarias para las actualizaciones de los distintos parámetros involucrados en la evolución de los algoritmos, y en particular, los valores de los pesos (pueden ser necesarias operaciones de acceso a memorias o registros internos, ...).

Los MCPS y MCUPS podrían ser considerados como equivalentes a los tradicionales MIPS ("*Millions of Instructions Per Second*") y MFLOPS ("*Millions of Floating Operations Per Second*") ampliamente utilizados para la comparación de las prestaciones de los ordenadores convencionales, aunque con los problemas añadidos de sus propias imprecisiones y excesiva dependencia de los modelos de RNA emulada. Así por ejemplo, el parámetro MCPS será función del propio modelo. En efecto un neurocomputador que emule una RNA recurrente y con fase de aprendizaje incluida, precisará de una mayor velocidad (más MCPS) para ejecutar una aplicación en el mismo tiempo que otro que esté emulando un modelo no recurrente

y con fase de ejecución únicamente. Por otro lado, y referente a los MCUPS, el tipo de operaciones principalmente involucradas en tal medida son las actualizaciones de los pesos, en el sentido siguiente:

$$W_{nuevo} = W_{antiguo} + \overset{\perp}{\delta} \cdot \overset{\perp}{\delta}^{T}$$

Es posible hallar en la literatura técnica referencias al número de MCUPS de ciertos neurocomputadores que no tienen en cuenta el tiempo empleado en el cálculo de $\overset{\perp}{\delta}$. La conclusión es entonces que no podríamos comparar directamente con otra máquina que efectivamente hubiera tenido en cuenta el cómputo de dichos parámetros.

A pesar de todo los parámetros MCPS y MCUPS han sido y son ampliamente utilizados a efectos de comparación de las prestaciones de los neurocomputadores.

Además de los parámetros y condiciones hasta aquí apuntadas, es posible referirnos a otras particularidades que afectan intrínsecamente a la velocidad de procesado de un neurocomputador [Cornu e Ienne, 1994]. En primer lugar se hace necesario indicar que la gran mayoría de las Matrices de Proceso que se integran en estos sistemas disponen de **unidades aritméticas enteras**, y por tanto, con una limitación inherente de la precisión. Ésto puede afectar a la velocidad en el sentido de que es necesario procurar un factor de escalado para tener un correcto mapeado entre el espacio de los parámetros de la aplicación y el de la representación interna de nuestra máquina. Al disponer solamente de **precisión limitada** en el neurocomputador, se hace necesario aplicar generalmente unos factores de escala que dificultan la evolución y convergencia de los propios algoritmos, debiendo pues realizar operaciones intermedias de supervisión de la evolución de los valores, que evitan dificultades pero que también contribuyen a hacer más lentas las ejecuciones desde un punto de vista global.

Otra particularidad va relacionada con la naturaleza de procesado "***batch***" (que podríamos traducir como "**diferido**", y se correspondería con la idea de realizar actualizaciones del valor de parámetros únicamente después de la presentación de un cierto número de vectores de entrada) que es intrínseca a un cierto número de algoritmos neuronales, y en particular los de aprendizaje, así como también a determinadas estructuras arquitecturales, y en particular a las basadas en "**Anillo Sistólico**". En efecto, para aprovechar al máximo las prestaciones de la arquitectura es necesario iniciar el procesado de un nuevo vector mientras se está todavía procesando el anterior. El problema reside en que no todos los algoritmos neuronales funcionan con una filosofía de procesado en "*batch*", que estaría más o menos adaptada a un buen número de máquinas, sino que algunos de ellos (cuantificación

vectorial, mapas autoorganizados, ...) funcionan respondiendo a una filosofía de procesado "***on-line***" (podemos traducirlo como "**en línea**", siendo la idea contrapuesta de la anterior y se corresponde con la actualización de parámetros después de cada presentación de un vector). Es necesario, pues, arbitrar una fase de adaptación de los algoritmos que evidentemente redunda en una disminución de prestaciones.

En [Cornu e Ienne, 1994], los autores hacen una propuesta de medida de las prestaciones de un neurocomputador que intenta balancear la mejora de velocidad que normalmente supone el uso de un neurocomputador, con la limitación de la eficiencia algorítmica que suponen una mayoría de sistemas neuronales. Se propone pues hacer uso de una medida unitaria que engloba ambos aspectos y que lleva a formular el **Incremento de velocidad $S_M(E_o)$** como la relación entre los tiempos necesarios a un computador convencional (cc) y a un sistema neuronal (hw) para alcanzar una determinada cota E_o de una cierta métrica de la convergencia de un determinado algoritmo (por ejemplo, el error en el caso de un algoritmo de aprendizaje supervisado):

$$S_M(E_o) = \frac{t_{cc}(E_o)}{t_{hw}(E_o)} = \frac{\tau_{cc}}{\tau_{hw}} A_M(E_o)$$

$$A_M(E_o) = \frac{k_{hw}(E_o)}{k_{cc}(E_o)}$$

$A_M(E_o)$ es una medida de la **eficiencia algorítmica**, calculada como la relación entre el número de iteraciones necesarias para alcanzar la convergencia en un neurocomputador (hw) y en un ordenador convencional (cc), y τ es el tiempo necesario para realizar una iteración en la plataforma considerada. Una buena implementación deberá tener una eficiencia algorítmica lo más cercana a la unidad posible.

La conclusión sería que esta nueva medida permite buscar un buen balanceado entre la mejora de velocidad intrínseca que supone el uso de los sistemas neuronales y la conservación de una buena eficiencia algorítmica. En efecto, la relación τ_{cc}/τ_{hw} se corresponde con la noción más tradicional de comparación mediante el parámetro MCUPS, pero que de alguna manera se "pondera" con la eficiencia algorítmica.

Como se observa existen gran cantidad de factores a considerar en la evaluación de las características del funcionamiento de los sistemas neurocomputador. Conviene indicar que desde el punto de vista del usuario de tales sistemas se dan algunas condiciones a tener en cuenta, unas por las propias exigencias del funcionamiento de los modelos de RNA (**Pre y Post-procesado de la información**), y otros cuando se considera la necesaria intervención del propio usuario (**entrada/salida y control de la evolución** del procesado). En la anterior figura 5

ya se vio que en todo sistema neuronal aparece la concurrencia de un ordenador convencional, que entre otras funciones realiza el interfase con el usuario. Otra circunstancia en la debe intervenir normalmente este computador central es en el proceso de entrada y salida de datos, sobre todo cuando a dichos procesos están asociados sendos procedimientos de pre y post tratamiento de la información. La figura 11 expresa dicha idea, y realza el concepto de que un sistema neuronal completo para el procesado de la información precisa, la mayoría de veces, de la concurrencia de dicho ordenador convencional para ser totalmente operativo. Se da la circunstancia de que en algunas soluciones comerciales existentes, consume más tiempo el proceso de comunicación entre ambas plataformas, que el dedicado al procesado de la información propiamente dicho. Ello ocasiona que el usuario deba cuidar de minimizar al máximo dichos procesos, ya que de lo contrario redundan en un tiempo global (entrada/salida, pre y post-procesado, comunicación y procesado propiamente dicho) muy alto.

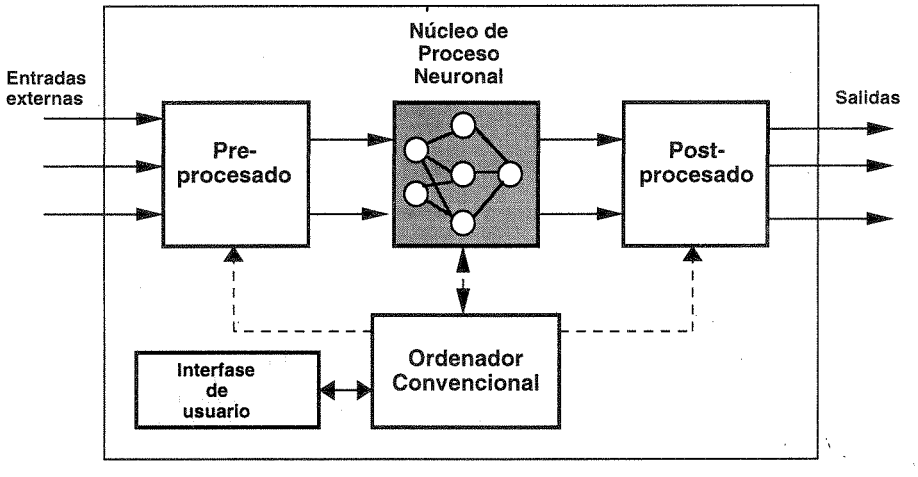

Figura 11. Partes de un Sistema Neuronal de procesado de la información.

4. ALGUNAS REALIZACIONES CONCRETAS

A lo largo del presente capítulo, se han hecho un número de consideraciones totalmente generales y válidas para gran cantidad de implementaciones prácticas del tipo neurocomputador. El objetivo del capítulo no es un análisis exhaustivo de arquitecturas de neurocomputador, ni tampoco la realización de comparación entre las existentes, sino más bien una presentación de las características que debe cumplir uno de tales sistemas. Sin embargo, es razonable hacer una breve presentación de algunos casos representativos, que cubran en parte el espectro de posibilidades apuntado hasta aquí.

Debido al trabajo de las grandes compañías y grupos universitarios en las tareas de investigación y desarrollo, empiezan a existir bastantes propuestas de neurocomputador. Algunos ejemplos son la máquina MY-NEUPOWER de Hitachi [Sato y col., 1993], la CNAPS de Adaptive Solutions [Hammerstrom, 1991], SYNAPSE de Siemens [Ramacher, 1992], SNAP de Hecht-Nielsen Comp. [Means y Lisenbee, 1991], por nombrar a algunas empresas, y MANTRA de *l'Ecole Polytechnique Fédérale de Lausanne* (EPFL, Suiza) [Viredaz y col., 1992], Procesador Toroidal de la Universidad de Nottingham (Inglaterra) [Jones y col., 1991] o DRA del Departamento de Ingeniería Electrónica de la UPC (Barcelona, España) [Castillo, 1992], por nombrar algunos desarrollos netamente universitarios.

En concreto centraremos la exposición en las siguientes realizaciones:

- **CNAPS** de Adaptive Solutions.
- **MANTRA** de EPFL.
- **DRA** de UPC.

De esta forma se reparte el análisis entre productos comerciales y prototipos de laboratorios universitarios, aparte de distribuir el interés entre realizaciones basadas en Arquitecturas Sistólicas unidimensionales y bidimensionales, y paralelas genéricas. Mientras que MANTRA y CNAPS son neurocomputadores que precisan del ordenador central para las operaciones de entrada/salida y de interfase de usuario, el DRA tiene la forma de una tarjeta aceleradora directamente conectable al bus de un ordenador.

4.1. CNAPS de Adaptive Solutions

La arquitectura propuesta por Adaptive Solutions, y denominada CNAPS (***C**onnected **N**etwork of **A**daptive **P**rocessor**S***) es una de las pioneras en el campo comercial con un parque de máquinas instaladas notable [Hammerstrom, 1991]. La Matriz de Proceso de esta arquitectura responde a una **Arquitectura General en Paralelo**, y concretamente en la presentada como **BBA** (Bus Distribuido) en el anterior apartado 2.

El circuito integrado (Neurochip) en el que se basa esta realización recibe el nombre de N64000 [Griffin y col., 1991], alberga en su interior 64 nodos de procesado y físicamente está constituido por más de 13 millones de transistores. La tecnología es de 0.8 mm CMOS, y está fabricado por Inova Microelectronics Co. Para poder asegurar el funcionamiento operativo de un *chip* de tal envergadura, se han incorporado varios niveles de redundancia en el diseño. Responde a una filosofía del tipo SIMD ("*Single Instruction stream Multiple Data stream*"), y se puede decir que es un

buen compromiso entre el diseño de un procesador neuronal y un procesador clásico. Entre sus mejores prestaciones figuran:

* La incorporación del aprendizaje en el propio *chip*.

* La modularidad que facilita la adaptación de la realización al tamaño que requiere el problema a tratar.

* La programabilidad para adaptar la arquitectura al tratamiento y emulación de una amplia variedad de modelos de RNA.

* Está planteada como una arquitectura general de tratamiento de señal, por lo que aparte, pueden ser mapeadas en ella aplicaciones de Procesado digital, de "*Pattern Recognition*" o de Lógica borrosa ("*Fuzzy logic*").

En parte, estas prestaciones se consiguen con la estructura mostrada en la figura 12 (donde los Nodos de Procesado vienen indicados por PNn). La propia figura nos sugiere una gran **regularidad** de la estructura y una **conectividad mínima** que reduce costos y hace más fiable al procesador.

Existen dos buses (OUTbus e INbus) que proporcionan interconexión en paralelo a todos los procesadores simultáneamente, mientras que un bus de comandos (PNCMDbus) de 31 bits es el responsable de indicar a cada procesador lo que debe hacer en cada momento.

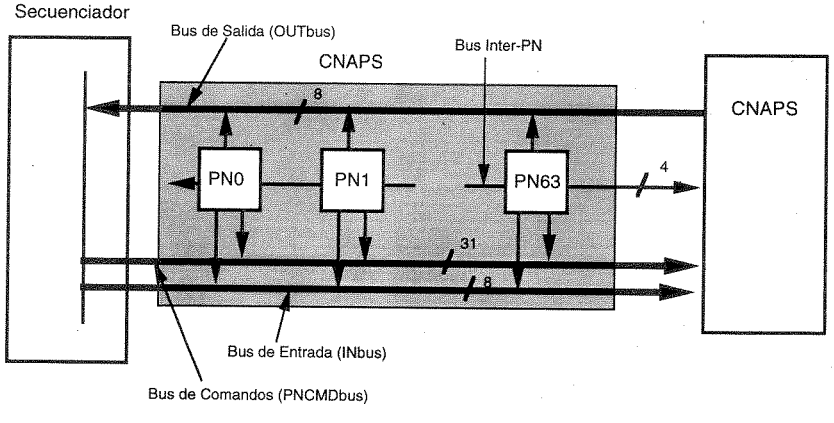

Figura 12. Arquitectura del CNAPS. Comunicación entre los neurochips.

El diagrama de bloques del Nodo de Procesado N64000 viene esquematizado en la figura 13. A señalar un multiplicador en complemento a dos de 9x16 bits, con salida a 24 bits, y un sumador/restador de 32 bits. Para almacenar la información se dispone de 32 registros de 16 bits y de una memoria interna de 4K bytes que puede ser accedida en modo de 8 y 16 bits.

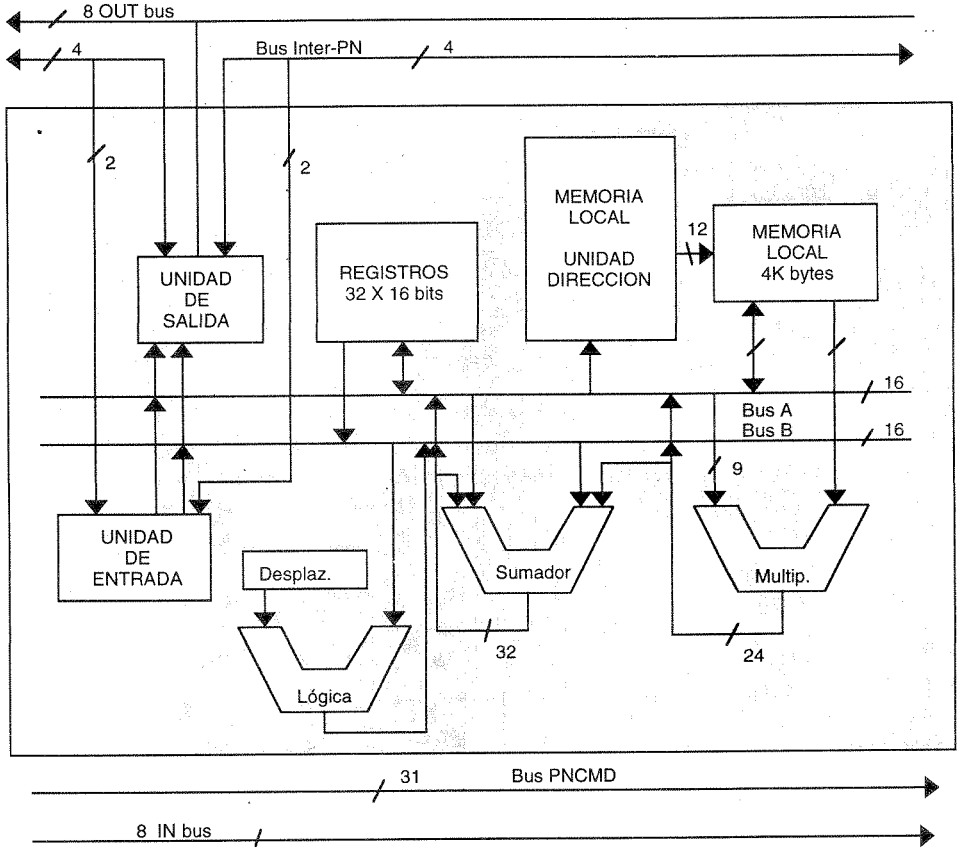

Figura 13. Diagrama de bloques del N64000.

Esta memoria RAM interna es la que se utiliza para almacenar los pesos de los modelos neuronales emulados, y puede resultar pequeña en algunos casos prácticos. Puede afirmarse que una de las limitaciones de esta máquina es su ineficiencia en el caso de emulación de redes virtuales que superen sus recursos internos. En el otro extremo de la balanza, como compensación, están su versatilidad y su programabilidad como principales ventajas.

Comercialmente se ofrece una plataforma (CNAPS Server) donde se ubica un número variable de *chips* N64000 (cada uno con 64 elementos de procesado), pudiendo constituir sistemas de prestaciones variables y de precios adaptados. La plataforma se conecta a un ordenador convencional a través de una red Ethernet. En este ordenador se realiza el interfase de usuario y la adecuación de datos.

El CNAPS Server, tiene unas prestaciones de velocidad para el procesado de algoritmos de tratamiento de señal y de clasificación de patrones, del orden de 100 veces superior al de un supercomputador CRAY-2 (el CNAPS/128 es capaz de ejecutar el algoritmo de *Back Propagation* a 1.7 billones de CPS, y 429 MCPS durante la fase de aprendizaje. El neurocomputador de gama más alta, el CNAPS/512 multiplica estas cantidades por un factor 3.5).

El CNAPS se suministra con su propio "*software*", denominado CodeNet que incluye ensamblador, depurador y librería de C. Su precio de base es de aproximadamente 5/6 millones de pesetas.

4.2. MANTRA de l'EPFL

El neurocomputador MANTRA [Viredaz y col., 1992], desarrollado y construido en la Escuela Politécnica Federal de Lausanne (EFPL) en Suiza, está configurado alrededor de una matriz sistólica bidimensional de procesadores específicos, denominados GENES IV. Esta particular arquitectura fue referenciada como **SSAA** en el anterior apartado 2, y esquematizada en la figura 9. La arquitectura intenta implementar un paralelismo de sinapsis entre los procesadores individuales de la matriz, en oposición a la filosofía de paralelismo entre las neuronas, ampliamente usado en los *arrays* sistólicos lineales. El sistema es capaz de emular eficientemente estructuras de Perceptrones Multicapa, entrenadas con el algoritmo de "*Back Propagation*", modelos de Hopfield y mapas topológicos auto-organizados (Kohonen).

La Matriz de Proceso del neurocomputador está constituido por un conjunto de GENES IV [Ienne, 1993], en forma de matriz cuadrada de procesadores elementales serie, conectados a sus cuatro vecinos más próximos mediante líneas de interconexión serie. El número total de elementos de procesado es 40x40, y la tecnología de fabricación es CMOS de 1 μm. La figura 14 muestra el principio de funcionamiento. Cada uno de los N x N elementos de procesado contiene un elemento w_{ij} de la matriz W, que puede corresponder a la matriz de pesos, o bien a una submatriz de ésta. Los pequeños círculos mostrados en la figura quieren indicar el necesario decalamiento temporal entre las distintas componentes de los vectores de entrada para poder obtener a la salida los vectores correctos, teniendo en cuenta los tiempos de desplazamiento de la información a través de la matriz bidimensional.

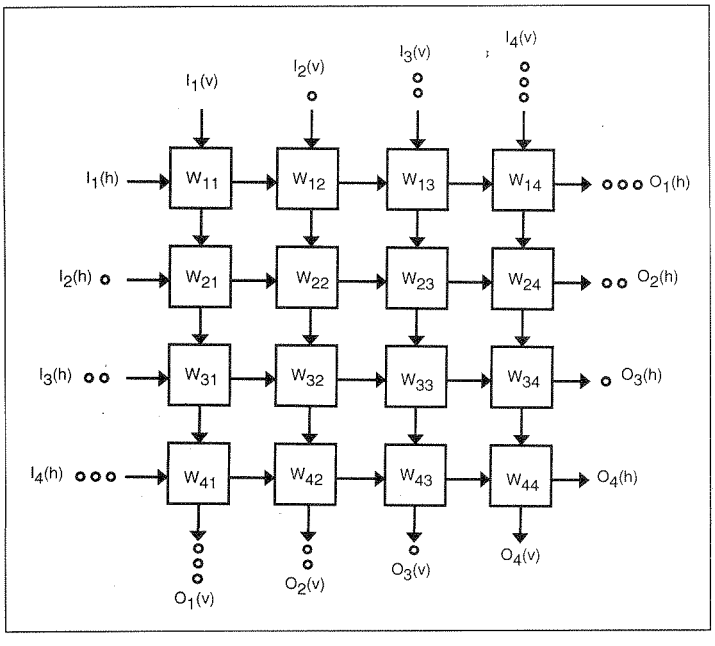

Figura 14. Funcionamiento de la matriz de proceso de MANTRA.

· En principio, GENES IV implementa seis instrucciones, aparte de la NOP (no operación) que dan como resultado un vector, o bien una matriz:

- Producto matriz-vector.
- Distancia euclídea.
- Elemento mínimo.
- Elemento máximo.
- Regla de Hebb.
- Regla de Kohonen.

Cada una de ellas toma los dos vectores I(h) e I(v) como entrada. En aquellas instrucciones que dan como resultado un vector, puede escogerse entre O(h) y O(v) como resultado. En las dos últimas, cuyo resultado es una matriz, se realiza una actualización de la matriz de pesos W.

La figura 15 muestra el diagrama de bloques correspondiente a la arquitectura de MANTRA. Es necesario hacer notar que la parte de control está compuesta por un sistema SISD ("*Single Instruction Single Data*"), basado en un microprocesador TMS 320C40.

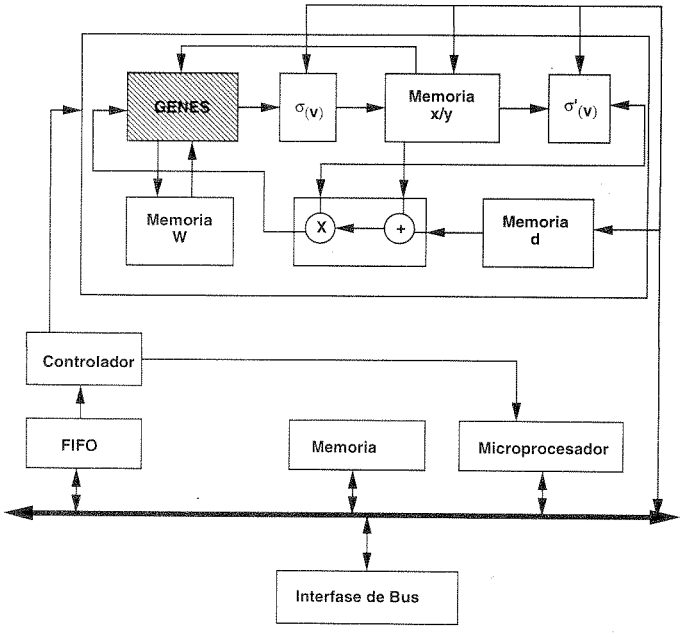

Figura 15. Diagrama de bloques del neurocomputador MANTRA.

* El sistema de memorias y FIFO permite la entrada y salida de datos, así como la memorización de resultados intermedios. Estas unidades son independientes para los tres tipos de datos: Pesos sinápticos, entradas/salidas y salidas deseadas.

* Se observan sendos recuadros que implementan respectivamente la función de activación de las neuronas y su derivada.

* Existe una unidad especialmente dedicada al cálculo de la señal error utilizada por la Regla Delta y la Regla de "*Back Propagation*". Para los otros algoritmos, esta unidad queda en "*bypass*" y se utiliza como convertidor paralelo-serie.

* Existen una serie de convertidores de formato serie a paralelo y viceversa, dadas las características de procesado en serie del *array* GENES IV.

Teniendo en cuenta que la frecuencia de reloj a la cual trabaja MANTRA es de 10 MHz., se presenta en la Tabla que sigue un resumen de las características de velocidad del neurocomputador para determinados modelos de RNA.

TABLA II

Modelo	Ejecución MCPS	Aprendizaje MCUPS
Perceptron, regla Delta	400	200
Back-Propagation	400	133
Modelo de Kohonen con min/max.	200	100

Sin embargo, MANTRA tiene algunas limitaciones muy serias en el momento actual:

* El interfase de usuario es todavía muy rudimentario, lo cual ocasiona problemas de utilización.

* Carece de un compilador eficaz, lo que hace necesaria su programación en lenguaje ensamblador.

* Como muchas arquitecturas bidimensionales es poco flexible y resulta problemático pasar de la emulación de un modelo neuronal a otro.

* Su eficacia es directamente dependiente de lo adaptada que esté la arquitectura física a la RNA a utilizar.

4.3. DRA de la UPC

DRA ("*Dynamic Ring Architecture*") [Castillo, 1992] es una arquitectura propuesta e implementada por el Departamento de Ingeniería Electrónica de la UPC para la eficaz emulación de los modelos de Perceptron Multicapa entrenados con el algoritmo de "*Back Propagation*". La plataforma desarrollada actualmente toma forma de una tarjeta aceleradora neuronal para ser conectada directamente al bus de un ordenador convencional del tipo PC (Computador Personal), el cual sirve como interfase de usuario así como de procesador de pre y post tratamiento de los datos. El propio computador personal realiza tareas de configuración de la tarjeta neuronal, y de control de la evolución de los algoritmos.

La arquitectura del DRA responde a la filosofía **SRAGB** presentada en el apartado 2. La Matriz de Proceso está compuesta por un cierto número de elementos de procesado (neurochips) que merced a sus puertos de comunicación pueden llegar a constituir un "anillo" físico de n elementos. La figura 16 muestra la arquitectura.

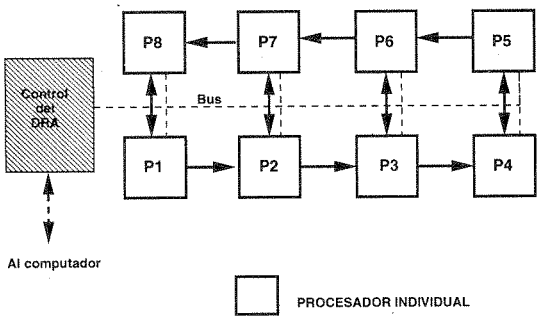

Figura 16. Arquitectura DRA («*Dynamic Ring Architecture*»).

Cada elemento de procesado es un neurochip especialmente diseñado para formar parte de esta matriz de proceso, con la anterior filosofía. El circuito integrado recibe el nombre de **UTAK** ("*Unit for Trained Adaptive Knowledge*") [Castillo y col., 1991] y es capaz de mapear hasta 12 neuronas. Está fabricado con tecnología CMOS de 1.2 mm de European Silicon Structures (ES2). En su estado actual puede implementar el funcionamiento de las neuronas de un Perceptron Multicapa sin aprendizaje interno ("*off-chip learning*"). Es totalmente digital y la frecuencia del reloj de trabajo es de 10MHz. La figura 17 muestra el diagrama de bloques del neurochip en cuestión.

Cada neurochip contiene esencialmente un multiplicador y un sumador, junto a una cierta cantidad de memoria RAM interna que sirve para el almacenamiento de las componentes de los vectores de peso. La precisión interna es de 8 bits, lo cual es suficiente para la implementación de la fase de ejecución del algoritmo. La función no lineal de salida es del tipo sigmoide, y está implementada en base a una tabla interna grabada en memoria ROM. Parte esencial de su constitución la forman tres puertos de comunicación programables (activables/desactivables), dos de los cuales son unidireccionales y un tercero que es bidireccional.

La arquitectura DRA constituida en base a estos elementos procesadores tiene una serie de características relevantes:

* El anillo constituido con los UTAKs emula una capa de la estructura perceptron multicapa, de forma que activando y desactivando unidades, el tamaño del anillo se adapta al tamaño de la capa a emular.

* La estructura completa de Perceptron Multinivel se emula de forma multiplexada en el tiempo (capa a capa en instantes sucesivos).

* El control del DRA a través del Bus Global de la estructura es el encargado de ir configurando en cada instante la función de un elemento de procesado dentro de la arquitectura, o bien de desactivar la unidad en cuestión.

* Las comunicaciones locales y el nivel de procesado asignado a cada UTAK permite un buen mapeado orientado a neurona de los distintos elementos de la RNA a emular.

* Las **características de velocidad de procesado aumentan de forma lineal con el número de neurochips** instalados.

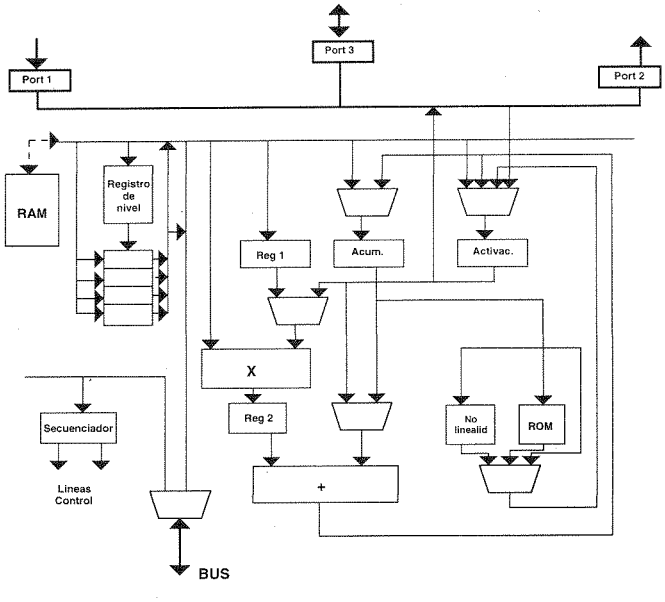

Figura 17. Diagrama de bloques del UTAK.

Una explicación detallada, tanto de la arquitectura como de los detalles internos de constitución del neurochip UTAK pueden hallarse en [Castillo, 1992]. La filosofía del mapeado del modelo de RNA sobre la plataforma física (DRA) es la de "orientada a neurona", sin embargo hay que hacer notar que cada neurochip UTAK permite el correcto mapeado de 3 neuronas distintas, dentro del mismo nivel o capa, y hasta 4 capas distintas. Ésto nos da la cifra de 12 neuronas por neurochip físico.

La velocidad de procesado de cada neurochip será función del número de neuronas que emule. En el caso de emular una única neurona (correspondencia uno a uno entre el neurochip y la neurona del modelo), la velocidad obtenida es de 1.5 MCPS por neurochip.

Actualmente se dispone de un prototipo de Acelerador Neuronal conectable directamente al bus de un PC. El acelerador está gestionado desde el propio PC con la ayuda de un "*software*" que hace de interfase de usuario. Este interfase está dotado de un **compilador de aplicaciones** que se encarga de mapear de manera automática el modelo de RNA sobre los neurochips físicos disponibles en la placa aceleradora, e independiza al usuario de los detalles técnicos concretos de control, configuración y gestión del DRA.

Los trabajos futuros y el desarrollo de la arquitectura DRA tienden hacia:

* La incorporación del algoritmo de aprendizaje "*Back Propagation*" en el propio neurochip ("*on-chip learning*").

* La modificación de la arquitectura interna del neurochip para permitir el mapeado eficiente de algoritmos neuronales de tipo evolutivo. Estos trabajos han culminado recientemente en la propuesta de una arquitectura del tipo RISC para un nuevo neurochip [Moreno, 1994].

5. CONCLUSIONES

El presente capítulo ha tratado de exponer las bases para las realizaciones físicas de modelos de RNA, y en particular de las denominadas Neurocomputadores. Se ha hecho una presentación de las características que resultan deseables en toda realización neuronal, y se han presentado las arquitecturas que posibilitan en parte dichos requisitos.

En el apartado 3 se ha hecho un breve repaso de los parámetros que normalmente son utilizados para cuantificar y comparar las diferentes realizaciones.

Se han considerado también las circunstancias que dificultan la estandarización y sistematización de tales medidas.

Por último se han escogido tres ejemplos de ilustración de todo lo dicho a lo largo del capítulo.

REFERENCIAS

Castillo, F., *Digital VLSI architectures for Neural Networks*, Tesis Doctoral, UPC, 1992.

Castillo, F., Cabestany, J. y Moreno, J.M., "An integrated circuit for Artificial Neural Networks", Prieto, A. (Ed.), *Lecture notes in Computer Science,* 540, Springer-Verlag, 1991, 328-332.

Castillo, F., Cabestany, J. y Moreno, J.M., "The Dynamic Ring Architecture", Aleksander, I. y Taylor, J. (Eds.), *Artificial Neural Networks*, 2, Elsevier Science Publishers, North Holland, 1992, 1439-1442.

Cornu, T. y Ienne, P., "Performance of Digital Neuro-computers", *Proceedings of the Fourth International Conference on Microelectronics for Neural Networks and Fuzzy Systems*, Turín, Italia, (1994), 87-93.

Glesner, M. y Pöchmüller, W., *Neurocomputers. An overview of Neural Networks in VLSI*, Chapman & Hall, 1994.

Griffin, M., Tahara, G., Knorpp, K. y Riley, B., "An 11-million transistor neural network execution engine", *IEEE Int. Conf. Solid State Circuits*, (1991), 180-181.

Hammerstrom, D., "A highly parallel digital architecture for neural networks emulation", Delgado-Frias, J.G. y Moore, W.R. (Eds.), *VLSI for Artificial Intelligence and Neural Networks*, Plenum-Press, 1991, 357-366.

Ienne, P., "Architectures for neuro-computers: Review and performance evaluation", *Technical Report* 93/21, LAMI-EPFL, 1993.

Jones, S., Sammut, K. y Hunter, J., "Toroidal Neural Network Processor: Architecture, operation, performance", *2nd Int. Conf. on Microelectronics for Neural Networks,* (1991), 1163-169.

Kung, S.Y., *Digital Neural Networks*, Prentice Hall, 1993.

Kung, H.T. y Leiserson, C.E., "Systolic Arrays for VLSI", *Sparse Matrix Symposium*, (1978), 256-282.

Means, R.W. y Lisenbee, L., "Extensible linear floating point SIMD neurocomputer array processor", *Proc. of IJCNN*, Vol. 1, Seattle, (1991), 587-592.

Moreno, M., *VLSI architectures for evolutive neural models*, Tesis Doctoral, UPC 1994.

Pino, B., Pelayo, F.J. y Prieto, A., "Implementación VLSI de Redes Neuronales Artificiales", *VI Escuela de Microelectrónica*, Universidad de Santander, septiembre, (1992).

Ramacher, U., "SYNAPSE- a neurocomputer that synthesizes neural algorithms on a parallel systolic engine", *Journal of Parallel and Distributed Computing*, 14, (1992), 306-318.

Sato, Y., Shibata, K., Asai, M., Ohki, M., Sugie, M., Sakaguchi, T., Hashimoto, M. y Kuwabara, Y., "Development of a high-performance general purpose neurocomputer composed of 512 digital neurons", *Proc. of the IJCNN*, Nagoya, Japón, (1993), Vol. 2, 1967-1970.

Seitz, C.L., "Concurrent VLSI architectures", *IEEE Trans. on Computers*, 33, 12, (1984), 1247-1264.

Treleaven, P.C., "Neurocomputers", *International Journal of Neurocomputing*, I, (1989), 4-31.

Valderrama, E. y Carrabina J., "Chips Neuronales", *en esta misma obra*, (1995).

Viredaz, M.A., Lehman, C., Blayo, F. y Ienne, P., "MANTRA: A multi-model neural network computer", *Proceedings of the 3rd Int. Workshop on VLSI for Neural Networks and Artificial Intelligence*, Oxford, (1992).